Extending Dimensional Modeling through the abstraction of data relationships and development of the Semantic Data Warehouse

by

Robert Hart

B.Sc., University of Alberta, 1986

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In the School of Health Information

Extending Dimensional Modeling through the abstraction of data relationships and development of the Semantic Data Warehouse.

by

Robert Hart

B.Sc., University of Alberta, 1986

Supervisory Committee

Dr. Alex Kuo, Supervisor

School of Health Information

Dr. Andre Kushniruk, Departmental Member

School of Health Information

Abstract

The Kimball methodology, often referred to as dimensional modelling, is well established in data warehousing and business intelligence as a highly successful means for turning data into information. Yet weaknesses exist in the Kimball approach that make it difficult to rapidly extend or interrelate dimensional models in complex business areas such as Health Care. This Thesis looks at the development of a methodology that will provide for the rapid extension and interrelation of Kimball dimensional models. This is achieved through the use of techniques similar to those employed in the semantic web. These techniques allow for rapid analysis and insight into highly variable data which previously was difficult to achieve.

Contents

List of Figures

List of Tables

Chapter Outline

Chapter 1: The Kimball Approach

Provides an introduction to the Kimball approach to dimensional modelling along with tools and techniques employed by Kimball. The four questions employed in star schema design and the integrated data warehouse are discussed as well as the limitations and proposed solution.

Chapter 2: Constraints and Limitations

List Constraints and limitations on the Research and Development work performed as part of this thesis. These include elements of data warehouse design and build such as data extraction, load, and transformation (ETL) as well as the lack of business analysis and other decisions that were not germane to the thesis topic.

Chapter 3: Literature Review

A literature review of the Kimball methodology and related areas. Several books written by Kimball are highlighted as well as a series of articles that Kimball describes as an introduction and overview of his methodology and business intelligence.

Chapter 4: Design Methods and Process

This chapter provides a review of the proposed Constellation methodology and design structures developed here. A detailed overview for each of the approaches and the relational table structures for implementation is provided.

Chapter 5: Source Data Sets

Introduces the four data sets used as part of this study. Each was chosen to represent different aspects of health services provided by a public health care system representing Emergency Services, Hospital Care, Home Care, and Residential Care.

Chapter 6: Dimensional Models

Reviews the separate dimensional models designed and built to prove the methodology. Separate dimensional models were built for each of the selected data sets. Conformed dimensions were used wherever possible giving us a functional Electronic Medical Record integrated data warehouse.

Chapter 7: Extension Development Build

Documents the design and build of the SQL transformation code and data structures used to implement the constellation methodology.

Chapter 8: Proof of Concept

Provides multiple examples as a proof of concept involving the selected data sets and models. Multiple patient cohorts are developed, value relationships, as well as a relationship between residential care assessments and emergency encounters. All the functionality provided as part of the methodology is tested and results provided.

Chapter 9: Evaluation of Appropriate Placement in Residential Care

A second proof of concept study that looks at recent work by the Government of British Columbia's Senior Advocate on the appropriate placement of seniors in Residential Care. This study compares the patient assessment data from home and residential care and draws different conclusions then those of the Seniors Advocate.

Chapter 10: Thesis Conclusion

A review of the thesis results and problems encountered during development. Also looks at future direction to move the methodology forward.

Introduction

Business Intelligence and the Kimball methodology [34], often referred to as dimensional modelling, are well established in data warehousing as a successful means of turning data into information. These techniques have been utilized in multiple business areas [33] such as banking, manufacturing, marketing, sales, healthcare and many others.

This success is not only due to the highly efficient data structures employed, but also the approach used in their design. This approach focusses on the business process [32] and the indicators used to measure the performance of that process. This is what forms the core of Kimball's "Star Schema" design.

But these methodologies are under increasing pressure to produce highly valuable information with ever shortened development times. Kimball himself recently wrote on the enduring nature of ETL (Extract Transform Load) and recognized that profound changes must be addressed [49] in order to meet increasing demands and describes how the catch phrase "Big Data" has become the norm with ever increasing volumes, variety, velocity, virtualization and value to that data.

The challenges related to variety in data are especially significant. Examples in the literature such as the work in Semantics and Big Data integration [66] or data linking [67] are common. Knoblock's article [66] is particularly interesting as it describes the integration of data sources at a schema level; but in its end discussion, points to the problems of linking data at a record level as an area requiring research. Yet even at the schema level the relationships are simplistic.

The concept of linked data [67] as discussed by Bizer et al. has key elements that provide a solution to the fundamental problems of extreme variety of data and linking at a record level. In linked data the concept of Resource Description Framework (RDF) triples (subject, predicate, and object) can be considered in terms of relational databases as relationships between a subject and an object or two entities to use database terminology. When working with relational databases these relationships are explicitly defined as part of the data structures and are both simplistic and fixed. A sales order entity is associated to a Customer entity in a relationship represented as a foreign key between these two entities (Customer 123 placed Sales order 723). In the abstract web of data, these relationships exist outside of the data sets and are frequently stored in a hub of relationships with the subject and object unique and the relationships potentially much more complex and dynamic.

Using the concepts of linked data it is possible to address the increasing demands of extreme data variety in a Kimball based data warehouse. To do this, the BI practitioner needs to go beyond the traditional development approach employed in the design of star schemas with traditional database tables and relationships [34] and ask the questions of how the business process and it's measures relates to other processes.

The objective of this work is to develop new methods which allow the rapid extension of a Kimball based star schema as well as to develop the ability to interrelate star schemas to provide extreme variety at higher velocity. This will be demonstrated through the development of four separate health related star schemas representing the Canadian Discharge Abstract Database [61, 62], the Continuing Care Reporting System (InterRai MDS 2.0 based assessment) [59, 60], the Home Care Reporting System [57, 58], and the National Ambulatory Care Reporting System [53, 54].

Separate Star Schemas will be developed for each respective data set as part of an enterprise architected data warehouse approach. These star schemas will then be extended using techniques based on the relational abilities of the underlying database and the abstract relationships within the data itself. The development of these methods will allow any data warehouse based on Kimball dimensional modelling to be rapidly extended with new data as well as provide valuable new insight into the information inside it.

Chapter 1. The Kimball Approach

The Kimball approach to the development of data warehousing [32] is one of the most successful techniques in the field of business intelligence. It has been employed in multiple business areas [32, 33] to provide information solutions at strategic, tactical, and operational levels. This success is due to the efficiency of the data structures involved, the relative ease at which those data structures can be developed, and the methods employed in their design.

The Kimball methodology employs an approach that is directly focused on the business processes of an organization. This methodology is designed to identify the information generated by those processes and structure it such that it becomes the central attribute of an analytical database structure directly available to the users in an easily accessible manor. The design pattern Kimball employs is known as dimensional modelling and the table structures generated are referred to as Star schemas.

1.1 Star Schema Design - The Four Questions

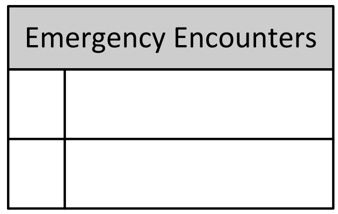
In using the Kimball approach the development methodology employs a series of questions [63] which are covered here. The answers to these questions are discovered through interviews with executives, business managers, and subject matter experts. These questions drive the design of the dimensional model and its development. Focusing on these questions helps make the Kimball process so successful. In essence, it eliminates much of the extraneous elements and focusses on the essential data required by a business to meet its information needs.

Question 1: What is the business process

The first question in the Kimball development methodology is the identification of the business process. This is the first building block of a Kimball dimensional model. The business process is the central element of the Kimball solution and is the basis for the creation of the central database table in a Kimball dimensional model referred to as the fact table. As an example, the fact table in Figure 1.1

represents Emergency Encounters for a typical Health Authority. It forms the central table for an emergency encounter star schema.

Figure 1.1: Emergency Encounter Fact Table



Fact tables represent the business process and their design is critical. Depending on the complexity of the business, multiple fact tables maybe required for a single process. In a truly complex business made up of multiple processes, this can result in a plethora of separate fact tables. A typical health organization will track payroll, general ledger, acute care, surgery, emergency, medications, home care, residential care, infections, mental health, scheduling, physician orders, lab results, and many other processes. In many situations fact tables can represent things other than business processes such as survey questionnaires but these situations are not as common.

Question 2: How do we measure the business process

The second question in the Kimball approach is how do we measure the activity and performance of the business process? In order to effectively manage a business process we must be able to measure it. This can be as simple as a count of occurrences, a sales amount, an average length of time, the duration of an event or a portion of that event, or any other element identified by the business. Measures are numeric and are included in the fact table as attributes.

In dimensional modelling, measures can take different forms, they can also exist at different levels. An assessment of a patient can provide a measure of that patient's health. Multiple assessments can estimate the health of a population. Taken over time, can also model the change in the health of that population due to the quality of care that the population receives.

Multiple business process measures can be included in a single fact table provided that those measures are captured within the same context and level of granularity. The measures must relate at the same transaction level as all other information in the fact table record. To continue the example of emergency encounters, we have four measures employed in the emergency encounter table.

1) A count of emergency encounters.

    This represents a volume measure of the number of emergency encounters. In many business processes a frequency count is common to measure the service demand or delivery.

2) The wait time in emergency.

    A key metric in many public healthcare systems is the measure of wait time, which is commonly how long a patient waits in emergency until they are seen and assessed by a physician. This is frequently compared statistically in terms of minimum, maximum, mode, median, average, etc.

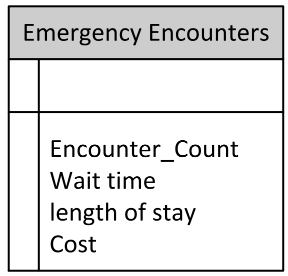
3) The total length of stay in emergency.

    This is the total length of time spent in emergency from the time the patient is registered to the time they are discharged, transferred to another facility, or admitted to acute care. As before, this is a statistical measure to look at how efficient an emergency department is. When an emergency department wishes to reduce wait times they need to know how long patients are staying and how different changes to emergency procedures can shorten that length of stay. What is the impact of opening additional emergency beds or adding additional staff to the emergency department?

4) The cost of the encounter.

This is a simple sum of the charges for the emergency encounter which can include items such as medications, medical imaging, lab costs, procedures, staff time and the duration of bed occupancy.

Figure 1.2: Emergency Encounter with Measures



These four measures are added to the fact table as separate attributes shown in Figure 1.2. Each of these attributes would be evaluated differently and are calculated using standard SQL aggregation functions or can be pre-calculated using technologies such as online analytical processing (OLAP) or statistical software.

Question 3: What is the grain

The next step in the process is the determination of the grain. The grain identifies the transaction level of the individual fact table records and is a fundamental part of the definition of the table. Each fact table represents a business process and the measures of that process are attributes of the fact table. Once the first two questions are answered, the grain of the fact table must be declared to properly define the table and to identify the transaction level of the records in it.

It is essential in the development of the fact table to define the granularity of the records that will be stored and to adhere to that definition. Although it is not difficult to store records at different levels of granularity in the same fact table, the resulting information is often difficult to understand and frequently results in the final product becoming unusable.

As an example, a typical home care referral system captures data records for home support hours, professional service visits, and adult day program visits. These records are all captured at a daily level and represent three separate measures that track the provision of home support services. A second aspect of the referral system is the tracking of the status or lifespan of the referral. The referral is requested, approved, rejected, actively receiving service, and closed on separate days. The length of time between different status changes is tracked as a performance measure. This information is part of the same referral system but at a completely different level of granularity. Although they could coexist in the same fact table it would be confusing to interact with the information and difficult to interpret the results. Two separate fact tables would be necessary in this situation.

The determination of the grain of the fact table is an important step in the Kimball approach. Preferably data is at as finely grained a level as possible. This provides the greatest capabilities for analysis and potentially the best results. If a retail chain wishes to manage staffing levels then it would need to know sales by date and time to determine peak demand on staffing resources. If sales are primarily during the evening and weekends or seasonal in nature, then staffing can be aligned based on that information.

Question 4: How do you define the measure

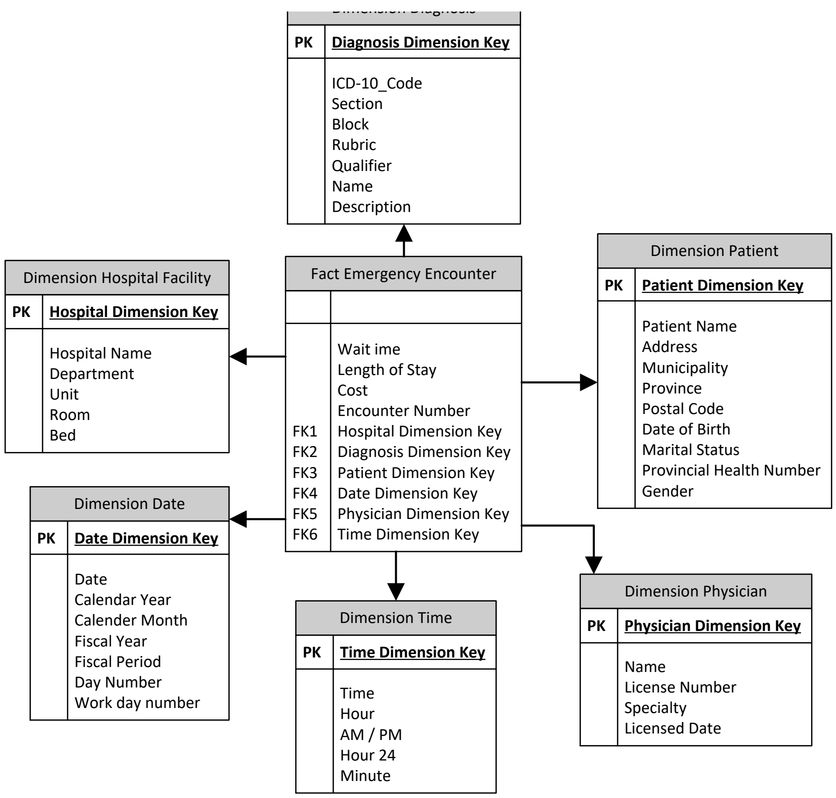
The final element in the process is to determine the dimensions. These can be considered as the attributes that define the measure. When a business perceives its processes, dimensions would be the aspects that they measure them by. A sales system would be measured by customers, date, time, store, product, sales person, and other attributes. An emergency encounter would be measured by patient, diagnosis, intervention, attending physician, emergency department bed location, date, time, and any other element used to define the encounter.

Identifying the attributes that define the measure also identifies the dimensions for the star schema. Each attribute is important and may form the basis of a dimension or be an attribute of a dimension. It is part of designing a star schema to both identify the attributes and structure them into dimension tables.

No attribute is trivial in this process. If the sale of a product varies by color, that attribute represents critical information to the business. It could represent the difference between a successful product and a failed one.

For our emergency encounter example, each key attribute that defines the encounter is created as a separate dimension. In this example, these attributes are date, time, patient, hospital facility, physician, and diagnosis. Other Individual attributes such as patient age or hospital bed can be included as separate attributes to existing dimensions. In general, dimensions are denormalized and structured such that they contain large descriptive fields and potentially numerous attributes. The dimensions represent all the information that defines each individual emergency encounter stored in the fact table.

Figure 1.3: Emergency Encounter Star Schema



In Figure 1.3 each of the dimensions is greatly expanded beyond a single attribute or field. As an example, the hospital facility dimension contains all the attributes that directly relate to the patient location in emergency. The hospital, the department, and the individual bed all identify the patient's location. This allows viewing the data by any of these individual attributes or, in the case of natural hierarchies, at different levels such that the aggregated values can be seen at the hospital, nursing unit, or room level using functionality commonly known as drill up/drill down [9,10]. You can look at average wait time for emergency encounters for a year, drill down and look at the average by fiscal quarter, and drill down further to look at it by month or even day of the week. Individual attributes can be naturally organized into dimensions based on the relationships between them [5, 46].

The design techniques employed in dimensional modeling shown here, are only part of the reason for its success. The resulting database structure, commonly referred to as a star schema, is also highly efficient from a performance perspective. Dimensions are intended to be wide and can contain multiple descriptive columns or large text fields but normally have relatively few records. Fact tables, by comparison, have a small number of attributes comprised of numeric measures and foreign keys to the dimensions and frequently contain a very large number of records. This allows a descriptive search through a dimension with a small number of records which then provides a filtered index search of the facts with a large number of records. The star schema is an optimal search structure from a performance perspective.

1.2 The Integrated Data Warehouse

The star schema has become synonymous with data warehouses in all business sectors but in looking at the approach an obvious limitation becomes apparent. If each of the business processes is represented by one or more star schemas, then the construction of dimensions and the information within them can become unmanageable. The existence of multiple dimensions representing the same information and the potential of different sources of that information represents significant challenges in developing data warehouse solutions.

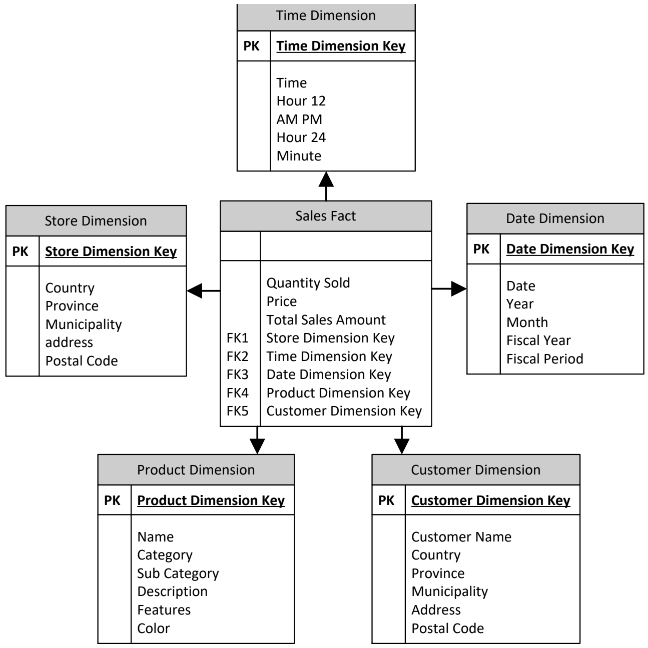
This problem was addressed by Kimball with the concept of the Integrated Data Warehouse [46, 7, 8]. Most businesses achieve data integration with varying levels of success. The reason for the lack of full success are often due to restrictions in available resources, compromises during development, changing business priorities, lack of commitment, strict business requirements, or the complexities of source systems.

It is critically important to understand the concepts behind the Integrated Data Warehouse and the need for data integration. If a business wishes to go beyond the basic star schema and take an enterprise level view of its processes and information, then it needs to understand the concepts and information requirements involved to accomplish those goals.

In an Integrated Data Warehouse we have separate star schemas for each data process. Kimball defines a data warehouse [32, 46] as the collection of multiple star schemas. Each star schema has its own unique fact table and measures different processes. What differentiates the integrated data warehouse is that the dimension tables associated with the fact tables are shared across all star schemas. From a business perspective this makes sense. Common entities such as products must exist across star schemas so that the associated information for sales and for returns can be related to the same product.

To illustrate this using two star schemas provided in Figure 1.3 and Figure 1.4, if a business reported product sales and product returns using two different product tables it would be impossible to associate the resulting information between sales and returns. To expand this further a business’s customers, dates, and stores should all be common between its star schemas. This is referred to in the Kimball approach as conformed dimensions [46, 32, 33].

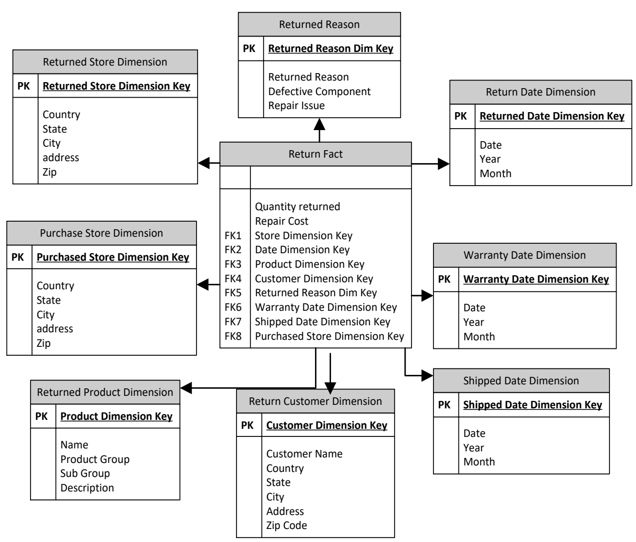
Figure 1.4: Sales Star Schema



The Sales fact table above measures the quantity, price, and total sales amount for a retail company.

These are measured by Store, Product, Customer, Date, and time.

Figure 1.5: Returns Star Schema

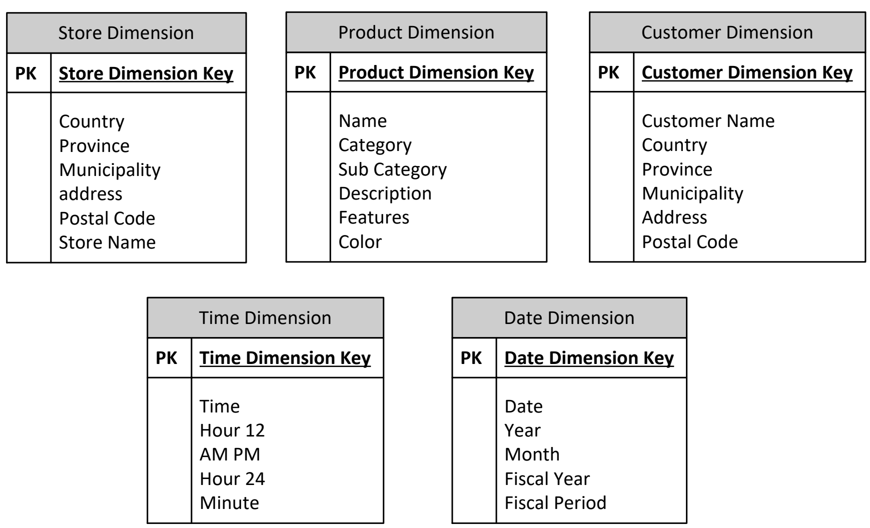


The Returns star schema above measure the quantity of products returned and the costs of repair. This is measured by Store, Product, Customer, Returned reason, and date.

These two Star schemas measure two very different business processes yet have a great deal in common: a customer who returns a product is the same customer who purchased it, the store that the product is returned to might be the same store that sold it and the product that was repaired is the same product that was purchased and returned. Even the date dimension must be conformed, situations where different calendars are used (Japan and numbering years according to the emperors reign) must be accounted for so that reporting is not affected.

The information that defines these business processes is common between them. In order to develop an integrated data warehouse the common elements that define the business transactions must become the common dimensions that we build our star schemas with. This is essential to allow proper reporting and analysis because for all analysis to be effective it must relate to the same things.

Figure 1.6: Common Dimensions



The dimensions above are shared across the star schemas. They represent the Store, Product, Customer, Date, and Time. Sharing these dimensions allows the sharing of information across the business and provides the same context to all business measures. If a hardware product for a door hinge is returned in higher volumes at several stores it is the same product that was sold at those stores. If these stores experience a drop in sales of that product it is the same store where products were returned. We now have information identifying a drop in sales of a product at a number of stores along with a high rate of returns. If we look at these returns and see a common reason for the return or failure of the product we can address those problems.

None of this is possible without the sharing of these dimensions. Conformed Dimensions is one of the cornerstones of the Kimball approach and is often associated with the concepts of master data management [32, 33, 46]. The Kimball approach has introduced tools to assist in the identification of conformed dimension and a method of illustrating the concepts involved known as the business matrix.

1.2.1 The Business Matrix

Within the Kimball approach the concept of the Business Matrix is used [64, 46] to assist in the development of the integrated data warehouse. The Business Matrix can help in visualizing the common information elements that go across business processes. It is essentially a crosstab report listing the business processes and measures by the dimensions that they are reported by.

Table 1.1: Sample Business Matrix

		Dimensions						
Business Process	Measure	Date	Time	Store	Product	Customer	Return Reason	Employee
Product Sales	Quantity Sold	X	X	X	X	X		X
	Total Sales Amount	X	X	X	X	X		X
	Price	X	X	X	X	X		X
Product Returns	Quantity Returned	X		X	X	X	X	X
	Repair Cost	X		X	X	X	X	X
Payroll	Hours	X	X	X				X
	Salary	X	X	X				X

The Business Matrix is an easy to use and understand tool that can help in the design of a data warehouse. It can be used to identify the common elements across the business processes. This information can then help prioritize items in the development process. Additional information requirements can be gathered as part of design to ensure that a dimension employed in the development cycle for one business process will meet the needs of a second business process. This commonality can reduce the overall development effort required for the data warehouse by allowing the reuse of many of the objects inside it.

1.2.2 Leveraging the Integrated Data Warehouse

When a business has achieved a high enough level of integration within its data warehouse, it can then report and analyze its information across different business processes. In doing this, there are caveats that must be understood or it can lead to misinformation. There are also difficulties involved in this exercise relating to the technical skillset of the business intelligence professional which will be demonstrated.

Kimball refers to the ability to query across multiple star schemas as drill across [46, 40]. He also explains the issues involved in performing these functions, most important of which is the context in which the query is performed. If the star schemas and business functions have no relationship between them or the queries are in a different context (Sales by store and returns by product) then the information would also be in a different context and likely meaningless.

To demonstrate the work involved, we will use the Sales and Returns star schemas illustrated in Figures 1.4 and 1.5 and the conformed dimensions from Figure 1.6 to create several SQL queries below.

Query 1: Sales by product and Month

```
Select  d.month, p.name, sum(f.Quantity_sold)
from    Sales_Fact f inner join
        date_dimension d on f.date_dimension_key = d.date_dimension_key inner join
        product_dimension p on f.product_dimension_key = p.product_dimension_key
Where d.year=2011
Group by d.month, p.name
Order by d.month, p.name
```

This first query above will select the total quantity sold for each product the results by product name and month for the year 2011.

Query 2: Returns by product and Month

```
Select   d.month, p.name, sum(f.Quantity_returned)
from     Returns_Fact f inner join
         date_dimension d on f.date_dimension_key = d.date_dimension_key inner join
         product_dimension p on f.product_dimension_key = p.product_dimension_key
Group by d.month, p.name
Order by d.month, p.name
```

This second query is similar to the first, but is selecting the quantity of products returned. It is here that we see the importance of context. These two queries would produce remarkably similar results but are in a different temporal context. Query 1 is filtered to the year 2011 while query two has no such filter, so the results would provide dissimilar information. In this situation returns would be across the entire history of the system.

Query 3: Sales and Returns by product and Month

```
Select   d.month, p.name, sum(f2.Quantity_sold) as units_sold,
         sum(f1.Quantity_returned) as units_returned
from     Returns_Fact f1 inner join
         date_dimension d on f1.date_dimension_key = d.date_dimension_key inner join
         product_dimension p on f1.product_dimension_key = p.product_dimension_key
         inner join Sales_Fact f2 on f2.date_dimension_key = d.date_dimension_key and
                              f2.product_dimension_key = p.product_dimension_key
Where d.year=2013
Group by d.month, p.name
Order by d.month, p.name
```

The above query will display the total quantity of units sold and returned for the year 2013. In all aspects, this is a legitimate query; however, it will return invalid results. This is due to the nature of the underlying business data and the SQL language itself. It is extremely complex to query across multiple star schemas and in some aspects it may not be possible to ensure the correct results. In this query, we are using inner joins between all tables. This means that all joins must be satisfied to return a record. For a sales record to be returned there must be a product record, a date record, AND a product return record for that same product and day. If there were no sales of that product on the same date that the product was returned, then there would be no results from the query. If product returns were not accepted on weekends, the above query would report no sales records on Saturdays or Sundays.

The proper way to perform this query is illustrated below.

Query 4: Sales and Returns by product and Month (proper Query)

```
Select  d.month, p.name, sum(f2.Quantity_sold) as units_sold,
        sum(f1.Quantity_returned) as units_returned
from    (select date_dimension_key, product_dmension_key, quantity_returned,
                     null as quantity_sold
             from Returns_Fact
        union
        select date_dimension_key, product_dmension_key, null as quantity_returned,
                     quantity_sold
             from Sales_Fact) f inner join
        date_dimension d on f.date_dimension_key = d.date_dimension_key inner join
        product_dimension p on f.product_dimension_key = p.product_dimension_key
Where d.year=2013
Group by d.month, p.name
```

Order by d.month, p.name

In the above example, we perform proper queries across the two star schemas and return the correct information. This is done in separate passes where we bring back the results from the two fact tables in two separate queries, then merge these two data sets together before joining to the conformed dimensions. The issues from the join conditions no longer apply. It is noted that this query is only possible through the use of conformed dimensions and a true integrated data warehouse.

The drill across functionality of the integrated data warehouse maybe the ultimate achievement in a Kimball based solution. The examples above also clearly illustrate the complexity in such queries and the difficulties in developing them. The effort involved in creating an integrated data warehouse and in bringing information back across star schemas is significant but the capability to look across business processes to view the larger picture show that the value in doing this is worth the investment.

1.3 Limitations in the Kimball approach

Many articles have been written in regards to limitations in the Kimball approach [19, 20, 24, 25] and dimensional modelling. Most, if not all, have been discredited by Kimball and others. There is however some truth to these articles as there are limits to an Integrated Kimball Data Warehouse.

There have been statements that a dimensional model may miss key relationships that exist in a relational model, that they are more difficult to extend than a relational data model, that they are designed to address a specific business need, or do not capture data at a fine enough detail. In Kimball's article "Myth Busters" he disputed [25] these statements as they are largely untrue. However these statements do point at some problems with the approach.

The Kimball dimensional model produces targeted star schemas. Each of these star schemas represents a specific business process. In large part, the focused approach to the business process and measures is what makes the Kimball approach so successful. The limitation within the Kimball approach is not dimensional modelling and the star schemas, it is the difficulty in interrelating and extending them. The focus of the star schema is the singular business process and does not look at the interrelationship between those business processes.

We have seen that a great deal can be accomplished in an integrated data warehouse but we have also seen that there are limits. As we have illustrated, it is complex to query across star schemas. Drill across is one of the few methods to relate business processes and that is not enough. We need to interrelate and extend star schemas at a level far beyond drill across. We need to be able to relate the measures of one star schema to the individual fact and dimension records of another and even associate fact records in order to achieve greater insight into business data, and to do all of this rapidly and dynamically.

In a recent article Kimball described the enduring nature of ETL [49] but that there is a need for new directions. He also described how the extreme variety, volume, velocity, and value of data are the challenges that are the driving force behind the need for these new directions. Kimball also wrote of the need for new ETL innovation and the emergence of the “Data Scientist”; the new emerging role of individuals in organizations who bring data together outside of the data warehouse for in depth analysis in order to provide new insight and direction. This is the need that must be addressed and the role that must be served. The Data warehouse must bring data together and enable new analysis. To do this it needs to support complex relationships between information represented in the underlying star schemas.

1.4 A Solution to the Limitations in a Kimball data warehouse

If the star schema is to be extended to meet these growing needs, then focus needs to be on the central element of the underlying database technology. The solution to extending star schemas is relationships. However, the creation of physical relationships in all their complexity would not be feasible; we need to find an alternative solution. In order to extend star schemas we need to be able to abstract the relationships between star schemas in a rapid manner. In effect, we need to be able to interrelate fact tables or dimension tables outside of the fixed relational database structure with which they are defined. Thus, developing the same techniques as linking data on the internet and the semantic web.

The key aspect to accomplishing this is to uniquely identify each record in a database just as each url address in the internet can be considered unique. This is not in the form of a primary key that identifies a single record in a table. Rather, this is a single field that crosses all tables allowing that single field to identify every individual record in the database across all tables as unique. In effect, a record can be considered a unique document and is identified as such.

This ability to identify all records uniquely, will allow us to abstract the relationships between the tables and the star schemas in our database. All relationships whether at a field, table, or star schema level can be abstracted and expressed as a SQL statement. This allows us to both extend existing star schema tables with additional information and interrelate them as required. This permits the creation of far more complex relationships then normally possible with relational database technology.

Chapter 2. Constraints and Limitations

This thesis deals with the extension and integration of disparate data sets in dimensional modelling and methods to interrelate different subject or information areas within a Kimball architected data warehouse. A data warehouse is a highly complex system and a comprehensive review of such a vast area is beyond the scope of this work. The focus is on methods to interrelate Kimball star schemas, which are the basis of a Kimball Integrated Data warehouse. Much of the work involved in building a data warehouse, such as the one outlined below, will not be covered as part of this work.

2.1 ETL

The complexities of building a data warehouse is beyond the scope of a simple thesis paper. The techniques involved in the programing aspect of Extract Transform and Load (ETL) alone fills entire volumes of the literature on data warehousing [34, 35]. Taking data and transforming it into information is not a simple task. Although some ETL techniques will be employed in the development of the prototype data warehouse solution, it is not the topic of this thesis which is focused on the methodology and the corresponding data modelling solution for interrelating disparate data sets.

Many of the aspects of data warehousing that involve cleaning and transforming the data, such as the identification of correct individuals as customers or clients, are not addressed here. The techniques involved in these tasks are established and in many cases, involve the use of commercial products or services [49]. Some are often best guess situations with no perfect solution. It is often not possible to correctly identify a customer or client from the data when only sparse information is available.

To avoid these dilemmas and other issues related to data cleansing, only clean data sets are employed [55, 57, 59, 61]. This removes a significant amount of effort involved in development that is unrelated to the methodology proposed here. In addition, only onetime full data loads are employed with no maintenance or update abilities.

2.2 Business Analysis

A large amount of the development of a data warehouse involves business analysis [32, 34]. Requirements gathering, business interviews, source data and systems evaluation, data profiling and analysis, subject area research, and even application analysis are often performed during this stage.

A minimal amount of these activities were performed as part of this work. Research articles, reference materials, [53 - 62] and previous experience with the source data subject areas were relied on to provide the design input for this portion. The research involved in this work does not attempt to redefine the Kimball approach or dimensional modelling, but merely looks at a method to extend the resulting structures of a Kimball data warehouse.

2.3 Dimensional Modelling

Basic dimensional modelling [32, 33] is described in this thesis. Some of the advanced structures involved in dimensional modelling and methods to model problem areas, such as ragged hierarchies, are not covered in this research as they are not germane to the subject.

The dimensional models proposed here represent possible solutions to the specific subject areas and problems involved. As argued by Simsion [63], data modelling is as much an art form as a science. Several data modelers, when presented with the same problems and requirements, will deliver multiple data solutions. The dimensional models developed are intended to represent possible solutions to the subject areas and are only complex enough to be representative of the subject matter.

2.4 Measures

The measures used in the prototype are based on the supplied literature. In the home care and continuing care reporting systems, CIHI standardizes the measures based on a standard patient population. The coefficients used in this calculation are unavailable, so this is not performed here.

2.5 Technology

The solutions proposed here can be applied to any database or technology platform. Different tools and products frequently require variations in approach to best utilize their abilities. Some have unique functionality that can be highly beneficial while others may lack functionality. Ultimately the selection of tools and technology are determined by functional requirements, cost, availability and personal bias.

For the purposes of this work the Microsoft product stack consisting of Microsoft SQL Server 2012, SQL Server Integration Server, SQL Server Analysis Server, and Microsoft Office Excel were selected. These tools were selected due to availability and familiarity with the products.

Chapter 3. Literature Review

The purpose of this review was to delve more deeply into Kimball's Dimensional modelling, with particular emphasis on methods to rapidly extend or develop star schema models as well as interrelate the information in our star schemas. Much of the current literature is focused on "Big Data" and Hadoop as well as the interpretation of large amounts of unstructured data such as the "Twitterverse" or other social media sources. Dimensional modelling, by comparison, is a well-established and proven methodology and not the focus of current research, making it difficult to find insightful research articles on the subject.

3.1 Methods

This review was performed online through multiple sources. The University of Victoria's Library search engine (Summon 2.0) which includes its catalogue, digitized selections, as well as citations and the full text from over 83% of scholarly journals was the primary source for much of this research. A second resource employed was Google Scholar, although significant overlap was noted between these search engines. The Kimball group and their online repository was a third resource. Dr. Kimball is recognized as the father of dimensional modelling and has remained very active in the subject area as a consultant on many data warehouse project, an educator through Kimball University, and a prolific writer. Books Including works by Kimball on Data Warehouse design and construction, several texts on Data Quality and Simsion's work on data modelling were also used as resources. In addition several online journals and open discussion forums were reviewed, although these proved to be of limited value. Finally, corporate resources such as IBM, SAP, QlikView, and Healthcatalyst were examined with Healthcatalyst being most noteworthy.

The online search catalogues were explored through the use of keyword searches. The terms searched for included "Star Schemas", "Data Warehouse", "Business Intelligence", "OLAP", or "Dimensional Modelling" used in conjunction with various adjectives such as "Extending", "Relating", "Limitations", "Problems with", or "Associating". Another query path involved the above search terms combined with "Healthcare", "Medicine", and "Medical" looking for areas of healthcare data warehouse research. For the most part these search terms proved ineffective. Individually the phrases would return articles on the subject but nothing was found on how to extend or associate dimensional data models. Multiple articles were found for Data Warehousing in the area of Healthcare but these also proved to be of limited value. Greater success was found when employing Dr. Kimball's name to find articles that referenced his work, although again this failed to locate any articles directly related to extending dimensional models.

Search results were reviewed for relevancy by reading there abstracts to determine if they were related to the subject of extending or relating star schema data models. Other articles of interest were those that potentially offered insight into techniques that related to star schema design or made note of limitations in dimensional modelling.

3.2 Review Results

3.2.1 Kimball's Works

The published works of Kimball are the best resource available on dimensional modelling. They include several books, countless articles, presentations, and educational materials. The difficulty in reviewing the works of Dr. Kimball is the volume of literature available with articles dating back to 1995. Because of this there are occasional conflicting statements caused by both evolving technology and methodology. One of the best sources for Kimball's work are his books [33, 34, 35, 46] which go into great detail on the subject of data warehousing.

3.2.1.1 Kimball Books

The first book recommended for an overall review of what is involved in building a data warehouse is The Data Warehouse Lifecycle Toolkit; Practical techniques for building data warehouse and business intelligence systems [34]. This book and the accompany course "The Data Warehouse / Business Intelligence Lifecycle in Depth" cover all aspects of what is involved in building and maintaining a data warehouse. This is not a technical manual on developing a business intelligence system, rather a guide book covering the conceptual planning, project management, roles and responsibilities, analysis, product selection, design, and build of the data warehouse through to practical techniques for report development. The book does not go into advanced techniques on dimensional modelling or Extract Transform Load development but provides a sufficient introduction to all the necessary subjects required for an organization to build a data warehouse system from a beginner to an intermediate level. It is an excellent review and is delivered from a practical business perspective.

The second book that should be considered is The Data Warehouse Toolkit, The complete guide to Dimensional Modelling [33]. This is an ideal book on the subject of designing star schemas and a highly practical guide for beginners or experts. It focuses on the methodology of dimensional modelling and is based on practical business applications. Every subject from the most basic dimension and fact tables to complex structures such as bridge tables or combination fact dimension tables, is illustrated and discussed through concrete examples from various industries. Even pitfalls and possible mistakes are illustrated with explanations of how and why these can occur and the preferred solution.

A third book that completes the essential Kimball data warehouse library is The Data Warehouse ETL Toolkit [35]. This book goes into greater depth on development concepts for building a data warehouse. As with the other books it is written from a practical perspective by experienced professionals and covers a variety of related topics such as audit logging, metadata, data warehouse architecture, data quality and real time ETL. Each section comes with useful tips, techniques, and helpful advice such as guidelines to build a back-out procedure as you build your load processes before failure might occur.

An optional fourth book is a complete collection of articles written by the Kimball group, The Kimball Group Reader [46]. This is a noted reference book on data warehousing and is an ideal source for design tips from the Kimball group. Many of these articles have been expanded with additional illustrations and text not available in the original published versions. Unlike the Kimball Group website, which has these articles arranged in chronological order, this book structures the articles around the conceptual areas of Data Warehouse design and construction with practical approaches to all applicable areas.

3.2.1.2 Kimball's Information Management Series

As previously described, there is a large volume of articles also available in industry journals and online. Prominent among those is a series of articles written for the Journal DM Review (later changed to Information Management). These articles are also available online at www.Kimballgroup.com and were republished in The Kimball Group Reader [46]. The order that these articles are reviewed follows his book The Data Warehouse Lifecycle Toolkit [34]; Practical techniques for building data warehouse and business intelligence systems described in the previous section.

The first article in this series was on Data Quality [1]. Although this article is not related to dimensional modelling, it is noted here as it was important in the development of the methodology proposed in this paper. This article explored the need for both a culture and a commitment to data quality within an organization. Kimball then went on to explore the possibility of capturing and measuring data quality within the data warehouse. This work was very reminiscent of Olson's [47] and Maydanchik's [48] books in terms of the organizational culture, commitment to data quality, and the information required in capturing and measure data quality events. The major difference in this article was that these events were transformed into a dimensional model allowing measurement of data quality not just capturing the events. The measurement of data quality is one of the most important requirements to ultimately addressing it within an organization. The approach in the article had one limitation, there is a need to relate and report the measurement of data quality within the context of the information inside the data warehouse. We also need to relate the measurement of data quality to all other measurements and dimensions available in the system. It was this need that drove development of the approaches in this thesis. This limitation can only be addressed through extending our star schema information and developing a data driven approach to relationships to support this extension.

The next article in the series examined the work required before beginning the development of a data warehouse [2]. He proposed ten important questions to look at and answer before starting. These deal with subjects such as requirements gathering, metadata, data profiling, long term support, security for the system and the information inside it, latency of the data and the most important factor to consider; the organizational commitment to the system both at an executive level and from staff. If an organization does not commit to its corporate systems and information, then the project will ultimately be limited in what it can achieve.

After considering these factors, the next issue considered in the article relates to scope and boundaries [3]. This includes defining the environment for the data warehouse, the responsibilities related to it, and the scope for the initial development. A data warehouse is a dynamic system that continuously grows and evolves. It cannot be built as a single project but must be approached as a long-term commitment.

Once these decisions have been understood and planned for, the tasks of building a data warehouse can begin [4]. The first step in this, as described by Dr. Kimball, is data wrangling. An organizations data can come in virtually any form and path. Mastering the flow of this data to bring it into the data warehouse is not a simple task and considerable effort can be expended. The source systems and the business functions must be exposed and understood. Data sources may be transactional systems using relational databases, message feeds such as HL7, text files, or any other possible source. Even within individual sources, irregularities might be present in the data that may affect its replication.

From this point begins the design and construction of the target solution. Preliminary design concepts would be proposed during requirements gathering but finalizing the design and its construction often occur as data wrangling is in process or nearly complete. Once the effort of capturing the business data is in process, it becomes possible to better recognize [6] an organizations fact and dimension data through data profiling and structure analysis. This is often apparent in the data and its structure. Textual attributes that describe the nature of a transaction or the elements of stable entities (Products, Procedures) are part of our dimensions. Numerical elements that are repeating in nature and found in entities that are natural cross reference tables are commonly facts or measures. The foundation of the data warehouse is the measurement event that produces the fact record and these transactions are commonly found at these cross reference points. It is the dimensions and facts that drive the user-interface experience. Kimball describes all of this through the example of a sales transaction system. This provides a very real world example of the information and the process.

The next two articles in the series [7, 8] describe the essential steps for the integrated enterprise data warehouse. The level of integration required to truly develop a system such as that described by Dr. Kimball cannot be achieved without a significant organizational commitment. This involves the development of data standards and definitions across an entire organization. Sales, Manufacturing, logistics, human resources, all departments within an organization must agree and adhere to the same definitions. All information related to business processes and measures within an organization must adhere to common reference definitions and standards where applicable. From an Information Technology perspective this is frequently considered under the category of data or information architecture and master data management.

Kimball goes on to describe the architecture of an integrated data warehouse and introduces tools to help achieve data integration such as the business matrix and conformed dimensions. He also introduces two roles within an organization to assist in both development and long term growth of the data warehouse; the dimension manager and the fact provider. Others names that could be used to describe these roles are data architect or information architect. Kimball also iterates that the key benefits of building an integrated enterprise data warehouse is a consistent view of the information that drives the organization, and ability to view business measures simultaneously across business processes using functionality such as drill across. This is a significant achievement in any organization as it requires both a vision of an organizations information flow and a commitment to achieving the goals of that vision.

After describing the integrated data warehouse Kimball explored some of the concepts of how users interact with the data warehouse in a two part article [9, 10]; Drill down to ask why. These articles do not just explore the basic BI tool functionality of drilldown in a hierarchical dimensional structure but examine the concepts of user interaction with a data warehouse to answer business questions and gain insight into the business processes and information. The interaction is essentially the same in that a user begins with the most basic of information provided and then progresses through increasing levels of analytical application stages to gain insight and answers to complex questions. Kimball discussed five stages to represent the levels of analytical application process. These stages start with basic report generation, to the identification of exceptions, determination of casual factors, modelling alternatives and tracking actions. These concepts show the value of what business intelligence and data warehousing can achieve, the goals for its development and measures of its success.

The next articles in the series described the concepts of slowly changing dimensions [11, 12]. These are actually advanced concepts and are the three basic design principles of maintaining dimensional data through data changes. Although this sounds trivial, it is a complex concept that must be considered when designing dimension tables. The goal of these concepts is to be able to display the results of a data warehouse query that reflects the correct values for business measures at a point in time. The ability to display a company's sales results by region, both before and after a reorganization, means having the correct address for a customer and a representation of sales areas at a point in time. This is reflected in data warehouse dimension objects by employing the techniques of slowly changing dimensions.

Kimball suggested three basic functionalities to provide this capability. The first is simply to ignore the requirement and not track any changes overwriting the dimensional data when changes occur, the second is to add additional fields to a table to reflect both states of the record. Finally, the third is to employ versioning within the table by expiring one record and creating a new version of the same record with the altered information. Other methods that employ combinations of these techniques are also possible and have been noted in other articles; but, the underlying purpose to reflect information at a point in time is the same.

The series continues with another article that has a dimensional focus [13] entitled “Judge Your BI Tool through Your Dimensions” which has several good points that any developer who follows the Kimball approach should take to heart. Although dimensions may be the smallest tables in a data warehouse, they are the heart of a data warehouse as they define the measures. They also implement the user interface as it is the navigation of facts provided through the dimensions that enables the Slice / Dice / Drillup / Drilldown abilities that are synonymous with business intelligence. A good Business Intelligence tool must be able to utilize the dimensions to navigate a star schema to provide a window to its fact table measures. Kimball goes on to describe this functionality most of which is well established but also notes an advanced technique. In this approach, a tool will traverse a fact table to apply constraints that have been set on other dimensions, then use those results. For example, we may want to develop a patient cohort by first navigating a patient assessment model, then examining emergency encounters.

The final article in the series [14] is one that focuses on fact tables. Kimball saves the topic of fact tables for last, as the earlier foundational work should be understood before proceeding. The first step outlined by Kimball in this article is to declare the grain of the fact table record. The grain is part of the description for the fact table record in the system. Whether this is the individual sale item at a store scanner or the daily timesheet entry for a service system, the grain is a key requirement to define the fact table. Once the grain is declared it becomes possible to associate dimensions to the fact table. In the Kimball methodology, the grain is declared before we begin to identify the dimensions for which the facts are measures.

Kimball then describes the three types of fact tables. Transaction grained, such as sales or timesheet entry, periodic snapshot, for areas such as account balance at a bank; and accumulating fact tables, which are for systems that capture multiple events for a process such as long running events and wait times. Transaction grained tables are usually additive in nature such as total sales, billable hours, or are designed to count events. Periodic snapshots are intended for situations such as account balances at month end or store warehouse inventory levels. Lastly, an accumulating snapshot is for situations such as a surgical event at a hospital that measures wait times. Surgical events frequently begin with a patient referral to a specialist and may contain other events such as examinations and tests, diagnosis, decision, booking, and the date of surgery. In this type of fact table what is frequently measured is wait time or duration of a business process. Another example of such a system is one employed for ambulance dispatch which also measures efficiency but at a much reduced scale.

3.2.1.3 Additional articles

Other notable works of Kimball includes those on ETL [15] such as “The 38 subsystems of ETL” which details the individual components or subsystems of a successful data warehouse. This article defines each of the components and is important for understanding the complexity of building a good system as more than 70% of the work in building a data warehouse involves these components. One example used, is gaining a better understanding of the replication of data and information from a source system into a data warehouse. The simplistic understanding that a data warehouse makes a “Copy” of the source data and the actuality of using change data capture to identify and only copy changed data are quite different. Version control, backup and recovery, security, error handling, data quality management, metadata management, dimension builders, aggregation builders, and surrogate key management are just a few examples of the complexity of this subject. Kimball lists each system and provides a definition for each of them; but, does not explore the subject to as great a depth as in his books.

These subsystems were later refined and categorized into four categories and thirty-four subsystems in a subsequent article written by Robert Becker [16] of the Kimball Group. The four categories include one that focuses on the extraction of data from source systems which includes three subsystems. A second category made up of five subsystems deals with value added components such as cleaning, data quality, and conforming dimensions. A third category of thirteen subsystems deals with delivering data into the final business intelligence layer and includes components such as slowly changing dimensions. The final , fourth category also contains thirteen subsystems which are dedicated to the management of a production data warehouse environment and are made up of areas such as backup and recovery, load scheduling, metadata management, and related components. This article lists many of the same components as Kimball's original work but the inclusion of a category structure is very beneficial to understanding the components.

There have also been numerous articles on "Real Time Data Warehousing" [27, 28, 29]. This area has been described in the literature over a considerable time. Each of these articles notes that real time systems require a new approach to the extraction, transformation, and loading of data. There is also a great deal of confusion as to what the term "real time" implies. Conceptually, a real time data warehouse is one that receives and transforms information into its target schema on a continual basis with very low latency. The traditional approach of overnight batch processing to load data once a day must change to a new architecture that processes information on a continual basis. Different methods to data extraction such as source database log mining or message based architectures are described. The limitation in these articles is that the process involves either a simple target database structure with minimal transformation or no transformation with the target being a copy of the source system in its native database structure for the purpose of operational reporting.

Another area found while researching was the concepts of Active Data Warehousing [26]. This involves leveraging data warehousing architecture and a business rules process to implement operational business changes when a situation or trigger occurs. Examples of this, are when a threshold for sales is not reached or when the volume of product returns grows to a certain threshold. Some of these can occur at near real time. The concepts involved are similar to those presented here in that a rules engine processes rules to identify a situation in star schema data (missed sales quota) and trigger an alert.

One article, by Costa et al., examines Parallel Processing of a star schema [30]; providing a good review of how data warehouse star schema queries scale out in a parallel database environment. This article reviewed the architecture of how a star schema is partitioned across multiple servers and how queries are then rewritten across multiple server nodes, with results being returned and merged on a controlling server. It goes on to discuss the scalability limitations involved, and suggests an alternative of further partitioning or denormalization of the fact table. The architecture of partitioning the fact table alone with full dimension tables on each node or alternate partitioning architectures is discussed. Also suggested is the concept of denormalizing the star schema into a flat structure. Ultimately the one statement that is most applicable is that "*Query processing can be improved by reducing the amount of data that each node has to process*". Not explored in this article is the concept of not partitioning the dimension tables, but locating them on one or more central nodes and then partitioning the fact table on multiple nodes. This offers the benefit of minimizing data on each of the query nodes with a fact table query based on dimensional key values; although in any approach, the size of the tables (Dimensions and Fact) and the partitioning choices play a key factor.

Another interesting article was SAMSTAR [31]. This article looked at the automated generation of a star schema from a source system entity relationship diagram. The concept is feasible and has been suggested by Kimball and Ross [32] as part of their lecture series on Dimensional Modelling. Riazate also suggested something similar [44] in his article on Matching Star Schemas. According to Ross, one of the areas to focus attention on when examining a source database system for inclusion in a data warehouse was cross reference tables. More precisely, those tables that lie at the intersection of multiple reference tables, especially those that contain additional attributes such as dates or numeric columns. An example is a Hospital Encounter which will relate to a hospital location, a patient, attending physician, diagnosis, and several other reference areas. Another example is a sales item which will relate to a product, customer, sales location, and also contain attributes for sales date or sales amount.

It is reasonable that a semi-automated approach to the design of a star schema could be possible. The unfortunate thing, is that such a tool would be entirely dependent on the quality of the source system data structure and the usage of that system which may differ from original design. This is frequently an issue with many systems and makes this approach less practical in implementation. Star Schema design is also not the major cost aspect in the development of a data warehouse. Still, such a tool combined with data profiling could be beneficial.

Multiple articles [36, 37, 38] related to Data Warehousing in Healthcare were also examined. Although not applicable to this thesis, they provided some interesting design concepts. Blechner's article [36] on a clinical research data warehouse and semantic information had some good design concepts, including the use of coding standards such as SNOWMED, LOIN, or HL7 CDA. However, there were some aspects that indicate a lack of understanding of some of the details of dimensional modelling. A parent child relationship within a fact table is unheard-of and indicates an issue with the granularity of the fact records themselves or the definition of the fact table.

Murphy's article on optimizing healthcare research data warehouse design [38] performed an evaluation of the use of a health research database at the Massachusetts General Hospital and how the majority of the needs could be met through the use of a dimensional model or star schema. Where this failed, for a small percentage of reports, was when searching for textual elements. It does show how star schema design was recognized as an effective solution for a health research data warehouse, but the need for a solution to the semantic and contextual elements [37] still exist.

Of particular interest was the article by Darmont and Olivier [37]. Although some aspects such as the storage of complex observations in the form of images, binary information, or other documents are currently not practical, other observations made in this article are worth noting. Darmont looked to advances in OLAP as required to relate some information. Specifically, he states "*Users must be able to display and exploit such relationships manually (which is currently the case) or automatically (here, we anticipate the advances of multimedia mining and the development of advanced OLAP operators)*". Darmont attempts to model complex relationships between observations, facts, and documents in what he describes as a "*Fuzzier Fact*" composed of multiple entities. This type of complexity is a challenge that normal data warehouse and OLAP technologies are not suited to deal with. However, this could be met through the abstraction of these relationships as proposed here. It is not possible to interrelate star schemas freely at variable levels, such as with a fuzzier fact, as the relationships can be too complex.

Other articles have demonstrated this and failed to correctly [39] relate fact tables in drill across functionality. The use and ability to combine result sets in SQL has existed in the standards for multiple years with functionality such as union or intersect. Yet it's usage is not understood and attempts to relate fact tables through inner joins on dimensions is often performed with incorrect results. Abello's article is one such example, although he does indicate that the only way for drill across to work is for the inner joins on the dimension tables to have a one-to-one relationship at the aggregate level. This is not true, it is possible but requires multiple SQL passes as Kimball demonstrates this in his article on the integrated data warehouse [7].

In Kimball's article on the logical foundation of dimensional modelling [41], he reviews the concepts of dimensional modelling as logical groupings of information. This description differs from his articles on the development of dimensional models, which describe the methodology used to identify the information utilized by a business process and how to fit that information into a dimensional model. The dimensional model contains all of the information required by a business process, with fact tables in third normal form and dimension tables in second normal form. The relationships in the data are still preserved, but take on a different form and often employ repeating values in the dimensions. The key element that this article brings forward is that the designers of a dimensional model must understand the data they are working with.

Ross and Kimball wrote an article on fact tables and the aggregation or consolidation of their values [43]. It is frequently considered best practice to capture fact table records at the lowest grain possible. The article examines whether this is really necessary. If a business captures several different facts (such as man hours, phone calls, estimated hours, or patient visits) at the same grain and with the same information, it is not necessary for this information to be captured in separate fact tables. In a sales order system individual line items are not necessary when they can be captured as quantity sold. Do estimated hours for a project need to be captured at a daily level or is weekly adequate? The important element is capturing the information at a level that the business requires in order to meet its measures.

Another article written by Knoblock and Szekely [66], looks at the world of big data and the problem of data integration. Although this article does not pertain to dimensional modelling, the processes and the problems it discusses have been aspects of Date Warehouse operations for decades. The work and data discussed are simplistic compared to the structures and data involved here, with no consideration of efficient structures, such as star schemas. This article does discuss how problems in data integration remain an issue. Schema level matching is shown with many examples, while record level matching remains a challenge and area of research. In short, the problems of data integration remain.

Bizer, Heath, and Berners-Lee work [67] is interesting in its exploration of linked data. At a root level, it discusses the same principles as those of relational databases. Explicitly defined, machine readable, linkages between data are relationships. The basis of the Resource Data Framework (RDF) triples are subject – predicate – object and is a major aspect of relational database design. Choosing to model these relationships outside of the fixed entity structure of an entity relationship diagram, as done in the semantic web, provides a solution to the problem of interrelating Star schema structures. In an RDF triple the subject and object are unique within the web of things as is the predicate of how they relate. The basis of linked Data is the unique identification and the ability to define new ways of relating (predicate) the data (subject and objects).

3.2.1.4 Criticisms of Dimensional Modelling and the Kimball Approach

There has been some criticism of Kimball's approach to Data Warehousing. Several of these criticisms have been refuted by Kimball in articles, such as his piece on total cost of ownership [19]. The premise of this article is to dispute those who look at the cost of a data warehouse in terms of labor and materials to instead ask the simple question: "What is the cost of a bad decision?" Kimball explores several aspects which he considers to be the true costs such as not having the information to make decisions, lacking partnership between IT and end users, or missing explicit end user focused cognitive and conceptual models. However, there is one element that Kimball lists that contradicts many of his other articles. He states that "the corporate data model is a waste of time that delays the data warehouse" and reasons that it is frequently an ideal model and not reflective of the true enterprise data. Well this may be true, it can also be argued that enterprise architecture and a corporate data model can help a business visualize its data assets which will assist in the development of a data warehouse and is at the core of his other work. A corporate data model and any other sources of metadata can help apply both business context and a framework for meeting the information needs of an organization.

An interesting view of business intelligence was put forth by R. Davenport in his technical whitepaper on ETL vs ELT [20]. Although this can be attributed to differences in semantics, there were several points in Davenport's paper that are valid. The basis of this paper is that what most data warehouses do is not Extract - Transform – Load (ETL) but rather Extract – Load – Transform (ELT). Davenport contends that the major effort in a data warehouse is extracting information from the source systems and loading it into the data warehouse. This is perhaps a valid statement in small systems where a single fact table can be completely populated in a single process or one that has a specific focus but not in an enterprise level integrated data warehouse.

Davenport defines the output of the ELT process as having a very narrowly defined goal; in essence this maybe a specific report or business requirement. This definition may apply to a very simple star schema but does not reflect the scope of an integrated data warehouse. Kimball defines a star schema fact table as one designed to measure a business process at a specific grain, but this does not imply a very narrowly defined goal. The design of a star schema is driven by the business requirements and can be narrow or broad based on those requirements. The scope of a business and the systems that support it cannot be summarized in a single star schema.

Davenport does have some valid points. A data warehouse and included star schemas are not a fixed deliverable, they are a system that requires support and ongoing development. The complexities of ETL development and those of data warehouse support and enhancement are not simple. A star schema can be difficult and time consuming to enhance which Kimball frequently does not adequately recognize. This can limit them when adapting to rapidly changing business requirements; however, Davenport fails to recognize that the supporting business systems would require significantly more effort than those required for the data warehouse.

It is a repeating theme in articles critical of the Kimball methodology that development effort and support of a data warehouse is too costly as stated in the HealthCatalyst literature and in Kimball's own articles [19, 25]. The effort to rapidly enhance star schemas is recognized as a limitation in star schema development by multiple authors; even Kimball [49] has suggested initial development using views, or other means, to produce results more rapidly. This requirement is supported in the methodologies proposed here as enhancing a star schema is essentially about relating information to it in a flexible and rapid manor.

In Chisholm's article "The Twin Towers of BI Babel," Chisholm does not directly criticize Dimensional modelling or the Kimball approach, but may point to one of the possible reasons for the failure of many data warehouse projects. It is not the methodology but rather a failure in information architecture.

Information systems development involves the abstraction of business processes and information into data structures and information systems. Chisholm describes the development of a data warehouse as the reversal of this process. The term Abstraction Translation Paradigm is used to describe this. While the concepts are interesting and insightful, there is no true solution offered. Nevertheless, recognizing the issues is helpful to the Business Intelligence solution designer.

Haughey's articles [51, 52] offer a good review of dimensional modelling, its application, and some advanced modelling problems, but fails to make valid criticisms of dimensional models. While it is true that the correct application of dimensional modelling or data modelling in general is foundational to the success of any systems project, the issues that Haughey attributes to dimensional modelling are failures in design and not technique or methodology. He criticizes dimensional modellers as being short-sighted and limited by an adherence to a narrow vision in their design of business solutions. He points to specific examples where alternate models to standard dimensional approaches performed better, but does not expand those examples to explain why they offered better performance. A specific example of one of Haughey's criticisms of dimensional modelling is a rapidly changing dimension. In one of his cases, records in a specific dimension change rapidly due to a single attribute. As described by Kimball, this attribute should not be included in this dimension. Haughey does not describe addressing this by moving the attribute to a separate dimension or a junk dimension but describes how a specific BI solution product does not support moving the attribute into the fact table. The criticism is unwarranted as this is clearly a modelling issue and not a limitation in dimensional modelling specifically. Haughey also explains a situation where normalized data warehouse structures performed as well as a dimensional model but does not provide information on the structure or hardware utilized in these tests.

An interesting set of whitepapers, articles, and presentations that advocates a different approach to data warehousing are published by Health Catalyst, a company that specializes in Health Sector Data Warehousing [22, 23, 24, 25]. This approach does not specifically criticize dimensional modelling rather it is critical of both Inmon's and Kimball's approaches to data warehousing. It sees both of these approaches as requiring extensive effort and advocates a third approach as being less labour intensive.

The approach put forth by Health Catalyst is identified by them as Late Binding and is not to be confused with the programming term of the same name. This approach advocates the minimal transformation of data as part of the data warehouse. In essence, the source system data and information is kept in its source relational structure and then analytical reporting structures are created from the source tables. This transformation is not in the traditional sense of ETL but rather in a late/transformational step similar to a database view or often employs a reporting or OLAP platform tool directly from source. They do advocate a star schema design and the reuse of objects but not the full transformational effort of a data warehouse or the Integrated Data Warehouse as defined by Kimball. The approach is more similar to an Inmon approach with individual data marts but lacks the foundational layer of Inmon's corporate data warehouse.

Health Catalyst targets the health sector and their product offering includes an initial "Start-up" platform and structures based on the major application providers. This offers an attractive opportunity to jump start a data warehouse environment with a turnkey solution based on the existing business systems within an organization. Although this approach is viable and certainly can produce deliverables in a timely manner, it has several limitations. The provided solution is reliant on the source system for its base data, relationships, and the quality of that system. When performing analytics in this approach, no effort is provided in the areas of data quality or validation. It is a rapid development approach which is actually supported by Kimball as a method for prototyping purposes. The approach is limited and is highly dependent on the quality of the source data structure which is not transformed into a relational corporate modal as with an Inmon approach. The source data structure is kept in its original form with the referential integrity (if existing), and all the potential problems from the source system present. The data in the structure is also dependant on the way that the source systems are employed and any custom use of fields or processes can be problematic. It is difficult to transform all information with the simple methods available with this approach. It is also not possible to merge data sets or provide complex answers to questions, which can be done with an Integrated Data Warehouse. That being said, they do advocate for a more complex transformation structure following a Kimball approach when required for situations such as a merged patient dimension and other conformed dimensions.

The major deficiency in this methodology is that it does not account for a higher level architected approach to data and information. An Inmon approach builds an enterprise data warehouse as a corporate relational model with all of the information transformed into this structure. From this separate data marts are created for the purposes of rapid reporting. A Kimball approach looks to an integrated data warehouse with conformed dimensions that ultimately are an architected solution supporting an enterprise view of information. The late binding approach sacrifices an enterprise level to information in favor of a model that supports rapid application development.

Kimball refuted many more of the criticisms of dimensional modelling in 2008 [25]. The first myth disputed was that a dimensional model could be missing key relationships that exist in the business system. However, all of the data required to both define and measure the processes for that business would be in the dimensional model. The fact table would be in third normal form and the dimensions second normal form. The relationship might be hierarchical in the dimension rather than represented in a data model; but it, and the associated data, would still be present.

The second criticism Kimball disputed was that dimensional models are not extensible and cannot easily accommodate rapidly changing business processes and that dimensional models could have negative impacts on data integration. Kimball refuted this by demonstrating how easily dimensional models can be extended and describing several methods that can be applied to do so. It is noted that changing a dimensional model can be accomplished much more rapidly than transforming a business application.

Next were statements that a dimensional model is built to address a specific business need and that it captures how people monitor their business, whereas a relational model mimics business processes. Kimball states that a dimensional model is not built for a specific business need or a single report, it is designed to capture a business measurement or event at a detail level. The format and structure of a star schema has no dependency on a final report. It is possible that this statement comes from confusion about the data warehouse marketplace and differences between a Kimball and an Inmon approach, where data marts are created to address a specific business need or report, and the concept that a star schema is a data mart. As identified by Kimball, a star schema captures business events at a specific level of granularity that measures that business process. It is not designed to meet the needs of a specific department and is designed from an enterprise perspective.

Another disputed myth is that in a dimensional model usually only one date is associated with time. This likely comes from the design concept of one dimension representing dates and a second (if required) representing time of day. Although these only exist as singular tables, they can be represented multiple times in a single fact table. The previously described accumulating snapshot that monitors a long running process is a very good example of this. Multiple dates are captured in a fact table representing the milestone events during that process.

Finally, Kimball disputes the argument that a relational model is preferred over a dimensional enterprise data model because it needs to capture data at a very low level of granularity. This is perhaps the most difficult to understand myth that Kimball disputes here. The Kimball methodology teaches that data should be captured at the lowest level of granularity possible. This allows the data to be aggregated in any combination or level required. It is possible that individuals misinterpret the aggregated results of a star schema query with the underlying granular data stored in the star schema and believe the data is pre-aggregated and individual records are lost which is not the case.

The majority of the criticism of the Kimball methodology and dimensional modelling can be attributed to a lack of understanding of the approach. The only criticisms that have a valid basis are those that are critical of the development effort in building and maintaining an enterprise data warehouse and those regarding relating information in a dimensional model. These criticisms do not take into account the far greater costs involved in building and maintaining the source computer systems. The demand for increasingly short cycles to IT development effort is not unique to data warehousing. The criticisms regarding the relational aspects of a dimensional model are also largely unjust, although there is a grain of truth to some of these criticisms. Kimball himself has stated that it is not possible to interrelate fact tables. This is largely true, as it is difficult and certainly beyond the simple referential integrity aspects of a relational database. There are; however, situations where we need to go beyond this and interrelate star schemas.

Chapter 4. Design Methods and Process

As previously described, Data Warehousing and Business Intelligence have become a mainstay for organizations to facilitate meeting their business information needs [32, 33, 34]. The demand for information from these systems is steadily growing in all areas [49]. HealthCare, as an example [22, 24, 36, 37], is increasingly demanding sophisticated answers to complex questions and other information needs in ever shorter timespans. Information requests and measures such as Data Quality, Patient Cohort's, and Complex Observations across multiple subject areas are the norm. The variety of data available and the statistical analysis being performed are major factors in driving this. They also represent significant risks as correctly defining relationships is critical to acquiring the proper information for this analysis.

In order to meet the complex requirements of relating information across subject areas in a timely fashion, new methods must be developed to go beyond the functionality of a Kimball data warehouse. The Integrated Data warehouse is the first step towards meeting these needs as has been documented in the literature for a number of years [7, 8, 32, 33, 34]. Not surprisingly, very few organizations accomplish a true integrated data warehouse as it requires both vision and commitment, whereas quick wins offer an easier path.

Even for those who do achieve an integrated data warehouse it is not enough. It is the foundation and must be recognized as only the first step. Businesses need to go further and we need the skills to accomplish this. The previous example on how to perform a drill across and the articles by Kimball show how to accomplish queries across multiple subject areas and how complex this can be. Not surprisingly there are articles that try to accomplish this in a single SQL query which will likely not return the correct results [39]. This is an inherent danger in the area of business intelligence and complex business systems. Without the proper skills and subject area knowledge, the risk of providing incorrect information and making bad decisions is always present. As the complexity increases, so does the risk.

Ultimately, extending our existing dimensional models to encompass new information is the main issue and the solution lies at the heart of information and our underlying technology.

4.1 Relationships

Relationships between database entities lie at the heart of our business systems and technology. Our source systems are all about relating information, the technology we use is relational database management systems, our dimensional models are based on relationships.

Given these facts, to extend our dimensional models we need to focus on how our information is related. The techniques described here present methods to both relate information to an existing dimensional model end interrelate different models. This will allow us to rapidly develop new business insights with minimal effort.

The difficulty, is that the relationships employed within a data warehouse star schema and databases in general are too simplistic to express the complexity required. As stated, the foundation for associating information and subject areas within our Business Intelligence systems is the Integrated Data Warehouse. We have a choice to build increasingly complex models and reports/data extraction routines or we can find an alternative by focusing on relationships within our data. The processes described here choose the latter option and explain how to build on the integrated data warehouse by abstracting our relationships. This abstraction will allow us to extend our dimensional models with new information as well as interrelate them. Thus, we can maintain subject specific star schemas and extend them as required.

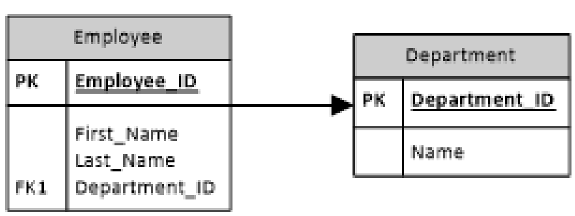
The method employed to abstract the relationships between star schemas is based on the functionality to relate information within the semantic web [73]. This is done through the Resource Description Framework (RDF) which offers a solution for extending dimensional models. RDF has features that can facilitate data relationships even when the underlying schemas are different. The development of this ability uses unique identifiers as are used in RDF triplets and creates a predicate or relationship object to establish the linkage between the subject and objects as described below.

4.2 Defining a Unique Key

One of the primary building blocks of a relational database is the relationships between database tables. This functionality is dependent on the ability to uniquely identify a record in a database table. The identifying column is known as a primary key and its values are unique within the table. Relationships are formed by creating a column in a second table known as a foreign key that, by definition, points to the primary key of the first table. Values for the foreign key column are restricted to those that occur in the primary key table column and the primary key table cannot have a record removed while its value exists in a dependent foreign key column. However values may exist that have no dependent foreign key.

An example of a simple relationship between two tables representing employees and departments is shown in figure below in Figure 4.1. In this case the unique identifier for the department is the column Department_ID. The foreign key is in the Employee table and uses the same name as the primary key. This represents a typical foreign key relationship in a relational database. It is a part of the physical database structure with integrity enforced by the database software. By definition the relationship is expressed as a simple equation of a=b.

Figure 4.1: Employee Department Relationship



Employee.Department_id = Department.Department_id

Not surprisingly, the secret to abstracting our relationships is our use of unique keys. However, to abstract relationships we need to go beyond the unique key within a table and the related columns in secondary tables to a unique key across all tables similar to unique URI addresses in the internet and the semantic web. If we do not do this, we remain within the constrained environment of referential integrity and relational structures defined by primary and foreign keys within a relational database. By employing a unique key across all our tables, relationships can be modelled outside our table structures. This is similar to RDF triplets with the relationship definition or predicate provided by a SQL statement that can be expanded beyond the simple equation of a=b shown in the previous employee/department example.

In this approach, the relationship is defined by a SQL statement and is represented in the results of that statement. These results can take multiple forms allowing us to relate information to an entity and thereby extend that entity or a join condition between two tables which enables us to form new relationships without implementing physical structure changes. This is not employed for all situations, but does allow us to extend the information in our dimensional models and join our tables and star schemas together outside of our fixed database structures in order to interrelate them in different ways.

We will look at how this is accomplished in the following sections. First we will look at how we can extend our star schemas by adding additional information to them. Then we will look at how we can interrelate our star schemas.

4.3 Extending Our Information

Extending information in our fact and dimension tables is much simpler conceptually then interrelating our star schemas. First we will look at a basic binary extension to a table which will identify records that match a certain condition. We will use the unique key to then relate information to our tables. This process involves four steps shown below.

4.3.1 Binary extension

Step One: Definition

Every table that is important to our dimensional models must have their records uniquely identified. Tables that do not require this (Date Dimension, Junk Dimensions, Time Dimension, etc.) do not require this unique key as information will not be related to them.

Figure 4.2: Typical Data Warehouse table

DataWarehouse_Table	
PK	<u>Unique_DataWarehouse_key</u>
	...

(Where DataWarehouse_Table is any required source, dimension, or fact table and Unique_DataWarehouse_Key is a unique key across all tables)

Step Two: Association

Once we have a unique Key across all required tables, we can create abstract association rules to extend the table with additional information.

Figure 4.3: Typical Data Warehouse table and Association Rule

DataWarehouse_Table	
PK	**<u>Unique_DataWarehouse_key</u>**
	...

Association Rule	
PK	**<u>Rule ID</u>**
	Sql Statement ...

As previously described, these rules are simple SQL statements. For example, if we have a dimension table of patients for a health authority and we needed to select a group of those patients based on their registration in a given program, this would require a SQL statement such as below.

Rule ID: Patient_Cohort_1

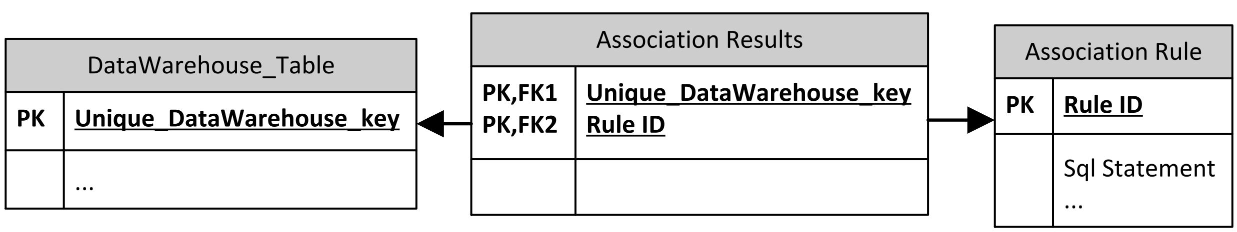
```
Select dp.Unique_DataWarehouse_Key
        from d_patient as dp
                        inner join criteria _table1 on condition 1
                        inner join Program_Criteria _table2 on condition 2
        Where criteria_condition
```

This statement is relatively simple as it only identifies those patients that form a particular cohort. This can be potentially more complex if additional constraints, such as date or other demographic criteria, is required.

Step Three: Rule Processing

Processing of the SQL rules is performed on a regular basis to capture results as shown below. This is accomplished through a simple automated process that will retrieve all of the records in the association rule table and process the individual SQL statements.

Figure 4.4: Association Results Table structure



In this situation for the patient cohort rule, we are simply capturing the unique data warehouse key for the patient dimension record and the rule identifier as shown in table 4.1.

Table 4.1: Association Results

Unique Date Warehouse Key	Rule Identifier
1243	Patient_Cohort_1
709234	Patient_Cohort_1
3456997	Patient_Cohort_1
9775298746	Patient_Cohort_1

Each of the above key values represents a row in the patient dimension for an individual that satisfies the cohort rule. At this point we have captured all of the necessary information to meet our business requirements and extend our dimensional model. All that remains is to populate the results into our star schemas.

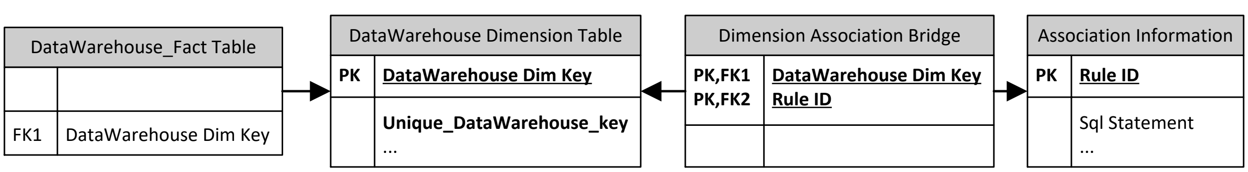
Step Four: Star Schema Population

The final step in our binary extension is to populate our star schema tables to relate our rule and the captured information to our dimension and fact tables. For our dimension tables, this can be easily accomplished with database views; but this requires additional work if it pertains to a fact table. Both structures are described below and identified as dimension or fact table bridge structures.

Permanently extending a dimension to capture the associated information in the dimension table would be optimal in a long term situation. The structure and techniques employed here allow the rapid development of this information with minimal effort and can also be used to capture information of a transitory nature.

Dimension Bridge Table

Figure 4.5: Dimension Association Structure



The structure above creates a dimension table for our association rules. This table uses a bridge or cross reference table between any of our standard dimensions and the association rule dimension. The table is derived from the association results table and can be expressed as a simple view from the results. To continue our previous example a view definition for our patient dimension is given below.

```
Create View patient_association_bridge as Select ar.rule_id, dp.Patient_dim_Key from d_patient as dp
                    inner join association_results as ar
```

on ar.unique_datawarehouse_key = dp.unique_datawarehouse_key

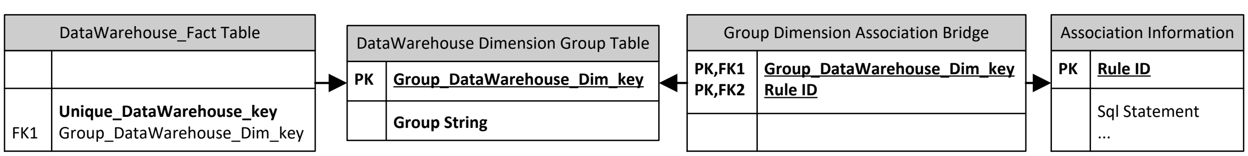
Part of the basis for this view definition is that the unique data warehouse key will only join to the table that contains it. Although the association results table may contain keys from multiple data warehouse tables, because the patient unique data warehouse key is unique across all tables, only those from the patient dimension will appear.

In the example above, the unique data warehouse key is not used as the primary key of the dimension table. Depending on the size of a data warehouse, our unique keys could grow to a large size (an eight byte integer is recommended) and our star schema keys are kept to a minimal size (2 bytes if possible) for performance reasons. A two byte reference key in a data warehouse fact table would take one quarter of the size and allow better performance from both a read and a comparison function when dealing with extremely large data volumes.

Fact Table

The primary difference between the dimension association and fact association is the necessity of building a bridge table structure with a group table related to the fact. This structure is identical to the dimension structure above, but is processed differently.

Figure 4.6: Fact Table Bridge Structure



In the case where we relate data to a fact table, we must build the structure to relate the association information dimension to the data warehouse fact table. This involves the generation of two tables a group table representing the existing combinations of association rules applied to the fact table and a bridge table that serves as a cross reference between the group table and the association rules.

This structure is explained in Kimball's article on the subject [32, 33, 46]. On the surface, it may seem unnecessarily complex; but in reality is higher performing then a cross reference between the fact table and the association table. This is due to the significant reduction in the number of records possible by representing distinct combinations instead of all cross reference records.

The difficult portion here is the group table. This table represents the combination of association rules that any particular record satisfies. It is populated through a custom developed function that concatenates the rule identifiers together to form a group string of all rule combinations that occurs. The development of this function is dependent on the database platform and is represented here as STRGROUP(). The SQL to create the grouping is provided below.

```
Create view Group_strings as
        Select Unique_DataWarehouse_Key, STRGROUP(rule_id) as StrGroup
        from AssociationResults
        group by Unique_DataWarehouse_Key
```

With this statement we now have the group string and the unique key it relates to. All that remains, is to populate the cross reference table between the rules and the group string from the view below.

```
Create view GroupDimensionBridge as
        Select Atab.StrGroup, Ar.Rule_ID
        (Select min(Unique_DataWarehouse_Key) as MinKey, StrGroup
        from Group_strings) as Atab inner join AssociationResults as Ar
              on Atab.MinKey=Ar.Unique_DataWarehouse_Key
```

The population of all tables is now complete. The structure is populated and each row that has been identified by any of our association rules is now associated with that rule and can be aggregated or filtered by that rule as required.

4.3.2 Value Extension

Step One: Definition

Step one in the process of associating a value to a data warehouse star schema is the same as before. We simply identify each record in our data warehouse uniquely.

Figure 4.7: Typical Data Warehouse table

DataWarehouse_Table	
PK	**Unique_DataWarehouse_key**
	...

Step Two: Association

As before, once we have a unique key across all required tables, we then create abstract association rules to extend the table with additional information. Unlike the binary association from our first example, in this situation we define a SQL statement that returns the unique data warehouse key for the table and the value we want to associate to that record.

Figure 4.8: Association Value Rule Table

DataWarehouse_Table	
PK	**Unique_DataWarehouse_key**
	...

Association Rule	
PK	**Rule ID**
	Sql Statement ...

As an example, if we have a fact table for product sales and wanted to associate the sales volume for the previous year to a product returns table, we could do this with the following select statement.

Rule ID: SalesVolume_Returns1

```
Select fr.Unique_DataWarehouse_Key, fs.TotalSales
        from f_Returns as fr
                inner join D_Product as dp on dp.product_dim_key=fr.product_dim_key
                inner join (select product_dim_key, sum(sales_units) as TotalSales from F_Sales
```

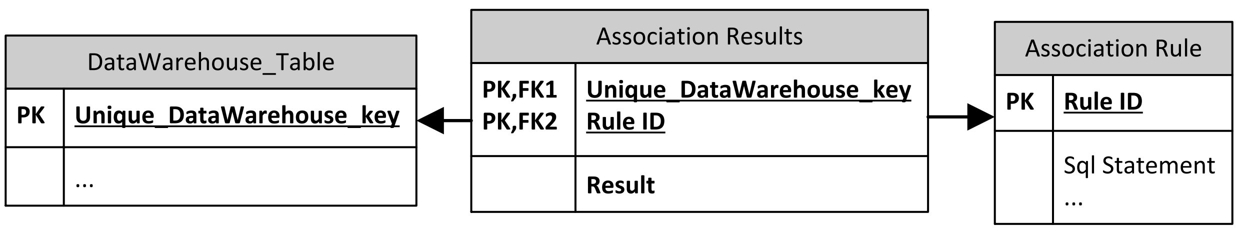
```
where sales_date>dateadd (year,getdate(),-1)
group by product_dim_key) as fs
        on fs.product_dim_key = dp.product_dim_key
```

The only difference between this and the previous example is that it returns a value along with the associated key.

Step Three: Rule Processing

Processing of the SQL rules is still performed on a regular basis. In this case, the results table stores the unique data warehouse key, the association rule identifier, and the result value. It is noted that the result value can be numeric or another data type.

Figure 4.9: Association by Value Results



In this example, for the sales volume rule, a possible group of values is provided below in Table 4.2.

Table 4.2: Association by Value Results

Unique Date Warehouse Key	Value	Rule Identifier
1243	1200	SalesVolume_Returns1
709234	1000	SalesVolume_Returns1
3456997	1300	SalesVolume_Returns1
9775298746	4200	SalesVolume_Returns1

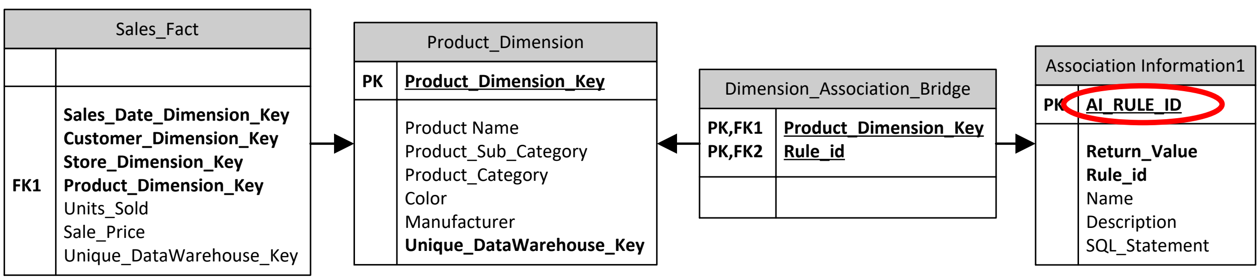
Each of the above key values represents a row in the product returns fact table and the value is the total number of units sold. At this point, we have captured all of the necessary information to meet our business requirement to associate our returns to the total sales volume. All that remains is to populate the results into our star schemas.

Step Four: Star Schema Population

The final step is to populate our star schema tables to relate our rule and the captured information to our dimension or fact tables. This process is similar to that employed for the binary extension, but differs in the association rule table. The cross reference table maintains the same structure but requires different processing to identify the association rule table record.

Dimension Bridge Table

Figure 4.10: Dimension by value table structure



As we can see, the dimension structure is the same as before with the exception that the association information table now includes a return value. Records in the product dimension are now cross referenced to the association information table based on Rule and value.

Shown in Table 4.3 is an example of the data in our association table.

Table 4.3: Association by Value table data

Rule_ID	Name	Description	SQL	Return Value
1	Product Returns	Units returned for 365 days	Select Product …	1200
2	Product Returns	Units returned for 365 days	Select Product …	1000
3	Product Returns	Units returned for 365 days	Select Product …	1300
4	Product Returns	Units returned for 365 days	Select Product …	4200

Only a single rule is shown in the table. In bringing these values into a star schema solution the records would most likely be organized in a hierarchy based on name and return value. Users would simply employ drill down techniques to show the required detail.

The query to populate the association information table is shown below. It is a simple join between our association rules and our results table. The key value (AI_Rule_ID) is a new sequential value representing the distinct combination of the Rule_ID and the returned value. This becomes the primary key of the new information table.

```
Select distinct ar.rule_id, ar.name, ar.sql_statement, ar.description, ares.result
        from AssociationRule as ar
        inner join AssociationResults as ares on ar.rule_id=ares.rule_id
```

The cross reference table is also populated from a simple SQL statement.

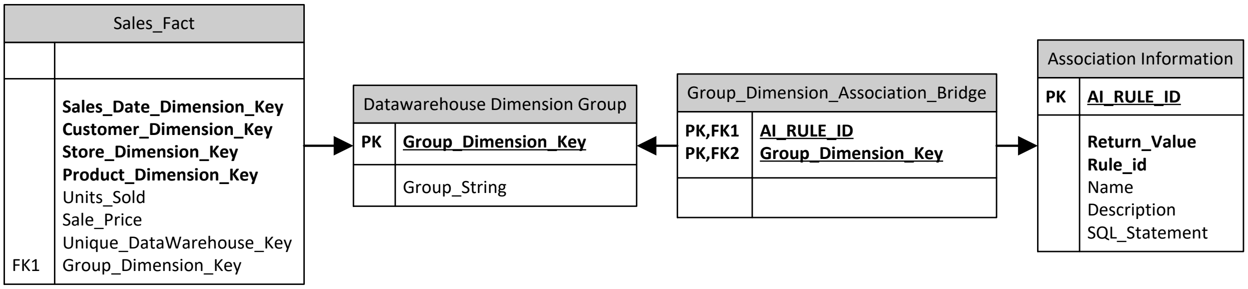
```
Select distinct ai.AI_Rule_id, ares.Unique_Datawarehouse_Key
        from AssociationInformation as ai
        inner join AssociationResults as ares on ai.rule_id=ares.rule_id
                                    and ares.Result=ai.return_value
```

These queries complete the population of the star schema tables and form the basis for the new association information dimension.

Fact Table

The fact table relationship is also modified in the same way as the dimension bridge. The change is the presence of the return value in the association information table; as was shown in the dimension bridge above. The remainder of the processing would be the same as before

Figure 4.11: Fact by Value bridge table structure



Associating to the fact table does provide additional functionality that was not present when associating to the product dimension above. We now have access to the additional information in the fact table that was not present with the product dimension alone. In relating to our product dimension, we selected the volume of returns for that product for the previous year from the current date. This is because the product dimension does not have a temporal aspect. When associating to the sales fact, we could use the sales date to look at returns prior to that date such as below.

```
Select fs.Unique_DataWarehouse_Key,
            (select sum(fr.quantity_returned) from f_returns as fr
                Where fr. product_dim_key =fs. product_dim_key
                     And fr.return_date between dateadd (year,fs.sales_date,-0.5) and
                                     dateadd (year,fs.sales_date,0.5) as Return_quantity
                              from F_Sales as fs
```

The association information table and bridge table would be populated in the same manner as the dimension bridge table. The difference for the fact table population is the need for a group table which is populated in a similar manner as before.

4.4 Associating our Star Schemas

Associating our dimensional models to each other is no more difficult than associating information to them. The complexity is in understanding the concepts represented in the relationship and the legitimacy of that relationship. It is strongly advised that whenever possible the user should restrict the usage of associating information to a value based option rather than establishing a full relationship. This will likely meet the majority of requests and will require the least effort. Significant misunderstandings in incorrectly relating information could result if relationships are established incorrectly or misinterpreted.

An example of complexity and understanding:

In a Healthcare data warehouse, we could have a dimensional model representing health assessments of our residential care patients. These would be routinely captured and measure a patient's health, the health of the patient population, and the quality of care the population receives. We also have a dimensional model used to capture emergency encounters at hospital emergency rooms. Developing an association between these two subject areas with the techniques below can be easily accomplished but what does it mean.

1) We could be looking at a patient's assessment before his emergency visit to determine a reason for the encounter or retrospectively assess the risk of an emergency encounter.

2) Alternatively we might be looking at a subsequent patient assessment to determine the impact of that event and results of possible interventions.

3) We might need to do both in an attempt to evaluate treatment options.

The complexity of these relationships is immediate. The relationship is obviously uni-directional and has distinct meaning. This is true in any database relationship but is much more complex here, as we could be relating entire star schemas and we must place context and meaning around that relationship.

Still, there is enormous potential value to this functionality and it is described as an option. A thorough understanding of the database structures and the meaning of the relationship is essential if we want to build a structure that is legitimate and correctly represents the information to the user.

Step One: Definition

As before the first step in relationships is to identify every record uniquely.

Figure 4.12: Typical Data Warehouse table

DataWarehouse_Table	
PK	**<u>Unique_DataWarehouse_key</u>**
	...

(Where DataWarehouse_Table is any required source, dimension, or fact table and Unique_DataWarehouse_Key is a unique key across all tables)

Step Two: Association

Once we have a unique key across all of our tables we can then create the abstract association rule to define the relationship and capture it. These rules are simple SQL statements that identify the source and destination unique data warehouse keys.

Figure 4.13: Data Warehouse table and Association Rule

DataWarehouse_Table	
PK	**<u>Unique_DataWarehouse_key</u>**
	...

Association Rule	
PK	**<u>Rule ID</u>**
	Sql Statement ...

As an example, in the provision of home care in the province of British Columbia, home care medical assessments are required on an annual basis. If we wanted to assess the provision of service hours and professional care visits by the medical assessment of that patient we could easily do this by selecting the most resent assessment prior to the visit.

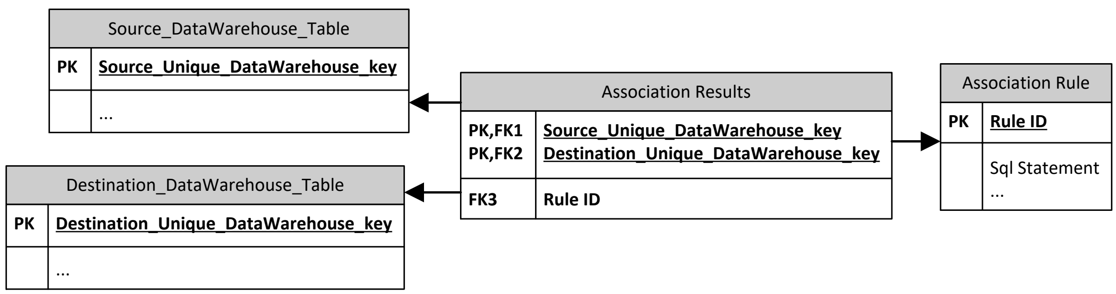
Rule ID: Service_Assessment

```
Select srv. Unique_DataWarehouse_Key,
        ( select top 1 asm. Unique_DataWarehouse_Key from f_assessment as asm
                Where asm.patient_key=srv.patient_key and asm.date_key<srv.date_key
                Order by asm.date_key desc) as assessment_ Unique_DataWarehouse_Key
        from f_service as srv
```

Step Three: Rule Processing

Processing of the SQL rules is performed on a regular basis to capture results as shown below. The difference here is that we capture two keys and define the direction of the relationship.

Figure 4.14: Association Rule results structure



As previously described these rules are simple SQL statements. Developing and executing them, to capturing their results is quite simple. The majority of the effort is in bringing the results into our data warehouse and reporting environment.

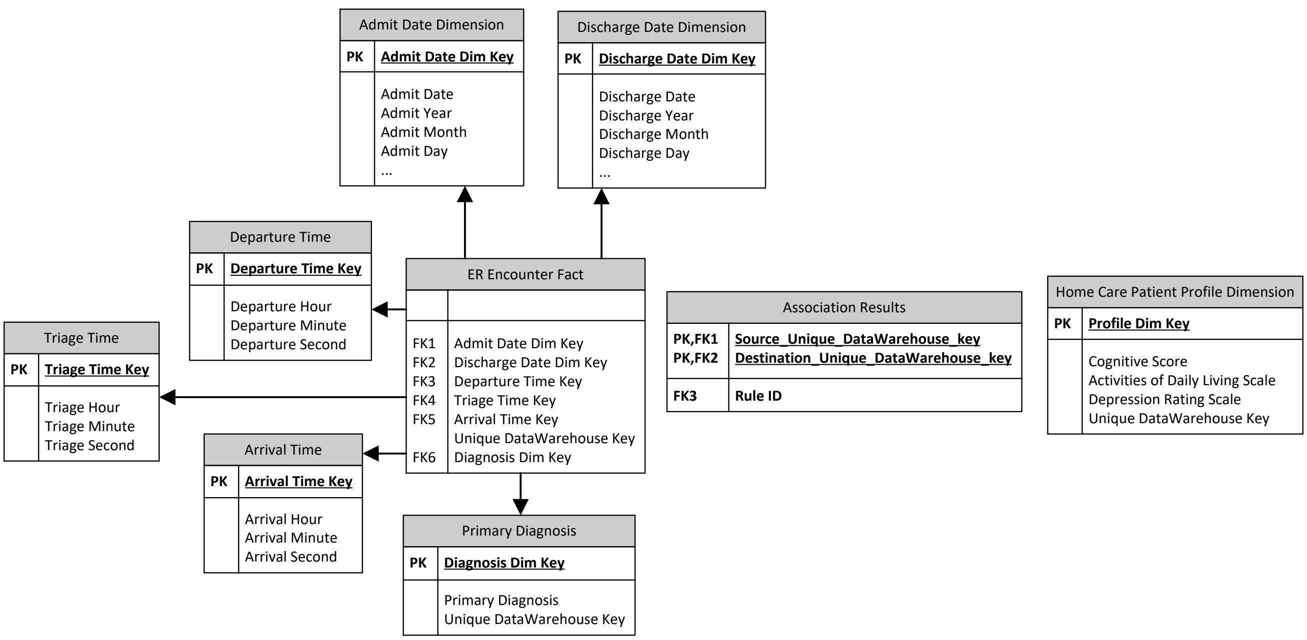
Step Four: Results

The implementation of the captured relationship into the environment is much more complex than associating information to a star schema object. It can be thought of as building a role playing dimension/view only in this circumstance it is a complete role playing star schema. How this information is reported is largely dependent on the tools being used. Some tools, such as Microsoft Tabular services, do not support many to many relationships and would require a different approach than the ones used here. This will be illustrated by two examples.

Example 1: Referencing a Dimension

In Figure 4.15 we have a source star schema representing Emergency Encounters and we want to access the health profile dimension associated with the patient assessment. The SQL rule to capture this relationship is to select the unique identifier from the emergency encounter and the unique identifier from the patient profile dimension where the patient assessment date is the most recent assessment prior to the emergency encounter. This is very similar to the example from step two.

Figure 4.15: Dimension Association example



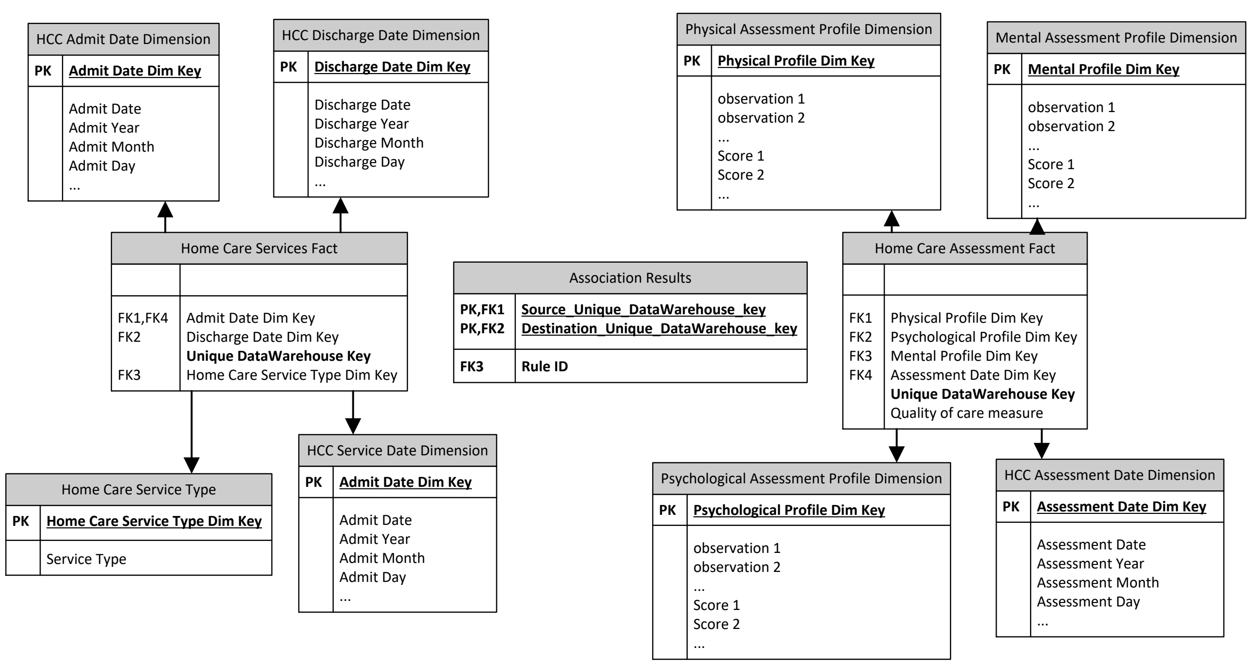
All of the information is available and shown in the above model. At this point, the only remaining issue is how to express this in our Business Intelligence tool. This is where the greatest effort will be required, and is dependent on the tool being used. If metadata is captured for the data warehouse and employed as part of the development of associations, this could be automated.

Example 2: referencing a star schema

Perhaps the most difficult situation to understand in a data warehouse would be the interrelationship of complete star schemas. The functionality goes beyond what is possible in an integrated data warehouse with drill across and is more about providing dimensional information from one star schema to another. Certainly, a fact tables measures would likely be of little value at a record level when joining two fact table records. What is involved is the joining of two separate fact tables so that the dimensions in one star schema can be used in the other. In essence, it is a bridge table relationship taken to an extreme level.

In the example in Figure 4.16, the dimensions for our patient assessment are brought into our home care patient services. This extends our example query from step two and shows how it allows us to look at our costs for providing care in terms of the population health. This could be used to predictively model the cost of patient care based on predicted population health.

Figure 4.16: Fact Association example



The insight into our costs in providing services should be obvious. Our home care services and patient assessment star schemas are both relatively simple star schemas. Neither needs to be expanded in terms of complexity, although our BI tool will require some effort. We have a simple solution that is able to offer significant increases to our abilities with minimal effort.

Any star schemas can be interrelated where the relationship can be defined as a SQL expression. Our star schemas can remain as uniform subject area constructs representing singular business functions or information subject areas. These subject areas can then be interrelated as required without the need to build new, larger constructs.

Chapter 5. Source Data Sets

As proposed, four separate data sets were requested from the Canadian Institute for Health Information (CIHI). This data was for the period of 2011 to 2013 and represented a single Health Authority with a geographic area of over 58,500 square kilometers and a population of over one million. These four data sets represented Emergency Services in the form of the National Ambulatory Care Reporting System (NACRS), Acute Care in the form of the Discharge Abstract Database (DAD), Home Services in the form of the Home Care Reporting System (HCRS), and Continuing Care in the form of the Continuing Care Reporting System (CCRS). These Four data sets represent four major areas for the provision of health services in Canada. By developing methods for interrelating disparate data sets such as these, it is hoped that new insights into the provision of health services and patient care can be explored.

5.1 NACRS

Field Level details for the supplied NACRS data set are provided in Appendix 1.

The NACRS contains data for Day Surgery, Outpatient / Community based Clinics, and Emergency Departments. Only Emergency Care visit data was available for the health authority chosen for this study. In addition to this; no Diagnosis, Intervention, Provider, Consultant, or any of the Emergency Level One optional fields were provided or populated.

The focus of this data set is the measurement of emergency volumes and wait times. As previously noted, no Diagnosis or intervention data was available which limits the information in this data set. Several Date and Time fields were provided which allows the calculation of wait times and the length of stay in Emergency.

5.2 Discharge Abstract Database

Field Level details for the supplied DAD data set are provided in Appendix 2.

The DAD captures the administrative, clinical, and demographic data for hospital discharges. CIHI restricts access to this data to specific fields and other information necessary for individual studies. For the purposes of this study, the data requested was the Administrative, Basic Demographic, Diagnosis, and Intervention data. The Case Mix Group and provider information were not supplied.

The focus of this data set will be to measure acute care volumes for the selected Health Authority. As no booking or referral date information is available, this data set does not have the necessary information to calculate hospital wait times.

5.3 Home Care Reporting System

Field Level details for the supplied HCRS data set are provided in Appendix 3.

The HCRS captures information related to the provision of health services primarily in the home environment, although services may be provided in different settings. This can be for short term care for patients recovering from surgery, long term care, support to those with chronic conditions, as well as other specialized programs such as palliative or rehabilitation.

The Home Care data provided for the study consisted of the Full Assessment and Episode information. This includes all observations, basic service volume information, scales, quality indicators, client assessment protocols, and disease diagnosis or problem conditions. Over 390 separate information fields were provided as part of the extract. No medication information was provided as part of the home care data set, but all other portions of the home care assessment data was included.

5.4 Continuing Care Reporting System

Field Level details for the supplied CCRS data set are provided in Appendix 4.

The CCRS captures information related to the provision of health services for individuals receiving continuing care services in a hospital or long term care homes in Canada. It is based on the InterRAI Minimum Data Set (MDS) version 2.0 which is a standardized medical assessment originally developed by a consortium of researches and was mandated by the 1987 U.S. Nursing Home Reform Act and is used for care planning and management of continuing care services.

The Continuing Care data provided for the Study consisted of the Assessment and Episode information. This includes all observations, Scales, Quality Indicators, Client Assessment Protocols, and disease diagnosis or problem conditions. The significant difference with the CCRS data over the HCRS data is the requirements for regular patient assessments. This allows the monitoring and evaluation of changes in patient health over time.

As with the HCRS data set no medication information was included in the extract.

Chapter 6. Dimensional Models Design and Build

Each of the four data sets was used to develop separate dimensional models. The development process followed the Kimball methodology and the resulting Star Schemas are shown below using Kimball's four question design process. The data models are based on the received data from CIHI and do not reflect additional information such as provider, intervention, or diagnosis unless such information was provided.

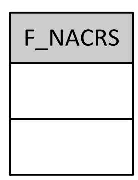
The use of conformed dimensions is noted in each section with a full description of these dimensions following the individual star schemas.

6.1 NACRS Emergency Care Star Schema.

1) What is the Business Process?

The business process is the provision of services for a hospital Emergency department.

Figure 6.1: Emergency Services Fact Table



2) How do we measure the business process?

The Emergency Care department measures are focused on service volumes and wait times. The volume of patients visiting the emergency departments, patients admitted into acute care, wait times, and the length of stay all represent measures for our emergency care star schema shown in Figure 6.2.

Figure 6.2: Emergency Fact Table with Measures

F_NACRS
LOS_HOURS WAIT_TIME_TO_PHYSICIAN_INITIAL_ASSESSMENT WAIT_TIME_TO_INPATIENT

3) What is the grain of the fact table?

Each record in our fact table represents a single patient registration in the Emergency Department. Even patients who leave Emergency without seeing a physician are included. If a patient leaves and returns, creating a second registration, it will be represented as two separate emergency visits.

4) What do we measure by?

The primary information used in the analysis of emergency encounters are Dates, Times, Facility, Patient, Visit Disposition, Triage Level, and whether the Patient was admitted via ambulance. Conformed dimensions that are essential to an integrated data warehouse are noted.

The Date Dimension (Conformed)

When examining these requirements in more detail it can be seen that for dates we are interested in multiple values reflecting Registration, Triage, Physician Assessment, Disposition, and when the Patient left the Emergency Department. This information yields our first conformed dimensions for dates and is shown in the Figure 6.3.

The Date dimension is the most common dimension in a Kimball Data Warehouse, though it is often misunderstood. The benefit of the Date Dimension is not in the date value; rather, in the metadata or information related to that date. Asides from natural hierarchies, such as Year – Month – Day, we may also have attributes such as day of the week, statutory holidays that affect pay scales, or lunar phase. It is also common to have different string values to reflect different date formats for standardized reporting.

Figure 6.3: The Date Dimension

D_Date	
PK	**Date_Dim_Key**
	Date_Value Date_DD_MMM_YYYY Date_YYYY_MM_DD Date_Sequence Calendar_Year Calendar_Quarter_ID Calendar_Quarter Calendar_Yr_Qtr Calendar_Yr_Qtr_ID Month_ID Month Month_Number Month_Short_Name Day_of_Week Day_of_Week_Short_Name Day_of_Week_Sort Weekend_Indicator Lunar_Day Lunar_Phase Lunar_Phase_Sort Day_of_Month

The Time Dimension (Conformed)

The next dimension we require is one reflecting time. Similar to the Date Dimension our Emergency Subject Area is interested in multiple time values reflecting the same Information requirements as in the date dimension. These include Time of Registration, Triage, Physician Assessment, Disposition, and when the Patient left the Emergency Department. This information yields our second conformed dimensions for times, and is shown in the Figure 6.4.

Figure 6.4: The Time Dimension

D_Time	
PK	**Time_Dim_Key**
	Hour_of_day Minute_of_Hour Period Twelve_Hour_Display Twentyfour_Hour_Display Time_of_Day Part_Of_Day

The Patient Dimension (Conformed)

The Patient Dimension, as with date and time, is a conformed dimension and reflects all patients in our system. As only minimal patient information is available in our supplied data sets. The dimension attributes are limited to the patient year of birth, gender, unique health care number, and the province that issued the health care number.

Figure 6.5: The Patient Dimension

D_Patient	
PK	**Patient_DIM_KEY**
	hcn_mbun HCN_Province Birth_Year Gender dw_seq_id

The Facility Dimension (Conformed)

The fourth Conformed Dimension is for our Hospitals or Care Facilities. As with our patient information only minimal information for facilities was provided with only one field in our data extracts for a single scrambled identifier representing the facility. This is reflected in our dimension which consists of a fictitious name and facility type based on the data set associated with the extract data.

Figure 6.6: Facility Dimension

D_Facility	
PK	**Facility_Dim_Key**
	System_Facility_Care_Number Facility_Name Facility_Type

The NACRS Flag Dimension

The last dimension in our Subject area is for NACRS low cardinality fields. This is a construct in the Kimball Methodology known as a “Junk Dimension,” also referred to as a flag dimension in this study. It is common practice in the Kimball approach for both performance and design simplicity to group low cardinality fields together in a single dimension, where each distinct combination of these values that exist in the source data is stored as a separate row. In our NACRS data set we have three low cardinality flags for Patient Admitted via ambulance (4 possible values), Triage Level (7 Possible Values), and Visit Disposition (13 Possible values). When we combine these fields into a single table it is found that the distinct combinations of these fields in the data is half of the product of the frequency of possible values for the fields. This produces a significantly smaller analysis structure in a typical On-Line Analytical Processing (OLAP) solution demonstrating its performance advantage.

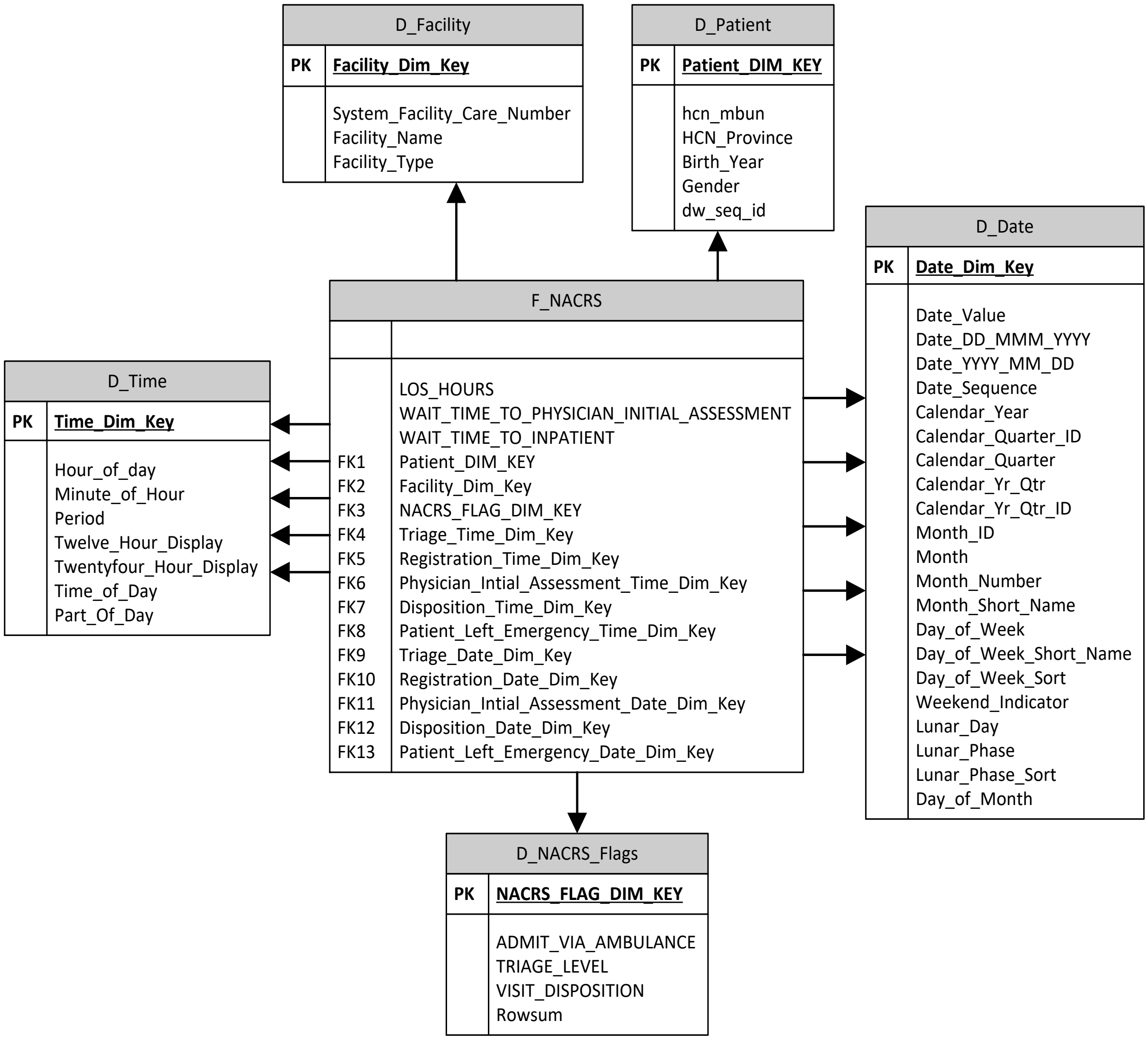
Figure 6.7: The Emergency Services Flags dimension

D_NACRS_Flags	
PK	**NACRS_FLAG_DIM_KEY**
	ADMIT_VIA_AMBULANCE TRIAGE_LEVEL VISIT_DISPOSITION Rowsum

Final NACRS Solution

When our fact table is combined with these dimensions, it creates the NACRS star schema solution for the emergency services area. This star schema allows us to report on emergency visits in detail or perform aggregate reporting by any of our dimension tables or fields. We can examine the count of Emergency records, the minimum length of stay, or average wait time and analyze this data by facility, year, or any combination of the available attributes.

Figure 6.8: Emergency Services Star Schema



As an example of the reporting possible against our NACRS star schema and the information it contains, Table 6.1 looks at the total number of emergency encounters for our solution by emergency facility and triage level at night. It is seen that facilities three and five are the busiest emergency departments, well facility six closes down its services in the evening.

Table 6.1: Night time Emergency Encounter count by Triage Level and Facility

Registration Time	Night

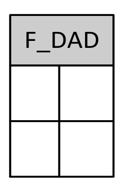
Encounter Count	Triage Level							
Facility	0	1	2	3	4	5	9	Grand Total
Emergency_Facility_1		164	5670	17801	10449	431	63	34578
Emergency_Facility_2	216	270	5329	14964	11264	860	26	32929
Emergency_Facility_3	193	722	10928	29260	18887	1856	10	61856
Emergency_Facility_4	5		39	297	1860	116		2317
Emergency_Facility_5	143	239	5637	22207	19135	2946	37	50344
Emergency_Facility_6			1				1	2
Emergency_Facility_7		104	2955	6986	7243	338	195	17821
Grand Total	**557**	**1499**	**30559**	**91515**	**68838**	**6547**	**332**	**199847**

6.2 Discharge Abstract Database Star Schema.

1) What is the business process?

The business process is the provision of services for hospital acute or alternate level of care.

Figure 6.9: Discharge Abstract Fact Table



2) How do we measure the business process?

The DAD measures are focused on service volumes and length of stay. The volumes of patients provided with hospital services, how many are admitted via emergency, how long a patient is in acute care, and what are the volumes and length of stay in alternate level of care represent the measures for our DAD star schema.

Figure 6.10: Discharge Abstract Fact Table with Measures:

F_DAD	
	Total_Lenth_Of_Stay_Days Acute_Length_of_Stay_Days Alternate_Level_of_Care_Length_of_Stay_Days Total_Special_Care_Unit_Length_Of_Stay_Hours Emergency_Department_Wait_Time_Hours Emergency_Department_Wait_Time_Minutes

3) What is the grain of the fact table?

Each record in our fact table represents a single patient discharge abstract record. If a patient is discharged and admitted to the same facility later that day, it will be represented as two separate abstract records.

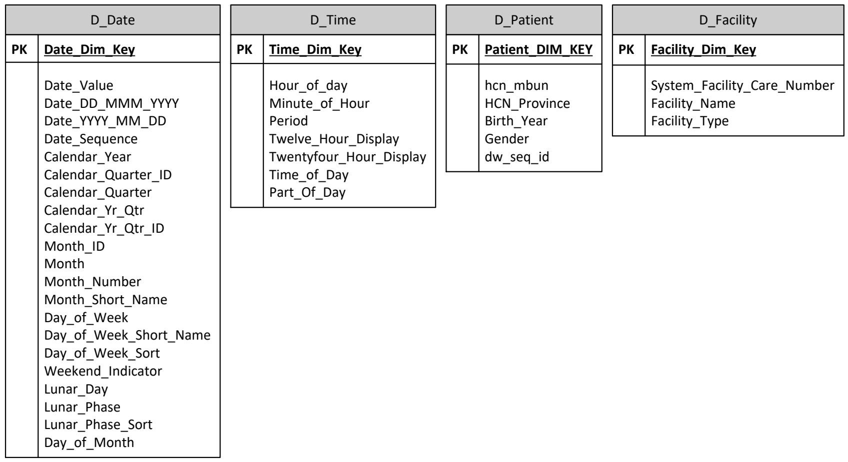
4) What do we measure by?

The primary information used in analysing abstract records are dates, times, patient, facility, patient service, diagnosis, and intervention. Additional information identifying if the abstract was for an emergency, the admission category, the type of entry, the discharge disposition, whether admission was via ambulance, or if the record was for a readmission are also provided.

Available Conformed Dimension

Four separate conformed dimensions were used representing Date, Time, Patient, and Facility. These were previously explained in our NACRS section and are shown in Figure 6.11.

Figure 6.11: Conformed Dimensions used with Discharge Abstract Star Schema

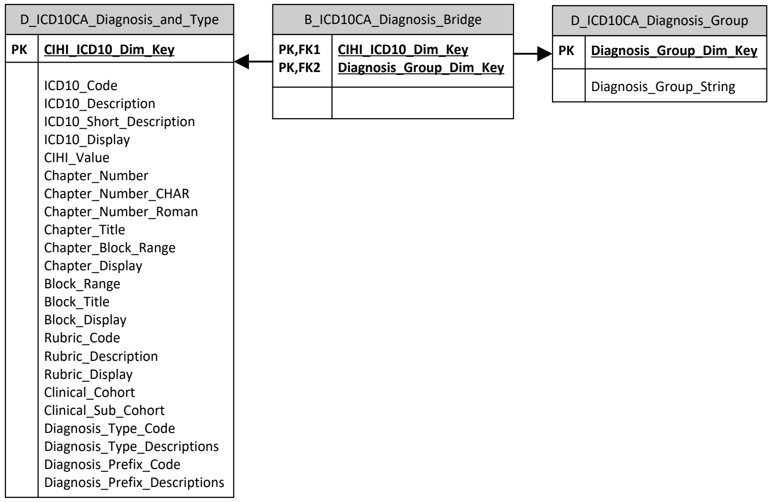


Diagnosis Dimension (Conformed)

The Diagnosis dimension was based on CIHI ICD-10-CA and is a conformed dimension. This dimension is shared with the Home Care and Residential Care assessments which identify additional diagnosis codes for patients using ICD-10-CA. Diagnosis is also de-normalized in that additional fields were added to represent clinical cohort, diagnosis type, and the diagnosis prefix code. There is also a natural hierarchy including chapter, block, rubric, and diagnosis which can provide additional functionality for aggregations.

Within the DAD data more than one diagnosis code might be provided as a patient may have multiple conditions that need to be documented. This means that the diagnosis area requires a bridge structure as in Figure 6.12 to accommodate the many-to-many relationship inherent in the data. As previously explained, this structure does not employ a standard relational cross reference table, but has individual records for each existing combination of diagnosis codes supplied in the data. The group table is frequently referred to as a helper table as it is not required when using SQL to query the table as the relational group key exist in the group and bridge tables, the group table assists in visualizing the table structure and also is required with certain query tools due to product dependencies.

Figure 6.12: ICD-10-CA Diagnosis Dimension Bridge Structure



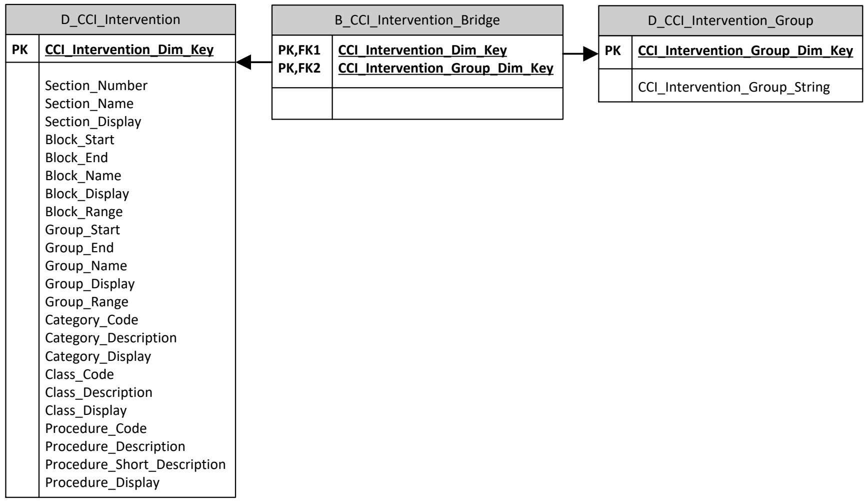
Intervention Dimension

The intervention dimension was based on the Canadian Classification of Health Interventions (CCI). It is not conformed and was not used with any other subject area although it represents an area of information that would be, if other areas such as our NACRS data included intervention information.

CCI is a classification system for use with health care procedures and is the companion classification system to ICD-10-CA. It includes a broad range of interventions including surgical, diagnostic procedures (imaging, tests, etc.), therapeutic, assessments, and counselling. As with diagnosis codes, a natural hierarchy in the form of a catalog structure exists in the data to facilitate navigation.

As multiple Interventions can be recorded as part of a DAD record a bridge structure is again required to facilitate the many-to-many relationship that exists in the data.

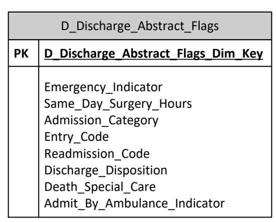
Figure 6.13: CIHI CCI Intervention Dimension Structure



Discharge Abstract Flags Dimension

The Discharge abstract data includes several low cardinality fields such as indicators for admission via ambulance, emergency, admission code, entry code, readmission, and discharge disposition. These are captured as a flag dimension shown in Figure 6.14.

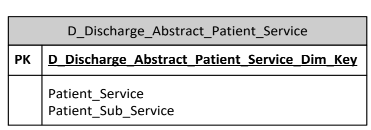
Figure 6.14: Discharge Abstract Flags Dimension



Discharge Abstract Patient Service Dimension

The final dimension captured was for patient service. This represents the main patient service and subservices provided such as general surgery, cardiology, or obstetrics.

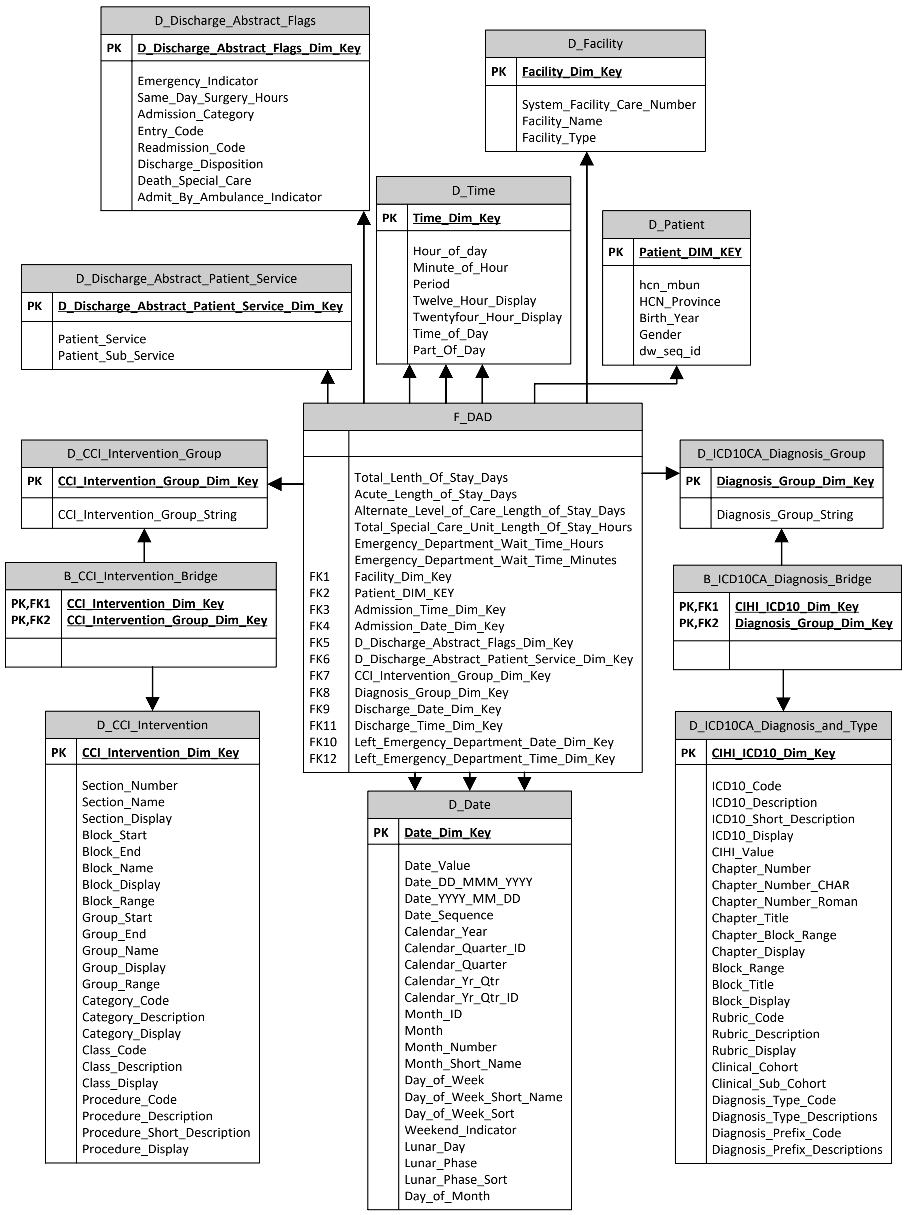
Figure 6.15: Discharge Abstract Patient Service



Final Discharge Abstract Solution

When the fact table is combined with these dimensions, it creates the DAD star schema solution. As can be seen, this structure is significantly more complex than the previous solution but provides the greatest flexibility in capturing the available information for reporting and analysis.

Figure 6.16: Discharge Abstract Star Schema.



6.3 CCRS Assessment Star Schema.

Modelling CCRS assessment data offers significant challenges when compared to the NACRS or DAD data previously discussed. This is due to the volume of information supplied in terms of the number of individual information elements. There are more than 500 distinct fields supplied as part of the CCRS assessment data. This volume of information presents problems not only in terms of modeling but in understanding and navigating. Should separate subject areas be developed that look at physical, cognitive, or psychological information or should all areas be combined? Should the grain of the fact be individual observations or at the assessment level?

Following the Kimball methodology and the business process outlined below, a design was arrived at that incorporated as much of the available data as possible at the level of the assessment. Decisions that influenced this design are provided as part of the design process.

1) What is the Business Process?

Unlike the previous subject areas the CCRS assessment data does not correspond to a direct business process; but instead, represents multiple processes at different organization levels. Specific business processes and associated work flows may exist within an organization to provide assessment services, but these are not reflected in the supplied data.

Assessment information is used for monitoring a patient's health condition, to assist in care planning for the patient, to manage service volumes, to monitor population health, and to look at the quality of care by viewing the changes in population health over time. As such, this subject area is used at multiple levels from direct patient care to strategic planning and the design needs to reflect this.

In each of these cases the measure and usage for the data is at an assessment level. For this reason the assessment was chosen as the key basis for the business process and the grain of the fact table.

Figure 6.17: CCRS Assessment Fact Table

F_CCRS_ASSESSMENT	

2) How do we measure the business process?

The CCRS assessment table has multiple measures. This includes a count of assessments, count of patients, count of service episodes, length of stay, service provision such as physical therapy, hospital stays, visits to an emergency department, visits by a physician, the number of changes to physician orders, and thirty-six separate quality indicators that look at changes in patient health. In total, eighty-five separate measure fields are included in the CCRS assessment.

It is noted that many of these measures are complex and involve both a numerator and a denominator. Others, such as service volumes for care provision or physical therapy, are not additive in nature but are statistical in that they indicate the level of care provided and not the total volume of service.

Figure 6.18: CCRS Assessment Fact Table with Measures

F_CCRS_ASSESSMENT
Length_Of_Stay
episode_id_mbun
assessment_id_mbun
PREVIOUS_AX_ID_mbun
P1BAB_MINS_SPEECH_THERAPY
P1BBB_MINS_OCCUPATION_THERAPY
P1BCB_MINS_PHYSICAL_THERAPY
P1BDB_MINS_RESPIRATORY_THERAPY
P1BEB_MINS_PSYCHO_THERAPY
P1BFB_MINS_RECREATION_THERAPY
P5_HOSPITAL_STAYS
P6_EMERGENCY_ROOM_VISITS
P7_DAYS_PHYSICIAN_VISITS
P8_DAYS_DOCTOR_ORDERS_CHANGED
QI_CAT02_D
QI_CAT02_N
QI_CNT04_D
QI_CNT04_N
QI_DRG01_D
QI_DRG01_N
QI_FAL02_D
QI_FAL02_N
QI_INF0X_D
QI_INF0X_N
QI_NUT01_D
QI_NUT01_N
QI_PAI0X_D
QI_PAI0X_N
QI_PRU05_D
QI_PRU05_N
QI_RES01_D
QI_RES01_N
QI_WGT01_D
QI_WGT01_N
QI_ADL01_D
QI_ADL01_N
QI_ADL05_D
QI_ADL05_N
QI_ADL06_D
QI_ADL06_N
QI_ADL1A_D
QI_ADL1A_N
QI_ADL5A_D
QI_ADL5A_N
QI_ADL6A_D
QI_ADL6A_N
QI_ADLD7_D
QI_ADLD7_N
QI_BEHD4_D
QI_BEHD4_N
QI_BEHI4_D
QI_BEHI4_N
QI_CNT02_D
QI_CNT02_N
QI_CNT03_D
QI_CNT03_N
QI_CNT2A_D
QI_CNT2A_N
QI_CNT3A_D
QI_CNT3A_N
QI_COG01_D
QI_COG01_N
QI_COG1A_D
QI_COG1A_N
QI_COM01_D
QI_COM01_N
QI_COM1A_D
QI_COM1A_N
QI_DEL0X_D
QI_DEL0X_N
QI_MOB01_D
QI_MOB01_N
QI_MOB1A_D
QI_MOB1A_N
QI_MOD4A_D
QI_MOD4A_N
QI_PAN01_D
QI_PAN01_N
QI_PRU06_D
QI_PRU06_N
QI_PRU09_D
QI_PRU09_N
QI_RSPX2_D
QI_RSPX2_N

3) What is the grain of the fact table?

The grain chosen for the fact table was an individual assessment. This decision was based on the measures which exist at an assessment level. Other designs, such as using individual observations, were considered; but as the measures for a patient's level of health and quality of care all exist at the level of the assessment, it provides the greatest functionality at this level.

4) What do we measure by?

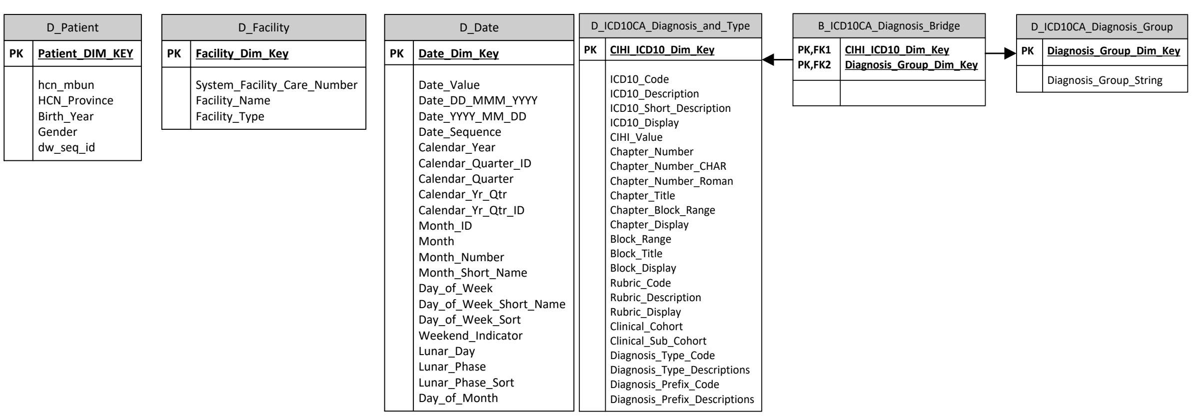
As with measures, this question presents unique challenges. Aside from the standard conformed dimensions for patient, date, and facility, there are hundreds of additional attributes to consider. As many attributes as possible were captured in order to provide the maximum functionality.

Two separate design patterns were used to include as many of the individual assessment attributes as possible. This was done through flag dimensions and bridge structures. The design structure was based on the assessment form and listed attributes in alphabetical order to allow for easy navigation and use. The key focus was organization and usability in order to provide as much functionality as possible.

Available Conformed Dimensions

Three separate conformed dimensions were used representing Date, Patient, and Facility. In addition the ICD-10-CA diagnosis bridge structure was also used. These were previously explained in our NACRS and DAD sections and are shown in Figure 6.19.

Figure 6.19: CCRS Assessment conformed dimensions



Flag Dimension Pattern

As described previously, flag dimensions are a design approach used to capture various low cardinality fields in a star schema as a single dimension. It is a simple approach that builds a single database table with a distinct combination of the individual fields as separate records.

In the case of the CCRS assessment data, we have nearly 500 separate fields including calculated scores, scales, quality indicators, and a large number of individual observations as low cardinality database fields. To capture many of these fields, multiple flag dimensions were created. These dimensions were organized based on CCRS field names and labels. Additional fields were added for name/display value and descriptions. The example table below is for the fields G2a through G3b.

Figure 6.20: CCRS Assessment dimension G2a through G3b

D_G2a_G3b_CCRS	
PK	**G2a_To_G3b_Dim_Key**
	G2A_BATHING_SELF G2A_BATHING_SELF_Name G2A_BATHING_SELF_Description G2B_BATHING_SUPPORT G2B_BATHING_SUPPORT_Name G2B_BATHING_SUPPORT_Description G3A_BALANCE_WHILE_STANDING G3A_BALANCE_WHILE_STANDING_Name G3A_BALANCE_WHILE_STANDING_Description G3B_BALANCE_WHILE_SITTING G3B_BALANCE_WHILE_SITTING_Name G3B_BALANCE_WHILE_SITTING_Description

Individual flag dimension tables were organized similar to this example. The dimension tables and columns were arranged based on alphabetical order and frequency count for distinct values of the columns. This was to achieve optimal usability to navigate the structure and locate information while providing optimal performance.

It is the number of dimension tables as well as the size and record counts within the dimension and fact tables that determines the overall performance of the solution. An optimal structure based on these variables could be calculated, but the more important aspect is usability. When this volume of information is included, the structure must be designed with a focus on usability and navigation. In order to easily locate the information required for analysis, that structure has to be organized with this focus.

Table 6.2 provides a listing of CCRS dimension tables and columns that were developed using the flag dimension pattern. In total forty-seven separate flag dimensions were created.

Table 6.2: CCRS Flag Dimension Tables

CCRS Dimension Name	CCRS Columns
D_B1_To_B4_CCRS	B1_COMATOSE, B2A_SHORT_TERM_MEMORY_OK, B2B_LONG_TERM_MEMORY_OK, B3A_CURRENT_SEASON, B3B_LOCATION_OF_OWN_ROOM, B3C_STAFF_NAMES_FACES, B3D_AWARE_IN_NURSING_HOME, B4_COGNITIVE_SKILLS
D_B5_To_B6_CCRS	B5A_EASILY_DISTRACTED, B5B_PERIODS_OF_ALT_PERCEPT, B5C_EPISODES_OF_DISORG_SPEECH, B5D_PERIODS_OF_RESTLESSNESS, B5E_PERIODS_OF_LETHARGY, B5F_MENTAL_FUNCTION_VARIES, B6_CHANGE_COGNITIVE_STATUS
D_C1_To_C7_CCRS	C1_HEARING, C2A_HEARING_AID_USED, C2B_HEARING_AID_NOT_USED, C2C_OTHER_RECEPT_COMM_TECH, C3A_SPEECH, C3B_WRITING_MESSAGES, C3C_SIGN_LANGUAGE, C3D_SIGNS_GESTURES, C3E_COMMUNICATION_BOARD, C3F_OTHER_EXPRESSION_MODE, C4_MAKING_SELF_UNDERSTOOD, C5_SPEECH_CLARITY, C6_UNDERSTANDS_OTHERS, C7_CHANGE_IN_COMMUNICATION
D_Caps_Section_One	ADL_CAP, CARDIO_RESPIRATORY_CONDITION_CAP, PAIN_CAP, PHYSICAL_RESTRAINTS_CAP, PRESSURE_ULCER_CAP, UNDERNUTRITION_CAP
D_Caps_Section_Two	ACTIVITIES_CAP, BEHAVIOUR_CAP, COGNITIVE_LOSS_CAP, COMMUNICATION_CAP, DELIRIUM_CAP, FALLS_CAP, MOOD_CAP, SOCIAL_RELATIONSHIP_CAP
D_Caps_Section_Three	APPROPRIATE_MEDICATIONS_CAP, BOWEL_CONDITIONS_CAP, DEHYDRATION_CAP, FEEDING_TUBE_CAP, NO_TRIGGERED_CAPS, URINARY_INCONTINENCE_CAP
D_CCRS_ASSESSMENT_FLAGS	AA8_ASSESSMENT_TYPE, ACTIVE_NEW_STATUS, DISCHARGE_FLAG_IND, DISCHARGE_REASON, DISCHARGE_SERVICE_TYPE, ENTRY_TYPE, EPISODE_AX_STATUS
D_D1_To_D3_CCRS	D1_VISION, D2A_SIDE_VISION_PROBLEMS, D2B_SEES_HALOS, D3_VISUAL_APPLIANCES
D_E1a_To_E1i_CCRS	E1A_NEGATIVE_STATEMENTS, E1B_REPETITIVE_QUESTIONS, E1C_REPETITIVE_VERBALIZATIONS, E1D_PERSISTENT_ANGER, E1E_SELF_DEPRECATION, E1F_EXPRESS_UNREALISTIC_FEAR, E1G_RECURRENT_STATEMENTS, E1H_REPEAT_HEALTH_COMPLAINTS, E1I_REPEAT_ANXIOUS_COMPLAINTS
D_E1j_To_E1p_CCRS	E1J_UNPLEASANT_MOOD_IN_MORNING, E1K_INSOMNIA, E1L_SAD_FACIAL_EXPRESSION, E1M_CRYING, E1N_REPEAT_PHYSICAL_MOVEMENTS, E1O_WITHDRAWAL_FROM_ACTIVITIES, E1P_REDUCED_SOCIAL_INTERACTION
D_E2_To_E4ba_CCRS	E2_MOOD_PERSISTENCE, E3_CHANGE_IN_MOOD, E4AA_WANDERING_FREQ, E4AB_WANDERING_ALTER, E4BA_VERBAL_ABUSE_FREQ, E4BB_VERBAL_ABUSE_ALTER
D_E4ca_To_E5_CCRS	E4CA_PHYSICAL_ABUSE_FREQ, E4CB_PHYSICAL_ABUSE_ALTER, E4DA_DISRUPTIVE_FREQ, E4DB_DISRUPTIVE_ALTER, E4EA_RESISTS_CARE_FREQ, E4EB_RESISTS_CARE_ALTER, E5_CHANGE_IN_BEHAVIOUR_SYMPTOM
D_F1a_To_F2b_CCRS	F1A_EASY_INTERACT_W_OTHER, F1B_EASY_PLANNED_ACTIVITY, F1C_EASY_SELF_INITIATE_ACTIVTY, F1D_ESTABLISH_OWN_GOALS, F1E_PURSUES_INVOLVEMENT, F1F_ACCEPTS_INVITATIONS, F2A_CONFLICT_W_STAFF, F2B_UNHAPPY_W_ROOMMATE
D_F2c_To_F3c_CCRS	F2C_UNHAPPY_W_OTHER_RESIDENTS, F2D_CONFLICT_W_FAMILY, F2E_NO_CONTACT_W_FAMILY, F2F_RECENT_LOSS_FAMILY, F2G_ADJUST_TO_ROUTINE_CHNG, F3A_IDENTIFY_PAST_ROLES, F3B_SAD_OVER_LOST_ROLES, F3C_PERCEIVES_DIFF_ROUTINE

D_G1aa_To_G1cb_CCRS	G1AA_BED_MOBILITY_SELF, G1AB_BED_MOBILITY_SUPPORT, G1BA_TRANSFER_SELF, G1BB_TRANSFER_SUPPORT, G1CA_WALK_IN_ROOM_SELF, G1CB_WALK_IN_ROOM_SUPPORT
D_G1da_To_G1fb_CCRS	G1DA_WALK_IN_CORRIDOR_SELF, G1DB_WALK_IN_CORRIDOR_SUPPORT, G1EA_LOCOMOT_ON_UNIT_SELF, G1EB_LOCOMOT_ON_UNIT_SUPPORT, G1FA_LOCOMOT_OFF_UNIT_SELF, G1FB_LOCOMOT_OFF_UNIT_SUPPORT
D_G1ga_To_G1hb_CCRS	G1GA_DRESSING_SELF, G1GB_DRESSING_SUPPORT, G1HA_EATING_SELF, G1HB_EATING_SUPPORT
D_G1ia_To_G1jb_CCRS	G1IA_TOILET_USE_SELF, G1IB_TOILET_USE_SUPPORT, G1JA_PERSONAL_HYGIENE_SELF, G1JB_PERSONAL_HYGIENE_SUPPORT
D_G2a_To_G3b_CCRS	G2A_BATHING_SELF, G2B_BATHING_SUPPORT, G3A_BALANCE_WHILE_STANDING, G3B_BALANCE_WHILE_SITTING
D_G4aa_To_G4cb_CCRS	G4AA_NECK_RANGE_OF_MOTION, G4AB_NECK_VOLUNTARY_MOVEMENT, G4BA_ARM_RANGE_OF_MOTION, G4BB_ARM_VOLUNTARY_MOVEMENT, G4CA_HAND_RANGE_OF_MOTION, G4CB_HAND_VOLUNTARY_MOVEMENT
D_G4da_To_G4fb_CCRS	G4DA_LEG_RANGE_OF_MOTION, G4DB_LEG_VOLUNTARY_MOVEMENT, G4EA_FOOT_RANGE_OF_MOTION, G4EB_FOOT_VOLUNTARY_MOVEMENT, G4FA_OTHER_LTD_RANGE_OF_MOTION, G4FB_OTHER_LTD_VOLUNTARY_LOSS
D_G5a_To_G7_CCRS	G5A_CANE_WALKER, G5B_WHEELED_SELF, G5C_OTHER_PERSON_WHEELED, G5D_WHEELCHAIR_PRIMARY_LOCOMOT, G6A_BEDFAST, G6B_BED_RAILS_FOR_BED_MOBILITY, G6C_LIFTED_MANUALLY, G6D_LIFTED_MECHANICALLY, G6E_TRANSFER_AID, G7_TASK_SEGMENTATION
D_G8a_To_G9_CCRS	G8A_RES_MORE_INDEPENDENCE, G8B_STAFF_MORE_INDEPENDENCE, G8C_SLOW_PERFORMING_TASKS, G8D_AM_PM_DIFFER_ADLS, G9_CHANGE_ADL_FUNCTION
D_H1a_To_H3b_CCRS	H1A_BOWEL_CONTINENCE_SELF, H1B_BLADDER_CONTINENCE_SELF, H2A_BOWEL_ELIMINATION_REGULAR, H2B_CONSTIPATION, H2C_DIARRHEA, H2D_FECAL_IMPACTION, H3A_SCHEDULED_TOILETING_PLAN, H3B_BLADDER_RETRAINING_PROGRAM
D_H3c_To_H4_CCRS	H3C_EXTERNAL_CATHETER, H3D_INDWELLING_CATHETER, H3E_INTERMITTENT_CATHETER, H3F_DID_NOT_USE_TOILET, H3G_PADS_BRIEFS_USED, H3H_ENEMAS_IRRIGATION, H3I_OSTOMY_PRESENT, H4_CHANGE_URINARY_CONTINENCE
D_J2a_To_J3j_CCRS	J2A_PAIN_SYMPTOMS_FREQ, J2B_PAIN_SYMPTOMS_INTENSITY, J3A_BACK_PAIN, J3B_BONE_PAIN, J3C_CHEST_PAIN, J3D_HEADACHE, J3E_HIP_PAIN, J3F_INCISIONAL_PAIN, J3G_JOINT_PAIN_NOT_HIP, J3H_SOFT_TISSUE_PAIN, J3I_STOMACH_PAIN, J3J_OTHER_PAIN
D_J4a_To_J5c_CCRS	J4A_FELL_IN_PAST_30_DAYS, J4B_FELL_IN_PAST_31_180_DAYS, J4C_HIP_FRACT_IN_LAST_180_DAYS, J4D_OTHER_FRACT, J5A_CONDITION_LEAD_TO_INSTABLE, J5B_EXPERIENCING_ACUTE_EPISODE, J5C_END_STAGE_DISEASE
D_K1a_To_K5a_CCRS	K1A_CHEWING_PROBLEM, K1B_SWALLOWING_PROBLEM, K1C_MOUTH_PAIN, K3A_WEIGHT_LOSS, K3B_WEIGHT_GAIN, K4A_COMPLAINS_ABOUT_TASTE, K4B_COMPLAINS_OF_HUNGER, K4C_LEAVES_FOOD_UNEATEN, K5A_PARENTERAL_IV
D_K5b_To_K6b_CCRS	K5B_FEEDING_TUBE, K5C_MECHANIC_ALTERED_DIET, K5D_ORAL_FEEDING, K5E_THERAPEUTIC_DIET, K5F_DIETARY_SUPPLEMENT, K5G_PLATE_GUARD, K5H_PLANNED_WEIGHT_CHANGE_PROG, K6A_TOTAL_CALORIES, K6B_AVERAGE_FLUIDS
D_L1a_To_L1f_CCRS	L1A_DEBRIS_IN_MOUTH, L1B_DENTURES_REMOVE_BRIDGE, L1C_NATURAL_TEETH_LOST, L1D_BROKEN_LOOSE_TEETH, L1E_INFLAMED_GUMS, L1F_DAILY_CLEANING_TEETH
D_M1a_To_M3_CCRS	M1A_STAGE1_ULCERS, M1B_STAGE2_ULCERS, M1C_STAGE3_ULCERS, M1D_STAGE4_ULCERS, M2A_STAGE_OF_PRESSURE_ULCER, M2B_STAGE_OF_STASIS_ULCER, M3_HISTORY_OF_RESOLVED_ULCERS
D_M4a_To_M5f_CCRS	M4A_ABRASIONS_BRUISES, M4B_BURNS, M4C_OPEN_LESIONS_NOT_ULCERS, M4D_RASHES, M4E_SKIN_DESENSITIZED_TO_PAIN, M4F_SKIN_TEARS_OR_CUTS, M4G_SURGICAL_WOUNDS, M5A_RELIEVING_DEVICE_CHAIR, M5B_RELIEVING_DEVICE_BED, M5C_TURNING_PROGRAM,

	M5D_NUTRITION_INTERVENTION, M5E_ULCER_CARE, M5F_SURGICAL_WOUND_CARE
D_M5g_To_M6f_CCRS	M5G_APPLY_DRESSINGS_NOT_FEET, M5H_APPLY_OINTMENTS_NOT_FEET, M5I_OTHER_PREVENT_NOT_FEET, M6A_HAS_FOOT_PROBLEM, M6B_INFECTION_OF_FOOT, M6C_OPEN_LESIONS_ON_FOOT, M6D_NAILS_CALLUSES_TRIMMED, M6E_RECEIVED_PREVENT_FOOT_CARE, M6F_APPLY_DRESSING_FOOT
D_N1a_To_N4c_CCRS	N1A_TIME_AWAKE_MORNING, N1B_TIME_AWAKE_AFTERNOON, N1C_TIME_AWAKE_EVENING, N2_AVERAGE_TIME_ACTIVITIES, N3A_PREF_ACT_OWN_ROOM, N3B_PREF_ACT_ACTIVITY_ROOM, N3C_PREF_ACT_INSIDE, N3D_PREF_ACT_OUTSIDE, N4A_PREF_ACT_CARDS_GAMES, N4B_PREF_ACT_CRAFTS, N4C_PREF_ACT_EXERCISE
D_N4d_To_N5b_CCRS	N4D_PREF_ACT_MUSIC, N4E_PREF_ACT_READING, N4F_PREF_ACT_SPIRITUAL, N4G_PREF_ACT_TRIPS, N4H_PREF_ACT_WALKING, N4I_PREF_ACT_WATCH_TV, N4J_PREF_ACT_GARDENING, N4K_PREF_ACT_TALKING, N4L_PREF_ACT_HELP_OTHERS, N5A_PREFER_CHANGE_IN_ACTIVITY, N5B_PREFER_CHANGE_IN_INVOLV
D_O1_O2_CCRS	O1_NUM_OF_MEDICATIONS, O2_NEW_MEDICATIONS
D_O3_O4f_CCRS	O3_DAYS_INJECTIONS, O4A_DAYS_ANTIPSYCHOTIC, O4B_DAYS_ANTIANXIETY, O4C_DAYS_ANTIDEPRESSANTS, O4D_DAYS_HYPNOTIC, O4E_DAYS_DIURETIC, O4F_DAYS_ANALGESIC
D_P1aa_P1bfa_CCRS	P1AA_CHEMOTHERAPY, P1AB_DIALYSIS, P1AC_IV_MEDICATION, P1AD_INTAKE_OUTPUT, P1AE_MONITOR_MEDICAL_CONDITION, P1AF_OSTOMY_CARE, P1AG_OXYGEN_THERAPY, P1AH_RADIATION, P1AI_SUCTIONING, P1AJ_TRACHEOSTOMY, P1AK_TRANSFUSIONS, P1AL_VENTILATOR_OR_RESPIRATOR, P1AM_ALCOHOL_DRUG_PROGRAM, P1AN_ALZHEIMER_CARE_UNIT, P1AO_HOSPICE_CARE, P1AP_PAEDIATRIC_UNIT, P1AQ_RESPITE_CARE, P1AR_TRAINING_COMMUNITY_SKILLS, P1BAA_DAYS_SPEECH_THERAPY, P1BBA_DAYS_OCCUPATION_THERAPY, P1BCA_DAYS_PHYSICAL_THERAPY, P1BDA_DAYS_RESPIRATORY_THERAPY, P1BEA_DAYS_PSYCHO_THERAPY, P1BFA_DAYS_RECREATION_THERAPY
D_P2a_To_P9_CCRS	P2A_SPEC_BEHAVIOR_SYMP_PROGRAM, P2B_EVAL_BY_LICENSED_SPECIALST, P2C_GROUP_THERAPY, P2D_RES_SPECIFIC_CHNGE_ENVIRO, P2E_REORIENTATION, P4A_FULL_BED_RAILS, P4B_OTHER_TYPES_OF_RAILS, P4C_TRUNK_RESTRAINT, P4D_LIMB_RESTRAINT, P4E_CHAIR_PREVENTS_RISING, P9_ABNORMAL_LAB_VALUES
D_P3_RehabDays_CCRS	P3A_REHAB_DAYS_ROM_PASSIVE, P3B_REHAB_DAYS_ROM_ACTIVE, P3C_REHAB_DAYS_SPLINT_ASSIST, P3D_REHAB_DAYS_BED_MOBILITY, P3E_REHAB_DAYS_TRANSFER, P3F_REHAB_DAYS_WALKING, P3G_REHAB_DAYS_DRESSING, P3H_REHAB_DAYS_EATING, P3I_REHAB_DAYS_AMPUTATION, P3J_REHAB_DAYS_COMMUNICATION, P3K_REHAB_DAYS_OTHER
D_Q1a_To_R1c_CCRS	Q1A_WANTS_RETURN_TO_COMMUNITY, Q1B_SUPPORT_POSITIVE_DISCHARGE, Q1C_STAY_SHORT_DURATION, Q2_CHANGE_IN_CARE_NEEDS, R1A_RES_PARTICIPATED_ASSESS, R1B_FAMILY_PARTICIPATED_ASSESS, R1C_OTHER_PARTICIPATED_ASSESS
D_Quality_Indicators_Section_Four_CCRS	QI_CNT3A_D, QI_CNT3A_N, QI_COG01_D, QI_COG01_N, QI_COG1A_D, QI_COG1A_N, QI_COM01_D, QI_COM01_N, QI_COM1A_D, QI_COM1A_N, QI_PAN01_D, QI_PAN01_N, QI_PRU06_D, QI_PRU06_N, QI_PRU09_D, QI_PRU09_N
D_Quality_Indicators_Section_One_CCRS	QI_CAT02_D, QI_CAT02_N, QI_CNT04_D, QI_CNT04_N, QI_DRG01_D, QI_DRG01_N, QI_FAL02_D, QI_FAL02_N, QI_INF0X_D, QI_INF0X_N, QI_NUT01_D, QI_NUT01_N, QI_PAI0X_D, QI_PAI0X_N, QI_PRU05_D, QI_PRU05_N, QI_RES01_D, QI_RES01_N, QI_WGT01_D, QI_WGT01_N

D_Quality_Indicators_Section_Three_CCRS	QI_BEHD4_D, QI_BEHD4_N, QI_BEHI4_D, QI_BEHI4_N, QI_CNT02_D, QI_CNT02_N, QI_CNT03_D, QI_CNT03_N, QI_CNT2A_D, QI_CNT2A_N, QI_DEL0X_D, QI_DEL0X_N, QI_MOD4A_D, QI_MOD4A_N
D_Quality_Indicators_Section_Two_CCRS	QI_ADL01_D, QI_ADL01_N, QI_ADL05_D, QI_ADL05_N, QI_ADL06_D, QI_ADL06_N, QI_ADL1A_D, QI_ADL1A_N, QI_ADL5A_D, QI_ADL5A_N, QI_ADL6A_D, QI_ADL6A_N, QI_ADLD7_D, QI_ADLD7_N, QI_MOB01_D, QI_MOB01_N, QI_MOB1A_D, QI_MOB1A_N, QI_RSPX2_D, QI_RSPX2_N
D_Scales_Chess_Pain_PURS_ABS_CCRS	ABS, CHESS, PAIN, PURS
D_Scales_Cognitive_Depression_Social_CCRS	CPS, DRS, ISE

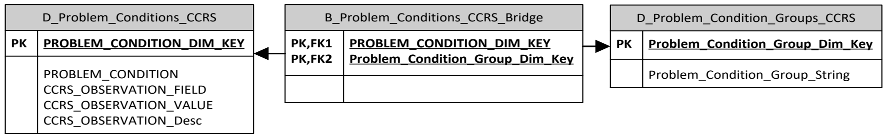
Bridge Dimension Pattern

An alternate approach employed was a bridge dimension structure. This was done for three separate sections of the CCRS assessment; Infections, Disease Diagnosis, and Problem Conditions. These three areas have multiple fields with a simple yes/no option indicating presence of the condition. The difference between this method and the flag approach is primarily that our observations are no longer individual data fields but individual records in a table. This allows for greater flexibility in that new values can be easily added but navigation can become more difficult.

Problem Condition Bridge Dimension Structure

Our first bridge structure represented in Figure 6.21 is for problem conditions. These are observations in CCRS for current conditions of concern that a patient in continuing care is experiencing. This includes indicators such as dizziness, fever, or hallucinations. Each of these fields is a simple yes/no indicator to represent the presence or absence of the condition.

Figure 6.21: Problem Conditions Dimension Bridge Structure.



The advantage of the bridge structure approach is that each of these observation questions is stored as a separate record and not represented as part of the database structure. This allows changes to this area with relative ease. Table 6.3 below shows the current values stored in the problem condition table.

Table 6.3: Problem Conditions

Problem Condition DIM KEY	Problem Condition	CCRS Observation Field
1	Weight gain or loss of 1.5 or more kilograms in last seven (7) days (3 lbs)	j1a
2	Inability to lie flat due to shortness of breath	j1b
3	Dehydrated; output exceeds input (refer to MDS User's Manual for more details)	j1c
4	Insufficient fluid; did NOT consume all/almost all liquids provided during last three (3) days	j1d
5	Delusions	j1e
6	Dizziness/Vertigo	j1f
7	Edema	j1g
8	Fever	j1h
9	Hallucinations	j1i
10	Internal bleeding	j1j
11	Recurrent lung aspirations in last 90 days	j1k
12	Shortness of breath	j1l
13	Syncope (fainting)	j1m
14	Unsteady gait	j1n
15	Vomiting	j1o

Infections Bridge Dimension Structure

Our second bridge structure, represented in Figure 6.22 is for Infections. These observations represent common infections in the Continuing Care environment including values such as Antibiotic resistant infections, Pneumonia, HIV, and several others. Data entry and use of these fields is identical to the problem conditions.

Figure 6.22: CCRS Infections Bridge Structure

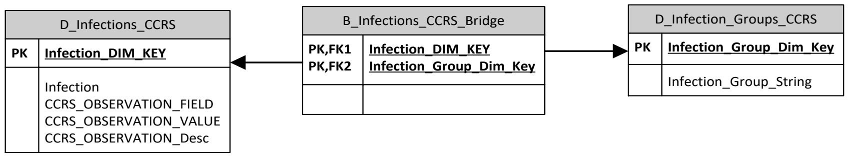


Table 6.4 below shows the infections and observation field identifiers from the CCRS specification. The functionality provided is the same as those of problem conditions.

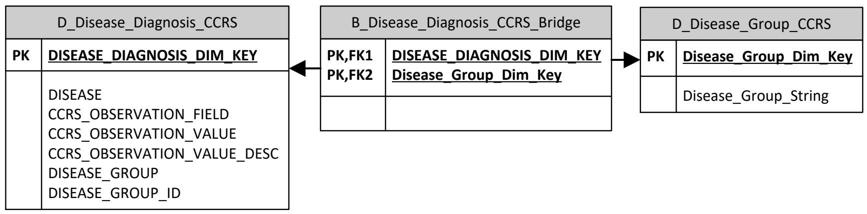
Table 6.4: CCRS Infections List.

Infection DIM KEY	Infection	CCRS Observation Field
1	Antibiotic resistant infection, e.g. Methicillin resistant staph	i2a
2	Cellulitis	i2b
3	Clostridium difficile (c. diff.)	i2c
4	Conjunctivitis	i2d
5	HIV infection	i2e
6	Pneumonia	i2f
7	Respiratory infection	i2g
8	Septicemia	i2h
9	Sexually transmitted diseases	i2i
10	Tuberculosis (active)	i2j
11	Urinary tract infection in last 30 days	i2k
12	Viral hepatitis	i2l
13	Wound infection	i2m

Diseases Bridge Dimension Structure

Our last bridge structure, represented in Figure 6.23 is for common disease conditions. The CCRS specification supports two methods to capture disease conditions. The first method is for common disease conditions and is represented here. It is a listing of forty seven observations that identify the presence of the disease using a simple present/not present response. This method also organizes the diseases into logical groupings for areas such as Neurological, Pulmonary, or Heart/Circulation. These diseases are not mapped to ICD-10-CA disease diagnosis codes or any other standard code set. A second method for entering additional diagnosis codes is based on ICD-10-CA code values and uses the conformed dimension previously developed. This represents a potential source of problems for analysis as we have two different systems employed for the same information. It would be a better solution to map these to a single conformed dimension, but this was not performed here due to expediency.

Figure 6.23: CCRS Disease Diagnosis Bridge Structure



The list of common diseases captured in CCRS is provided below in table 6.5. The diseases are organized into several groups in a natural hierarchy that is reflected in the data. This hierarchy is used to aggregate values and to filter at a group level if required.

Table 6.5: CCRS Common Disease Diagnosis

Disease Diagnosis DIM Key	Disease Group	Disease	CCRS Observation Field
1	Endocrine/Metabolic/Nutritional	Diabetes mellitus	i1a
2	Endocrine/Metabolic/Nutritional	Hyperthyroidism	i1b
3	Endocrine/Metabolic/Nutritional	Hypothyroidism	i1c
4	Heart/Circulation	Arteriosclerotic heart disease (ASHD)	i1d
5	Heart/Circulation	Cardiac dysrhythmia	i1e
6	Heart/Circulation	Congestive heart failure	i1f
7	Heart/Circulation	Deep vein thrombosis	i1g
8	Heart/Circulation	Hypertension	i1h
9	Heart/Circulation	Hypotension	i1i
10	Heart/Circulation	Peripheral vascular disease	i1j
11	Heart/Circulation	Other cardiovascular disease	i1k
12	Musculoskeletal	Arthritis	i1l
13	Musculoskeletal	Hip fracture	i1m
14	Musculoskeletal	Missing limb (e.g. amputation)	i1n
15	Musculoskeletal	Osteoporosis	i1o
16	Musculoskeletal	Pathological bone fracture	i1p
17	Neurological	Amyotropic lateral sclerosis	i1q
18	Neurological	Alzheimer's disease	i1r
19	Neurological	Aphasia	i1s
20	Neurological	Cerebral palsy	i1t
21	Neurological	Cerebrovascular accident (stroke)	i1u
22	Neurological	Dementia other than Alzheimer's disease	i1v

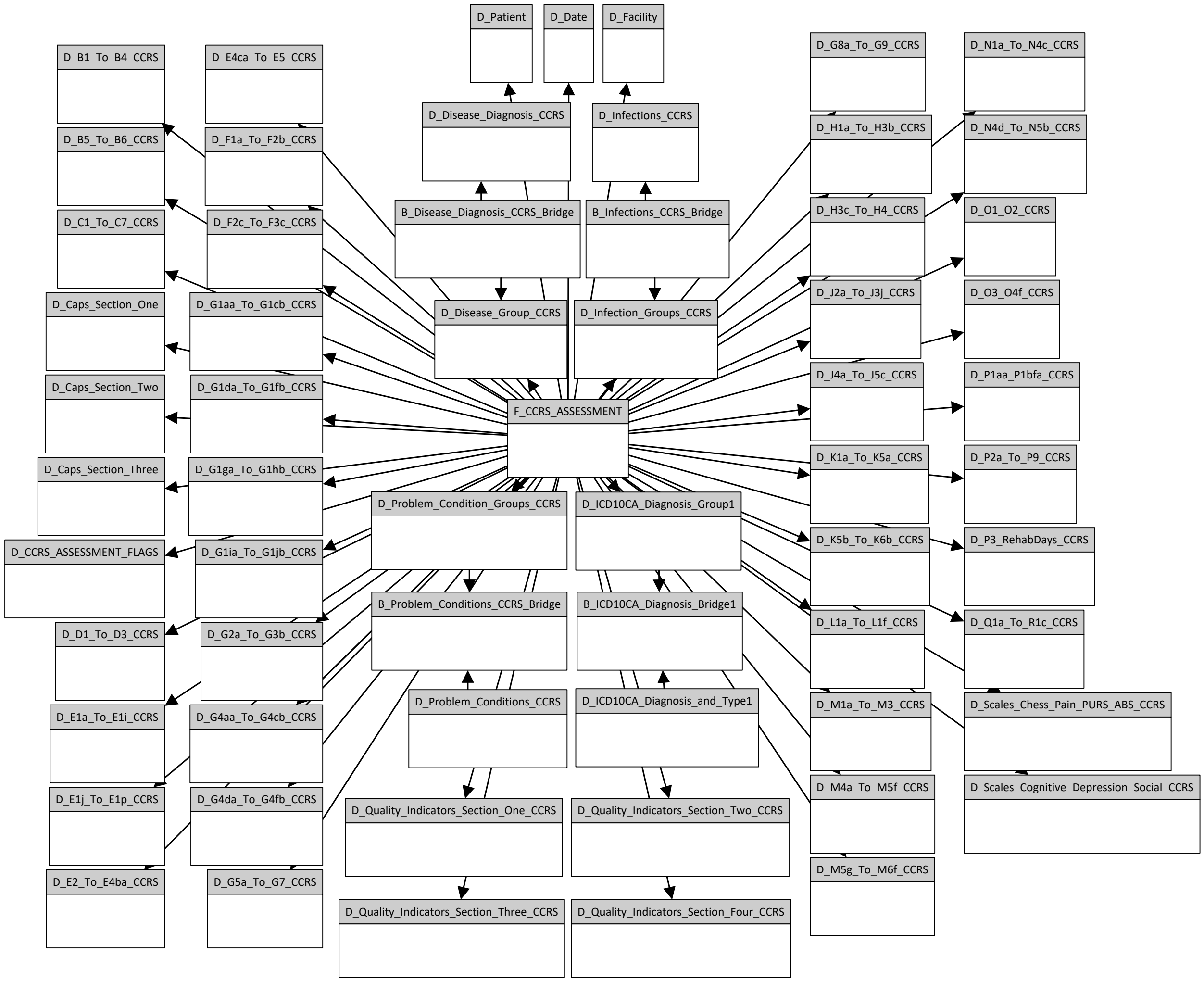
23	Neurological	Hemiplegia/Hemiparesis	i1w
24	Neurological	Huntington's chorea	i1x
25	Neurological	Multiple sclerosis	i1y
26	Neurological	Paraplegia	i1z
27	Neurological	Parkinson's disease	i1aa
28	Neurological	Quadriplegia	i1bb
29	Neurological	Seizure disorder	i1cc
30	Neurological	Transient ischemic attack	i1dd
31	Neurological	Traumatic brain injury	i1ee
32	Psychiatric/Mood	Anxiety Disorder	i1ff
33	Psychiatric/Mood	Depression	i1gg
34	Psychiatric/Mood	Bipolar Disorder	i1hh
35	Psychiatric/Mood	Schizophrenia	i1ii
36	Pulmonary	Asthma	i1jj
37	Pulmonary	Emphysema/COPD	i1kk
38	Sensory	Cataracts	i1ll
39	Sensory	Diabetic retinopathy	i1mm
40	Sensory	Glaucoma	i1nn
41	Sensory	Macular Degeneration	i1oo
42	Other	Allergies	i1pp
43	Other	Anemia	i1qq
44	Other	Cancer	i1rr
45	Other	Gastrointestinal disease	i1ss
46	Other	Liver disease	i1tt
47	Other	Renal failure	i1uu

Final CCRS Solution

When our Fact Table is combined with all of the above dimensions, we have our CCRS Star Schema solution as shown in Figure 6.24. This star schema allows us to report on the Continuing Care patient population and quality of care being provided within the health authority.

The CCRS star schema is significantly larger than is normal within a data warehouse. It also contains multiple complex measures that are not normal in business intelligence. This makes for a very large star schema, but also provides significant potential, a great deal of information and flexibility.

Figure 6.24: CCRS Star Schema



6.4 HCRS Assessment Star Schema.

The HCRS assessment data is similar to the CCRS data and offers the same challenges. There are approximately 400 distinct fields supplied as part of the HCRS assessment data, making this data set slightly less complicated but only in terms of volume. The design approach selected for the HCRS subject area is the same as selected for the CCRS area, incorporating as much of the HCRS data as possible at the level of the assessment.

1) What is the Business Process?

The HCRS assessment data does not correspond to a direct business process; but instead, represents the measure of a patient's health and the provision of home care services. As with the CCRS area, this data is used to measure the health of a patient, the type of care received, the service level of care, the health of the patient population, and the quality of care being delivered. However, the data is captured on an irregular basis usually reflecting significant changes in health and is therefore not as functional as the CCRS subject area for tracking population health.

Figure 6.25: HCRS Assessment Fact Table

F_HCRS_ASSESSMENT	

2) How do we measure the business process?

The HCRS assessment table has multiple measures. This includes a count of assessments, count of patients, count of service episodes, length of stay, service provisioning, nursing visits, home care, hospital stays, and visits to an emergency department. In addition, seventeen separate quality indicators that look at changes in patient health are included. In total seventy separate measure fields are encompassed in the HCRS assessment. As with the CCRS assessment, these measures are complex with some involving a numerator and denominator, some intended for statistical calculations, and some represent average measures of weekly service volumes.

Figure 6.26: HCRS Assessment Fact Table with Measures

F_HCRS_ASSESSMENT
P1aA_Home_Health_Aides_Days P1aB_Home_Health_Aides_Hours P1aC_Home_Health_Aides_Mins P1bA_Visiting_Nurses_Days P1bB_Visiting_Nurses_Hours P1bC_Visiting_Nurses_Mins P1cA_Homemaking_Services_Days P1cB_Homemaking_Services_Hours P1cC_Homemaking_Services_Mins P1dA_Meals_Days P1dB_Meals_Hours P1dC_Meals_Mins P1eA_Volunteer_Services_Days P1eB_Volunteer_Services_Hours P1eC_Volunteer_Services_Mins P1fA_Physical_Therapy_Days P1fB_Physical_Therapy_Hours P1fC_Physical_Therapy_Mins P1gA_Occupational_Therapy_Days P1gB_Occupational_Therapy_Hours P1gC_Occupational_Therapy_Mins P1hA_Speech_Therapy_Days P1hB_Speech_Therapy_Hours P1hC_Speech_Therapy_Mins P1iA_Day_Care_or_Day_Hospital_Days P1iB_Day_Care_or_Day_Hospital_Hours P1iC_Day_Care_or_Day_Hospital_Mins P1jA_Social_Worker_in_Home_Days P1jB_Social_Worker_in_Home_Hours P1jC_Social_Worker_in_Home_Mins G3a_Informal_Help_Hours_Weekdays G3b_Informal_Help_Hours_Weekend SINCE_LAST_AX_DAYS HC_IP_Flag HC_QI_Flag HC_InadequateMeal_N HC_InadequateMeal_D HC_WeightLoss_N HC_WeightLoss_D HC_Dehydration_N HC_Dehydration_D HC_MedReview_N HC_MedReview_D HC_NoAsstDevice_N HC_NoAsstDevice_D HC_RehabPotential_N HC_RehabPotential_D HC_Falls_N HC_Falls_D HC_Isolation_N HC_Isolation_D HC_Delirium_N HC_Delirium_D HC_NegativeMood_N HC_NegativeMood_D HC_DailyPain_N HC_DailyPain_D HC_PainControl_N HC_PainControl_D HC_Neglect_N HC_Neglect_D HC_Injury_N HC_Injury_D HC_Vaccination_N HC_Vaccination_D HC_Hospital_N HC_Hospital_D HC_Incidence_6 HC_Incidence_12

3) What is the grain of the fact table?

The grain chosen for the fact table was at the level of an individual assessment. This decision was based on the measures which exist at an assessment level.

4) What do we measure by?

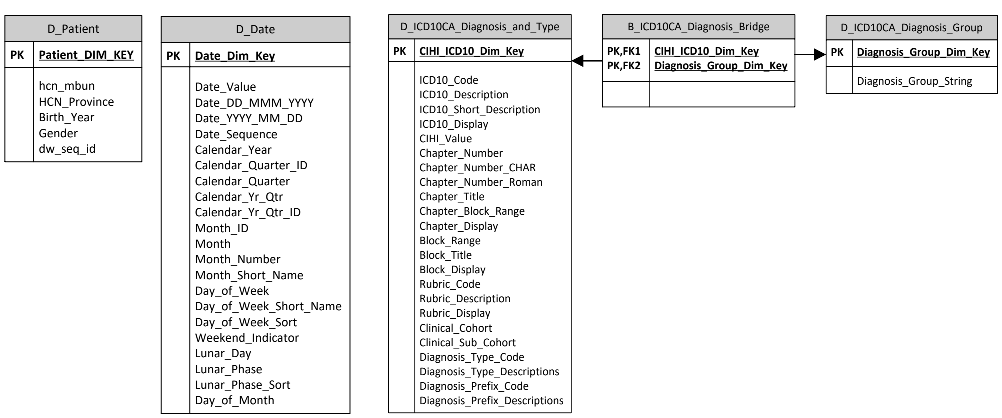
This question presents the same challenges as the CCRS subject area. Conformed dimensions for patient and date were used but there are hundreds of additional attributes to capture. As many attributes as possible were captured in order to provide maximum functionality.

Unlike the CCRS subject area, only the flag dimension pattern was used and no bridge table structures, other than our conformed disease diagnosis, were employed.

Available Conformed Dimension

Only two conformed dimensions were used, representing Date and Patient. In addition to these the HCRS assessment data includes ICD-10-CA diagnosis codes which map to our conformed Diagnosis Bridge Structure.

Figure 6.27: HCRS Assessment Conformed Dimensions



Flag Dimension Pattern

In building the HCRS schema nearly forty separate flag dimensions were created. These follow the same pattern used for the CCRS subject area. Tables were organized in alphabetical order with observations, quality indicators, Client Assessment Protocols (CAPS), and scales in separate tables. Additional fields were included for display values and descriptions. A full description of these fields is provided in appendix three.

Table 6.6: HCRS FLAG Dimension tables

HCRS Dimension Name	HCRS Columns
D_J1q_to_J1ac_HCRS	J1aa, J1ab, J1ac, J1q, J1r, J1s, J1t, J1u, J1v, J1w, J1x, J1y, J1z
D_K1a_to_K3h_HCRS	K1a, K1b, K1c, K1d, k1e, K2a, K2b, K2c, K2d, K2e, K2f, K3a, K3b, K3c, K3d, K3e, K3f, K3g, K3h
D_K4a_to_K6b_HCRS	K4a, K4b, K4c, K4d, K4e, K5, K6a, K6b
D_K7a_to_K9f_HCRS	K7a, K7b, K7c, K8a, K8b, K8c, K8d, K8e, K8f, K9a, K9b, K9c, K9d, K9e, K9f
D_L1a_to_M1d_HCRS	L1a, L1b, L1c, L2a, L2b, L2c, L2d, L3, M1a, M1b, M1c, M1d
D_Misc_Indicators_HCRS	AX_IN_HOSPITAL_IND_CODE, CAREGIVER_BURDEN_IND_CODE, CLIENT_FIRST_AX_IND_CODE, CLIENT_LAST_AX_IND_CODE, EMERGENT_CARE_VISIT_IND_CODE, END_OF_LIFE_IND_CODE, EPISODE_FIRST_AX_IND_CODE, EPISODE_LAST_AX_IND_CODE, ER_VISIT_IND_CODE, INFORMAL_CAREGIVER_IND_CODE, OVRNGHT_HOSPTAL_VST_IND_CODE, PRIOR_RESIDENT_CARE_IND_CODE
D_N1_to_N5e_HCRS	N1, N2a, N2b, N3a, N3b, N3c, N3d, N3e, N3f, N4, N5a, N5b, N5c, N5d, N5e
D_O1a_to_O2b_HCRS	O1a, O1b, O1c, O1d, O1e, O1f, O1g, O1h, O1i, O2a, O2b
D_P2a_to_P2p_HCRS	P2a, P2aa, P2b, P2c, P2d, P2e, P2f, P2g, P2h, P2i, P2j, P2k, P2l, P2m, P2n, P2o, P2p
D_P2q_to_P2z_HCRS	P2q, P2r, P2s, P2t, P2u, P2v, P2w, P2x, P2y, P2z
D_P3a_to_P7_HCRS	P3a, P3b, P3c, P3d, P4a, P4b, P4c, P5, P6, P7
D_Q1_to_Q4_HCRS	Q1, Q2a, Q2b, Q2c, Q2d, Q3, Q4

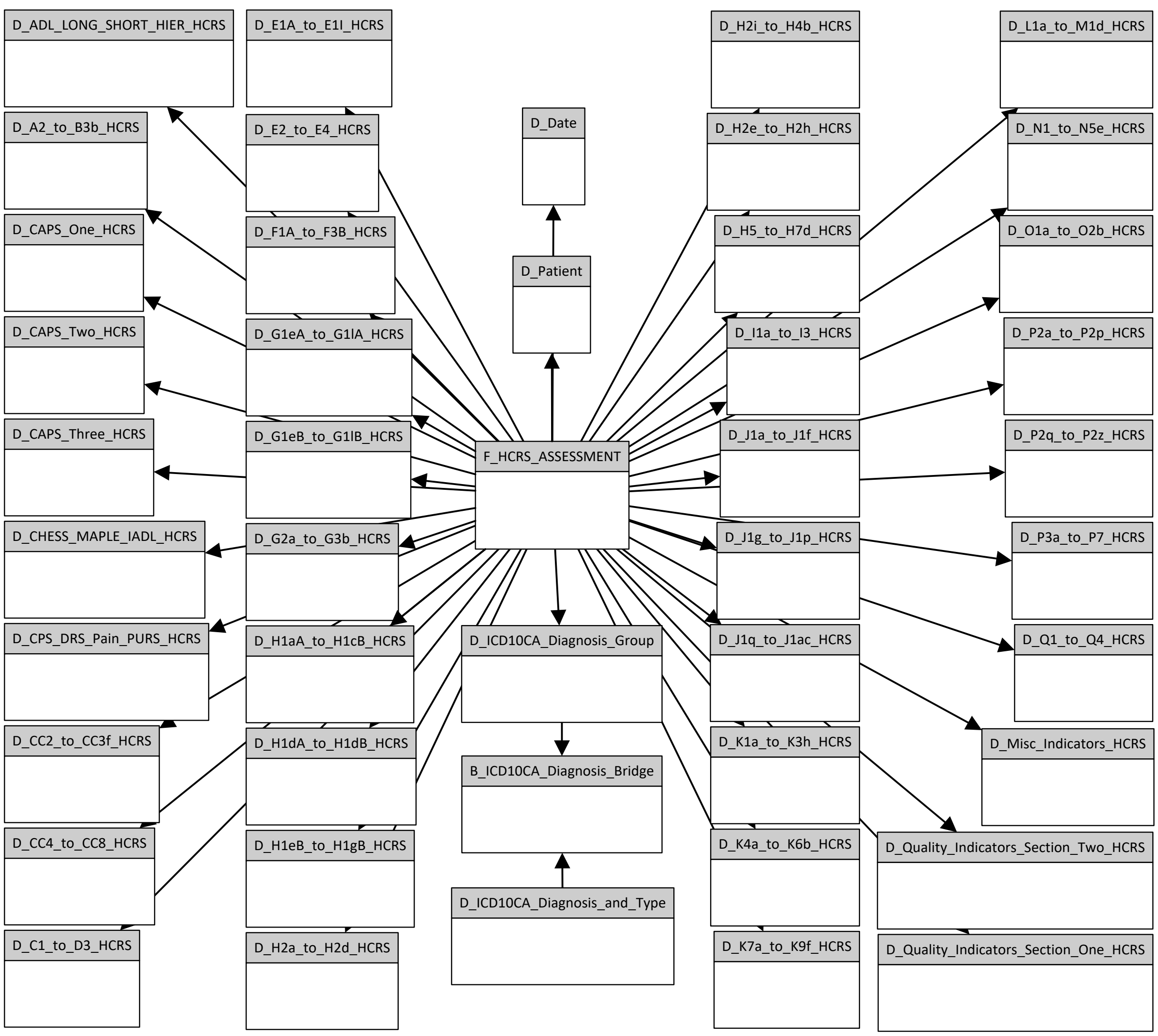
D_Quality_Indicators_Section_One_HCRS	HC_Dehydration_D, HC_Dehydration_N, HC_Falls_D, HC_Falls_N, HC_InadequateMeal_D, HC_InadequateMeal_N, HC_IP_Flag, HC_Isolation_D, HC_Isolation_N, HC_MedReview_D, HC_MedReview_N, HC_NoAsstDevice_D, HC_NoAsstDevice_N, HC_QI_Flag, HC_RehabPotential_D, HC_RehabPotential_N, HC_WeightLoss_D, HC_WeightLoss_N
D_Quality_Indicators_Section_Two_HCRS	HC_DailyPain_D, HC_DailyPain_N, HC_Delirium_D, HC_Delirium_N, HC_Hospital_D, HC_Hospital_N, HC_Incidence_12, HC_Incidence_6, HC_Injury_D, HC_Injury_N, HC_NegativeMood_D, HC_NegativeMood_N, HC_Neglect_D, HC_Neglect_N, HC_PainControl_D, HC_PainControl_N, HC_Vaccination_D, HC_Vaccination_N
D_A2_to_B3b_HCRS	A2, B1a, B1b, B2a, B2b, B3a, B3b
D_ADL_LONG_SHORT_HIER_HCRS	ADL_hier_hc, ADL_long_hc, ADL_short_hc
D_C1_to_D3_HCRS	C1, C2, C3, C4, D1, D2, D3
D_CAPS_One_HCRS	Abuse_CAP2_HC, ADL_CAP2_HC, Behaviour_CAP2_HC, Bowel_CAP2_HC, Cardio_CAP2_HC, Cognitive_CAP2_HC, Communication_CAP2_HC
D_CAPS_Three_HCRS	Medication_CAP2_HC, Mood_CAP2_HC, Pain_CAP2_HC, Physical_Activity_CAP2_HC, Social_CAP2_HC, Support_CAP2_HC, Ulcer_CAP2_HC, Urinary_CAP2_HC
D_CAPS_Two_HCRS	Dehydration_CAP2_HC, Delirium_CAP2_HC, Environment_CAP2_HC, Falls_CAP2_HC, Feeding_CAP2_HC, IADL_CAP2_HC, Institution_CAP2_HC
D_CC2_to_CC3f_HCRS	CC2, CC3a, CC3b, CC3c, CC3d, CC3e, CC3f
D_CC4_to_CC8_HCRS	CC4, CC5, CC6, CC7, CC8
D_CHESS_MAPLE_IADL_HCRS	Chess_hc, IADL_Difficulty_hc, IADL_Inv_HC, maple_hc
D_CPS_DRS_Pain_PURS_HCRS	CPS_hc, DRS_hc, pain_hc, PURS_hc
D_E1A_to_E1I_HCRS	E1a, E1b, E1c, E1d, E1e, E1f, E1g, E1h, E1i
D_E2_to_E4_HCRS	E2, E3a, E3b, E3c, E3d, E3e, E4
D_F1A_to_F3B_HCRS	F1a, F1b, F2, F3a, F3b
D_G1eA_to_G1lA_HCRS	G1eA, G1fA, G1gA, G1hA, G1iA, G1jA, G1kA, G1lA
D_G1eB_to_G1lB_HCRS	G1eB, G1fB, G1gB, G1hB, G1iB, G1jB, G1kB, G1lB
D_G2a_to_G3b_HCRS	G2a, G2b, G2c, G2d, G3a, G3b
D_H1aA_to_H1cB_HCRS	H1aA, H1aB, H1bA, H1bB, H1cA, H1cB
D_H1dA_to_H1dB_HCRS	H1dA, H1dB, H1eA, H1fA
D_H1eB_to_H1gB_HCRS	H1eB, H1fB, H1gA, H1gB
D_H2a_to_H2d_HCRS	H2a, H2b, H2c, H2d
D_H2e_to_H2h_HCRS	H2e, H2f, H2g, H2h
D_H2i_to_H4b_HCRS	H2i, H2j, H3, H4a, H4b
D_H5_to_H7d_HCRS	H5, H6a, H6b, H7a, H7b, H7c, H7d
D_I1a_to_I3_HCRS	I1a, I1b, I2a, I2b, I2c, I3
D_J1a_to_J1f_HCRS	J1a, J1b, J1c, J1d, J1e, J1f

D_J1g_to_J1p_HCRS	J1g, J1h, J1i, J1j, J1k, J1l, J1m, J1n, J1o, J1p

Final HCRS Solution

A model of the HCRS Star Schema tables is provided below in Figure 6.28. Field names are listed in table 6.5 and not included in the model for purposes of legibility. The size and scale of the HCRS subject area is significant and other design options were considered, but this offered the most flexible option at the level of granularity used for the capture of the measures.

Figure 6.28: HCRS Star Schema



Chapter 7. Extension Development Build

As documented in section four, our methodology involves the development of a relation engine that will be used to associate information to our subject area star schemas to extend and interrelate our data warehouse subject areas, providing new insight into health services. This involves developing a solution to identify our records, store and process SQL relation rules, capture the results of these relation rules, and process these results into our star schema subject areas. The data structure and processes to execute this are documented below.

7.1 Identify the records

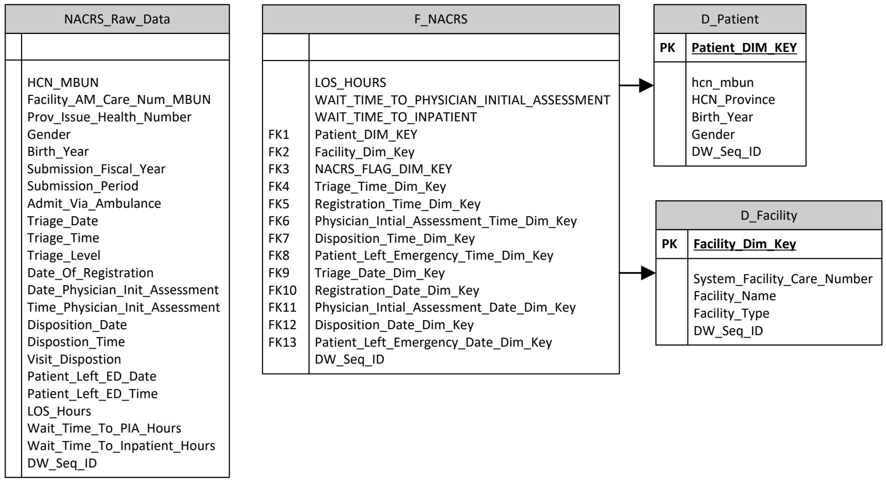
The first step in developing the new relation engine was to identify each record uniquely. This is not a simple, table-based primary key, but rather a unique identifier across all database tables similar to the RDF triplets in the semantic web. The use of a unique key across all tables is necessary, as it greatly reduces the complexity of establishing new relationships by removing the table from the equation. Relationships in our relation engine are based on unique identifiers and an expression in the form of a SQL statement.

All unique values were created through the use of a single database sequence shown below.

```
Create Sequence SRCDAT.Unique_Identifier start with 1 increment by 1 no cycle no maxvalue;
Select Next Value for SRCDAT.Unique_Identifier;
```

The sequence was used to populate a database field that was created for each table and populated as part of the raw data input into the database. This field, named DW_Seq_ID, has the same name in all tables. Figure 7.1 shows the NACRS source table and the target database star schema table. The DW_Seq_ID was populated in the source table as it was loaded, and this value flowed through our data transformation into the target Fact table. This allows enhanced functionality in monitoring and controlling the extract, transform, and load process and increases metadata functionality.

Figure 7.1: Unique Record Identifier Samples

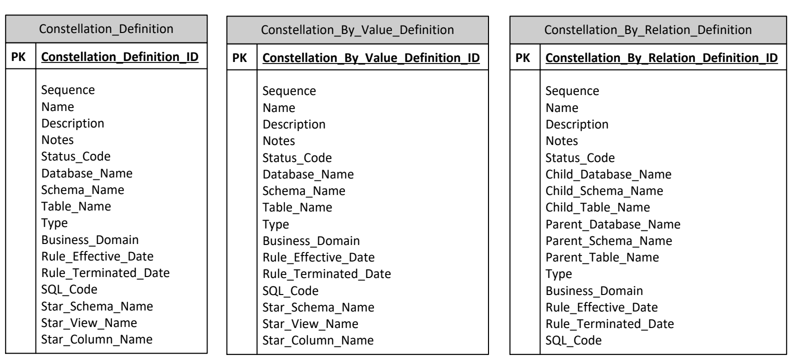


In some situations, such as Patient or Facility, the DW_Seq_ID was populated directly in the target star schema dimension table as no corresponding source data existed.

7.2 Relation Storage System

The first step in building the new relation engine is to create a database table to store our SQL expressions for processing. To simplify the development, three separate tables were created to store the expressions. These tables were based on the level of required functionality to either simply identify records that meet a condition, relates a value to a record, or relates two separate records to each other that meet a given criteria. These tables are shown below in Figure 7.2.

Figure 7.2: Constellation Rule Storage.



These tables provide minimal additional information to support the management of the relation rules. In a true robust system, additional metadata with a full relational structure to manage the validation rules and the processing of the rules would be created. A description of the fields is provided in Table 7.1 below.

Table 7.1: Constellation Rule Table Columns

Column Name	Description
Constellation_Definition_ID	Primary Key for the Constellation Definition Table
Constellation_By_Value_Definition_ID	Primary Key for the Constellation by Value Definition Table
Constellation_By_Relation_Definition_ID	Primary Key for the Constellation by Relation Definition Table
Sequence	A sequence value to control the order for processing Constellation Rules
Name	The name of the Constellation rule
Description	A description of the Constellation Rule and its purpose
Notes	Any notes or documentation pertaining to the Constellation Rule
Status_Code	The Status of the Constellation Rule. Is the rule active, Under Development, or is it disabled?
Database_Name	The database that contains the table that is the target of the rule.
Schema_Name	The schema that owns the table that is the target of the rule.
Table_Name	The target Table of the Constellation Rule.
Type	The type of Constellation Rule (Dimension or Fact Table)
Business_Domain	The Business domain for the validation rule, Is this for emergency services, Residential Care, etc.?
Rule_Effective_Date	The effective date for the Constellation Rule
Rule_Terminated_Date	The expiry date for the Constellation Rule
SQL_Code	The SQL of the Constellation Rule.
Star_Schema_Name	For Value and Record identification rules. An optional target schema for creating a Dynamic view
Star_View_Name	For Value and Record identification rules. An optional target Dynamic view
Star_Column_Name	For Value and Record identification rules. An optional target column in the Dynamic view
Parent_Database_Name	For reference relationships, The database for the Parent table in the relationship
Parent_Schema_Name	For reference relationships, The schema containing the Parent table in the relationship
Parent_Table_Name	For reference relationships, The Parent table in the relationship
Child_Database_Name	For reference relationships, The database for the Child table in the relationship
Child_Schema_Name	For reference relationships, The schema containing the Child table in the relationship
Child_Table_Name	For reference relationships, The Child table in the relationship

7.3 Relation Rules

There are three distinct types of relationship rules. Our first rule type is used to identify a record that meets a condition and is ideal for situations such as a patient cohort. The second rule type involves associating a value to a record. This requires the identification of the record and capturing the associated value. The final rule type is used to associate two records. This type of rule must capture the parent and child record identifiers and the parent-child nature of the relationship.

7.3.1 Constellation Record Identification

The first type of rule we examine is for identifying records that meet a condition. This type of rule is ideal for the identification of subject area records that meet a given condition and that cannot be established in the star schema that represents that subject area. A good example of such a rule would be to identify all emergency encounters in our NACRS data set for patients who are also registered in home care. The form of this SQL statement is to select the unique identifier from a table where it meets the criteria. Three examples are provided below.

Name: Emergency Patient in Home Care

This rule selects the dw_seq_id from the NACRS fact table where the patient is in Home care at the admission date for the NACRS encounter.

Select fn.dw_seq_id from star.dbo.F_NACRS as fn inner join star.dbo.F_HCRS_Assessment as fah on fn.patient_dim_key=fah.Patient_DIM_KEY and fn.Registration_Date_Dim_Key between fah.Admission_Date_Dim_Key and (case when fah.Discharge_Date_Dim_Key<0 then 99999999 else fah.Discharge_Date_Dim_Key end)

Name: Emergency Patient in Residential Care

This rule is also for the NACRS subject area identifies records where the patient is in residential care at the admission date for the NACRS encounter.

Select fn.dw_seq_id from star.dbo.F_NACRS as fn inner join star.dbo.F_CCRS_Assessment as fac on fn.patient_dim_key=fac.Patient_DIM_KEY and fn.Registration_Date_Dim_Key between fac.Entry_Date_Dim_Key and (case when fac.Discharge_Date_Dim_Key<0 then 99999999 else fac.Discharge_Date_Dim_Key end)

Name: DAD Abstract Patient in Residential Care

The last rule is for the DAD subject area and identifies records where the patient is in residential care at the admission date for the DAD encounter.

Select distinct fd.dw_seq_ID from star.dbo.F_DAD as fd inner join star.dbo.F_CCRS_Assessment as fac on fd.patient_dim_key=fac.Patient_DIM_KEY and fd.Admission_Date_Dim_Key between fac.Entry_Date_Dim_Key and (case when fac.Discharge_Date_Dim_Key<0 then 99999999 else fac.Discharge_Date_Dim_Key end)

These rules are all based on patient cohorts; but due to date constraints, identify records from a fact table for that cohort. To identify the dimension table record in this situation would have potentially resulted in misleading information for records that do not meet the temporal factor of the rule.

7.3.2 Constellation by Value Record

The Constellation Value rules are to associate information to records. These rules function by identifying the unique record identifier and the information value pair in a query. This type of rule is ideal for associating information from one star schema, such as a value from a patient's recent assessment, with a different star schema table. An example would be the Change in Health, End stage disease, Signs and Symptoms (CHESS) score from a home care assessment with emergency encounters. Another example would be the aggregate quality of care score for residential care patients to the DAD fact table.

Two more examples below extend the Home Care and Residential Care star schemas by associating the aggregate count of emergency encounters to the respective fact table.

Name: Emergency Encounters Last 90 Days

This rule selects the dw_seq_id from the HCRS fact table and the count of records from the NACRS table representing emergency encounters for that patient in the 90 days prior to the assessment.

select distinct fah.DW_SEQ_ID ,(select count(*) from star.dbo.F_NACRS as fn inner join star.dbo.D_Date as dd on fn.Registration_Date_Dim_Key=dd.Date_Dim_Key where fn.patient_dim_key=fah.Patient_DIM_KEY and dd.Date_Sequence between df.Date_Sequence-90 and df.Date_Sequence) as value from star.dbo.F_HCRS_ASSESSMENT as fah inner join star.dbo.d_date as df on df.Date_Dim_Key=fah.Assessment_Reference_Date_Dim_Key

Name: Emergency Encounters Last 90 Days

Similar to our previous rule, this rule selects the dw_seq_id from the CCRS fact table and the count of records from the NACRS table representing emergency encounters for that patient in the 90 days prior to the assessment.

```
select distinct fah.DW_SEQ_ID ,(select count(*) from star.dbo.F_NACRS as fn inner join star.dbo.D_Date as dd on
fn.Registration_Date_Dim_Key=dd.Date_Dim_Key where fn.patient_dim_key=fah.Patient_DIM_KEY and dd.Date_Sequence between
df.Date_Sequence-90 and df.Date_Sequence) as value  from star.dbo.F_CCRS_ASSESSMENT as fah inner join star.dbo.d_date as df on
df.Date_Dim_Key=fah.Assessment_Date_Dim_Key
```

7.3.3 Constellation by Relation Rule

The constellation by relation rules are to associate two records together. These rules function by returning two unique record identifiers as a parent and a child in a relationship. This type of rule can be used to associate records from separate star schemas such as a lab result to a patient assessment. Alternatively a rule can be used to associate two records from the same star schema such as a Discharge Abstract to a previous Abstract record. The primary use for this functionality in our study is to interrelate star schemas by associating different assessment information to hospital emergency or discharge abstract records.

Name: Prior Emergency Encounter

The constellation rule below associates the most recent prior Hospital Emergency encounter for a patent with the patient assessment in continuing care. This can be useful to determine the impact of that encounter with the assessment. No criteria for a limit on how recent that encounter was included, but could easily be added.

```
select distinct dw_seq_id as child_dw_seq_id,isnull((select top 1 dw_seq_id from star.dbo.F_NACRS as fn where
fn.patient_dim_key=fca.Patient_DIM_KEY and fn.Registration_Date_Dim_Key<fca.Assessment_Date_Dim_Key order by
fn.Registration_Date_Dim_Key desc),-1) as parent_dw_seq_id from Star.dbo.F_CCRS_ASSESSMENT as fca
```

Name: Next Emergency Encounter

Similar to our first rule, this constellation rule associates a Hospital Emergency encounter for a patient with the previous assessment in continuing care. This can be useful to examine any changes to the patient's health or potential reasons that may have triggered the emergency encounter. Again, no criteria for a limit on how recent the encounter is was included, but could easily be added.

```
select distinct dw_seq_id as child_dw_seq_id,isnull((select top 1 dw_seq_id from star.dbo.F_NACRS as fn where
fn.patient_dim_key=fca.Patient_DIM_KEY and fn.Registration_Date_Dim_Key>fca.Assessment_Date_Dim_Key order by
fn.Registration_Date_Dim_Key asc),-1) as parent_dw_seq_id from Star.dbo.F_CCRS_ASSESSMENT as fca
```

Name: Next DAD encounter

This constellation rule associates a Hospital Discharge Abstract record for a patent with the previous patient assessment in continuing care. As with emergency encounters this can be useful to examine the assessment to determine factors that may have impacted the hospital encounter. No criteria for a time limit on how recent that encounter was is included, but could easily be added.

select distinct dw_seq_id as child_dw_seq_id ,isnull((select top 1 dw_seq_id from star.dbo.F_Dad as fd where fd.patient_dim_key=fca.Patient_DIM_KEY and fd.admission_Date_Dim_Key>fca.Assessment_Date_Dim_Key order by fd.admission_Date_Dim_Key asc),-1) as parent_dw_seq_id from Star.dbo.F_CCRS_ASSESSMENT as fca

Name: Prior DAD encounter

The final constellation rule example associates a residential assessment record with a prior Discharge abstract database record. As in previous examples, this is useful for examining the effects of the hospital encounter on the patient's health. Whether they have improved or not, and what impact the encounter had on the patient's health and wellbeing.

select distinct dw_seq_id as child_dw_seq_id, isnull((select top 1 dw_seq_id from star.dbo.F_DAD as fd where fd.patient_dim_key=fca.Patient_DIM_KEY and fd.Discharge_Date_Dim_Key<fca.Assessment_Date_Dim_Key order by fd.Discharge_Date_Dim_Key desc),-1) as parent_dw_seq_id from Star.dbo.F_CCRS_ASSESSMENT as fca

7.4 Relation Rule Processing

Three separate database procedures to execute the validation rules were created. These procedures read the stored SQL rules from our constellation tables, execute those rules dynamically, and capture the results. They can be run on a nightly basis or interactively, based on requirements. All three procedures are provided in Appendix 5. A pseudo code version is supplied below based on associating a value to a record. All procedures follow a similar structure and process.

String Constellation_Rule;

Integer Constellation_Rule_ID;

String SQL_Statement;

```
Create Cursor Get_Rules as
  Select SQL_Code , Constellation_Rule_ID from Constellation_By_Value_Rules;
 Open Get_Rules;
 Fetch Get_Rules into Constellation_Rule, Constellation_ID;
 While fetch_status=0
  Begin
   SQL_Statement = ' with constellation_code as ( ' + Constellation_Rule + ')
                    merge Constellation_by_value as target
                     using ( select distinct dw_seq_id, value from constellation_code)
                     as source on source.dw_seq_id=target.dw_seq_id
                              and source.value = target.value
                     when not matched by target insert (dw_seq_id,value)
                    when not matched by source and
Target.constellation_rule_id=Source.constellation_rule_id  then delete;'
   sp_executesql SQL_Statement
   Fetch Get_Rules into Constellation_Rule, Constellation_ID;
  End;
```

The process above is straight forward. A cursor is created that reads constellation rules from the database table they are stored in. The SQL from these constellations is then combined with a database merge statement that takes the results of the constellation query and merges it into the target results table. This SQL statement combines a SQL Insert, Update, and Delete into a single, efficient SQL statement. This is then executed and the returned data is stored in the Constellation results table. This completes the execution of our constellation rule and the cursor moves to the next rule to continue processing. It is a simple, straightforward process to execute these rules. Moving the results of the rule queries into our Kimball structured Star Schema database is all that remains.

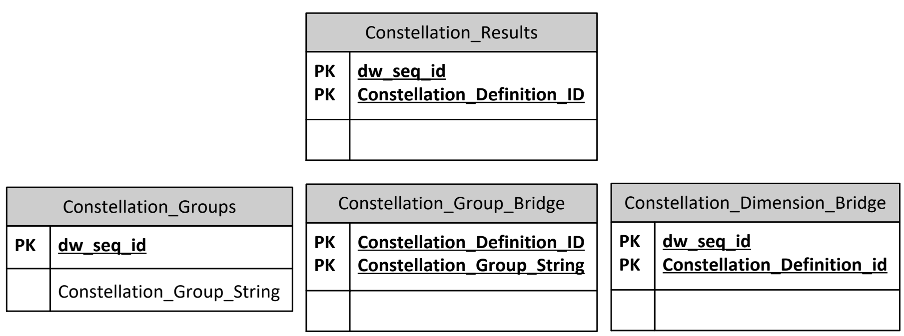
7.5 Relation Results Processing

The process of moving data from the stored results of our constellation processing into the targeted structure of our reporting Star Schema database involves multiple steps but is straight forward. This process must be integrated into the overall processing of the system in order to ensure the data is properly maintained with no loss of information. Two separate workflows are explained below for identify records and associating values to a record. The third method for creating relationships is accomplished with views and is largely dependent on the toolset used for analyzing the data.

7.5.1 Processing the identification of records.

Processing of the constellation data for identifying records is performed by two procedures. The first procedure involves populating staging tables to create the constellation groups for use with our fact records. A second procedure populates a parallel table for a dimension record cross reference table. These tables and the source results table for identifying records are shown in Figure 7.3.

Figure 7.3: Constellation Definition Results and Staging tables.



The fact table procedure involves populating a group results table that identifies each of the records that belong to a fact table and satisfies a constellation rule by the combination of rules that were satisfied. This is done by selecting the unique dw_seq_id from our results table with an aggregate function to create a sorted combined group string of the constellation definitions satisfied. The SQL for this process is below.

```
Merge Constellation_ Groups as target using
 (select cr.dw_seq_id, ,dbo.SortConcatenate(cr.Constellation_Definition_ID) as Group_string
     from constellation_results as cr
         inner join Constellation_Definition as cd on cr.Constellation_Definition_ID=cd.Constellation_Definition_ID
    where cd.type= 'FACT BRIDGE' group by dw_seq_id) as source
on source.dw_seq_id=target.dw_seq_id
When not matched by target then insert (dw_seq_id, Constellation_Group_String)
     values (souce.dw_seq_id, source.Group_String)
When matched and Source.Group_String != target,Constellation_Group_String
     then update set target. Constellation_Group_String= Source.Group_String
When not matched by source then delete;
```

This SQL statement employs a custom aggregate function, SortConcatenate which takes the individual definition identifiers and turns them into a single concatenated string. The source for this function is provided in Appendix 6.

The next step in this process is to build a bridge table between the new group combinations and the individual constellation definitions. This is accomplished in SQL by taking the minimum unique identifier in our new constellation group table with the group string as an aggregate query. This subquery is then joined to the results table to return the distinct constellation definitions with the group string and forms the bridge table or cross reference between the group combinations and constellation definitions. This query is provided below as a SQL merge statement.

```
Merge Constellation_Group_Bridge as Target using
 (SELECT distinct cj.Constellation_Definition_ID,cg. Constellation_Group_String FROM
   (SELECT  min([DW_Seq_ID]) as DW_SEQ_ID , Constellation_Group_String
     FROM Constellation_ Groups group by Constellation_Group_String) as cg
  inner join constellation_results as cr on cr.DW_Seq_ID=cg.DW_Seq_ID ) as Source
on source.constellation_group_string=target.Constellation_group_string
  and source.Constellation_Definition_id = target.Constellation_Definition_id
when not matched by target then insert (Constellation_Definition_ID,Constellation_Group_String)
     values (Source.Constellation_Definition_ID,Source.Constellation_Group_String)
```

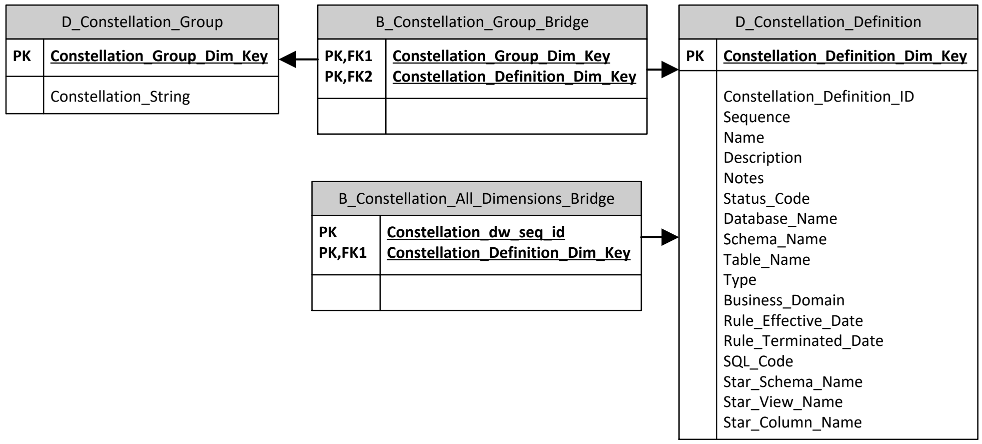
when not matched by source then delete

The parallel process for dimensions is simpler and involves populating a staging table for the dimension cross references. Dimension records do not use a bridge table structure as created in the fact table process. Instead, a simple cross reference table, containing the unique identifiers for any dimension record that satisfies a constellation rule and the definition identifier of that rule is used. This is shown in the SQL merge statement below.

```
Merge Constellation_Dimension_Bridge as Target using
    (select distinct cr.dw_seq_id, cr.Constellation_Definition_ID  from constellation_results as cr
        inner join Constellation_Definition as cd
           on cr.Constellation_Definition_ID=cd.Constellation_Definition_ID
        where cd.type= 'DIMENSION BRIDGE') as Source
on source.dw_seq_id=target.dw_seq_id
   and source.Constellation_Definition_id = target.Constellation_Definition_id
when not matched by target then insert (dw_Seq_id, Constellation_Definition_ID)
       values (Source.dw_seq_id,Source.Constellation_Definition_ID)
when not matched by source then delete
```

With this, the staging tables are complete. All that remains is to populate the star database objects that are used for reporting with our Star Schemas. This step involves individual SQL merge statements that populate each table. In total four separate star schema tables exist, representing the bridge structure and the cross reference table between the dimensions and the constellation definitions. These database tables are shown in Figure 7.4 and the SQL statements are listed following the figure.

Figure 7.4: Constellation Star Schema Objects



The Constellation Star Schema objects are all based on staging or definition objects previously discussed and populated from these database table. The SQL statements are listed below.

The first SQL Statement populates the Group dimension with the distinct groups selected from the staging Constellation_by_Group_Bridge table created in our staging process

```
merge Star.Constellation_Data.D_CONSTELLATION_GROUP as target
using
( SELECT distinct Constellation_GROUP FROM Constellation_By_Group_BRIDGE ) as source
on source.Constellation_GROUP=target.CONSTELLATION_STRING
when not matched by target then insert (CONSTELLATION_STRING)
values (source.Constellation_GROUP) ;
```

Once the bridge table is populated, the second step is to populate the constellation dimension table. This inserts new records or updates existing records if a change has occurred based on the source definition table.

```
merge star.Constellation_Data.D_Constellation_Definition as target
using
(SELECT Constellation_Definition_ID, Name, Description, Notes, Status_Code, Database_Name,
     Schema_Name, Table_Name, type, Business_Domain, Rule_Effective_Date,
    Rule_Terminated_Date, Star_Schema_Name, Star_View_Name, Star_Column_Name
 FROM Constellation.Constellation_Build.Constellation_Definition
```

```
  WHERE type in ('DIMENSION','FACT BRIDGE')
) as source
on source.Constellation_Definition_ID=target.Constellation_Definition_ID
when not matched by target then
insert ( Constellation_Definition_ID, Constellation_Definition_NAME, Constellation_Definition_DESCRIPTION, Constellation_Definition_NOTES, Rule_Effective_Date, Rule_Terminated_Date, Database_Name, Schema_Name, Table_Name, Status_Code, Type, Business_Domain, Star_Schema_Name, Star_View_Name, Star_Column_Name)
values (source.Constellation_Definition_ID, source.Name, source.Description, source.Notes, source.Rule_Effective_Date, source.Rule_Terminated_Date, source.Database_Name, source.Schema_Name, source.Table_Name, source.Status_Code, source.type, source.Business_Domain, source.Star_Schema_Name, source.Star_View_Name, source.Star_Column_Name)
when matched and
(isnull(target.Constellation_Definition_NAME,'NVL')!=isnull(SOURCE.NAME,'NVL') or
 isnull(target.Constellation_Definition_DESCRIPTION,'NVL')!=isnull(SOURCE.DESCRIPTION,'NVL') or
 isnull(target.Constellation_Definition_NOTES,'NVL')!=isnull(SOURCE.NOTES,'NVL') or
 isnull(target.Rule_Effective_Date,cast('1799-09-01' as datetime)) != isnull(Source.Rule_Effective_Date,cast('1799-09-01' as datetime)) or
 isnull(target.Rule_Terminated_Date,cast('1799-09-01' as datetime)) != isnull(Source.Rule_Terminated_Date,cast('1799-09-01' as datetime)) or
 isnull(target.Database_Name,'NVL')!=isnull(SOURCE.Database_Name,'NVL') or
 isnull(target.Schema_Name,'NVL')!=isnull(SOURCE.Schema_Name,'NVL') or
 isnull(target.Table_Name,'NVL')!=isnull(SOURCE.Table_Name,'NVL') or
 isnull(target.Status_Code,'NVL')!=isnull(SOURCE.Status_Code,'NVL') or
 isnull(target.Type,'NVL')!=isnull(SOURCE.Type,'NVL') or
 isnull(target.Business_Domain,'NVL')!=isnull(SOURCE.Business_Domain,'NVL') or
 isnull(target.Star_Schema_Name,'NVL')!=isnull(SOURCE.Star_Schema_Name,'NVL') or
 isnull(target.Star_View_Name,'NVL')!=isnull(SOURCE.Star_View_Name,'NVL') or
 isnull(target.Star_Column_Name,'NVL')!=isnull(SOURCE.Star_Column_Name,'NVL')) then
        update set
target.Constellation_Definition_NAME=SOURCE.NAME,
target.Constellation_Definition_DESCRIPTION=SOURCE.DESCRIPTION,
target.Constellation_Definition_NOTES=SOURCE.NOTES,
target.Rule_Effective_Date=Source.Rule_Effective_Date,
target.Rule_Terminated_Date=Source.Rule_Terminated_Date,
target.Database_Name=SOURCE.Database_Name,
target.Schema_Name=SOURCE.Schema_Name,
target.Table_Name=SOURCE.Table_Name,
target.Status_Code=SOURCE.Status_Code,
```

```
target.Type=SOURCE.Type,
target.Business_Domain=SOURCE.Business_Domain,
target.Star_Schema_Name=SOURCE.Star_Schema_Name,
target.Star_View_Name=SOURCE.Star_View_Name,
target.Star_Column_Name=SOURCE.Star_Column_Name;
```

Then, the bridge table between the Constellation dimension and the constellation group dimension must be populated. This is based on the bridge staging table, but requires joins to our new dimension tables to retrieve the new dimension keys. The bridge table structure for the constellation star schema objects used for fact table relationships is now complete.

```
merge Star.Constellation_Data.B_CONSTELLATION_BRIDGE as target
using
(
SELECT distinct cd.Constellation_Definition_DIM_KEY
     ,cg.CONSTELLATION_GROUP_DIM_KEY
  FROM Constellation.Constellation_Data.Constellation_By_Group_BRIDGE cgb
inner join Star.Constellation_Data.D_Constellation_Definition cd on cd.Constellation_Definition_ID =
cgb.Constellation_Definition_ID
inner join Star.Constellation_Data.D_CONSTELLATION_GROUP cg on
cg.CONSTELLATION_STRING=cgb.Constellation_GROUP
) as source on
(source.Constellation_Definition_DIM_KEY=target.Constellation_Definition_DIM_KEY and
source.CONSTELLATION_GROUP_DIM_KEY=target.CONSTELLATION_GROUP_DIM_KEY)
when not matched by target then
insert (Constellation_Definition_DIM_KEY,CONSTELLATION_GROUP_DIM_KEY)
values (source.Constellation_Definition_DIM_KEY,source.CONSTELLATION_GROUP_DIM_KEY);
```

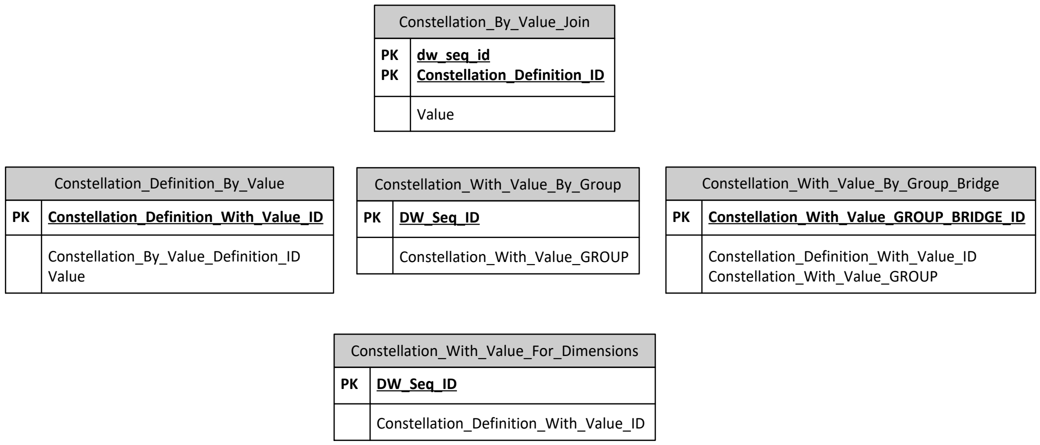
The last step is to populate the constellation all dimension bridge. This is done with a simple select statement from the all dimension bridge that was previously created in the staging area. The dimension contains the unique keys for dimension records that link to the constellation definition dimension. The benefits of our unique key are evident, as any dimension containing a unique key that is represented in the bridge table will join to the bridge.

```
merge star.Constellation_Data.B_Constellation_ALL_DIMENSIONS_BRIDGE as target

using

(select distinct cfd.DW_Seq_ID,cd.Constellation_Definition_DIM_KEY from
Constellation.Constellation_Data.Constellation_For_Dimensions as cfd

inner join Star.Constellation_Data.D_Constellation_Definition cd on cd.Constellation_Definition_ID =
cfd.Constellation_Definition_ID) as source

on

(source.Constellation_Definition_DIM_KEY=target.Constellation_Definition_DIM_KEY and

source.DW_Seq_ID=target.Constellation_DW_Seq_ID)

when not matched by target  then

insert (Constellation_Definition_DIM_KEY,Constellation_DW_Seq_ID)

values (source.Constellation_Definition_DIM_KEY,source.DW_Seq_ID)

when not matched by source then delete;
```

7.5.2 Processing the constellation value records.

Processing of the constellation data for associating a value to our records follows the same pattern as previously described for record identification. The process is more complicated in this situation, as we need to account for the unique identifier, rule definitions, and the value for each definition. As with record identification, the first step involves populating staging tables to create the constellation groups. Separate tables containing value combinations for use with our fact records and dimension records are populated. These tables and the source table for identifying records are shown in Figure 7.5. The key goal in this situations is to create a new structure for the constellation rules combined with values from the target dimension. The bridge structure then maps the fact table records to the constellation rule with value pair

Figure 7.5: Constellation by Value results and staging tables.



Populating the staging Constellation_Definition_By_Value table takes the constellation rule definition and combines it with the values returned from rule processing. This table serves as the base for our target dimension and much of the remaining processing.

```
merge Constellation_Data.Constellation_Definition_By_Value as target
using (
  SELECT distinct cbvd.Constellation_By_Value_Definition_ID , cbvj.Value
 FROM Constellation_Build.Constellation_By_Value_Definition as cbvd
 inner join Constellation_Build.Constellation_By_Value_Join as cbvj
 on cbvd.Constellation_By_Value_Definition_ID=cbvj.Constellation_By_Value_Definition_ID
) as source
on source.Constellation_By_Value_Definition_ID=target.Constellation_By_Value_Definition_ID and
source.Value=target.Value
when not matched by target then
insert (Constellation_By_Value_Definition_ID,Value)
values (source.Constellation_By_Value_Definition_ID,source.Value)
when not matched by source then delete;
```

The next step is to populate a group string table for the individual uniquely identified records. This is similar to the group table in the identification of records, but involves the new primary key from the previous table Constellation_Definition_By_Value. This is because we are mapping to the new definition/value pair combination.

```
truncate table Constellation_Data.Constellation_With_Value_By_Group;

INSERT INTO Constellation_Data.Constellation_With_Value_By_Group
        (DW_Seq_ID, Constellation_With_Value_GROUP)
SELECT cbvj.DW_SEQ_ID ,staging.dbo.SortConcatenate( scdv.Constellation_Definition_With_Value_ID )
  FROM Constellation_Build.Constellation_By_Value_Definition as cbvd
  inner join Constellation_Build.Constellation_By_Value_Join as cbvj
  on cbvd.Constellation_By_Value_Definition_ID=cbvj.Constellation_By_Value_Definition_ID
  inner join Constellation_Data.Constellation_Definition_By_Value as scdv
  on scdv.Constellation_By_Value_Definition_ID=cbvj.Constellation_By_Value_Definition_ID
      and scdv.Value=cbvj.Value
  where cbvd.type='FACT BRIDGE'  group by cbvj.DW_SEQ_ID
```

Once we have our group table and the target dimension for the constellation definition/value pair the only remaining task is to create a staging table for the bridge between them.

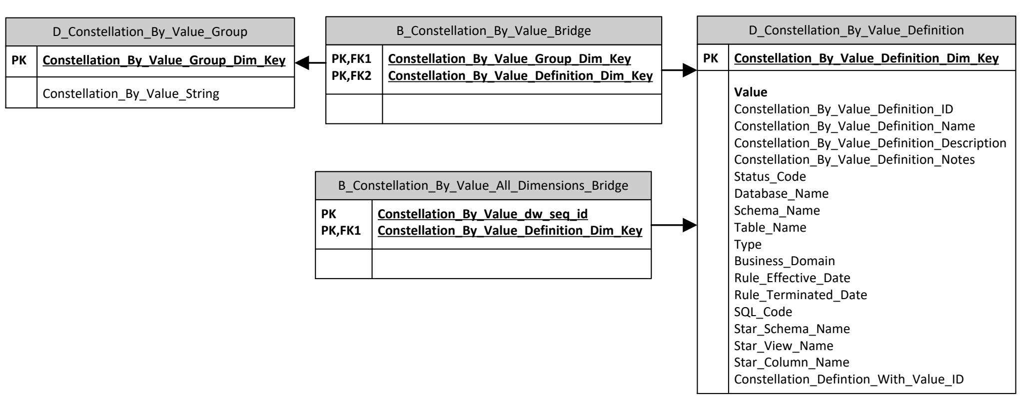
```
truncate table Constellation_Data.Constellation_With_Value_By_Group_BRIDGE

INSERT INTO Constellation_Data.Constellation_With_Value_By_Group_BRIDGE
        (Constellation_Definition_With_Value_ID, Constellation_With_Value_GROUP)
SELECT distinct cdv.Constellation_Definition_With_Value_ID ,cg.Constellation_With_Value_GROUP
  FROM
( SELECT  min([DW_Seq_ID]) as DW_SEQ_ID ,Constellation_With_Value_GROUP
FROM Constellation_Data.Constellation_With_Value_By_Group group by Constellation_With_Value_GROUP
) as cg
inner join Constellation_Build.Constellation_By_Value_Join as cj on cj.DW_Seq_ID=cg.DW_Seq_ID
inner join Constellation_Data.Constellation_Definition_By_Value as cdv
on cj.Constellation_By_Value_Definition_ID=cdv.Constellation_By_Value_Definition_ID and cj.Value=cdv.Value
```

The last step in our staging processing is to populate a bridge table for dimension records. This table contains the unique identifier for any dimension and the individual definition/value pair we populated previously.

```
truncate table Constellation_Data.Constellation_With_Value_For_Dimensions

INSERT INTO Constellation_Data.Constellation_With_Value_For_Dimensions
     (DW_Seq_ID, Constellation_Definition_With_Value_ID)
SELECT distinct cbvj.DW_SEQ_ID ,scdv.Constellation_Definition_With_Value_ID
 FROM Constellation_Build.Constellation_By_Value_Definition as cbvd
 inner join Constellation_Build.Constellation_By_Value_Join as cbvj
 on cbvd.Constellation_By_Value_Definition_ID=cbvj.Constellation_By_Value_Definition_ID
 inner join Constellation_Data.Constellation_Definition_By_Value as scdv
 on scdv.Constellation_By_Value_Definition_ID=cbvj.Constellation_By_Value_Definition_ID
 and scdv.Value=cbvj.Value   where cbvd.type='DIMENSION'
```

The processing of the staging tables is complete. At this point, the only remaining processing is the population of the star schema objects that represent the constellation by value structure in our reporting star database. This structure is identical to the previous structure used for the identification of records, with the exception that the constellation definition dimension now represents a definition for a field and a value. The target structure is shown in Figure 7.6 and the procedures to populate the structure and description follow.

Figure 7.6: Constellation by Value Star Schema Structures



Populating the constellation by value table structure follows the same steps used to populate the previous constellation star schema objects for record identification. The primary difference between the two structures was the addition of the value which was addressed in the staging process. The only significant difference is that the definition table must be joined to the staging tables in order to include the new value in the constellation by value dimension

Our first step in populating the constellation by value structure is the processing for the group dimension. This requires a query of the staging table for the distinct definition and value combinations that were concatenated together.

```
merge D_CONSTELLATION_BY_VALUE_GROUP as target
using (
SELECT distinct Constellation_With_Value_GROUP FROM Constellation_With_Value_By_Group_BRIDGE ) as source
on source.Constellation_With_Value_GROUP=target.CONSTELLATION_BY_VALUE_STRING
when not matched by target then
insert (CONSTELLATION_BY_VALUE_STRING)
values (source.Constellation_With_Value_GROUP) ;
```

Step two is the population of the constellation by value dimension. As noted, this is the only significant difference from our previous process, as the dimension must contain the associated values. The query returns all records from our definition table for the majority of the fields and the value from the staging table that contains the constellation by value definition identifier and value combination. This is an inner join, which means that any constellation by value rule that has no associated values will not return any records. The SQL merge statement will insert new results records and update old records where information has changed. It will not remove records that no longer exist due to potential referential integrity concerns.

```
merge D_Constellation_By_Value_Definition as target using
(SELECT cd.Constellation_By_Value_Definition_ID, cd.Name, cd.Description, cd.Notes, cd.Status_Code,
    cd.Database_Name, cd.Schema_Name, cd.Table_Name, cd.type, cd.Business_Domain,
```

```
    cd.Rule_Effective_Date, cd.Rule_Terminated_Date, cd.Star_Schema_Name, cd.Star_View_Name,
    cd.Star_Column_Name, cv.value, cv.Constellation_Definition_With_Value_ID
 FROM Constellation_By_Value_Definition as cd inner join Constellation_Definition_By_Value as cv
  on cv.Constellation_By_Value_Definition_ID=cd.Constellation_By_Value_Definition_ID
) as source
on source.Constellation_By_Value_Definition_ID=target.Constellation_By_Value_Definition_ID
and source.value=target.value
when not matched by target then
insert (Constellation_By_Value_Definition_ID, Constellation_By_Value_Definition_NAME,
    Constellation_By_Value_Definition_DESCRIPTION, Constellation_By_Value_Definition_NOTES,
    Rule_Effective_Date, Rule_Terminated_Date, Database_Name, Schema_Name, Table_Name,
    Status_Code, Type, Business_Domain, value, Constellation_Definition_With_Value_ID,
    Star_Schema_Name, Star_View_Name, Star_Column_Name)
values (source.Constellation_By_Value_Definition_ID, source.Name, source.Description, source.Notes,
    source.Rule_Effective_Date, source.Rule_Terminated_Date, source.Database_Name,
    source.Schema_Name, source.Table_Name, source.Status_Code, source.type,
    source.Business_Domain, Source.value, Source.Constellation_Definition_With_Value_ID,
    Source.Star_Schema_Name, Source.Star_View_Name, Source.Star_Column_Name)
when matched and(
    isnull(target.Constellation_By_Value_Definition_NAME,'NVL')!=isnull(SOURCE.NAME,'NVL') or
isnull(target.Constellation_By_Value_Definition_Description,'NVL')!=isnull(Source.Description,'NVL') or
        isnull(target.Constellation_By_Value_Definition_NOTES,'NVL')!=isnull(SOURCE.NOTES,'NVL') or
        isnull(target.Rule_Effective_Date,cast('1799-09-01' as datetime)) !=
isnull(Source.Rule_Effective_Date,cast('1799-09-01' as datetime)) or
        isnull(target.Rule_Terminated_Date,cast('1799-09-01' as datetime)) != isnull(Source.Rule_Terminated_Date,cast('1799-
09-01' as datetime)) or
        isnull(target.Database_Name,'NVL')!=isnull(SOURCE.Database_Name,'NVL') or
        isnull(target.Schema_Name,'NVL')!=isnull(SOURCE.Schema_Name,'NVL') or
        isnull(target.Table_Name,'NVL')!=isnull(SOURCE.Table_Name,'NVL') or
        isnull(target.Status_Code,'NVL')!=isnull(SOURCE.Status_Code,'NVL') or
        isnull(target.Type,'NVL')!=isnull(SOURCE.Type,'NVL') or
        isnull(target.Business_Domain,'NVL')!=isnull(SOURCE.Business_Domain,'NVL') or
        isnull(target.Constellation_Definition_With_Value_ID,-9)!=isnull(SOURCE.Constellation_Definition_With_Value_ID,-9) or
       isnull(target.Star_Schema_Name,'NVL')!=isnull(SOURCE.Star_Schema_Name,'NVL') or
       isnull(target.Star_View_Name,'NVL')!=isnull(SOURCE.Star_View_Name,'NVL') or
       isnull(target.Star_Column_Name,'NVL')!=isnull(SOURCE.Star_Column_Name,'NVL')
```

```
) then      update set
     target.Constellation_By_Value_Definition_NAME=SOURCE.NAME,
     target.Constellation_By_Value_Definition_DESCRIPTION=SOURCE.DESCRIPTION,
     target.Constellation_By_Value_Definition_NOTES=SOURCE.NOTES,
     target.Rule_Effective_Date=Source.Rule_Effective_Date,
     target.Rule_Terminated_Date=Source.Rule_Terminated_Date,
     target.Database_Name=SOURCE.Database_Name,
     target.Schema_Name=SOURCE.Schema_Name,
     target.Table_Name=SOURCE.Table_Name,
     target.Status_Code=SOURCE.Status_Code,
     target.Type=SOURCE.Type,
     target.Business_Domain=SOURCE.Business_Domain,
     target.Star_Schema_Name=SOURCE.Star_Schema_Name,
     target.Star_View_Name=SOURCE.Star_View_Name,
     target.Star_Column_Name=SOURCE.Star_Column_Name;
```

The next step, is to populate the bridge table between the group dimension and the constellation with value dimension. This is primarily based on the staging table that contains all of the required information, but must be joined to our new dimension tables to retrieve the keys for those tables. The SQL merge statement will insert new records and will delete existing records that are no longer in the source result set.

```
merge B_CONSTELLATION_BY_VALUE_BRIDGE as target
using (
SELECT distinct cd.Constellation_By_Value_Definition_DIM_KEY     ,cg.CONSTELLATION_BY_VALUE_GROUP_DIM_KEY
  FROM Constellation_With_Value_By_Group_BRIDGE cgb
   inner join D_Constellation_By_Value_Definition cd
      on cd.Constellation_Definition_With_Value_ID = cgb.Constellation_Definition_With_Value_ID
   inner join D_CONSTELLATION_BY_VALUE_GROUP cg
      on cg.CONSTELLATION_BY_VALUE_STRING=cgb.Constellation_With_Value_GROUP
) as source on (
source.Constellation_By_Value_Definition_DIM_KEY =
                                    target.Constellation_By_Value_Definition_DIM_KEY and
source.CONSTELLATION_BY_VALUE_GROUP_DIM_KEY =
```

```
                    target.CONSTELLATION_BY_VALUE_GROUP_DIM_KEY
)
when not matched by target then
insert (Constellation_By_Value_Definition_DIM_KEY,CONSTELLATION_BY_VALUE_GROUP_DIM_KEY)
values (source.Constellation_By_Value_Definition_DIM_KEY,
            source.CONSTELLATION_BY_VALUE_GROUP_DIM_KEY)
when not matched by source then delete;
```

The final step in the process is to populate the all dimensions bridge table that serves as a cross reference between any dimension table and the constellation by value table. This is populated with the unique record identifier for the dimension record and the primary key for the associated constellation by value dimension record.

```
merge B_Constellation_By_Value_ALL_DIMENSIONS_BRIDGE as target
using (
select distinct cfd.DW_Seq_ID,cd.Constellation_By_Value_Definition_DIM_KEY from
Constellation_With_Value_For_Dimensions as cfd
inner join D_Constellation_By_Value_Definition cd on cd.Constellation_Definition_With_Value_ID =
cfd.Constellation_Definition_With_Value_ID
) as source on (
source.Constellation_By_Value_Definition_DIM_KEY=target.Constellation_By_Value_Definition_DIM_KEY and
source.DW_Seq_ID=target.Constellation_By_Value_DW_Seq_ID )
when not matched by target  then
insert (Constellation_By_Value_Definition_DIM_KEY,Constellation_By_Value_DW_SEQ_ID)
values (source.Constellation_By_Value_Definition_DIM_KEY,source.DW_Seq_ID)
when not matched by source then delete;
```

This concludes the population of all the constellation objects. With the record identification and value structures populated we proceed to look at a proof of concept with queries and results.

Chapter 8. Proof of Concept Tests

With our selected subject area star schemas complete and all of the data structures and procedures in place, it is now possible to prove the concepts involved and perform a true study based on the use of constellations. For initial testing, multiple queries were developed and then used against our available subject areas. These queries employed both Constellation Identification and Value Association.

8.1 Constellation for Record Identification

Five separate tests of Constellation for Record Identification were performed. All of these tests identify a patient cohort, but each is based on different sources. Four of these queries have temporal elements that necessitate their use against a fact table. These queries identify a Discharge Abstract or Emergency NACRS encounter where the patient was registered in either home or residential care on the day of the encounter. The fifth query identifies records in the patient dimension, where the patient transitioned directly from an Alternate Level of Care (ALC) Hospital encounter to Residential Care.

Table 8.1: Constellation Record Identification Rules

Name	Type	SQL Code
Emergency Patient registered in Home Care	FACT BRIDGE	Select distinct fn.dw_seq_id from star.dbo.F_NACRS as fn inner join star.dbo.F_HCRS_Assessment as fah on fn.patient_dim_key=fah.Patient_DIM_KEY and fn.Registration_Date_Dim_Key between fah.Admission_Date_Dim_Key and (case when fah.Discharge_Date_Dim_Key<0 then 99999999 else fah.Discharge_Date_Dim_Key end)
Emergency Patient registered in Residential Care	FACT BRIDGE	Select distinct fn.dw_seq_id from star.dbo.F_NACRS as fn inner join star.dbo.F_CCRS_Assessment as fac on fn.patient_dim_key=fac.Patient_DIM_KEY and fn.Registration_Date_Dim_Key between fac.Entry_Date_Dim_Key and (case when fac.Discharge_Date_Dim_Key<0 then 99999999 else fac.Discharge_Date_Dim_Key end)
Discharge Abstract Patient registered Home Care	FACT BRIDGE	Select distinct fd.dw_seq_id from star.dbo.F_DAD as fd inner join star.dbo.F_HCRS_Assessment as fah on fd.patient_dim_key=fah.Patient_DIM_KEY and fd.Admission_Date_Dim_Key between fah.Admission_Date_Dim_Key and (case when fah.Discharge_Date_Dim_Key<0 then 99999999 else fah.Discharge_Date_Dim_Key end)
Discharge Abstract Patient registered Residential Care	FACT BRIDGE	Select distinct fd.dw_seq_ID from star.dbo.F_DAD as fd inner join star.dbo.F_CCRS_Assessment as fac on fd.patient_dim_key=fac.Patient_DIM_KEY and fd.Admission_Date_Dim_Key between fac.Entry_Date_Dim_Key and (case when fac.Discharge_Date_Dim_Key<0 then 99999999 else fac.Discharge_Date_Dim_Key end)
Patient Direct to Residential Care from ALC	DIMENSION	select distinct dp.dw_seq_id from star.dbo.D_Patient dp inner join star.dbo.F_DAD as fd on dp.Patient_DIM_KEY=fd.Patient_DIM_KEY inner join star.dbo.d_date as daddd on fd.Discharge_Date_Dim_Key=daddd.Date_Dim_Key inner join star.dbo.F_CCRS_ASSESSMENT as fca on dp.Patient_DIM_KEY=fca.Patient_DIM_KEY inner join star.dbo.d_date as fcaad on fca.Entry_Date_Dim_Key=fcaad.Date_Dim_Key where daddd.Date_Sequence<=fcaad.Date_Sequence and daddd.Date_Sequence>fcaad.Date_Sequence-7

Each of these tests is reviewed along with the results of the queries. The information is then used with the subject area Star Schema to demonstrate how this new information can be utilized.

8.1.1 Rule 1: Emergency Patient Registered in Home Care

Our first query selects the unique identifier for an emergency encounter in our NACRS fact table where that patient is in home care on the day that they were registered in emergency. The test query below returns 39423 emergency encounter records. A small subset of these records is provided in Table 8.2. The query joins between the NACRS and HCRS Assessment fact tables where it is the same patient and the Emergency registration date is between the date of admission and date of discharge from home care. For records where there is no discharge date from home care it is assumed that the patient is still in the program. Those with no admission are assumed to have been admitted to the home care program prior to 2010 when the HCRS assessment system and data was first supplied to CIHI.

```
Select distinct
 fn.dw_seq_id
 ,fn.patient_dim_key
 ,fn.Registration_Date_Dim_Key as Emergency_Registration_Date
 ,fah.Admission_Date_Dim_Key as Home_Care_Admission_Date
 ,fah.Discharge_Date_Dim_Key as Home_Care_Discharge_Date from
star.dbo.F_NACRS as fn
inner join star.dbo.F_HCRS_Assessment as fah
on fn.patient_dim_key=fah.Patient_DIM_KEY
and fn.Registration_Date_Dim_Key between fah.Admission_Date_Dim_Key
and (case when fah.Discharge_Date_Dim_Key<0 then 99999999 else
fah.Discharge_Date_Dim_Key end)
```

This query selects distinct records as the result set is a Cartesian product. Multiple records exist in the NACRS and HCRS Assessment tables for patients so the query will return a large number of records. However, we are only interested in distinct records from our NACRS table.

Table 8.2: NACRS Emergency Encounters for Home Care Patients

dw_seq_id	patient Dimension Key	Emergency Registration Date	Home Care Admission Date	Home Care Discharge Date
180008	9003	20110622	20050613	-1
503650	544821	20140115	20121005	20140512
587428	9939	20130608	20091105	20130729
381559	328149	20130730	20111229	-1

357854	10241	20120204	20120117	20131030
816744	205865	20120916	20031008	20140305
920887	413969	20120703	20110705	20130320
922828	4497	20130228	20120322	20130726
569340	68582	20140328	20050225	-1
178620	5585	20110512	20100628	-1
237956	409455	20111104	19980501	-1
565895	368079	20120103	20100618	-1
39838	316992	20120918	20000601	-1
184905	1974	20110921	20110729	-1
242716	9720	20111112	20050712	20131218
205936	8627	20110829	20100518	20130820
496286	149462	20110916	20100721	20121121
840579	260846	20120522	20110908	20141016
39836	316992	20130301	20000601	-1
82355	178284	20120908	20100323	-1

The DW_SEQ_ID is the unique identifier from our NACRS data and is the field of interest. In the results above, it can be seen that the Emergency Registration is between the home care admission and discharge date for each of these records (note: -1 for discharge date indicates the patient is not discharged).

8.1.2 Rule 2: Emergency Patient Registered in Residential Care

Our second query selects the unique identifier for an emergency encounter in our NACRS fact table where that patient is in residential care on the day that they were registered in emergency. The test query below, returns 10868 emergency encounter records, a sample of which is provided in Table 8.3. The query is similar to the one used for home care encounters. The NACRS table is joined to the Residential Care assessments based on the patient identifier where the date of registration in emergency occurred between the dates of entry and discharge for residential care (Note: A value of -1 for a discharge date indicates that no date of discharge was provided and it is assumed the patient is still in residential care)

```
Select distinct
```

```sql
fn.dw_seq_id
 ,fn.patient_dim_key
 ,fn.Registration_Date_Dim_Key as Emergency_Registration_Date
 ,fac.Entry_Date_Dim_Key as Residential_Care_Admission_Date
 ,fac.Discharge_Date_Dim_Key as Residential_Care_Discharge_Date
from star.dbo.F_NACRS as fn
inner join star.dbo.F_CCRS_Assessment as fac on
fn.patient_dim_key=fac.Patient_DIM_KEY and
fn.Registration_Date_Dim_Key between fac.Entry_Date_Dim_Key and
(case when fac.Discharge_Date_Dim_Key<0 then 99999999 else
  fac.Discharge_Date_Dim_Key end)
```

As with the home care query, this select statement would normally return a Cartesian product. However, as we are only interested in the unique identifier for the NACRS emergency record, the distinct values returned are all that is required.

Table 8.3: NACRS Emergency Encounters for Residential Care Patients

dw_seq_id	patient Dimension Key	Emergency Registration Date	Residential Care Admission Date	Residential Care Discharge Date
726094	1698	20120425	20110909	-1
751436	4065	20120819	20020313	-1
68962	8743	20111107	20080815	20120705
333588	4238	20131221	20131017	20150401
477467	1228	20120109	20111227	-1
885361	10836	20120427	20110302	20120503
672091	2036	20121206	20120918	20130402
448967	7547	20110923	20071108	-1
176607	9009	20120709	20120514	20120710
366702	8588	20111203	20060620	20141129
451656	2485	20110430	20110303	-1
37292	6906	20130208	20121025	-1
356627	4769	20130913	20130802	20131023
428769	5612	20120330	20101116	-1
313159	4000	20110729	20101029	20140327
454377	6633	20120112	20111219	20141029
239419	8273	20140226	20120216	20140920
356756	952	20140218	20130725	20150701
607038	6505	20140112	20111027	-1
673658	857	20130909	20100830	20131031

8.1.3 Rule 3: Discharge Abstract Patient registered in Home Care

Our third rule looks at the Discharge Abstract table to identify patients that are also in home care. This query selects the unique identifier for a discharge abstract record in our DAD fact table where that patient is in home care on the day that they were admitted. The query below returns 25447 discharge abstract records, a subset of which is provided in Table 8.4. The query is similar in structure and filtering to those used for emergency encounters. The DAD table is joined to the Home Care assessments based on the patient identifier and the date of registration for the discharge abstract record occurring between the dates of admission and discharge from home care.

```sql
Select distinct
fd.dw_seq_id
,fd.Patient_DIM_KEY
,fd.Admission_Date_Dim_Key
,fah.Admission_Date_Dim_Key
,fah.Discharge_Date_Dim_Key
from star.dbo.F_DAD as fd
inner join star.dbo.F_HCRS_Assessment as fah on fd.patient_dim_key=fah.Patient_DIM_KEY
and fd.Admission_Date_Dim_Key between fah.Admission_Date_Dim_Key and
(case when fah.Discharge_Date_Dim_Key<0 then 99999999
   else fah.Discharge_Date_Dim_Key end)
```

Table 8.4: Discharge Abstract records Where Patient in Home Care

dw_seq_id	patient Dimension Key	Discharge Abstract Admission Date	Home Care Admission Date	Home Care Discharge Date
1037622	21142	20110403	20070718	20110723
1519881	6293	20130604	20110301	20140310
1317602	194988	20120426	20081104	-1
995057	2441	20110715	20070725	20110830
1222813	7591	20121119	-2	20131029
1405447	379299	20130401	20070607	-1
1405510	13048	20130410	20110407	20130425
965580	331649	20110620	20110223	-1
966035	5153	20110421	20101213	20131009
1186481	65442	20120712	20120628	-1
985208	10000	20120109	20120106	20131016
1495105	314163	20131010	20110525	20140320
990414	225245	20110615	20110408	20110726
1211022	336874	20120513	20110715	-1
1223841	321541	20120602	20101020	-1
1448496	382997	20130916	20100628	-1
1158372	51988	20110531	20050701	-1
1411954	416054	20130624	20110314	-1
1414322	224382	20130626	20090305	-1
1044491	412	20111218	20100618	20120828

As with the NACRS data, the use of a distinct clause on the query is important as the query will return multiple records and we are only interested in the unique identifiers. In addition, anomalies in the data such as home care encounters that begin and end during a hospital stay and overlapping home care episodes were found in the data. Such data quality issues are expected in large complicated data sets. These issues can lead to duplicate records which can be addressed using constellation to identify records that reflect data quality issues.

8.1.4 Rule 4: Discharge Abstract Patient registered in Residential Care

Our fourth rule looks at the Discharge Abstract table and identifies hospital patients that are also in residential care. The query is similar to those presented before and returns 8495 discharge abstract records. A small subset of the data results is provided in Table 8.5. The query selects the unique identifier from the DAD table which is joined to the Residential Care assessments based on the patient identifier and the date of registration for the discharge abstract record occurring between the dates of entry and discharge for residential care.

```
Select distinct
fd.dw_seq_id
,fd.Patient_DIM_KEY
,fd.Admission_Date_Dim_Key
,fac.Entry_Date_Dim_Key
,fac.Discharge_Date_Dim_Key
from star.dbo.F_DAD as fd
inner join star.dbo.F_CCRS_Assessment as fac on fd.patient_dim_key=fac.Patient_DIM_KEY
and fd.Admission_Date_Dim_Key between fac.Entry_Date_Dim_Key and
(case when fac.Discharge_Date_Dim_Key<0 then 99999999
   else fac.Discharge_Date_Dim_Key end)
```

Again, the distinct clause is used to limit the results to the unique identifier for the Discharge Abstract record. A Cartesian product between abstracts and residential care assessments exists as multiple records exist in both tables due to overlapping dates and anomalies in the data such as overlapping residential care episodes.

Table 8.5: Discharge Abstract records where Patient in Residential Care

dw_seq_id	Patient Key	Discharge Abstract Admission Date	Residential Care Admission Date	Residential Care Discharge Date
1141925	7444	20110429	20090319	20140401
1491153	899	20130726	20130326	20130815
1062098	2366	20120117	20111221	20140820
1561912	10546	20130829	20020313	20140205
984684	10530	20120310	20100831	20130313
1281978	523	20121220	20120914	20140309
1204796	4682	20130214	20090827	20130304
1307996	1372	20121205	20120312	20130215
1445918	4577	20130818	20111110	20150304
1457250	3242	20140122	20140115	-1
1324662	10612	20120621	20080331	20140502
1054042	4180	20110621	20090501	20111001
1192601	927	20120913	20120703	20130827
1259749	10032	20120704	20110304	20130725
1209977	4106	20120325	20100609	20120907
1610256	2770	20130704	20080317	20150309
1046927	3088	20120317	20081219	-1
996925	194	20110721	20070814	20110814
970306	9936	20110823	20090825	20111007
1126035	5503	20120302	20110627	-1

8.1.5 Rule 5: Patient admitted directly to Residential Care from Hospital Alternate Level of Care

Our fifth rule looks at a patient cohort as well. In this case, we are identifying a cohort of patients that are not based on temporal elements that require focusing on the fact records for NACRS emergency encounters or Discharge abstract records. Instead, this rule identifies the patients in the dimension directly so that the cohort can be used across all of our fact tables for the complete patient history.

The query selects the patient unique identifier from the patient dimension and joins it to both the Discharge Abstract table and the Residential Care assessment table for that patient. Filtering is then done to limit the records to those where the patient discharge date in the abstract record is the same as the entry date in Residential Care. Because our dates are numeric representations and do not use a date data type, the query joins to the date dimension to utilize a numeric sequence field and allow for a calculation of the difference between the two dates. This was done in testing to evaluate the period of time between the discharge from hospital and admission to residential care. The query returns 2831 records, a subset is provided in Table 8.6.

```
select distinct
dp.dw_seq_id
,fd.Discharge_Date_Dim_Key
,fac.Entry_Date_Dim_Key
,fac.Discharge_Date_Dim_Key
,fcaad.Date_Sequence-daddd.Date_Sequence
from star.dbo.D_Patient dp
inner join star.dbo.F_DAD as fd on dp.Patient_DIM_KEY=fd.Patient_DIM_KEY
inner join star.dbo.d_date as daddd on fd.Discharge_Date_Dim_Key=daddd.Date_Dim_Key
inner join star.dbo.F_CCRS_ASSESSMENT as fac on dp.Patient_DIM_KEY=fac.Patient_DIM_KEY
inner join star.dbo.d_date as fcaad on fac.Entry_Date_Dim_Key=fcaad.Date_Dim_Key
where
fd.Discharge_Date_Dim_Key=fac.Entry_Date_Dim_Key
```

A distinct clause is used again to limit the number of returned records to the unique identifier for the patient. This is due to the grain declared for the residential care assessment table which represents Assessments and not the Episode of care. A second anomaly existed for a small number of records where a patient was previously in residential care and was discharged before going to hospital at a later date. If we were interested in a patient's first episode, this would need to be accounted for and would require additional information. The supplied data for this research was for a specific time period so this was not performed.

Table 8.6: Patient directly admitted to Residential Care from Hospital

dw_seq_id	Abstract Discharge Date	Residential Care Entry Date	Residential Care Discharge Date	Delay in Residential Admission
4756212	20120424	20120424	20150701	0
4756213	20120131	20120131	20120426	0
4756225	20121204	20121204	20131211	0
4756232	20110913	20110913	20120319	0
4756233	20130911	20130911	20150701	0
4756236	20130604	20130604	20130720	0
4756245	20130319	20130319	-1	0

4756246	20130325	20130325	20131001	0
4756247	20110401	20110401	20120206	0
4756248	20120418	20120418	-1	0
4756250	20110615	20110615	20110704	0
4756255	20110509	20110509	20150128	0
4756256	20111117	20111117	20130311	0
4756258	20130215	20130215	-1	0
4756265	20130621	20130621	-1	0
4756270	20121011	20121011	20150114	0
4756272	20110428	20110602	20110707	0
4756276	20121126	20130201	20131225	0
4756277	20120828	20121003	20150619	0
4756279	20131009	20131204	20140701	0

8.1.6 Constellation for Record Identification Results

With the completion of the queries and the processing provided by constellation, we can now bring this data into our NACRS emergency encounters and Discharge Abstract subject areas. Once this data is included through constellation, it can easily be used within a Business Intelligence or OLAP environment. All of the resulting tables are produced using Microsoft SQL Server Analysis server with Microsoft Excel as a front end display tool. All of these tables could be produced through any common BI toolset.

Table 8.7 shows the encounter count, total length of stay, and the average length of stay for emergency encounters. Examining the data shows that the home and residential care patients make up approximately five percent of the emergency encounters, but have an average length of stay that is double that of the patients that are not in the defined cohorts. It is assumed that this is indicative of the health of these cohorts; but without diagnosis or intervention data, analysis is not possible.

Table 8.7: Emergency Encounter Count, Total Length of Stay, and Average Length of Stay by defined Cohort

Row Labels	Encounter Count	LOS HOURS	Average LOS Hours
None	898629	4005696.75	4.457564523
Patient Home Care	39372	365724.9688	9.288960905
Patient in Residential Care	10868	112192.5	10.32319654
Grand Total	**946582**	**4460454.5**	**4.712169152**

Table shows the same Cohorts filtered for those patients who are admitted via ground ambulance and includes the count of encounters, average wait time to Physician Initial Assessment (PIA), and the average wait time to inpatient admission.

Table 8.8: Emergency Encounter Count, Average Wait time for Physician Assessment and Inpatient Admission

ADMIT VIA AMBULANCE	G

Row Labels	Encounter Count	Average Wait Time to PIA Hours	Average Wait Time to Inpatient Hours
None	153538	0.737734816	2.869944248
Patient Home Care	21891	0.800437216	5.538795909
Patient in Residential Care	9062	0.74169604	5.185175303
Grand Total	**182620**	**0.745256544**	**3.282955385**

In this table, the wait time for physician assessment varies little between the cohorts. The majority of the emergency encounters from residential care are shown to arrive by ambulance, which is expected as this cohort now resides in a care facility. Additional diagnosis and intervention information would be required to determine why there is a difference in average admission wait times.

Table 8.9 looks at the number of days in Alternate Level of Care, Acute Care, and the count of abstract records by the primary CCI intervention for the discharge abstract subject area filtered to the cohort of patients in residential care. It is immediately apparent that Musculoskeletal Interventions on the Hip and Leg are the single largest primary intervention for acute days in hospital for this patient group. Several observations and indicators in the Residential Care Assessments area relate to this and show the need for prevention of falls in residential care.

Table 8.9: Residential Care Patients Primary CCI Intervention

Row Labels	Residential Care Patient ALC DAYS	Abstract Count	ACUTE DAYS
1, Physical/Physiological Therapeutic Interventions	**422**	**3433**	**22827**
1.AA - 1.BZ, Therapeutic Interventions on the Nervous System	**78**	**1235**	**2056**
1.CC - 1.CZ, Therapeutic Interventions on the Eye and Ocular Adnexa	**0**	**345**	**15**
1.EA - 1.FX, Therapeutic Interventions on the Orocraniofacial Region	**0**	**36**	**41**
1.GA - 1.GZ, Therapeutic Interventions on the Respiratory System	**15**	**175**	**3050**
1.HA - 1.LZ, Therapeutic Interventions on the Cardiovascular System	**7**	**290**	**3136**
1.MA - 1.MZ, Therapeutic Interventions on the Lymphatic System	**0**	**1**	**27**
1.NA - 1.OZ, Therapeutic Interventions on the Digestive and Hepatobiliary Tracts and Other Sites within the Abdominal Cavity NEC	**86**	**363**	**4576**
1.PB - 1.RZ, Therapeutic Interventions on the Genitourinary System	**53**	**320**	**2468**
1.SA - 1.WZ, Therapeutic Interventions on the Musculoskeletal System	**168**	**588**	**6930**
1.SA - 1.SZ, Therapeutic Interventions on the Spine, Trunk and Pelvis	24	41	745
1.TA - 1.TZ, Therapeutic Interventions on the Shoulder and Arm (excluding hand and wrist)	0	16	116
1.UB - 1.UZ, Therapeutic Interventions on the Hand and Wrist	0	8	14
1.VA - 1.VZ, Therapeutic Interventions on the Hip and Leg	144	499	5807
1.WA - 1.WV, Therapeutic interventions on the Ankle and Foot	0	24	248
1.YA - 1.YZ, Therapeutic Interventions on the Skin, Subcutaneous Tissue and Breast	**0**	**66**	**155**
1.ZX - 1.ZZ, Therapeutic Interventions on the Body NEC	**15**	**14**	**373**
2, Diagnostic Interventions	**105**	**265**	**1783**
3, Diagnostic Imaging Interventions	**0**	**51**	**316**
Grand Total	**2848**	**8451**	**63577**

Our last example for the area of constellation by identification is Table 8.10 which looks at the Depression Rating Scale for patients on their admission assessment (Assessment type 1) to residential care. Anecdotal evidence from home and community care professionals has suggested that patients admitted direct to residential care have higher levels of depression, as they have difficulty in transitioning to the environment compared to those who transition from home care. These results however, show no significant difference in the patient cohort for those patients admitted to residential care from hospital.

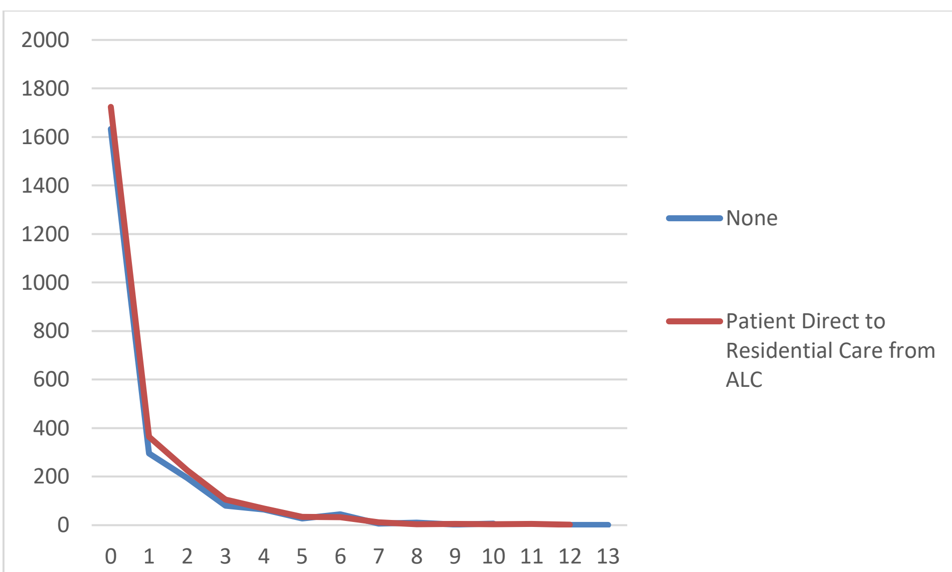
Table 8.10: Depression rating Scale CCRS initial Assessment by Patient Cohort (Direct admit from ALC)

AA8 ASSESSMENT TYPE	1

CCRS ASSESSMENT Count															
Row Labels	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**Grand Total**
None	1633	296	194	81	64	28	45	6	10	3	6		1	1	2368
Patient Direct to Residential Care from ALC	1724	364	226	106	69	34	33	12	4	5	4	5	2		2588
Grand Total	**3357**	**660**	**420**	**187**	**133**	**62**	**78**	**18**	**14**	**8**	**10**	**5**	**3**	**1**	**4956**

An alternate view of the data is shown in the Figure 8.1. It is evident that there is little difference in the two patient cohorts in terms of depression scale, at the time of the initial assessment.

Figure 8.1: Depression rating Scale CCRS initial Assessment by Patient Cohort (Direct admit from ALC)



8.2 Constellation by Value

Seven separate tests were performed for constellation value association. Three of these tests are based on temporal factors and form part of a fact bridge table structure. Two of these associate the number of emergency encounters to our Home and Residential care assessments. The third provides a residential care assessment sequence number for individual patients by order of the date of assessments. The final four associate selected residential care quality indicators to the facility.

Table 8.11: Constellation Queries by Value.

Name	type	SQL Code
Emergency Encounters Last 90 Days	FACT BRIDGE	select distinct fah.DW_SEQ_ID ,(select count(*) from star.dbo.F_NACRS as fn inner join star.dbo.D_Date as dd on fn.Registration_Date_Dim_Key=dd.Date_Dim_Key where fn.patient_dim_key=fah.Patient_DIM_KEY and dd.Date_Sequence between df.Date_Sequence-90 and df.Date_Sequence) as value from star.dbo.F_HCRS_ASSESSMENT as fah inner join star.dbo.d_date as df on df.Date_Dim_Key=fah.Assessment_Reference_Date_Dim_Key
Emergency Encounters Last 90 Days	FACT BRIDGE	select distinct fah.DW_SEQ_ID ,(select count(*) from star.dbo.F_NACRS as fn inner join star.dbo.D_Date as dd on fn.Registration_Date_Dim_Key=dd.Date_Dim_Key where fn.patient_dim_key=fah.Patient_DIM_KEY and dd.Date_Sequence between df.Date_Sequence-90 and df.Date_Sequence) as value from star.dbo.F_CCRS_ASSESSMENT as fah inner join star.dbo.d_date as df on df.Date_Dim_Key=fah.Assessment_Date_Dim_Key
Patient Assessment Number	FACT BRIDGE	select dw_seq_id ,ROW_NUMBER() over (partition by patient_dim_key order by fca.ASSESSMENT_DATE_DIM_KEY) as value from F_CCRS_ASSESSMENT as fca order by patient_dim_key,ASSESSMENT_DATE_DIM_KEY
Facility Late-Loss ADL Worsened Score	DIMENSION	select df.dw_seq_id ,(sum(QI_ADL01_N) * 100) / sum(QI_ADL01_D) as value from F_CCRS_ASSESSMENT as fca inner join D_Facility as df on df.Facility_Dim_Key=fca.Facility_Dim_Key group by df.Facility_Dim_Key
Facility Patients Falling Quality Indicator	DIMENSION	select df.dw_seq_id ,(sum(QI_FAL02_N) * 100) / sum(QI_FAL02_D) as value from F_CCRS_ASSESSMENT as fca inner join D_Facility as df on df.Facility_Dim_Key=fca.Facility_Dim_Key group by df.dw_seq_id
Facility Cognitive Loss Worsened Score	DIMENSION	select df.dw_seq_id ,(sum(QI_COG01_N) * 100) / sum(QI_COG01_D) as value from F_CCRS_ASSESSMENT as fca inner join D_Facility as df on df.Facility_Dim_Key=fca.Facility_Dim_Key group by df.dw_seq_id
Facility Mood Worsened Score	DIMENSION	select df.dw_seq_id ,(sum(QI_MOD4A_N) * 100) / sum(QI_MOD4A_D) as value from F_CCRS_ASSESSMENT as fca inner join D_Facility as df on df.Facility_Dim_Key=fca.Facility_Dim_Key group by df.dw_seq_id

In the following sections, each of the three fact bridge queries will be reviewed separately and the four Dimension queries, which are identical except in return value, will be examined as a group.

8.2.1 Emergency Encounters Last 90 Days for Home Care Patient on date of Assessment

Our first query associates the number of emergency encounters experienced over the previous ninety days with the home care assessment for a patient. It does this by selecting the unique assessment record identifier and employing a subquery to select the count of records from the NACRS emergency encounter fact table for that patient where the date of the emergency encounter is within the last 90 days.

```
select  fah.DW_SEQ_ID ,
   (select count(*) from star.dbo.F_NACRS as fn
        inner join star.dbo.D_Date as dd on fn.Registration_Date_Dim_Key=dd.Date_Dim_Key
        where fn.patient_dim_key=fah.Patient_DIM_KEY
        and dd.Date_Sequence between df.Date_Sequence-90 and df.Date_Sequence) as value
    , fah.Patient_dim_key
    ,fah.Assessment_Reference_Date_Dim_Key
 from star.dbo.F_HCRS_ASSESSMENT as fah
       inner join star.dbo.d_date as df on df.Date_Dim_Key=fah.Assessment_Reference_Date_Dim_Key
```

Results from the query above are shown in Table 8.12. The Patient identifier and Assessment Date are provided as reference.

Table 8.12: Emergency Encounters Count Last 90 Days for Home Care Assessment

DW_SEQ_ID	Emergency Encounters	Patient Key	Assessment Date
4625427	0	366083	20101116
4626622	0	363358	20111209
4629579	0	5435	20100824
4640942	1	220399	20130102
4641904	1	2556	20130307
4641987	1	324471	20130131
4644597	0	688338	20120830
4645077	0	5476	20110301
4647707	1	7039	20111019
4630762	0	687462	20100420
4634761	0	2890	20101017
4635133	0	286894	20110222
4635696	0	1304	20110127
4639417	0	201927	20110902

4638394	0	46348	20111220
4620857	0	2538	20100909
4623376	0	331997	20120201
4625571	0	3988	20120926
4631477	0	74973	20100729
4638568	4	231041	20120824

As a cross check for the last record in Table 8.12 the query below will select the emergency encounters for the patient identified as 231041 for the time period from May 26th to August 24th of 2012 and return the results in Table 8.13. This represents 90 days prior to the Assessment in August. This shows that our first query is returning the correct result count (Four Encounters)

```
select * from star.dbo.F_NACRS as fn
inner join star.dbo.D_Date as dd on fn.Registration_Date_Dim_Key=dd.Date_Dim_Key
cross join (select * from star.dbo.D_Date where Date_Dim_Key=20120824 ) as df
where fn.patient_dim_key=231041 and dd.Date_Sequence between df.Date_Sequence-90 and df.Date_Sequence
```

Table 8.13: Emergency Encounters for Patient 231041 between 20120526 and 20120824

Patient Key	Registration Date	dw_seq_id	LOS Hours	Facility Key
231041	20120816	86780	16.4	6
231041	20120712	86786	9.1	3
231041	20120724	86785	2.4	3
231041	20120601	86787	14.6	3

8.2.2 Emergency Encounters Last 90 Days for Residential Care Patient on date of Assessment

Our second query is very similar to our first, it associates the number of emergency encounters experienced over the previous ninety days with the residential care assessment for a patient. The same method is employed which selects the unique assessment record identifier and then uses a subquery to select the count of records from the NACRS emergency encounter fact table for the same patient where the date of the emergency encounter is within the last 90 days. A subset of the results from the query is provided in Table 8.14.

```
select fah.DW_SEQ_ID
  ,(select count(*) from star.dbo.F_NACRS as fn
```

```
    inner join star.dbo.D_Date as dd on fn.Registration_Date_Dim_Key=dd.Date_Dim_Key
    where fn.patient_dim_key=fah.Patient_DIM_KEY
    and dd.Date_Sequence between df.Date_Sequence-90 and df.Date_Sequence) as value
      fah.Patient_dim_key
     ,fah.Assessment_Reference_Date_Dim_Key
from star.dbo.F_CCRS_ASSESSMENT as fah
inner join star.dbo.d_date as df on df.Date_Dim_Key=fah.Assessment_Date_Dim_Key
```

Table 8.14: Emergency Encounters for 90 days Prior to Residential Care Assessment

DW_SEQ_ID	Emergency Encounters	Patient Key	Assessment Date
4746454	0	9519	20140114
4728534	0	3845	20121101
4726947	0	9038	20121113
4726777	0	9038	20120410
4708187	0	8261	20110410
4755666	0	4053	20120810
4690129	1	11007	20111019
4741360	0	10441	20131020
4749857	1	6835	20131223
4712995	0	5940	20120125
4703336	0	7816	20120511
4742711	0	5271	20130605
4688278	0	4845	20121101
4693760	0	9033	20140118
4705609	0	9579	20130826
4742590	0	2703	20120929
4731985	0	951	20130603
4722893	0	9267	20130409
4691871	0	7813	20130812
4736701	0	7687	20130827

8.2.3 Residential Care Assessment Sequence Number by Assessment date

Our third query is also based on the Residential Care Assessment. In this query, a sequential number is created against our residential care assessments ordered by assessment date for each patient. This was done for the purpose of looking at changes in our patient population over time. By developing a quick numerical sequence, the changes in the patient population over time can be tracked and different cohorts can be compared. Individual patients can also be examined to identify any changes in health during a course of treatment or other interventions.

```
select dw_seq_id
   ,ROW_NUMBER() over (partition by patient_dim_key order by fca.ASSESSMENT_DATE_DIM_KEY ) as value
  ,patient_dim_key
  ,ASSESSMENT_DATE_DIM_KEY
  ,Facility_Dim_Key
from F_CCRS_ASSESSMENT as fca
order by patient_dim_key,ASSESSMENT_DATE_DIM_KEY
```

The query is simple and employs the SQL Server Ranking function ROW_NUMBER(), the numbering is partitioned for different patients and ordered by the assessment date. Similar functionality exists in other relational databases, though some variations may exist. Results are provided in Table 8.15.

Table 8.15: CCRS Assessment Sequence Number for Patient by Assessment Date

dw_seq_id	Assessment Number	Patient Key	Assessment Date	Facility Key
4694030	1	1	20110521	59
4694095	2	1	20110802	59
4694166	3	1	20111026	59
4694241	4	1	20120122	59
4694310	5	1	20120419	59
4699781	1	2	20110626	52
4699951	2	2	20110919	52
4700053	3	2	20111213	52
4700160	4	2	20120307	52
4700274	5	2	20120619	52
4700368	6	2	20120912	52
4700500	7	2	20121206	52
4700605	8	2	20130301	52
4700726	9	2	20130525	52
4711636	1	3	20110401	46
4711513	2	3	20110701	46
4711724	3	3	20110929	46
4711903	4	3	20111228	46
4711982	5	3	20120327	46
4711542	6	3	20120701	46

8.2.4 Facility Quality Indicator Scores from Residential Care

Our final four queries for value association are based on CCRS. Several quality indicators have been developed by the Canadian Institute for Health Information (CIHI) that look at the changes in a patient's health. These quality indicators compare the current assessment for a patient with the previous assessment. Specifically they examine changes in ADL scores, Depression, Cognitive Performance, and other aspects in order to test if a patient's health improved or worsened. The results of these individual quality tests are aggregated and calculated as a percentage of the population based on which improved or worsened over a period of time. CIHI does these calculations as quarterly measures, though each quarter contains a year's worth of data to ensure a large enough volume for statistical calculations. For this study, the aggregate calculation for the entire data set was performed and the value returned was associated to our facility identifier.

```
select  df.dw_seq_id
    ,(sum(QI_ADL01_N) * 100) / sum(QI_ADL01_D) as value
from F_CCRS_ASSESSMENT as fca
    inner join D_Facility as df on df.Facility_Dim_Key=fca.Facility_Dim_Key
group by df.dw_seq_id
```

This query calculates the quality indicator for ADL improved as a percentage value. The aggregate result value is then grouped by the unique identifier for the facility to give us the association results.

Four separate quality indicator calculations were done. Late Loss ADL score worsened, Mood Worsened, Patient falls in the previous 30 days, and Cognitive Loss Worsened. These were combined in one query to provide the results in Table 8.16.

```
select  df.dw_seq_id
    ,(sum(QI_ADL01_N) * 100) / sum(QI_ADL01_D) as Late_Loss_value
    ,(sum(QI_FAL02_N) * 100) / sum(QI_FAL02_D) as Patient_Falls_value
    ,(sum(QI_COG01_N) * 100) / sum(QI_COG01_D) as Cognitive_Loss_value
    ,(sum(QI_MOD4A_N) * 100) / sum(QI_MOD4A_D) as Mood_Worsened_value
from F_CCRS_ASSESSMENT as fca
    inner join D_Facility as df on df.Facility_Dim_Key=fca.Facility_Dim_Key
group by df.dw_seq_id
```

Table 8.16: CCRS Assessment Quality Indicators by Facility.

dw_seq_id	Late Loss	Patient Falls	Cognitive Loss	Mood Worsened
8429196	15	34	2	11
8429197	26	16	20	15
8429198	21	4	19	11
8429199	10	17	7	6
8429200	23	12	17	13
8429201	11	11	10	6

8429202	14	3	12	13
8429203	17	14	11	6
8429204	4	2	12	12
8429205	10	12	11	9
8429206	4	10	4	9
8429207	18	14	13	7
8429208	36	20	18	16
8429209	16	14	15	14
8429210	12	14	10	6
8429211	18	4	16	16
8429212	17	11	16	14
8429213	18	13	7	5
8429214	50	0	0	50
8429215	23	13	24	27

8.2.5 Constellation by Value results

With the completion of the queries and constellation processing, the results can now be used within a BI or OLAP environment. Table 8.17 shows the count of home care assessments by frequency of emergency encounters in the 90 days prior to the home care assessment. The assessment field P4b, representing the number of emergency encounters entered on the assessment is also provided for comparison.

Examining the data shows the discrepancy between these two values. Actual Emergency Encounters, as reported in NACRS, are higher than reported in HCRS. This is understandable given the nature of the HCRS field and that field P4b is likely a verbal response. This discrepancy shows the need for interrelating data sets in order to retrieve accurate information.

Table 8.17: Assessment Count by NACRS and HCRS Emergency Encounters

Assessment Count	HCRS Emergency Encounter										
NACRS											
Emergency encounters	0	1	2	3	4	5	6	7	8	9	Grand Total
0	14134	1865	339	125	49	15	16	7	4	11	15683
1	3378	935	78	24	6	3			2	1	4327
2	966	338	107	19	3	1	1	1		1	1418
3	277	103	59	25	2		1	1		2	469
4	101	42	25	16	9		1				194
5	36	15	6	10	3	4	1	1		1	75
6	15	7	4	2	1	4	4				37
7	7	2	4	1		1				1	16
8	6	3	1	1	1	1	1				14
9	3	1	1		1					1	7
10	2	1		1		1				2	7
11	1	1									2
12	4					1				1	6
13	1	1								1	3
15	1		1								2
16	1										1
17			1							1	2
18	1	1									2
25	1										1
Grand Total	16955	3132	605	218	72	30	24	9	5	23	19282

Table 8.18 shows the emergency encounter for continuing care assessments. Again, we see discrepancies between emergency encounters as reported in the continuing care assessments and those reported in NACRS. The discrepancies in this case are not as significant, as it is likely that records would exist in the continuing care facility that tracked this information.

Table 8.18: Assessment Count by NACRS and CCRS Emergency Encounters

Assessment Count	CCRS Emergency Encounters									
NACRS										
Emergency encounter	0	1	2	3	4	5	10	11	13	Grand Total
0	60415	737	71	12	3	1	2	3	1	61245
1	5503	980	25	5	1					6514
2	1076	206	75	5	1					1363
3	228	49	27	11	2					317
4	63	11	3	3	2					82
5	30	5			1	1				38
6	13	1								14
7	7	2								9
8	5			1						6
9	2									2
10	2									2
14	1									1
36	1									1
Grand Total	69594	1991	201	37	10	2	2	3	1	69594

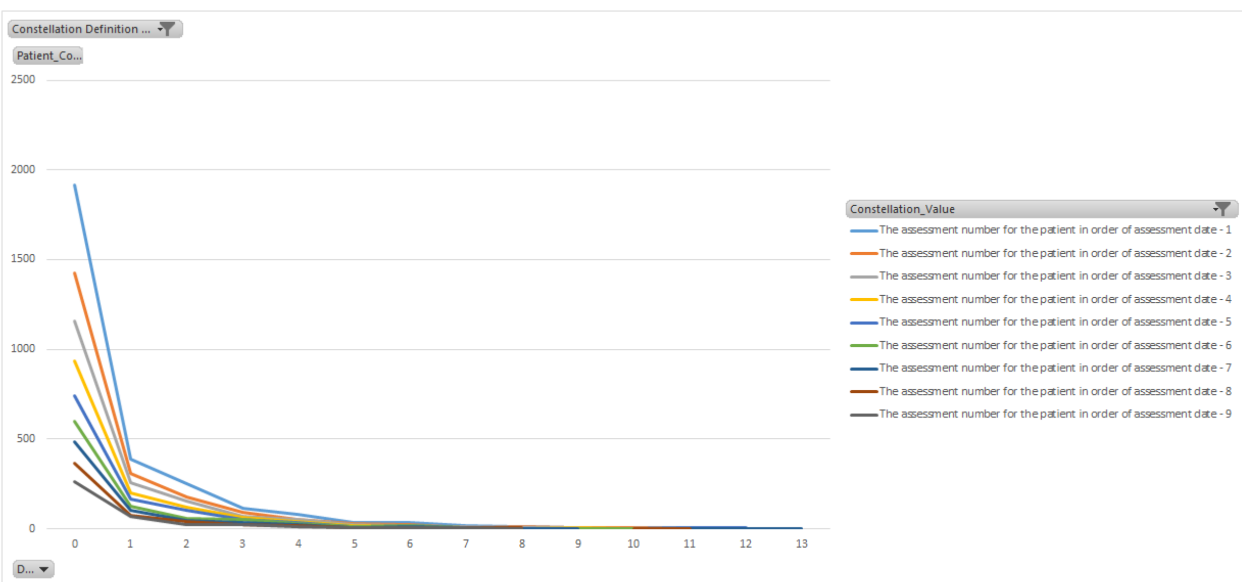
In the next example we combine two separate associations with the continuing care assessment data to demonstrate the additional functionality of combining more than one association rule. In table 8.18 we look at the patient cohort for direct admission to residential care from hospital alternate level of care and the depression of those patients over time. The time factor is provided by an association rule that identifies the order of assessments for the patient. Observing this data we see that there is no real change in depression of this group over time although this is more apparent in Figure 8.2.

Table 8.19: Depression Rating Scale for Direct ALC Admit Patients by Assessment Number

Cohort	Patient Direct to Residential Care from ALC														
Patient_Count	Depression Rating Scale														
Assessment Number	0	1	2	3	4	5	6	7	8	9	10	11	12	13	Grand Tota
1	1915	386	250	115	80	33	32	15	12	5	6	6	1	1	2857
2	1428	308	175	89	53	29	22	7	9	5	4	2	2		2133
3	1160	259	154	67	49	20	15	10	7	3		2	1		1747
4	937	201	117	65	39	19	15	8	1	4	2	1	1		1410
5	739	168	100	53	36	11	16	5	2	2	1	3	3		1139
6	598	127	58	51	30	11	15	6	1	2	1	1	1		902
7	483	102	47	36	24	7	12	3	1	1		1	2	1	720
8	365	74	39	25	18	7	7	4	3		1	1		2	546
9	260	71	24	20	10	6	3	3	1						398
Grand Total	2276	840	579	313	211	98	84	39	31	16	14	13	6	3	2857

Below is the matching graph for Table 8.18.

Figure 8.2: Depression Rating Scale for Direct ALC Admit Patients by Assessment Number



Although this example does not show any change in the population, the functionality of this constellation value could be valuable for following treatment programs and outcomes for selected patient populations.

The final example of the use of constellation by value investigates the CCRS assessment quality indicators. This table brings in the percentage of patients with cognitive loss and the percentage with mood deterioration. The results show that the deterioration in both quality measures seems to follow the same trend.

Table 8.20: Patient Count by Facility Cognitive Loss and Mood Deterioration

Patient_Count	Percentage with Cognitive Loss																						
Percentage With Mood deterioration	0	2	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	23	24	Grand To
0	5																						5
2						184																	184
3							595	299															893
4						406																	406
5				99		437																	533
6						489			369	251													1104
7										546	307	287											1139
8																		198					198
9			169			122			175	223													688
10						265			90														355
11		26									416			115		272		136		451			1413
12						367					161								391	502			1417
13											135					127							262
14					150									152	255								557
15																			393	299			691
16															125	133	42						300
17																275							275
18													145								149		293
19								150															150
20																			98				98
21																	109						109
22																101					91		192
24														140									140
25														115									115
27																						146	146
50	2																						2
Grand Total	7	26	169	99	150	2243	595	449	634	1019	1019	287	145	518	380	907	151	334	879	1248	240	146	11053

8.3 Constellation by Relation

Our last tests for constellation were performed for constellation by relation. From a SQL perspective, this form of association can vary in complexity. The test examples used here are relatively simplistic;

however even with simple SQL statements, the difficulty is in understanding the nature of the information and the relationships. These queries normally involve defining complex relationships between Star Schema fact tables, though in some rare situations these rules could be used to relate dimension tables.

Between star schema subject areas the relationships can be exceedingly complex. For example, when Hospital Discharge Abstract records are related to a patient assessment is it to look for the following assessment to see the result of an intervention, or to look for a prior assessment to see the cause or diagnosis. A patient assessment may relate to multiple Hospital Discharge Abstract records in multiple ways. In essence we have a many-to-many-to-many relationship. This becomes limiting from a technology perspective in that most BI tools are not capable of dealing with this level of complexity. Some tools, such as SQL Server Analysis Server, support this level of complexity as discussed by Tkachuk [74] but these are not common.

Table 8.21: Test Constellation Reference Rules

Name	Child Table	Parent Table	SQL Code
Prior Emergency Encounter	F_CCRS_ASSESSMENT	F_NACRS	select distinct dw_seq_id as child_dw_seq_id,isnull((select top 1 dw_seq_id from star.dbo.F_NACRS as fn where fn.patient_dim_key=fca.Patient_DIM_KEY and fn.Registration_Date_Dim_Key<fca.Assessment_Date_Dim_Key order by fn.Registration_Date_Dim_Key desc),-1) as parent_dw_seq_id from Star.dbo.F_CCRS_ASSESSMENT as fca
Next Emergency Encounter	F_CCRS_ASSESSMENT	F_NACRS	select distinct dw_seq_id as child_dw_seq_id,isnull((select top 1 dw_seq_id from star.dbo.F_NACRS as fn where fn.patient_dim_key=fca.Patient_DIM_KEY and fn.Registration_Date_Dim_Key>fca.Assessment_Date_Dim_Key order by fn.Registration_Date_Dim_Key asc),-1) as parent_dw_seq_id from Star.dbo.F_CCRS_ASSESSMENT as fca
Next DAD encounter	F_CCRS_ASSESSMENT	F_DAD	select distinct dw_seq_id as child_dw_seq_id ,isnull((select top 1 dw_seq_id from star.dbo.F_Dad as fd where fd.patient_dim_key=fca.Patient_DIM_KEY and fd.admission_Date_Dim_Key>fca.Assessment_Date_Dim_Key order by fd.admission_Date_Dim_Key asc),-1) as parent_dw_seq_id from Star.dbo.F_CCRS_ASSESSMENT as fca

Prior DAD encounter	F_CCRS_ASSESSMENT	F_DAD	select distinct dw_seq_id as child_dw_seq_id, isnull((select top 1 dw_seq_id from star.dbo.F_DAD as fd where fd.patient_dim_key=fca.Patient_DIM_KEY and fd.Discharge_Date_Dim_Key<fca.Assessment_Date_Dim_Key order by fd.Discharge_Date_Dim_Key desc),-1) as parent_dw_seq_id from Star.dbo.F_CCRS_ASSESSMENT as fca

8.3.1 Relating Continuing Care Assessment to NACRS Emergency Encounter

For the purposes of this test, the relationship between the patient continuing care assessment and the following NACRS emergency encounter was created in order to look at determining factors from the assessment that could have led to the emergency encounter.

```
select distinct dw_seq_id as child_dw_seq_id,
        isnull((select top 1 dw_seq_id from star.dbo.F_NACRS as fn
                where fn.patient_dim_key=fca.Patient_DIM_KEY
                        and fn.Registration_Date_Dim_Key>fca.Assessment_Date_Dim_Key
          order by fn.Registration_Date_Dim_Key asc),-1) as parent_dw_seq_id
        ,fca.Patient_DIM_KEY
        ,fca.Assessment_Date_Dim_Key
from Star.dbo.F_CCRS_ASSESSMENT as fca
```

The SQL statement selects the unique sequence identifier from the assessment table and performs a subquery on the NACRS emergency encounter fact table to find the unique identifier for the first emergency encounter after the assessment date for the patient. This query uses the top clause to return only a single record from a sorted result set as demonstrated in Table 8.22.

Table 8.22: Constellation Relation query results, CCRS child with following NACRS Encounter

child_dw_seq_id	parent_dw_seq_id	Patient_DIM_KEY	Assessment_Date_Dim_Key
4750892	602518	4391	20120916
4719027	914641	6396	20120402
4693760	-1	9033	20140118
4748606	-1	4385	20120222
4708721	-1	9545	20120608
4711419	-1	10420	20131212
4730740	-1	8624	20130424
4729360	445826	4311	20120816
4732611	-1	6023	20111117
4733488	-1	1103	20120910

4700180	346799	6996	20120209
4724707	-1	2820	20120719
4749976	683262	568	20110616
4739114	-1	9516	20120425
4752618	514004	9261	20110505
4701588	-1	9097	20111103
4750951	-1	10769	20120713
4736192	607229	590	20121026
4730877	382466	2131	20130803
4733078	-1	9499	20110519

To test these results, we modify the subquery to return all the NACRS records for three patients based on the assessment date. This query selects all of the NACRS Emergency Encounters for the patients with identifiers 590, 2131, and 4311 based on the date of their assessments. It then orders those encounters by the registration date.

```
select dw_seq_id,patient_dim_key,Registration_Date_Dim_Key,facility_dim_key,LOS_HOURS from star.dbo.F_NACRS
as fn
    where (fn.patient_dim_key = 590 and fn.Registration_Date_Dim_Key>20121026)
                    or (fn.patient_dim_key = 2131 and fn.Registration_Date_Dim_Key>20130803)
                    or (fn.patient_dim_key = 4311 and fn.Registration_Date_Dim_Key>20120816)
                     order by fn.patient_dim_key,fn.Registration_Date_Dim_Key asc
```

In examining the results in Table 8.23 it can be seen that the results of our first query are correct; the first encounter after the assessment date was selected in each case. However, a careful examination of the results indicate a significant period of time may have passed between the assessment date and the emergency encounter. In such situations, we may want to limit the timeframe to a smaller period; but for testing purposes these results show that the functionality works.

Table 8.23: Emergency NACRS records for Selected Patients and dates

dw_seq_id	Patient key	Registration Date	Facility Key	LOS HOURS
607229	590	20130527	2	4.6
382466	2131	20130809	5	13.2
382464	2131	20140124	5	8.9
382465	2131	20140127	5	8.8
445826	4311	20131127	3	2.3

Table 8.24 shows the value of constellation by relation. The measure is the number of encounters by the Emergency Facility and the Home care clients MAPLE score. This table is not adjusted for population but is provided to demonstrate the functionality available. Any dimension field from the Home care assessments is available for evaluation with our Emergency Encounter subject area providing a wealth of information.

Table 8.24: Encounter Count by facility and MAPLE Score for Home Care Patients.

Encounter Count	Home Care Maple Score					
Facility	1	2	3	4	5	Grand Tota
Emergency_Facility_1	229	645	2113	2402	1047	6436
Emergency_Facility_2	308	380	1993	1570	843	5094
Emergency_Facility_3	942	1130	4541	4591	1769	12973
Emergency_Facility_4	94	67	353	265	100	879
Emergency_Facility_5	1199	1263	3226	3088	1114	9890
Emergency_Facility_6	268	295	1177	1431	727	3898
Emergency_Facility_7					1	1
Grand Total	3040	3780	13403	13347	5601	39171

Chapter 9. Evaluation of Appropriate Placement in Residential Care

9.1 Seniors Advocate Study, Province of British Columbia

In 2015, the Office of the Seniors Advocate for the Province of British Columbia published a series of reports. Two of those reports [68, 69] looked at what was described as "the inappropriate placement in residential care of higher functioning seniors who could live more independently with changes to home care and assisted living". The report based its analysis on the residential care and home care reporting data, comparing seniors across multiple services and jurisdictions. One of its conclusions was that as many as 15% of seniors in residential care could have their needs met more appropriately in a different environment.

As part of this government study, three client profiles were developed representing light physical and cognitive care needs, dementia care needs, and higher physical care needs. These profiles were compared between those receiving services in a home care environment and those receiving care in a residential care facility. In addition, a comparison between British Columbia and the jurisdictions of Ontario and Alberta was performed for the residential care data. It was found in this comparison that the percentage of patients in residential care who met the criteria for the developed client profiles was only five percent of the population for these jurisdictions.

Each of the patient profiles and the criteria for inclusions is explained below. Specific calculations and SQL statements to identify these cohorts based on the previously documented Star Schema structures are provided in Appendix 7. These SQL statements were provided by the Vancouver Island Health authority and based on the Seniors Advocate report.

i) Light Physical and Cognitive Care Needs

This cohort represents clients requiring relatively low care needs, retaining a high level of both cognitive and physical abilities. The population is determined by low scores for ADL

(Activities of Daily Living), self-performance hierarchy, Cognitive Performance Scale (CPS), Change in Health, End stage disease, Signs and Symptoms (CHESS) scale, and a negative indicator for wandering.

ii) Dementia Care Needs

The second cohort is intended to capture those individuals whose needs could be met in a dementia care setting. This population represents individuals with a diagnosis of Alzheimer's or other dementia, a low score for the ADL long form scale, intact to moderate cognitive performance, no aggressive behavior, no psychological conditions, not receiving oxygen treatment, and not having complete bladder incontinence.

iii) Higher Physical Care Needs

The last cohort represents those requiring higher physical care and is also referred to as assisted living plus. This group have a higher score for ADL scale, a high level of cognitive performance, no aggressive behavior, is not receiving oxygen treatment, no psychological conditions, and does not experience wandering.

Each of these three cohorts was identified in our data set for continuing care assessments. As with the analysis performed by the Office of the Seniors Advocate, it represented approximately 15% of the assessments in the data set. It is noted that the cohorts are not mutually exclusive in their calculation, but this study followed the specifications provided. This information is shown in Table 9.1, with the additional indicator of Q1a from the assessment that provides information as to whether the client wishes to remain in long term care or return to the community.

Table 9.1: Residential Care Assessments by Cohort and desire to return to community.

F CCRS ASSESSMENT Count	Q1a Wants to return to community		
Row Labels	**No**	**Yes**	**Grand Total**
Not in Identified Cohort	58366	701	59067
I) Light Care patients in CCRS	6296	97	6393
II) Assisted Living Plus patients in CCRS	4918	65	4983
III) Dementia Care Needs patients in CCRS	3926	82	4008
Grand Total	**68715**	**879**	**69594**

As shown in Table 9.1, the data provided validates the cohort calculations developed in the seniors advocate study. Approximately 15% of the assessments in residential care fall within the three defined cohorts as individuals who could have their care needs met in an alternate level of care. This table also shows that the majority of the population does not want to return to the community.

9.2 Evaluating Correct Placement in Residential Care Based on Home Care Assessment

What was not covered in the Seniors Advocate report, was an analysis of the home care assessment data for the selected cohorts prior to placement in residential care that was used to determine the client needs and priority. If the premise is that these patients were inappropriately placed in residential care, then an analysis of the information used to determine their needs must be undertaken. It needs to be determined if the initial placement in residential care was appropriate or, as suggested by the government study, that the clients in these three cohorts were inappropriately placed. Assessment data needs to be reviewed to determine if there was a change/improvement in the client’s health during their stay in residential care that makes it possible for them to receive care in a different setting?

9.2.1 MAPLE (Method of Assigning Priority Levels) Score

One key indicator used in placement of clients in residential care is the Method of Assigning Priority Levels (MAPLE) score [72, 71, 58]. Developed in Canada, MAPLE is one of the Screening algorithms included in the InterRAI Home Care Instrument [57, 58]. The MAPLE score is based on the Home Care Assessment data and assigns clients to one of five levels based on their risk of adverse outcomes. The highest priority level is based on the presence of ADL impairment, cognitive impairment, behavioral problems, and the InterRai Home Risk Client Assessment protocol, while those at the lowest level have no major functional problems and are considered self-reliant [71]. Both the analysis performed by the Seniors Advocate and the MAPLE score use many of the same standardized scales such as the ADL Hierarchy and the CPS [72].

Using the functionality developed here, it is a simple matter to associate the MAPLE score from the most recent home care assessment prior to placement in residential care with our three cohorts as shown in Table 9.2.

Table 9.2: Residential Care Assessments by Cohort and HCRS MAPLE Score.

F CCRS ASSESSMENT Count	MAPLE Scale						
Row Labels	No Value	1 - Low	2 -Mild	3 - Moderate	4 High	5 Very High	Grand Total
Light Care patients	3609	28	71	929	1150	606	6393
Assisted Living Plus	2678	18	53	801	983	450	4983
Dementia Care Needs	1831	3	7	393	1003	771	4008
Grand Total	**5539**	**28**	**82**	**1408**	**2161**	**1309**	**10527**

It is immediately obvious that although no data is available in our home care data set for many of the members of our three cohorts. There is a significant difference in the information provided from the prior home care assessment when compared to the residential care assessment data for those where data is available. The majority of clients were assessed as Moderate to Very High needs in their most recent home care assessment, yet have been identified as inappropriately placed in residential care.

Based on the discrepancy between Tables 9.1 and 9.2, it is apparent that a more detailed analysis of the data is in order.

9.3 Detail Analysis of Previous Home Care Assessment

Our first step in performing a more detailed analysis is to relate our CCRS assessment to the HCRS assessment. For each CCRS assessment, a relationship was created to the most recent prior HCRS assessment for the same patient. In addition, only initial admission assessments in residential care (within 14 days) were included in the analysis to minimize the time between assessments and reduce the possibility of a significant health change. For a measure of the population, the distinct count of patients was used. Table 9.3 provides a count of distinct patients for their initial residential care assessment and their MAPLE scores from the previous home care assessment.

Table 9.3: Residential Care Patients by Cohort and HCRS MAPLE Score.

AA8 ASSESSMENT TYPE	1

Distinct Patient Count	**Maple Scale**					
Row Labels	**1 - Low**	**2 - Mild**	**3 - Moderate**	**4 -High**	**5 – Very High**	**Grand Total**
Light Care patients	7	7	109	138	79	340
Assisted Living Plus patients	4	7	107	126	58	302
Dementia Care Needs patients			56	154	106	316
Grand Total	**7**	**10**	**178**	**291**	**176**	**662**

9.3.1 Examination of ADL Hierarchy

Table 9.4 in the detail analysis looks at the ADL Hierarchy Scale between the initial assessments in residential care and the same scale from the previous home care assessment. The ADL Self Performance Hierarchy [58, 59] is a seven point scale from 0 to 6 used to represent the disablement process by grouping a patient's performance into separate stages from complete independence to total dependence. The ADL Hierarchy scale was used in both the MAPLE score calculation and in the development of the Light Care Patient cohorts by the Seniors Advocate.

Table 9.4: Residential Care Patients by Cohort and ADL Self Performance Hierarchy.

AA8 ASSESSMENT TYPE	1 - Initial

Distinct Patient Count	**HCRS ADL Hierarchy**						
CCRS Patient Cohort / ADL Hierarchy	**0**	**1**	**2**	**3**	**4**	**5**	**Grand Total**
Light Care Patients	**118**	**69**	**109**	**29**	**10**	**5**	**340**
0 – Independent	70	43	68	15	4	2	202
1 – Supervision Required	49	26	43	14	6	3	141
Assisted Living Plus Patients	**96**	**47**	**115**	**26**	**12**	**6**	**302**
0 – Independent	42	22	47	9	1	1	122
1 – Supervision Required	34	11	28	4	5	3	85
2 – Limited Impairment	19	13	36	11	5	2	86
3 – Extensive assistance level 1		1	3	2			6
4 – Extensive assistance level 2			1				1
5 – Dependent	1		1		1		3
Dementia Care Needs Patients	**96**	**99**	**93**	**18**	**9**	**1**	**316**
0 – Independent	37	40	33	6	3	1	120
1 – Supervision Required	47	37	41	5	4		134
2 – Limited Impairment	11	20	20	7	2		60
3 – Extensive assistance level 1	1	2	1	1			5
5 – Dependent	1						1
Grand Total	**207**	**161**	**211**	**55**	**21**	**7**	**662**

The difference, shown in Table 9.4, between the ADL Hierarchy calculated on the residential care initial assessment and the most recent home care assessment prior to admission to residential care is substantial. It is most significant in terms of the light care patients, as it is used in that calculation. Light care patients are defined as being independent or requiring some supervision in their ADL Hierarchy performance, yet nearly 45% of these patients have a higher placement on the ADL Hierarchy in the previous home care assessment than they do in residential care.

9.3.2 Examination of Cognitive Performance Scale

A second scale that is used in the calculation of all three of the Continuing Care cohorts and in the MAPLE score, is the patients CPS [58, 59]. This scale has seven points similar to the ADL hierarchy and ranges from intact to very severe impairment.

Table 9.5: Residential Care Patients by Cohort and Cognitive Performance Scale

AA8 ASSESSMENT TYPE	1 - Initial

Distinct Patient Count	**HCRS CPS**							
CCRS Patient Cohort/CPS Score	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**Grand Total**
Light Care patients	**45**	**39**	**149**	**91**	**7**	**9**		**340**
0 – Intact	38	15	76	25	4	1		159
1 – Borderline Intact	7	24	73	66	3	8		181
Assisted Living Plus patients	**39**	**45**	**134**	**74**	**4**	**5**	**1**	**302**
0 – Intact	33	20	67	19	2	1		142
1 – Borderline Intact	6	25	67	55	2	4	1	160
Dementia Care Needs patients	**1**	**13**	**116**	**151**	**11**	**24**		**316**
0 – Intact		1	14	10				25
1 – Borderline Intact		4	26	30	2	4		66
2 – Mild Impairment	1	4	39	60	4	6		114
3 – Moderate Impairment		5	39	55	7	14		120
4 – Moderate/Severe Impairment								
5 – Severe Impairment								
6 – Very Severe Impairment								
Grand Total	**55**	**65**	**262**	**229**	**18**	**32**	**1**	**662**

Again, there is a substantial difference between the prior home care assessment and the initial assessment in residential care. Light care and assisted living plus patients are defined as having intact to borderline intact cognitive performance, yet prior to admission in continuing care the majority of these patients were mild to moderately impaired. By comparison, the dementia care patients are closer though still scaled higher in home care than residential care.

9.3.3 Examination of Change in Health, End-Stage Disease and Symptoms, and Signs Score

The next scale examined is the CHESS [58, 59]. This scale is meant to measure the instability in a patient's health. It is only used in the identification of the Light Care patient group and is not used in the MAPLE score. Only patients identified as having no or low instability are considered as being part of the group.

Table 9.6: Residential Care Patients by Cohort and CHESS Score

AA8 ASSESSMENT TYPE	1 - Initial

Distinct Patient Count CCRS Patient Cohort/CHESS Score	HCRS CHESS Score 0	1	2	3	4	5	Grand Total
Light Care patients	**50**	**87**	**130**	**48**	**24**	**1**	**340**
0 – No Instability	39	69	102	36	19		265
1 – Minimal Instability	10	13	20	11	5	1	60
2 – Low Instability	2	5	9	1			17
Assisted Living Plus patients	**45**	**84**	**117**	**38**	**16**	**2**	**302**
0 – No Instability	32	70	87	26	13	1	229
1 – Minimal Instability	11	11	24	9	3	1	59
2 – Low Instability	2	3	6	3			14
Dementia Care Needs patients	**40**	**79**	**151**	**32**	**13**	**1**	**316**
0 – No Instability	35	67	116	24	12	1	255
1 – Minimal Instability	4	8	27	8	1		48
2 – Low Instability	1	3	9	1			14
3 – Moderate Instability							
4 – High Instability		1					1
5 – Very High Instability							
Grand Total	**90**	**175**	**272**	**86**	**36**	**3**	**662**

Once more it is apparent that the home care assessments score the patients higher than the residential care assessments. More than 100 of the patients who were identified in residential care as having no instability were listed as low instability in home care, while 19 were listed as high instability. However, nearly 80% (267) of the patients would still qualify as light care based on their CHESS score from the last home care assessment.

9.3.4 Examination of ADL Long form

The ADL long form [58, 59] scale is used in the calculation of the Assisted Living Plus and the Dementia Care Needs cohorts. The scale includes 29 points and is more detailed then the other ADL scales. The ADL long form scale is calculated from the ADL performance on several functional measures including dressing, eating, mobility, toilet use, bed mobility, transfers, dressing, and personal hygiene.

Table 9.7: Residential Care Patients by Cohort and ADL Long Form Scale

AA8 ASSESSMENT TYPE	1-Initial

Distinct Patient Count CCRS Patient Cohort / ADL Long Form	HCRS ADL Long form 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	Grand Total
Light Care patients	**97**	**18**	**35**	**31**	**26**	**23**	**14**	**18**	**21**	**9**	**9**	**11**	**6**	**2**	**3**	**2**	**3**	**2**	**2**	**1**	**4**	**1**	**1**	**1**	**340**
0	56	11	18	19	14	12	9	9	12	4	2	4	1	2	1		1				1			1	177
1	16	4	5	6	4			2	2	1	2	5	2		1	1			1		1	1	1		55
2	12	2	9	3	5	5	1	6	2	3	1	2	1			1	2			1	1				57
3	6	1	1	2	2	4	2	1	2		1														22
4	3	1	1	1		1		1	3				1					2	1						15
5	3						2				2		1		1										9
6	2		1		1	1					1														6
7								1																	1
8										1															1
9																					1				1
Assisted Living Plus patients	**71**	**11**	**33**	**17**	**23**	**25**	**17**	**16**	**25**	**13**	**8**	**12**	**8**	**6**	**1**	**4**	**4**	**1**	**2**		**3**	**1**	**1**		**302**
0	33	2	11	8	12	7	8	7	7	2	2	4		2			1								106
1	10	2	5	1	1			1	2	1	1	3	2						1		1	1	1		33
2	8	1	9		4	5	2	4	2	3	1	1	1	1		1	1				1				45
3	4	1	3	2	2	4		1	4	2				1											24
4	8	3	2	3	1	4	2		4	4	1	2	2			1		1	1						39
5	5		1	2	1	1	3	2	5	2	2		1		1		1								27
6	3	2	2	1	2	4	2	2	1		1	2	2	2		2	1				1				30
Dementia Care Needs patients	**78**	**37**	**53**	**21**	**23**	**28**	**21**	**14**	**14**	**6**	**2**	**4**	**3**	**2**	**2**	**3**	**1**		**1**		**2**		**1**		**316**
0	24	16	16	7	6	7	10	4	3		1	3			1		1								99
1	15	3	9	3	5	3	1	2	3	1													1		46
2	18	7	15	7	5	9	2	2	2	1			1	1		2					2				74
3	12	4	6	2	3	5	2	2	3	1	1				1										42
4	13	7	7	3	4	4	6	4	4	3		1	2	1		1			1						61
Grand Total	**164**	**55**	**84**	**50**	**47**	**53**	**35**	**31**	**45**	**19**	**11**	**18**	**12**	**7**	**6**	**6**	**5**	**2**	**2**	**1**	**6**	**1**	**1**	**1**	**662**

When comparing the ADL long form scale from the residential care assessment with the values from the home care assessment, nearly a third of the patients in the cohorts would not have met the criteria for the cohort definition.

9.3.5 Depression Rating Scale

One other scale of interest in the evaluation of the cohorts and shown in Table 9.8 is the Depression Rating Scale (DRS) [58, 59]. This scale is not used in the maple score or in the calculation of our cohorts, but is an indicator of the level of depression or anxiety the patient is feeling. A score of three or more indicates a potential or actual problem with depression.

The patient cohorts all show a higher level of depression in home care than is indicated in continuing care. As an example for the patients in our three cohorts, more than 80% have a depression scale value of zero with the highest being 90% on their residential care assessment. This can be compared to home care, where the highest percentage of patients to have a DRS score of zero is 61%.

In all three cohorts the patient population is less depressed in a residential care facility. According to the assessment data indicated earlier in Table 9.1, they are happier in facility care and generally want to be there. The majority of patients indicated that they did not want to leave residential care.

Table 9.8: Residential Care Patients by Cohort and DRS

AA8 ASSESSMENT TYPE	1 - Initial

Distinct Patient Count	HCRS - DRS												
CCRS Patient Cohort / DRS	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**Grand Total**
Light Care patients	**180**	**44**	**41**	**22**	**21**	**8**	**6**	**9**	**2**	**2**	**2**	**3**	**340**
0	158	39	32	20	12	7	5	4	2	1	2		282
1	10	1	5		5			4		1		2	28
2	7	1	2	1	1							1	13
3	1	2	1										4
4	2		1		1	1							5
5	1			1	1								3
6	1						1	1					3

8	1												1
9					1								1
12		1											1
Assisted Living Plus patients	**185**	**38**	**36**	**16**	**9**	**6**	**2**	**7**		**2**	**1**		**302**
0	170	36	28	16	7	6	2	5		2	1		273
1	7	1	5		1			2					16
2	7		3		1								11
4	1												1
5		1											1
Dementia Care Needs patients	**177**	**51**	**32**	**20**	**17**	**7**	**7**	**1**	**2**	**2**			**316**
0	144	37	27	18	13	4	7	1	1	2			254
1	14	8	2	1	3	1			1				30
2	13	2	3	1	1	1							21
3	1	2											3
4	3			1		1							5
5	2	1											3
6		1											1
7		1											1
Grand Total	**365**	**98**	**70**	**38**	**41**	**14**	**12**	**11**	**4**	**3**	**3**	**3**	**662**

9.3.6 Individual Field Values Home Care Assessment Living Arrangement

The last table presented as part of the evaluation of the Home Care data is living arrangements from the previous assessment. Field O2b of the home care assessment indicates whether the client or primary caregiver feels that the client would be better off in another living arrangement. A significant majority of clients and caregivers believe that they would be better off with different living arrangements.

Table 9.9: Residential Care Patients by Cohort and HCRS Field O2b Living Arrangements

AA8 ASSESSMENT TYPE	1 - Initial

Distinct Patient Count	**HCRS O2b**				
CCRS Patient Cohort	**No**	**Client Only**	**Caregiver Only**	**Client and Caregiver**	**Grand Total**
Light Care patients	51	18	65	206	340
Assisted Living Plus patients	53	17	42	190	302
Dementia Care Needs patients	51	13	103	149	316
Grand Total	**104**	**29**	**168**	**361**	**662**

9.4 Analysis of Previous Hospital Discharge Abstract Record

An obvious discrepancy between the last home care assessment and the initial assessment in residential care exists. Each of the scales examined here show the client population health is poorer in home care when compared to the initial assessment in residential care. A possible reason for this discrepancy would be a health event requiring hospitalization. For this reason, the DAD [62] data was evaluated for any events for the three patient cohorts prior to admission.

This evaluation looked at intervention and diagnosis information to determine if a major health event had occurred for the patient. The selection criteria was restricted so that only DAD records where a corresponding HCRS record also existed were selected.

A portion of the patient cohorts did not have a discharge record, indicating that no hospital stay had occurred during the time frame for the data or geographical area of the study. In addition, a surprising number of records did not have any intervention data associated with them. A discharge abstract record existed, but no intervention information was available. Given the selection criteria and results, the majority of the three cohorts that did have a DAD record fell in this area.

Table 9.10: Residential Care Patients by Cohort and Intervention

AA8 ASSESSMENT TYPE	1 - Initial			
Distinct Patient Count	**CCRS Cohort**			
Row Labels	**Light Care patients**	**Assisted Living Plus patients**	**Dementia Care Needs patients**	**Grand Total**
1, Physical/Physiological Therapeutic Interventions	53	56	43	102
2, Diagnostic Interventions	11	13	10	20
3, Diagnostic Imaging Interventions	5	4	6	13
No Intervention Entered	199	163	160	359
No Discharge Abstract Record	159	130	172	325
Grand Total	**427**	**366**	**391**	**819**

After reviewing the intervention data, further analysis to determine the reason for the lack of intervention information for such a large number of records was performed. The DAD record was examined to determine if the primary length of stay was in Alternate Level of care (ALC) or Acute care (AC). This information showed that between 31% and 44% of the DAD records where no intervention occurred were coded as ALC.

Table 9.11: Residential Care Patients by Cohort, Intervention, and Type of Stay

AA8 ASSESSMENT TYPE	1 Initial			
Patient Count	**Column Labels**			
Row Labels	**Light Care patients**	**Assisted Living Plus patients**	**Dementia Care Needs patients**	**Grand Total**
1, Physical/Physiological Therapeutic Interventions	**53**	**56**	**43**	**102**
Acute Stay	42	49	34	83
ALC Stay	11	7	9	19

2, Diagnostic Interventions	**11**	**13**	**10**	**20**
Acute Stay	11	13	7	17
ALC Stay			3	3
3, Diagnostic Imaging Interventions	**5**	**4**	**6**	**13**
Acute Stay	3	1	3	6
ALC Stay	2	3	3	7
Not determined	**199**	**163**	**160**	**359**
Acute Stay	122	113	89	218
ALC Stay	77	50	71	141
No Discharge Abstract Record	**159**	**130**	**172**	**325**
Grand Total	**427**	**366**	**391**	**819**

The diagnosis information did not have the issues of incomplete data that interventions did. However, the diagnosis data contradicted the CCRS assessment data in a number of areas. As stated previously, 36% to 44% of patients had no DAD record. For those that did have diagnosis information, there were a number of elements of note. For the Light Care patient cohort, the group was defined as not subject to mental health issues, yet a significant portion were diagnosed with mental and behavioral issues. The cognitive performance scale on the residential care initial assessment indicates the patient as having intact or borderline intact cognitive performance, yet the home care assessment and the DAD both contradict this.

Table 9.12: Residential Care Patients by Cohort and Diagnosis

AA8 ASSESSMENT TYPE	1 - Initial			

Distinct Patient Count	**Column Labels**			
Row Labels	**Light Care patients**	**Assisted Living Plus patients**	**Dementia Care Needs patients**	**Grand Total**
I, A00-B99, Certain infectious and parasitic diseases	**6**	**8**	**7**	**13**
II, C00-D48, Neoplasms	**8**	**8**	**3**	**11**

III, D50-D89, Diseases of the blood and blood-forming organs and certain disorders involving the immune mechanism			**2**	**2**
IV, E00-E90, Endocrine, nutritional and metabolic diseases	**6**	**9**	**5**	**11**
V, F00-F99, Mental and behavioural disorders	**58**	**27**	**54**	**111**
F00-F09, Organic, including symptomatic, mental disorders	**32**	**19**	**44**	**73**
F01, Vascular dementia	8	3	8	15
F03, Unspecified dementia	13	7	19	30
F05, Delirium, not induced by alcohol and other psychoactive substances	10	8	16	25
F06, Other mental disorders due to brain damage and dysfunction and to physical disease	1	1	1	3
F10-F19, Mental and behavioural disorders due to psychoactive substance use	**4**	**1**	**6**	**10**
F20-F29, Schizophrenia, schizotypal and delusional disorders	**10**	**2**	**2**	**13**
F30-F39, Mood [affective] disorders	**12**	**5**	**2**	**15**
VI, G00-G99, Diseases of the nervous system	**20**	**15**	**32**	**49**
VII, H00-H59, Diseases of the eye and adnexa	**2**	**2**	**3**	**6**
IX, I00-I99, Diseases of the circulatory system	**42**	**43**	**24**	**66**
I10-I15, Hypertensive diseases		**1**		**1**
I20-I25, Ischaemic heart diseases	**4**	**4**	**2**	**8**
I26-I28, Pulmonary heart disease and diseases of pulmonary circulation	**2**	**1**		**2**
I30-I52, Other forms of heart disease	**21**	**22**	**13**	**33**
I60-I69, Cerebrovascular diseases	**10**	**9**	**6**	**16**
I70-I79, Diseases of arteries, arterioles and capillaries	**3**	**3**	**1**	**3**
I80-I89, Diseases of veins, lymphatic vessels and lymph nodes, not elsewhere classified	**1**	**1**	**1**	**1**
I95-I99, Other and unspecified disorders of the circulatory system	**1**	**2**	**1**	**2**
X, J00-J99, Diseases of the respiratory system	**16**	**20**	**6**	**30**
XI, K00-K93, Diseases of the digestive system	**13**	**15**	**11**	**25**
XII, L00-L99, Diseases of the skin and subcutaneous tissue	**4**	**4**	**3**	**7**
XIII, M00-M99, Diseases of the musculoskeletal system and connective tissue	**8**	**10**	**6**	**15**
XIV, N00-N99, Diseases of the genitourinary system	**7**	**6**	**6**	**14**
XVIII, R00-R99, Symptoms, signs and abnormal clinical and laboratory findings, not elsewhere classified	**33**	**24**	**19**	**54**

XIX, S00-T98, Injury, poisoning and certain other consequences of external causes	**23**	**24**	**19**	**44**
XXI, Z00-Z99, Factors influencing health status and contact with health services	**23**	**22**	**20**	**40**
-No Value-	**159**	**130**	**172**	**325**
Grand Total	**427**	**366**	**391**	**819**

9.5 Study Conclusions

The report by the seniors advocate evaluated patient assessment data and concluded that patients in continuing care could have their needs met in alternate care environments. The evaluation here contradicts this conclusion. Although the patient assessment data used in the study verified the calculations by the seniors advocate, several issues were encountered in the data.

When this data was expanded to include additional data sets, such as the Home Care Assessment and DAD, several problems were encountered that contradict the data available in the continuing care assessments alone. The patient cohorts identified all showed worse scores on the ADL Hierarchy, CPS, and CHESS in the home care environment than in continuing care. In addition to this, a portion of all three cohort populations had a history of ALC care in hospital attributed to problems, such as mental health, that are not reflected or shown in the cohort design and continuing care assessments.

It is possible that the home care assessment data may have had quality issues such as those identified in Hirdei's evaluation [70] of Ontario home care data. As the cohort members expressed a strong preference for moving into a continuing care environment, it is also possible that the assessment information may have been skewed by the patient's responses to questions and tests. Finally, the discharge abstract data showed hospital admissions for alternate level of care and mental health issues that may reflect a patient's health to be failing or a breakdown in the home care environment.

The seniors advocate report identified a need for alternate care services for those whose needs, according to the assessment information, could be met in an alternate environment. The continuing care data alone did support those conclusions; but with the abilities to expand beyond a single data set and interrelate information as demonstrated in this thesis, a more comprehensive view of the patient population can be provided which contradicts the conclusion.

The complexities of health care information require enhanced methods for working with data. The methods employed here allowed for the rapid expansion and analysis of health care data across home care, hospital care, and continuing care environments to arrive at new conclusions that were unavailable in the original study. This capability can be used to provide insights and functionality in the analysis of health information that previously required significant effort or was not possible due to the complexity of the data and the simplified tools available to work with that data.

Chapter 10. Thesis Conclusions

The goal of this thesis was the development of a new methodology to both extend and enhance an integrated enterprise data warehouse built following the Kimball methodology. The methods proposed and developed here were successfully proven to accomplish this and can enable better insight into complex data and information.

10.1 Success

The first success of this development effort was an enterprise level data warehouse encompassing emergency, acute care, and long term care services. The star schema structures developed are representative of the foundational work for an electronic medical record data warehouse. The resulting structures employed conformed dimensions and achieved what Kimball defined as an Integrated Enterprise Data Warehouse which is a significant achievement.

Each of the separate star schemas developed were fully functional information subject areas. The data involved was a subset of the national data used for strategic and tactical planning for the delivery of health care services in Canada. The only limitation to this study data was in terms of geographical and temporal restrictions witch limited only the data volume not its diversity or complexity.

The second success for the project was the development of the semantic relationship engine itself. The project successfully was able to rapidly select and associate information across multiple subject areas. Cohorts were easily defined against the DAD and NACRS data based on the patient registration in the home or continuing care program. It was also possible to easily identify patients within programs who transitioned from home care to facility based long term care and examine the differences between those populations.

At its most sophisticated the engine was able to associate between separate subject areas. In the final study, this technique was used to associate a patient's final home care assessment with his initial assessment in a continuing care facility. This provided the ability to evaluate the effects of the care level transition on that patient's health as well as the decisions related to the provision of care. The study looked at a report on what was described as the inappropriate placement of seniors in residential care. With the ability to compare the patient assessment prior to this placement and their initial assessment in residential care it was possible to show that the initial placement in residential care was not inappropriate based on the home care assessment used in determining the patient's needs. This was due to the ability to interrelate these data sets which the initial study was unable to do.

10.2 Risks and Limitations

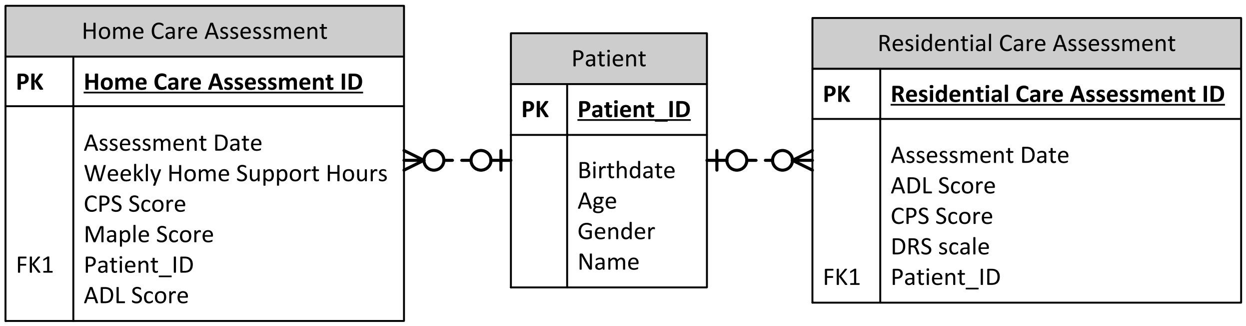
10.2.1 Data and Structure

Despite the successes of building an integrated electronic medical record data warehouse, developing the methodology, and proving the functionality there remain constraints and limitations on its usage.

The most significant risks in employing the methodology is the structure of the data and knowing its appropriate use. This is not simply a matter of technical skills or abilities but is based on an understanding of the data, the nature of the underlying data structures and their meaning.

In the study we compared a patient's initial assessment in residential care with the final assessment from home care prior to admission. This study selected the final home care assessment that would have been used to determine the patient's needs and priority. This assessment would have been performed prior to admission into residential care. The results showed a distinct disconnect between the two subject areas as the Home care assessment showed the patient to be in more serious condition then the initial assessment in residential care. This is a valid and appropriate use of this data but the constraints used in developing the relationship and underlying nature of the structure of the data needed to be understood before proceeding.

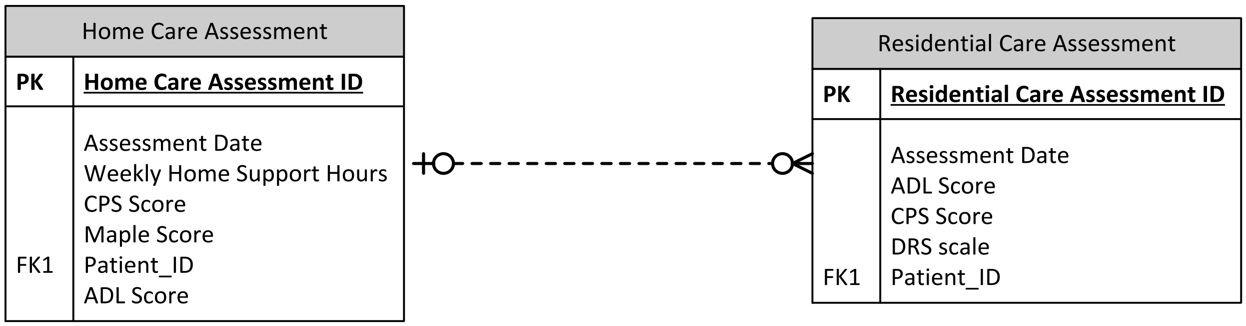
Figure 10.1: Patient Home care and Residential Care Assessments



As shown in Figure 10.1 the underlying data structure between the Home Care and Residential care data is a many to many relationship. A patient may have multiple home care assessments as well as multiple residential care assessments. In Kimball's article on Drill Across he states that it is effectively impossible to resolve the many to many relationship to return valid results when joining fact tables and this correct. Only under limited situations can a relationship be established and these situations are dependent on the underlying data. Even when such a relationship is possible the use of the underlying data is still limited.

In our situation we linked the residential care assessment data to the most recent home care assessment prior to placement in residential care based on supplied assessment dates. This in effect filtered the assessment data into a one to many relationship as shown in Figure 10.2. Only one home care assessment was returned and related to each individual residential care assessment record.

Figure 10.2: Home Care Assessment related to Residential Care Assessment



However, as residential care assessments remained the base table of the query, multiple home care assessment records would still have been returned. Any use of data fields from the home care records for aggregate calculations could return invalid results because multiple records would exist. Only the base table at the grain of the query can be used for aggregate calculations. Looking at Figure 10.1 again; it can be seen that a field representing weekly home support hours exists in the table. If we return multiple home care records because two initial residential care assessments exist, then the sum of the total hours of home support being provided would be doubled for that patient.

10.2.2 Tools and Technology Limitations

A second less obvious limitation is the tools and technology. This seems unlikely given the constant flow of new products and technologies in the era of Big Data, but in many ways remains a major issue. The data structures involved are complex and represent multiple cascading many to many relationships. Several of today's Adhoc or OLAP query tools do not have the ability to work with structures this complex or require additional effort. Using the Microsoft toolsets [74, 75] or others such as Cognos [76] can work but each requires that specific techniques be employed or other steps performed in order to function correctly.

As an example; Microsoft's SQL Server Analysis Server can work with many to many relationships without any special configuration but will enforce mandatory joins and uses the underlying referential integrity declared in the source database or defined in the model. By comparison, the BISM or Cognos model do not handle the many to many relationships directly but must depend on programming effort or data model changes during setup and configuration.

In the study performed here SQL Server Analysis server was used. The product works with many to many relationships but may require effort to configure depending on complexity. Under normal operation analysis server will correctly role play dimensions but does not do this with bridge structures requiring manual configuration for each instance of the structure. This required the creation of separate views for each of our subject areas for the Constellation tables. In addition, NULL or missing values must be accounted for using techniques such as a placeholder record for Null or default not found values in dimension and bridge tables. This allowed for facilitating queries to return all records when comparing cohorts to the total population. In the study this was required in the home care assessment fact table to allow it to bridge the other fact table with the home care assessment dimensions.

Regardless of which tool is used, all will likely have issues or idiosyncrasies that may require skilled knowledge or abilities to correctly employ. The more dangerous tools are those that require a knowledgeable SQL user and that do not recognize the nature of the underlying table structures and generate invalid results. A typical query wizard such as SQL Server Management Studio will allow many to many joins in a query and assume the end user has the skills and understanding to correctly interpret the results. Even the most skilled SQL experts can and will make this mistake.

10.3 Future Direction

The next steps for the future development of this methodology is to both publish the concepts [77,78] and to deploy them in a full production environment. The methodologies developed here have been successfully used with Island Health of British Columbia and are also employed at both Fraser Health and Vancouver Coastal as part of their Data Quality efforts. Presentations to the British Columbia Provincial Health Service Authority, Vancouver Coastal Health Authority, and the Fraser Valley Health authority are scheduled to help expand the use of the methodology in those areas. Documentation and source code have been shared with the Government of Norway and others through conferences and correspondence with additional papers and presentations planned.

The future expansion of this methodology is likely to occur as part of development in Electronic Medical Record systems in British Columbia. Each health authority as well as the province have developed Enterprise Data Warehouses for reporting, research, and analysis. All have encountered challenges due to the complexity of the data and all have expressed interest in the techniques developed here.

Although the need for these techniques can be considered as limited to extremely complex data. In today's information age increasing volume and complexity to data is inevitable and is leading to the demand for this type of functionality. The development of semantic relationships between dimensional models as done here have been proven to work. Insights into patient care and analysis of data can be performed in a timelier manner with less development through these semantic relationships and facilitate better health research in the future.

Appendix 1: NACRS (National Ambulatory Care Reporting System)

Table A1.1 represents the data fields provided by CIHI for the NACRS data. Additional fields that were provided as part of the data set but that were not populated are not included.

Table A1.1: NACRS Fields

Field Name	Data Type	Description
HCN_MBUN	Numeric	Unique Patient Identifier
Facility_AM_Care_Num_MBUN	Numeric	Unique Facility Identifier
Prov_Issue_Health_Number	Character	The Province that Issued the Health Care number for the Patient
Gender	Character	The Gender of the Patient
Birth_Year	Numeric	The year of Birth for the Patient
Submission_Fiscal_Year	Numeric	The CIHI Fiscal Year of the Record Submission
Submission_Period	Numeric	The CIHI Fiscal Period of the Record Submission
Admit_Via_Ambulance	Character	An indicator for Patients Admitted via ambulance (Ground, Air, Sea)
Triage_Date	Date	Date of the Patient Triage
Triage_Time	Time	Time of the Patient Triage
Triage_Level	Numeric	Numeric the Patient Was Triaged
Date_Of_Registration	Date	The Date the Patient was Registered
Registration_Time	Time	The Time the Patient was Registered
Date_Physician_Init_Assessment	Date	Date the Patient was Initially Assessed by Physician
Time_Physician_Init_Assessment	Time	Time the Patient was Initially Assessed by Physician
Disposition_Date	Date	Date of Disposition for the Patient Visit
Dispostion_Time	Time	Time of Disposition for the Patient Visit
Visit_Dispostion	Numeric	Disposition of the Visit (Discharge Disposition)
Patient_Left_ED_Date	date	Date the Patient left the Emergency Department
Patient_Left_ED_Time	Time	Time the Patient left the Emergency Department
LOS_Hours	Numeric	The Length of Stay for the Patient visit in hours
Wait_Time_To_PIA_Hours	Numeric	The Wait time for the Patient until Initial Physician Assessment
Wait_Time_To_Inpatient_Hours	Numeric	The Wait time for the Patient until admitted (Inpatient)

Appendix 2: DAD (Discharge Abstract Database)

Three separate data files were provided for each year of the study. These data sets represented the Hospital Discharge, the Interventions, and the diagnoses for the patient. Field names and descriptions for each data file are provided below.

Table A2.1: DAD File One: Discharge Abstract Record

Field Name	Data Type	Description
hcn_mbun	Numeric	Unique Patient Identifier
DAD_TRANSACTION_id_mbun	Numeric	Unique DAD Record Within The Fiscal Year
DAD_INST_CODE_RAN	Numeric	Unique Facility Identifier
GENDER_CODE	Character	The Gender of the Patient
BIRTHYEAR	Numeric	The year of Birth for the Patient
FISCAL_YEAR	Numeric	The CIHI Fiscal Year of the Record Submission
FISCAL_PERIOD	Numeric	The CIHI Fiscal Period of the Record Submission
MAIN_PATIENT_SERVICE	Numeric	The main patient service based on disease and diagnosis.
MAIN_PATIENT_SUBSERVICE	Numeric	An optional further deliniation of patient service types
MR_DIAG_ICD10_CODE	Character	The Major Diagnosis Code (ICD-10-CA coding)
PRINC_INTERV_CCI_CODE	Character	The Principle Intervention (CCI Coding)
SAME_DAY_SURGERY_HOURS	Numeric	The length of stay in hours for same day surgery
TOTAL_LOS_DAYS	Numeric	Total Length of Stay in Days
ACUTE_LOS_DAYS	Numeric	Length of Stay in Acute Care
ALC_LOS_DAYS	Numeric	Length of Stay in Alternate Level of Care
ADMISSION_DATE	Date	The Date the Patient was Admitted
ADMISSION_TIME	Time	The Time the Patient was Admitted
ADMISSION_CATEGORY	Character	The Patient Classification on Admission
ENTRY_CODE	Character	The Point of Entry for the Patient
READMISSION_CODE	Numeric	For Acute Care Abstracts information about the patient's previous admissions.
DISCHARGE_DATE	Date	The Date the Patient was Discharged
DISCHARGE_TIME	Time	The Time the Patient was Discharged
DEATH_SPECIAL_CARE	Character	Flag to Indicate Death in Special Care Unit
WEIGHT	Numeric	The Weight of the Newborn in Grams
ADMIT_BY_AMBULANCE_IND	Character	An indicator for admission via ambulance.
TOTAL_SCU_LOS_HOURS	Numeric	Total Length of Stay in Hours in a Special Care Unit
DISCHARGE_DISPOSITION	Numeric	The Disposition at Discharge
ED_WAIT_TIME	Numeric	The total time waiting in emergency (hours)
ED_LEAVING_DATE	Date	The Date the patient left the Emergency Department
ED_LEAVING_TIME	Time	The Time the patient left the Emergency Department
ED_WAIT_MINUTE	Numeric	The total time waiting in emergency (minutes)

Table A2.2: DAD File Two: Discharge Abstract Diagnosis (ICD-10-CA Code) Fields

Field Name	Data Type	Description
DAD_TRANSACTION_id_mbun	Numeric	A foreign key to the DAD Transaction record.
*DIAG*_SEQ_ID	Numeric	The sequence identifier for the Diagnosis
DIAG_ICD10_CODE	Character	The Diagnosis Code (ICD-10-CA coding)
DIAG_PREFIX	Character	The diagnosis Prefix Code (Questionable, Paliative, Post-Admit …)
DIAG_TYPE_CODE	Character	The Diagnosis Type code indicating the impact on care for the diagnosis (Main, Secondary, Type 2 …)
DIAG_CODING_CLASS	Numeric	The coding class for the diagnosis code (Exclusively 0)
DIAG_CLUSTER	Character	A code assigned to indicate when more than one diagnosis code is required to describe the condition

Table A2.3: DAD File Three: Discharge Abstract Intervention Codes (CCI Code) Fields

Field Name	Data Type	Description
DAD_TRANSACTION_id_mbun	Numeric	A foreign key to the DAD Transaction record.
EPISODE_SEQ_ID	Numeric	The Intervention Episode identifier
INTERV_SEQ_ID	Numeric	The Sequence of the Interventions within the episode.
INTERV_CCI_CODE	Character	the CCI Intervention Code
EPISODE_START_DATE	Date	The date the Intervention started
EPISODE_START_TIME	Time	The time the Intervention started
EPISODE_END_DATE	Date	The date the Intervention ended
EPISODE_END_TIME	Time	The time the Intervention ended
EPISODE_DURATION_MINS	Numeric	the duration of the intervention
INTERV_CCI_DESC	Character	The description of the intervention

Appendix 3: HCRS (Home Care Reporting System)

Two separate files were provided for the Home Care Reporting Data. The first file represented Episodes of Care while the second file contained individual assessment records. Fields are provided below.

Table A3.1: HCRS File One Fields

Field Name	Data Type	Description
HCN_MBUN	Numeric	Unique Patient Identifier
CLIENT_EPISODE_ID_MBUN	Numeric	Unique identifier for the Home Care Episode
X6	Numeric	Home Care Acceptance Date
X30	Character	Discharge date from Home Care
CLIENT_PROVINCE	Date	Patient Home Province
BB1	Time	Patient Gender
AA3b	Date	Province for Patient Health Care Card
BIRTH_YEAR	Time	Patient Year of Birth

Individual Observations, Quality Indicators, and calculated scales were provided in the assessment file with fields listed below.

Table A3.2: HCRS File Two Fields

Field Name	Data Type	Description
A1	Date	Client Episode Assessment Reference Date
CC1	Date	Home Care Case Open/Reopened Date
Client_Province	Character	Province code
A2	Numeric	Reason for Assessment
B1a	Numeric	Memory Recall Ability Short Term
B1b	Numeric	Memory Recall Ability Procedural
B2a	Numeric	Cognitive Skills for Daily Decision Making
B2b	Numeric	Cognitive Skills for Daily Decision Making (Worsening Flag)
B3a	Numeric	Indications of Delirium (Last 7 Days)
B3b	Numeric	Indications of Delirium (Last 90 Days)
BB1	Character	Gender
BB8j	Numeric	BB8j - Source client_rfp table
BB8k	Numeric	BB8k - Source client_rfp table
C1	Numeric	Hearing
C2	Numeric	Making Self Understood
C3	Numeric	Ability to understand others

C4	Numeric	Communication decline
CC2	Numeric	Reason for Referral
CC4	Numeric	Time since last Hospital Stay
CC5	Numeric	Where lived at time of Referral
CC6	Numeric	Who lived with at time of Referral
CC7	Numeric	Prior Residential Care Facility Placement
CC8	Numeric	Residential History
CC3a	Numeric	Patient Understanding of Goals of Care (Nursing)
CC3b	Numeric	Patient Understanding of Goals of Care (Monitoring)
CC3c	Numeric	Patient Understanding of Goals of Care (Rehabilitation)
CC3d	Numeric	Patient Understanding of Goals of Care (Client/Family Education)
CC3e	Numeric	Patient Understanding of Goals of Care (Family Respite)
CC3f	Numeric	Patient Understanding of Goals of Care (Palliative)
D1	Numeric	vision code
D2	Numeric	vision limitation flag
D3	Numeric	visual decline flag
E2	Numeric	Mood Decline
E4	Numeric	Changes in Behaviour Symptoms
E1a	Numeric	Indicators of Depression, Anxiety, Sad Mood (A feeling of Sadness or Depression)
E1b	Numeric	Indicators of Depression, Anxiety, Sad Mood (Persistant Anger with Self or Others)
E1c	Numeric	Indicators of Depression, Anxiety, Sad Mood (Expressions of Unrealistic Fears)
E1d	Numeric	Indicators of Depression, Anxiety, Sad Mood (Repetitive Health Complaints)
E1e	Numeric	Indicators of Depression, Anxiety, Sad Mood (Repetitive anxious Complaints, Concerns)
E1f	Numeric	Indicators of Depression, Anxiety, Sad Mood (Sad, Pained, Worried Facial Expressions)
E1g	Numeric	Indicators of Depression, Anxiety, Sad Mood (Recurrent Crying, Tearfulness)
E1h	Numeric	Indicators of Depression, Anxiety, Sad Mood (Withdrawel from Activities or Interest)
E1i	Numeric	Indicators of Depression, Anxiety, Sad Mood (Reduced Social Interaction)
E3a	Numeric	Behaviour Symptons (Wandering)
E3b	Numeric	Behaviour Symptons (Verbally Abusive)
E3c	Numeric	Behaviour Symptons (Physically Abusive)

E3d	Numeric	Behaviour Symptons (Socially Inappropriate/Disruptive Behavioural Symptoms)
E3e	Numeric	Behaviour Symptons (Resists Care)
F2	Numeric	Change in Social Activity
F1a	Numeric	Social Involvement (at ease with others)
F1b	Numeric	Social Involvement (Openly expresses conflict or anger)
F3a	Numeric	Isolation (Length of time alone during the day)
F3b	Numeric	Isolation (Client indicates feeling lonely)
G1eA	Numeric	Informal Helper Primary (Lives with Client)
G1eB	Numeric	Informal Helper Secondary (Lives with Client)
G1fA	Numeric	Informal Helper Primary (Relationship to Client)
G1fB	Numeric	Informal Helper Secondary (Relationship to Client)
G1gA	Numeric	Informal Helper Primary Advice or emotional support
G1gB	Numeric	Informal Helper Secondary Advice or emotional support
G1hA	Numeric	Informal Helper Primary IADL care
G1hB	Numeric	Informal Helper Secondary IADL care
G1iA	Numeric	Informal Helper Primary ADL care
G1iB	Numeric	Informal Helper Secondary ADL care
G1jA	Numeric	Informal Helper Primary Additional Support Emotional
G1jB	Numeric	Informal Helper Secondary Additional Support Emotional
G1kA	Numeric	Informal Helper Primary Additional Support IADL Care
G1kB	Numeric	Informal Helper Secondary Additional Support IADL Care
G1lA	Numeric	Informal Helper Primary Additional Support ADL Care
G1lB	Numeric	Informal Helper Secondary Additional Support ADL Care
G2a	Numeric	Client Caregiver Status (Unable to Continue)
G2b	Numeric	Client Caregiver Status (Unsatisfied with Support)
G2c	Numeric	Client Caregiver Status (Expresses feelings of Distress)
G2d	Numeric	Client Caregiver Status (None of the Above)
G3a	Numeric	Extent of Informal Help/Hours of care (Weekdays)

G3b	Numeric	Extent of Informal Help/Hours of care (Weekends)
H3	Numeric	ADL decline flag
H5	Numeric	stair climbing code
H1aA	Numeric	Source client IADL Self Performance (Meal Preparation)
H1aB	Numeric	Source client IADL Difficulty (Meal Preparation)
H1bA	Numeric	Source client IADL Self Performance (Ordinary Houswork)
H1bB	Numeric	Source client IADL Difficulty (Ordinary Houswork)
H1cA	Numeric	Source client IADL Self Performance (Managing Finances)
H1cB	Numeric	Source client IADL Difficulty (Managing Finances)
H1dA	Numeric	Source client IADL Self Performance (Managing Medications)
H1dB	Numeric	Source client IADL Difficulty (Managing Medications)
H1eA	Numeric	Source client IADL Self Performance (Phone Use)
H1eB	Numeric	Source client IADL Difficulty (Phone Use)
H1fA	Numeric	Source client IADL Self Performance (Shopping)
H1fB	Numeric	Source client IADL Difficulty (Shopping)
H1gA	Numeric	Source client IADL Self Performance (Transportation)
H1gB	Numeric	Source client IADL Difficulty (Transportation)
H2a	Numeric	Source client ADL Self Performance (Mobility in Bed)
H2b	Numeric	Source client ADL Self Performance (Transfer)
H2c	Numeric	Source client ADL Self Performance (Locomotion in Home)
H2d	Numeric	Source client ADL Self Performance (LOCOMOTION OUTSIDE OF HOME)
H2e	Numeric	Source client ADL Self Performance (DRESSING UPPER BODY)
H2f	Numeric	Source client ADL Self Performance (DRESSING LOWER BODY)
H2g	Numeric	Source client ADL Self Performance (EATING)
H2h	Numeric	Source client ADL Self Performance (TOILET USE)
H2i	Numeric	Source client ADL Self Performance (PERSONAL HYGEINE)

H2j	Numeric	Source client ADL Self Performance (BATHING)
H4a	Numeric	Source client locomotion Indoors
H4b	Numeric	Source client locomotion Outdoors
H6a	Numeric	Stamina went ouside days code
H6b	Numeric	Stamina physical activity hours code
H7a	Numeric	Source client functional potential (Client believes cabable of increased functional independence)
H7b	Numeric	Source client functional potential (Caregiver believes Client is capable of increased functional independence)
H7c	Numeric	Source client functional potential (Good prospects of Recovery)
H7d	Numeric	Source client functional potential (None of the Above)
I3	Numeric	Bowel incontinence
I1a	Numeric	Bladder continence
I1b	Numeric	Worsening of Bladder Incontinance
I2a	Numeric	Source client bladder device Pads
I2b	Numeric	Source client bladder device Catheter
I2c	Numeric	Source client bladder device None of the above
J1a	Numeric	Diseases Cerebrovascular accident
J1aa	Numeric	Diseases Renal failure
J1ab	Numeric	Diseases Thyroid disease (hyper or hypo_)
J1ac	Numeric	Diseases NONE OF THE ABOVE
J1b	Numeric	Diseases Congestive heart failure
J1c	Numeric	Diseases Coronary artery disease
J1d	Numeric	Diseases Hypertension
J1e	Numeric	Diseases Irregularly irregular pulse
J1f	Numeric	Diseases Peripheral vascular disease
J1g	Numeric	Diseases Alzheimer's
J1h	Numeric	Diseases Dementia other than Alzheimer's disease
J1i	Numeric	Diseases Head trauma
J1j	Numeric	Diseases Hemiplegia/hemiparesis
J1k	Numeric	Diseases Multiple Sclerosis
J1l	Numeric	Diseases Parkinsonism
J1m	Numeric	Diseases Arthritis
J1n	Numeric	Diseases Hip fracture
J1o	Numeric	Diseases Other fractures
J1p	Numeric	Diseases Osteoporosis
J1q	Numeric	Diseases Cataract

J1r	Numeric	Diseases Glaucoma
J1s	Numeric	Diseases Any psychiatric diagnosis
J1t	Numeric	Diseases HIV infection
J1u	Numeric	Diseases Pneumonia
J1v	Numeric	Diseases Tuberculosis
J1w	Numeric	Diseases Urinary tract infection
J1x	Numeric	Diseases Cancer
J1y	Numeric	Diseases Diabetes
J1z	Numeric	Diseases Emphysema/COPD/asthma
J2a	Character	ICD10 Diagnosis
J2b	Character	ICD10 Diagnosis
J2c	Character	ICD10 Diagnosis
J2d	Character	ICD10 Diagnosis
K5	Numeric	FALLS FREQUENTLY
K1a	Numeric	Preventive Health Measures (Blood Pressure Measured)
K1b	Numeric	Preventive Health Measures (Influenza Vaccination)
K1c	Numeric	Preventive Health Measures (Tests for Blood in stool or Screening Endoscopy)
K1d	Numeric	Preventive Health Measures (Breast Exam/Mammography)
k1e	Numeric	Preventive Health Measures (None of the Above)
K2a	Numeric	Problem Conditions Present (Diarrhea)
K2b	Numeric	Problem Conditions Present (Urinating Issues)
K2c	Numeric	Problem Conditions Present (Fever)
K2d	Numeric	Problem Conditions Present (Loss of appetite)
K2e	Numeric	Problem Conditions Present (Vomiting)
K2f	Numeric	Problem Conditions Present (None of the Above)
K3a	Numeric	Problem Conditions (Chest Pain)
K3b	Numeric	Problem Conditions (No bowel movement 3 days)
K3c	Numeric	Problem Conditions (Dizzines/light headed)
K3d	Numeric	Problem Conditions (Edema)
K3e	Numeric	Problem Conditions (Shortness of Breath)
K3f	Numeric	Problem Conditions (Delusions)
K3g	Numeric	Problem Conditions (Hallucinations)
K3h	Numeric	Problem Conditions (None of the Above)
K4a	Numeric	pain frequency
K4b	Numeric	pain intensity code
K4c	Numeric	pain disruption flag
K4d	Numeric	pain character code

K4e	Numeric	adequate medication code
K6a	Numeric	Danger of Fall Unsteady Gait
K6b	Numeric	Danger of Fall Client Limits Activity
K7a	Numeric	Lifestyle (Drinking/Smoking Concerns)
K7b	Numeric	Lifestyle (Drinking/Smoking troubles)
K7c	Numeric	Lifestyle Smoked or Chewed Tobacco Daily
K8a	Numeric	Health Status (Client believes poor health)
K8b	Numeric	Health Status (Conditions of Unstability)
K8c	Numeric	Health Status (Experiencing flare-up)
K8d	Numeric	Health Status (Treatment Changed)
K8e	Numeric	Health Status (end stage disease)
K8f	Numeric	Health Status (None of the Above)
K9a	Numeric	Other Status Indications (Fearful of a family member or caregiver)
K9b	Numeric	Other Status Indications (Unusually poor hygiene)
K9c	Numeric	Other Status Indications (Unexplained injuries, broken bones, or burns)
K9d	Numeric	Other Status Indications (Neglected, abused, or mistreated)
K9e	Numeric	Other Status Indications (Physically restrained)
K9f	Numeric	Other Status Indications (None of the Above)
L3	Numeric	nutrition hydration status Swallowing
L1a	Numeric	nutrition hydration status Unintended weight loss
L1b	Numeric	nutrition hydration status Severe Malnutrition
L1c	Numeric	nutrition hydration status Morbid Obesity
L2a	Numeric	Source client consumption (one or fewer meals daily)
L2b	Numeric	Source client consumption (Noticeable decrease)
L2c	Numeric	Source client consumption (Insufficient fluid)
L2d	Numeric	Source client consumption (Enteral tube feeding)
M1a	Numeric	Oral Status (Problem Chewing)
M1b	Numeric	Oral Status (Mouth is dry)
M1c	Numeric	Oral Status (Problem brushing teath)
M1d	Numeric	Oral Status (None of the above)
N1	Numeric	Skin Problems
N2a	Numeric	Pressure Ulcer
N2b	Numeric	Stasis Ulcer
N3a	Numeric	Other Skin Problems Burns
N3b	Numeric	Other Skin Problems Open Lesions

N3c	Numeric	Other Skin Problems Tears or cuts
N3d	Numeric	Other Skin Problems Surgical Wound
N3e	Numeric	Other Skin Problems Corns, calluses, structural problems, infections, fungi
N3f	Numeric	Other Skin Problems (None of the Above)
N4	Numeric	History of Resolved Pressure Ulcer
N5a	Numeric	Wound / Ulcer Care Antibiotics, systemic or topical
N5b	Numeric	Wound / Ulcer Care Dressings
N5c	Numeric	Wound / Ulcer Care Surgical wound care
N5d	Numeric	Wound / Ulcer Care Other wound/ulcer care
N5e	Numeric	Wound / Ulcer Care (None of the Above)
O1a	Numeric	Home Environment (Lighting in evening)
O1b	Numeric	Home Environment (Flooring and carpeting)
O1c	Numeric	Home Environment (Bathroom and toilet room)
O1d	Numeric	Home Environment (Kitchen)
O1e	Numeric	Home Environment (Heating and cooling)
O1f	Numeric	Home Environment (Personal safety)
O1g	Numeric	Home Environment (Access to home)
O1h	Numeric	Home Environment (Access to rooms in house)
O1i	Numeric	Home Environment (None of the Above)
O2a	Numeric	Living Arrangment Recent change in living arrangement
O2b	Numeric	Living Arrangment Client/Caregiver believes client better off with change to arrangements
P1aA	Numeric	Formal Care (# of Days) Home health aides
P1aB	Numeric	Formal Care (Hours) Home health aides
P1aC	Numeric	Formal Care (Minutes) Home health aides
P1bA	Numeric	Formal Care (# of Days) Visiting nurses
P1bB	Numeric	Formal Care (Hours) Visiting nurses
P1bC	Numeric	Formal Care (Minutes) Visiting nurses
P1cA	Numeric	Formal Care (# of Days) Homemaking services
P1cB	Numeric	Formal Care (Hours) Homemaking services
P1cC	Numeric	Formal Care (Minutes) Homemaking services
P1dA	Numeric	Formal Care (# of Days) Meals
P1dB	Numeric	Formal Care (Hours) Meals
P1dC	Numeric	Formal Care (Minutes) Meals
P1eA	Numeric	Formal Care (# of Days) Volunteer services
P1eB	Numeric	Formal Care (Hours) Volunteer services
P1eC	Numeric	Formal Care (Minutes) Volunteer services
P1fA	Numeric	Formal Care (# of Days) Physical therapy
P1fB	Numeric	Formal Care (Hours) Physical therapy

P1fC	Numeric	Formal Care (Minutes) Physical therapy
P1gA	Numeric	Formal Care (# of Days) Occupational therapy
P1gB	Numeric	Formal Care (Hours) Occupational therapy
P1gC	Numeric	Formal Care (Minutes) Occupational therapy
P1hA	Numeric	Formal Care (# of Days) Speech therapy
P1hB	Numeric	Formal Care (Hours) Speech therapy
P1hC	Numeric	Formal Care (Minutes) Speech therapy
P1iA	Numeric	Formal Care (# of Days) Day care or day hospital
P1iB	Numeric	Formal Care (Hours) Day care or day hospital
P1iC	Numeric	Formal Care (Minutes) Day care or day hospital
P1jA	Numeric	Formal Care (# of Days) Social worker in home
P1jB	Numeric	Formal Care (Hours) Social worker in home
P1jC	Numeric	Formal Care (Minutes) Social worker in home
P2a	Numeric	Special Treatments, Therapies, Programs. (Oxygen)
P2aa	Numeric	Special Treatments, Therapies, Programs. (None of the Above)
P2b	Numeric	Special Treatments, Therapies, Programs. (Respirator for assistive breathing)
P2c	Numeric	Special Treatments, Therapies, Programs. (All other respiratory treatments)
P2d	Numeric	Special Treatments, Therapies, Programs. (Alcohol/drug treatment program)
P2e	Numeric	Special Treatments, Therapies, Programs. (Blood transfusion)
P2f	Numeric	Special Treatments, Therapies, Programs. (Chemotherapy)
P2g	Numeric	Special Treatments, Therapies, Programs. (Dialysis)
P2h	Numeric	Special Treatments, Therapies, Programs. (IV infusion – central)
P2i	Numeric	Special Treatments, Therapies, Programs. (IV infusion – peripheral)
P2j	Numeric	Special Treatments, Therapies, Programs. (Medication by injection)
P2k	Numeric	Special Treatments, Therapies, Programs. (Ostomy care)
P2l	Numeric	Special Treatments, Therapies, Programs. (Radiation)
P2m	Numeric	Special Treatments, Therapies, Programs. (Tracheostomy care)
P2n	Numeric	Special Treatments, Therapies, Programs. (Exercise therapy)

P2o	Numeric	Special Treatments, Therapies, Programs. (Occupational therapy)
P2p	Numeric	Special Treatments, Therapies, Programs. (Physical therapy)
P2q	Numeric	Special Treatments, Therapies, Programs. (Day centre)
P2r	Numeric	Special Treatments, Therapies, Programs. (Day hospital)
P2s	Numeric	Special Treatments, Therapies, Programs. (Hospice care)
P2t	Numeric	Special Treatments, Therapies, Programs. (Physician or clinic visit)
P2u	Numeric	Special Treatments, Therapies, Programs. (Respite care)
P2v	Numeric	Special Treatments, Therapies, Programs. (Daily nurse monitoring)
P2w	Numeric	Special Treatments, Therapies, Programs. (Nurse monitoring less than daily)
P2x	Numeric	Special Treatments, Therapies, Programs. (Medical alert bracelet or electronic security alert)
P2y	Numeric	Special Treatments, Therapies, Programs. (Skin treatment)
P2z	Numeric	Special Treatments, Therapies, Programs. (Special diet)
P3a	Numeric	Management of Equipment (Oxygen)
P3b	Numeric	Management of Equipment (IV)
P3c	Numeric	Management of Equipment (Catheter)
P3d	Numeric	Management of Equipment (Ostomy)
P4a	Numeric	Visits in last 90 days or since last Assessment (Hospital)
P4b	Numeric	Visits in last 90 days or since last Assessment (Emergency Department)
P4c	Numeric	Visits in last 90 days or since last Assessment (Emergent Care)
P5	Numeric	Treatment Goals Met
P6	Numeric	Overall Change in Needs
P7	Numeric	Trade Offs
Q1	Numeric	Number of Medications
Q2a	Numeric	Receipt of Psychotropic Medication (Antipsychotic/Neuroleptic)
Q2b	Numeric	Receipt of Psychotropic Medication (Anxiolytic)
Q2c	Numeric	Receipt of Psychotropic Medication (Antidepressant)

Q2d	Numeric	Receipt of Psychotropic Medication (Hypnotic)
Q3	Numeric	Medical Oversight
Q4	Numeric	Compliance/Adherence with Medications
ADL_long_hc	Numeric	Activities of Daily Living Long Form Calculation
ADL_short_hc	Numeric	Activities of Daily Living Short Form Calculation
ADL_hier_hc	Numeric	Activities of Daily Living Hierarchy Calculation
Chess_hc	Numeric	Change in Health, End Stage Disease and Symptoms and Signs Score
CPS_hc	Numeric	Cognitive Performance Scale
DRS_hc	Numeric	Depression Rating Scale
IADL_Inv_HC	Numeric	Instrumental Activities of Daily Living Involvement Scale
IADL_Difficulty_hc	Numeric	Instrumental Activities of Daily Living Difficulty Scale
pain_hc	Numeric	Pain Scale
PURS_hc	Numeric	Pressure Ulcer Risk Scale
maple_hc	Numeric	Method For Assigning Priority Levels Score
Physical_Activity_CAP2_HC	Numeric	Physical Activity Promotion Client Assessment Protocol
IADL_CAP2_HC	Numeric	Instrumental Activities of Daily Living Client Assessment Protocol
ADL_CAP2_HC	Numeric	Activities of Dialy Living Client Assessment Protocol
Environment_CAP2_HC	Numeric	Home Environment Optimization Client Assessment Protocol
Institution_CAP2_HC	Numeric	Institutional Risk Client Assessment Protocol
Cognitive_CAP2_HC	Numeric	Cognitive Loss Client Assessment Protocol
Delirium_CAP2_HC	Numeric	Delirium Client Assessment Protocol
Communication_CAP2_HC	Numeric	Communication Client Assessment Protocol
Mood_CAP2_HC	Numeric	Mood Client Assessment Protocol
Behaviour_CAP2_HC	Numeric	Behaviour Client Assessment Protocol
Abuse_CAP2_HC	Numeric	Abusive Relationship Client Assessment Protocol
Support_CAP2_HC	Numeric	Informal Support Client Assessment Protocol
Social_CAP2_HC	Numeric	Social Relationship Client Assessment Protocol
Falls_CAP2_HC	Numeric	Falls Client Assessment Protocol
Pain_CAP2_HC	Numeric	Pain Client Assessment Protocol
Ulcer_CAP2_HC	Numeric	Pressure Ulcer Client Assessment Protocol
Cardio_CAP2_HC	Numeric	Cardio Respiratory Conditions Client Assessment Protocol
Dehydration_CAP2_HC	Numeric	Dehydration Client Assessment Protocol

Feeding_CAP2_HC	Numeric	Feeding Tube Client Assessment Protocol
Medication_CAP2_HC	Numeric	Appropriate Medication Client Assessment Protocol
Urinary_CAP2_HC	Numeric	Urinary Incontinence Client Assessment Protocol
Bowel_CAP2_HC	Numeric	Bowel Conditions Client Assessment Protocol
AX_IN_HOSPITAL_IND_CODE	Character	Assessment In hospital
END_OF_LIFE_IND_CODE	Character	End of Life Indicaotr
OVRNGHT_HOSPTAL_VST_IND_CODE	Character	Overnight Hospital Visit indicator
ER_VISIT_IND_CODE	Character	Emergency Department Visit Indicator
EMERGENT_CARE_VISIT_IND_CODE	Character	Emergent Care Visit Indicator
INFORMAL_CAREGIVER_IND_CODE	Character	Informal Caregiver Indicator
CAREGIVER_BURDEN_IND_CODE	Character	Caregiver under Burden Indicator
PRIOR_RESIDENT_CARE_IND_CODE	Character	Patient Prior in Residential Care
CLIENT_FIRST_AX_IND_CODE	Character	Client First Assessment Indicator
CLIENT_LAST_AX_IND_CODE	Character	Client Last Assessment
SINCE_LAST_AX_DAYS	Numeric	Days since last assessment
EPISODE_FIRST_AX_IND_CODE	Character	First Assessment current Episode
EPISODE_LAST_AX_IND_CODE	Character	Last Assessment current Episode
HC_IP_Flag	Numeric	Home Care Quality Indicator Intake Profile Flag
HC_QI_Flag	Numeric	Home Care Quality Indicator Inclusion Flag
HC_InadequateMeal_N	Numeric	Home Care Quality Indicator Prevalence of Inadequate Meals
HC_InadequateMeal_D	Numeric	Home Care Quality Indicator Prevalence of Inadequate Meals
HC_WeightLoss_N	Numeric	Home Care Quality Indicator Prevalence of Weight Loss
HC_WeightLoss_D	Numeric	Home Care Quality Indicator Prevalence of Weight Loss
HC_Dehydration_N	Numeric	Home Care Quality Indicator Prevalence of Dehydration
HC_Dehydration_D	Numeric	Home Care Quality Indicator Prevalence of Dehydration
HC_MedReview_N	Numeric	Home Care Quality Indicator Prevalence of Not Receiving a Medication Review by a Physician
HC_MedReview_D	Numeric	Home Care Quality Indicator Prevalence of Not Receiving a Medication Review by a Physician
HC_NoAsstDevice_N	Numeric	Home Care Quality Indicator Prevalence of No Assistive Device Among Clients with Difficulty in Locomotion

HC_NoAsstDevice_D	Numeric	Home Care Quality Indicator Prevalence of No Assistive Device Among Clients with Difficulty in Locomotion
HC_RehabPotential_N	Numeric	Home Care Quality Indicator Prevalence of ADL/Rehabilitation Potential and No Therapies
HC_RehabPotential_D	Numeric	Home Care Quality Indicator Prevalence of ADL/Rehabilitation Potential and No Therapies
HC_Falls_N	Numeric	Home Care Quality Indicator Prevalence of Falls
HC_Falls_D	Numeric	Home Care Quality Indicator Prevalence of Falls
HC_Isolation_N	Numeric	Home Care Quality Indicator Prevalence of Social Isolation
HC_Isolation_D	Numeric	Home Care Quality Indicator Prevalence of Social Isolation
HC_Delirium_N	Numeric	Home Care Quality Indicator Prevalence of Delirium
HC_Delirium_D	Numeric	Home Care Quality Indicator Prevalence of Delirium
HC_NegativeMood_N	Numeric	Home Care Quality Indicator Prevalence of Negative Mood
HC_NegativeMood_D	Numeric	Home Care Quality Indicator Prevalence of Negative Mood
HC_DailyPain_N	Numeric	Home Care Quality Indicator Prevalence of Disruptive or Intense Daily Pain
HC_DailyPain_D	Numeric	Home Care Quality Indicator Prevalence of Disruptive or Intense Daily Pain
HC_PainControl_N	Numeric	Home Care Quality Indicator Prevalence of Inadequate Pain Control Among Those with Pain
HC_PainControl_D	Numeric	Home Care Quality Indicator Prevalence of Inadequate Pain Control Among Those with Pain
HC_Neglect_N	Numeric	Home Care Quality Indicator Prevalence of Neglect/Abuse
HC_Neglect_D	Numeric	Home Care Quality Indicator Prevalence of Neglect/Abuse
HC_Injury_N	Numeric	Home Care Quality Indicator Prevalence of Any Injuries
HC_Injury_D	Numeric	Home Care Quality Indicator Prevalence of Any Injuries
HC_Vaccination_N	Numeric	Home Care Quality Indicator Prevalence of Not Receiving Influenza Vaccination
HC_Vaccination_D	Numeric	Home Care Quality Indicator Prevalence of Not Receiving Influenza Vaccination

HC_Hospital_N	Numeric	Home Care Quality Indicator Prevalence of Hospitalization
HC_Hospital_D	Numeric	Home Care Quality Indicator Prevalence of Hospitalization
HC_Incidence_6	Numeric	Home Care Quality Indicator
HC_Incidence_12	Numeric	Home Care Quality Indicator

Appendix 4: CCRS (Continuing Care Reporting System)

Two separate files were also provided for the Continuing Care Reporting Data. The first file represented Episodes/admissions to Continuing Care while the second file contained individual assessment records.

Table A4.1: CCRS File One Fields

Field Name	Data Type	Description
HCN_MBUN	Numeric	Unique Patient Identifier
EPISODE_ID_MBUN	Numeric	Unique identifier for the Continuing Care Episode
facility_code_mbun	Numeric	Unique Identifier for the Facility
PROVINCE_CODE	Character	Province
AA5B_PROV_ISSUE_HEALTH_CARD	Character	Province Issuing Health Care Card
LAST_TRANSFER_DATE	Date	Last Patient Transfer Date
AA2_SEX_CODE	Character	Patient Gender
CONSISTENT_SEX_IND	Numeric	Consistent Gender
ENTRY_DATE	Date	Patient Entry Date
ENTRY_TYPE	Numeric	Patient Entry Type
DISCHARGE_DATE	Date	Discharge Date
DISCHARGE_FLAG_IND	Numeric	Discharged Flag Indicator
DISCHARGE_SERVICE_TYPE	Numeric	Discharge Service Type
DISCHARGE_REASON	Numeric	Discharge Reason
DISCHARGE_LOS_DAYS	Numeric	Length of Stay at Discharge
EPISODE_AX_STATUS	Numeric	Episode Assessment Status
AB4_RESIDENT_POSTAL_CODE	Character	Patient Postal Code of Residence
RES_PROVINCE	Character	Patient Province of Residence
FISCAL_QUARTER_ENTRY	Character	CIHI Fiscal Quarter of Entry
FISCAL_YEAR_ENTRY	Numeric	CIHI Fiscal Year of Entry
FISCAL_QUARTER_DISCHARGE	Character	CIHI Fiscal Quarter of Discharge
FISCAL_YEAR_DISCHARGE	Numeric	CIHI Fiscal Year of Discharge
ASSUMED_DISCHARGE_DATE	Date	Assumed Discharge Date
BIRTH_YEAR	Numeric	Patient Year of Birth

Individual Observations, Quality Indicators, and calculated scales were provided in the assessment file with fields listed below.

Table A4.2: CCRS File Two Fields

Field Name	Data Type	Description
episode_id_mbun	Numeric	Unique Indicator for the Continuing care Episode
assessment_id_mbun	Numeric	Unique Indicator for the Assessment
PREVIOUS_AX_ID_mbun	Numeric	Unique Indicator for the Previous Assessment for the Patient
facility_code_mbun	Numeric	Unique identifier for the Facility
PROVINCE_CODE	Character	The Province Code
ACTIVE_NEW_STATUS	Numeric	ACTIVE_NEW_STATUS
ASSESSMENT_DATE	Date	The Date of the Assessment
AA8_ASSESSMENT_TYPE	Numeric	The type of Assessment (Quarterly, Annual, Initial, etc.)
FISCAL_YEAR_AX	Numeric	CIHI Fiscal Year
FISCAL_QUARTER_AX	Character	CIHI Fiscal Quarter
DAY_IND	Numeric	Day Indicator
QUARTER_IND	Numeric	Quarter Indicator
AX_ANNUAL_FACILITY_IND	Numeric	Assessment Annual Facility Indicator (Used in Calculations of Quality Indicators)
AX_ANNUAL_SECTOR_IND	Numeric	Assessment Annual Sector Indicator (Used in Calculations of Quality Indicators)
AX_PREV_QTR_IND	Numeric	Assessment Previous Quarter Indicator (Used in Calculations of Quality Indicators)
B1_COMATOSE	Numeric	Flag for resident comatose status
B2A_SHORT_TERM_MEMORY_OK	Numeric	Short-term memory OK/appears to recall after 5 minutes
B2B_LONG_TERM_MEMORY_OK	Numeric	Long-term memory OK/appears to recall long past
B3A_CURRENT_SEASON	Numeric	Memory/Recall ability: Current season
B3B_LOCATION_OF_OWN_ROOM	Numeric	Memory/Recall ability: Location of own room
B3C_STAFF_NAMES_FACES	Numeric	Memory/Recall ability: Staff names/faces
B3D_AWARE_IN_NURSING_HOME	Numeric	Memory/Recall ability: That he/she is in a facility
B4_COGNITIVE_SKILLS	Numeric	Cognitive Skills for Daily Decision-Making
B5A_EASILY_DISTRACTED	Numeric	Indicators of Delirium/Periodic Disordered Thinking/Awareness: EASILY DISTRACTED
B5B_PERIODS_OF_ALT_PERCEPT	Numeric	Indicators of Delirium/Periodic Disordered Thinking/Awareness: PERIODS OF ALTERED PERCEPTION OR AWARENESS OF SURROUNDINGS
B5C_EPISODES_OF_DISORG_SPEECH	Numeric	Indicators of Delirium/Periodic Disordered Thinking/Awareness: EPISODES OF DISORGANIZED SPEECH
B5D_PERIODS_OF_RESTLESSNESS	Numeric	Indicators of Delirium/Periodic Disordered Thinking/Awareness: PERIODS OF RESTLESSNESS
B5E_PERIODS_OF_LETHARGY	Numeric	Indicators of Delirium/Periodic Disordered Thinking/Awareness: PERIODS OF LETHARGY

B5F_MENTAL_FUNCTION_VARIES	Numeric	Indicators of Delirium/Periodic Disordered Thinking/Awareness: MENTAL FUNCTION VARIES OVER THE COURSE OF THE DAY
B6_CHANGE_COGNITIVE_STATUS	Numeric	Change in Cognitive Status (previous 90 days or since last assessment)
C1_HEARING	Numeric	Hearing
C2A_HEARING_AID_USED	Numeric	Communication Devices/Techniques: Hearing aid, present and used regularly
C2B_HEARING_AID_NOT_USED	Numeric	Communication Devices/Techniques: Hearing aid, present and not used regularly
C2C_OTHER_RECEPT_COMM_TECH	Numeric	Communication Devices/Techniques: Other receptive communication techniques used.
C3A_SPEECH	Numeric	Modes of Expression: Speech
C3B_WRITING_MESSAGES	Numeric	Modes of Expression: Writing messages to express or clarify needs
C3C_SIGN_LANGUAGE	Numeric	Modes of Expression: American sign language or Braille
C3D_SIGNS_GESTURES	Numeric	Modes of Expression: Signs/gestures/sounds
C3E_COMMUNICATION_BOARD	Numeric	Modes of Expression: Communication board
C3F_OTHER_EXPRESSION_MODE	Numeric	Modes of Expression: Other mode of expression
C4_MAKING_SELF_UNDERSTOOD	Numeric	Making self understood
C5_SPEECH_CLARITY	Numeric	Speech Clarity
C6_UNDERSTANDS_OTHERS	Numeric	Understands Others
C7_CHANGE_IN_COMMUNICATION	Numeric	Change In Communication
D1_VISION	Numeric	Vision: Indicate the resident's ability to see close objects in adequate light and with glasses, if used.
D2A_SIDE_VISION_PROBLEMS	Numeric	Side vision problems, decreased peripheral vision, e.g. leaves food on one side of tray, difficulty travelling, bumps into people and objects, misjudges placement of chair when seating self
D2B_SEES_HALOS	Numeric	Experiences any of the following: sees halos or rings around lights; sees "curtains" over eyes
D3_VISUAL_APPLIANCES	Numeric	Indicate whether the resident uses any of the following: glasses, contact lenses or a magnifying glass.
E1A_NEGATIVE_STATEMENTS	Numeric	Verbal Expressions of Distress: Resident makes negative statements
E1B_REPETITIVE_QUESTIONS	Numeric	Verbal Expressions of Distress: Repetitive questions
E1C_REPETITIVE_VERBALIZATIONS	Numeric	Verbal Expressions of Distress: Repetitive verbalizations
E1D_PERSISTENT_ANGER	Numeric	Verbal Expressions of Distress: Persistent anger with self or others
E1E_SELF_DEPRECATION	Numeric	Verbal Expressions of Distress: Self deprecation
E1F_EXPRESS_UNREALISTIC_FEAR	Numeric	Verbal Expressions of Distress: Expressions of what seem to be unrealistic fears

E1G_RECURRENT_STATEMENTS	Numeric	Verbal Expressions of Distress: Recurrent statements that something terrible is about to happen
E1H_REPEAT_HEALTH_COMPLAINTS	Numeric	Verbal Expressions of Distress: Repetitive health complaints
E1I_REPEAT_ANXIOUS_COMPLAINTS	Numeric	Verbal Expressions of Distress: Repetitive anxious complaints/concerns (non-health related)
E1J_UNPLEASANT_MOOD_IN_MORNING	Numeric	Sleep-cycle Issues: Unpleasant mood in morning
E1K_INSOMNIA	Numeric	Sleep-cycle Issues: Insomnia/change in usual sleep pattern
E1L_SAD_FACIAL_EXPRESSION	Numeric	Sad, Apathetic, Anxious Appearance: Sad, pained, worried facial expressions
E1M_CRYING	Numeric	Sad, Apathetic, Anxious Appearance: Crying, tearfulness
E1N_REPEAT_PHYSICAL_MOVEMENTS	Numeric	Sad, Apathetic, Anxious Appearance: Repetitive physical movements
E1O_WITHDRAWAL_FROM_ACTIVITIES	Numeric	Loss of Interest: Withdrawal from activities of interest.
E1P_REDUCED_SOCIAL_INTERACTION	Numeric	Loss of Interest: Reduced social interaction
E2_MOOD_PERSISTENCE	Numeric	Mood Persistence: Indicate whether one or more indicators or depression, anxiety or sad mood were not easily altered by attempts to "cheer up", console, or reassure the resident over last seven (7) days.
E3_CHANGE_IN_MOOD	Numeric	Change In Mood: Indicate whether resident's mood status has changed as compared to status of 90 days ago (or since last assessment if less than 90 days).
E4AA_WANDERING_FREQ	Numeric	WANDERING (moved with no rational purpose, seemingly oblivious to needs or safety)
E4AB_WANDERING_ALTER	Numeric	WANDERING (moved with no rational purpose, seemingly oblivious to needs or safety)
E4BA_VERBAL_ABUSE_FREQ	Numeric	VERBALLY ABUSIVE BEHAVIOURAL SYMPTOMS (others were threatened, screamed at, cursed at)
E4BB_VERBAL_ABUSE_ALTER	Numeric	VERBALLY ABUSIVE BEHAVIOURAL SYMPTOMS (others were threatened, screamed at, cursed at)
E4CA_PHYSICAL_ABUSE_FREQ	Numeric	PHYSICALLY ABUSIVE BEHAVIOURAL SYMPTOMS (others were hit, shoved, scratched, sexually abused)
E4CB_PHYSICAL_ABUSE_ALTER	Numeric	PHYSICALLY ABUSIVE BEHAVIOURAL SYMPTOMS (others were hit, shoved, scratched, sexually abused)
E4DA_DISRUPTIVE_FREQ	Numeric	SOCIALLY INAPPROPRIATE/DISRUPTIVE BEHAVIOURAL SYMPTOMS (made disruptive sounds, noisiness, screaming, self-abusive acts, sexual behaviour or disrobing in public, smeared/threw food/feces, hoarding, rummaging through others belongings)

E4DB_DISRUPTIVE_ALTER	Numeric	SOCIALLY INAPPROPRIATE/DISRUPTIVE BEHAVIOURAL SYMPTOMS (made disruptive sounds, noisiness, screaming, self-abusive acts, sexual behaviour or disrobing in public, smeared/threw food/feces, hoarding, rummaging through others belongings)
E4EA_RESISTS_CARE_FREQ	Numeric	RESISTS CARE (resisted taking medications/injections, ADL assistance, or eating)
E4EB_RESISTS_CARE_ALTER	Numeric	RESISTS CARE (resisted taking medications/injections, ADL assistance, or eating)
E5_CHANGE_IN_BEHAVIOUR_SYMPTOM	Numeric	Change In Behavioural Symptoms
F1A_EASY_INTERACT_W_OTHER	Numeric	At ease interacting with others
F1B_EASY_PLANNED_ACTIVITY	Numeric	At ease doing planned or structured activities
F1C_EASY_SELF_INITIATE_ACTIVTY	Numeric	At ease doing self-initiated activities
F1D_ESTABLISH_OWN_GOALS	Numeric	Establishes own goals
F1E_PURSUES_INVOLVEMENT	Numeric	Pursues involvement in life of facility, e.g. makes/keeps friends; involved in group activities; responds positively to new activities; assists at religious services
F1F_ACCEPTS_INVITATIONS	Numeric	Accepts invitations into most group activities
F2A_CONFLICT_W_STAFF	Numeric	Covert/open conflict with or repeated criticism of staff
F2B_UNHAPPY_W_ROOMMATE	Numeric	Unhappy with roommate
F2C_UNHAPPY_W_OTHER_RESIDENTS	Numeric	Unhappy with residents other than roommate
F2D_CONFLICT_W_FAMILY	Numeric	Openly expresses conflict/anger with family/friends
F2E_NO_CONTACT_W_FAMILY	Numeric	Absence of personal contact with family/friends
F2F_RECENT_LOSS_FAMILY	Numeric	Recent loss of close family member/friend
F2G_ADJUST_TO_ROUTINE_CHNG	Numeric	Does not adjust easily to change in routines
F3A_IDENTIFY_PAST_ROLES	Numeric	Strong identification with past roles and life status
F3B_SAD_OVER_LOST_ROLES	Numeric	Expresses sadness/anger /empty feeling over lost roles/status
F3C_PERCEIVES_DIFF_ROUTINE	Numeric	Resident perceives that daily routine (customary routine, activities) is very different from prior pattern in the community
G1AA_BED_MOBILITY_SELF	Numeric	How resident moves to and from lying position, turns side to side, and positions body while in bed
G1AB_BED_MOBILITY_SUPPORT	Numeric	How resident moves to and from lying position, turns side to side, and positions body while in bed
G1BA_TRANSFER_SELF	Numeric	How resident moves between surfaces. to/from: bed, chair, wheelchair, standing position (EXCLUDE to/from bath/toilet)
G1BB_TRANSFER_SUPPORT	Numeric	How resident moves between surfaces. to/from: bed, chair, wheelchair, standing position (EXCLUDE to/from bath/toilet)

G1CA_WALK_IN_ROOM_SELF	Numeric	How resident walks between locations in his/her room
G1CB_WALK_IN_ROOM_SUPPORT	Numeric	How resident walks between locations in his/her room
G1DA_WALK_IN_CORRIDOR_SELF	Numeric	How resident walks in corridor on unit
G1DB_WALK_IN_CORRIDOR_SUPPORT	Numeric	How resident walks in corridor on unit
G1EA_LOCOMOT_ON_UNIT_SELF	Numeric	How resident moves between locations in his/her room and adjacent corridor on same floor. If in wheelchair, selfsufficiency once in chair
G1EB_LOCOMOT_ON_UNIT_SUPPORT	Numeric	How resident moves between locations in his/her room and adjacent corridor on same floor. If in wheelchair, selfsufficiency once in chair
G1FA_LOCOMOT_OFF_UNIT_SELF	Numeric	How resident moves to and returns from off-unit locations, e.g. areas set aside for dining, activities, or treatments. If facility has only one floor, how resident moves to and from distant areas on the floor. If in wheelchair, self-sufficiency once in ch
G1FB_LOCOMOT_OFF_UNIT_SUPPORT	Numeric	How resident moves to and returns from off-unit unit locations
G1GA_DRESSING_SELF	Numeric	How resident puts on, fastens, takes off all items of street clothing, including donning/removing prosthesis
G1GB_DRESSING_SUPPORT	Numeric	How resident puts on, fastens, takes off all items of street clothing, including donning/removing prosthesis
G1HA_EATING_SELF	Numeric	How resident eats and drinks (regardless of skill). Includes intake of nourishment by other means, (e.g. tube feeding, total parenteral nutrition)
G1HB_EATING_SUPPORT	Numeric	How resident eats and drinks (regardless of skill). Includes intake of nourishment by other means, (e.g. tube feeding, total parenteral nutrition)
G1IA_TOILET_USE_SELF	Numeric	How resident uses the toilet room (or commode, bed pan, urinal); transfers on/off toilet, cleanses, changes pad, manages ostomy or catheter, adjusts clothes
G1IB_TOILET_USE_SUPPORT	Numeric	How resident uses the toilet room (or commode, bed pan, urinal); transfers on/off toilet, cleanses, changes pad, manages ostomy or catheter, adjusts clothes
G1JA_PERSONAL_HYGIENE_SELF	Numeric	How resident maintains personal hygiene, including combing hair; brushing teeth; shaving; applying makeup; washing/drying face, hands, and perineum (EXCLUDE baths and showers)
G1JB_PERSONAL_HYGIENE_SUPPORT	Numeric	How resident maintains personal hygiene, including combing hair; brushing teeth; shaving; applying makeup; washing/drying face, hands, and perineum (EXCLUDE baths and showers)

G2A_BATHING_SELF	Numeric	Bathing Self: Indicate how the resident takes full body bath/shower, sponge bath, and transfer in/out of tub/shower.
G2B_BATHING_SUPPORT	Numeric	Bathing Support: Indicate how the resident takes full body bath/shower, sponge bath, and transfer in/out of tub/shower.
G3A_BALANCE_WHILE_STANDING	Numeric	Balance While Standing
G3B_BALANCE_WHILE_SITTING	Numeric	Balance While Sitting
G4AA_NECK_RANGE_OF_MOTION	Numeric	Neck Range Of Motion
G4AB_NECK_VOLUNTARY_MOVEMENT	Numeric	Neck Voluntary Movement
G4BA_ARM_RANGE_OF_MOTION	Numeric	Arm Range Of Motion
G4BB_ARM_VOLUNTARY_MOVEMENT	Numeric	Arm Voluntary Movement
G4CA_HAND_RANGE_OF_MOTION	Numeric	Hand Range Of Motion
G4CB_HAND_VOLUNTARY_MOVEMENT	Numeric	Hand Voluntary Movement
G4DA_LEG_RANGE_OF_MOTION	Numeric	Leg Range Of Motion
G4DB_LEG_VOLUNTARY_MOVEMENT	Numeric	Leg Voluntary Movement
G4EA_FOOT_RANGE_OF_MOTION	Numeric	Foot Range Of Motion
G4EB_FOOT_VOLUNTARY_MOVEMENT	Numeric	Foot Voluntary Movement
G4FA_OTHER_LTD_RANGE_OF_MOTION	Numeric	Other Ltd Range Of Motion
G4FB_OTHER_LTD_VOLUNTARY_LOSS	Numeric	Limitation or loss in other joints not listed
G5A_CANE_WALKER	Numeric	Cane/walker/crutch
G5B_WHEELED_SELF	Numeric	Wheeled self
G5C_OTHER_PERSON_WHEELED	Numeric	Other person wheeled
G5D_WHEELCHAIR_PRIMARY_LOCOMOT	Numeric	Wheelchair primary mode of locomotion
G6A_BEDFAST	Numeric	Bedfast all or most of time
G6B_BED_RAILS_FOR_BED_MOBILITY	Numeric	Bed rails used for bed mobility or transfer
G6C_LIFTED_MANUALLY	Numeric	Lifted manually
G6D_LIFTED_MECHANICALLY	Numeric	Lifted mechanically
G6E_TRANSFER_AID	Numeric	Transfer aid (e.g. slide board, trapeze, cane, walker, brace)
G7_TASK_SEGMENTATION	Numeric	Task Segmentation
G8A_RES_MORE_INDEPENDENCE	Numeric	Resident believes self to be capable of increased independence in at least some ADLs
G8B_STAFF_MORE_INDEPENDENCE	Numeric	Direct care staff believe resident is capable of increased independence in at least some ADLs
G8C_SLOW_PERFORMING_TASKS	Numeric	Resident able to perform tasks/activity but is very slow
G8D_AM_PM_DIFFER_ADLS	Numeric	Difference in ADL Self-Performance or ADL Support comparing mornings to evenings
G9_CHANGE_ADL_FUNCTION	Numeric	Change ADL Function
H1A_BOWEL_CONTINENCE_SELF	Numeric	Bowel Continence Self: Control of bowel movement, with appliance or bowel continence programs, if employed
H1B_BLADDER_CONTINENCE_SELF	Numeric	Bladder Continence Self: Control of urinary bladder function (if dribbles, volume insufficient to soak through underpants), with

		appliances (e.g. oley) or continence programs, if used
H2A_BOWEL_ELIMINATION_REGULAR	Numeric	Bowel elimination pattern regular at least one movement every three (3) days
H2B_CONSTIPATION	Numeric	Constipation
H2C_DIARRHEA	Numeric	Diarrhea
H2D_FECAL_IMPACTION	Numeric	Fecal impaction
H3A_SCHEDULED_TOILETING_PLAN	Numeric	Any scheduled toileting plan
H3B_BLADDER_RETRAINING_PROGRAM	Numeric	Bladder retraining program
H3C_EXTERNAL_CATHETER	Numeric	External (condom) catheter
H3D_INDWELLING_CATHETER	Numeric	Indwelling catheter
H3E_INTERMITTENT_CATHETER	Numeric	Intermittent catheter
H3F_DID_NOT_USE_TOILET	Numeric	Did not use toilet room/commode/urinal
H3G_PADS_BRIEFS_USED	Numeric	Pads/briefs used
H3H_ENEMAS_IRRIGATION	Numeric	Enemas/irrigation
H3I_OSTOMY_PRESENT	Numeric	Ostomy present
H4_CHANGE_URINARY_CONTINENCE	Numeric	Change Urinary Continence
I1A_DIABETES_MELLITUS	Numeric	Diabetes Mellitus
I1B_HYPERTHYROIDISM	Numeric	Hyperthyroidism
I1C_HYPOTHYROIDISM	Numeric	Hypothyroidism
I1D_ARTERIO_HEART_DISEASE	Numeric	Arteriosclerotic Heart Disease
I1E_CARDIAC_DYSRHYTHMIAS	Numeric	Cardiac Dysrhythmias
I1F_CONGESTIVE_HEART_FAILURE	Numeric	Congestive Heart Failure
I1G_DEEP_VEIN_THROMBOSIS	Numeric	Deep Vein Thrombosis
I1H_HYPERTENSION	Numeric	Hypertension
I1I_HYPOTENSION	Numeric	Hypotension
I1J_PERIPHERAL_VASC_DISEASE	Numeric	Peripheral Vasc Disease
I1K_OTHER_CARDIOVASC_DISEASE	Numeric	Other Cardiovascular Disease
I1L_ARTHRITIS	Numeric	Arthritis
I1M_HIP_FRACT	Numeric	Hip Fracture
I1N_MISSING_LIMB	Numeric	Missing Limb
I1O_OSTEOPOROSIS	Numeric	Osteoporosis
I1P_PATHOLOGICAL_BONE_FRACT	Numeric	Pathological Bone Fract
I1Q_AMYOTROPHIC_LAT_SCLEROSIS	Numeric	Amyotrophic Lateral Sclerosis
I1R_ALZHEIMERS	Numeric	Alzheimers
I1S_APHASIA	Numeric	Aphasia
I1T_CEREBRAL_PALSY	Numeric	Cerebral Palsy
I1U_CEREBROVASC_ACCIDENT	Numeric	Cerebrovascular Accident
I1V_DEMENTIA_NOT_ALZHEIMERS	Numeric	Dementia Not Alzheimers
I1W_HEMIPLEGIA_HEMIPARESIS	Numeric	Hemiplegia Hemiparesis
I1X_HUNTINGTONS_CHOREA	Numeric	Huntingtons Chorea
I1Y_MULTIPLE_SCLEROSIS	Numeric	Multiple Sclerosis
I1Z_PARAPLEGIA	Numeric	Paraplegia
I1AA_PARKINSONS_DISEASE	Numeric	Parkinsons Disease

I1BB_QUADRIPLEGIA	Numeric	Quadriplegia
I1CC_SEIZURE_DISORDER	Numeric	Seizure Disorder
I1DD_TRANSIENT_ISCHEMIC_ATTACK	Numeric	Transient Ischemic Attack
I1EE_TRAUMATIC_BRAIN_INJURY	Numeric	Traumatic Brain Injury
I1FF_ANXIETY_DISORDER	Numeric	Anxiety Disorder
I1GG_DEPRESSION	Numeric	Depression
I1HH_MANIC_DEPRESSIVE	Numeric	Manic Depressive
I1II_SCHIZOPHRENIA	Numeric	Schizophrenia
I1JJ_ASTHMA	Numeric	Asthma
I1KK_EMPHYSEMA	Numeric	Emphysema
I1LL_CATARACTS	Numeric	Cataracts
I1MM_DIABETIC_RETINOPATHY	Numeric	Diabetic Retinopathy
I1NN_GLAUCOMA	Numeric	Glaucoma
I1OO_MACULAR_DEGENERATION	Numeric	Macular Degeneration
I1PP_ALLERGIES	Numeric	Allergies
I1QQ_ANEMIA	Numeric	Anemia
I1RR_CANCER	Numeric	Cancer
I1SS_GASTROINTESTINAL_DISEASE	Numeric	Gastrointestinal Disease
I1TT_LIVER_DISEASE	Numeric	Liver Disease
I1UU_RENAL_FAILURE	Numeric	Renal Failure
I2A_ANTIBIOTIC_RESIST_INFECT	Numeric	Antibiotic resistant infection, e.g. Methicillin resistant staph
I2B_CELLULITIS	Numeric	Cellulitis
I2C_CLOSTRIDIUM_DIFFICILE	Numeric	Clostridium difficile (c. diff)
I2D_CONJUNCTIVITIS	Numeric	Conjunctivitis
I2E_HIV_INFECTION	Numeric	HIV infection
I2F_PNEUMONIA	Numeric	Pneumonia
I2G_RESPIRATORY_INFECTION	Numeric	Respiratory infection
I2H_SEPTICEMIA	Numeric	Septicemia
I2I_SEXUALLY_TRANSMIT_DISEASES	Numeric	Sexually transmitted diseases
I2J_TUBERCULOSIS	Numeric	Tuberculosis (active)
I2K_URINARY_TRACT_INFECTION	Numeric	Urinary tract infection in last 30 days
I2L_VIRAL_HEPATITIS	Numeric	Viral hepatitis
I2M_WOUND_INFECTION	Numeric	Wound infection
I3A_OTHER_DIAG	Character	Other Diag
I3B_OTHER_DIAG	Character	Other Diag
I3C_OTHER_DIAG	Character	Other Diag
I3D_OTHER_DIAG	Character	Other Diag
I3E_OTHER_DIAG	Character	Other Diag
I3F_OTHER_DIAG	Character	Other Diag
J1A_WEIGHT_FLUCTUATION	Numeric	Weight gain or loss of 1.5 or more kilograms (3 lbs) in previous 7 days
J1B_INABILITY_TO_LIE_FLAT	Numeric	Inability to lie flat due to shortness of breath
J1C_DEHYDRATED	Numeric	Dehydrated; output exceeds input

J1D_INSUFFICIENT_FLUIDS	Numeric	Insufficient fluid; did NOT consume all/almost during last three (3) days
J1E_DELUSIONS	Numeric	Delusions
J1F_DIZZINESS	Numeric	Dizziness/Vertigo
J1G_EDEMA	Numeric	Edema
J1H_FEVER	Numeric	Fever
J1I_HALLUCINATIONS	Numeric	Hallucinations
J1J_INTERNAL_BLEEDING	Numeric	Internal bleeding
J1K_RECURRENT_LUNG_ASPIRATIONS	Numeric	Recurrent lung aspirations in last 90 days
J1L_SHORTNESS_OF_BREATH	Numeric	Shortness of breath
J1M_SYNCOPE	Numeric	Syncope (fainting)
J1N_UNSTEADY_GAIT	Numeric	Unsteady gait
J1O_VOMITING	Numeric	Vomiting
J2A_PAIN_SYMPTOMS_FREQ	Numeric	Pain Symptoms Frequency
J2B_PAIN_SYMPTOMS_INTENSITY	Numeric	Pain Symptoms Intensity
J3A_BACK_PAIN	Numeric	Back pain
J3B_BONE_PAIN	Numeric	Bone pain
J3C_CHEST_PAIN	Numeric	Chest pain while doing usual activities
J3D_HEADACHE	Numeric	Headache
J3E_HIP_PAIN	Numeric	Hip pain
J3F_INCISIONAL_PAIN	Numeric	Incisional pain
J3G_JOINT_PAIN_NOT_HIP	Numeric	Joint pain (other than hip)
J3H_SOFT_TISSUE_PAIN	Numeric	Soft tissue pain, e.g. lesion, muscle
J3I_STOMACH_PAIN	Numeric	Stomach pain
J3J_OTHER_PAIN	Numeric	Pain in other site not listed above
J4A_FELL_IN_PAST_30_DAYS	Numeric	Fell in past 30 days
J4B_FELL_IN_PAST_31_180_DAYS	Numeric	Fell in past 31 to 180 days
J4C_HIP_FRACT_IN_LAST_180_DAYS	Numeric	Hip fracture in last 180 days
J4D_OTHER_FRACT	Numeric	Other fracture in last 180 days
J5A_CONDITION_LEAD_TO_INSTABLE	Numeric	Conditions/diseases make resident's cognitive, ADL, behaviour patterns unstable (fluctuating, precarious, deteriorating)
J5B_EXPERIENCING_ACUTE_EPISODE	Numeric	Resident experiencing an acute episode or a flare-up recurrent or chronic problem
J5C_END_STAGE_DISEASE	Numeric	End-stage disease, six (6) months or less to live
K1A_CHEWING_PROBLEM	Numeric	Chewing problem
K1B_SWALLOWING_PROBLEM	Numeric	Swallowing problem
K1C_MOUTH_PAIN	Numeric	Mouth pain
K2A_HEIGHT	Numeric	Height of the patient
K2B_WEIGHT	Numeric	Weight of the patient
K3A_WEIGHT_LOSS	Numeric	Weight loss 5% or more in last 30 days; or 10% or more in last 180 days
K3B_WEIGHT_GAIN	Numeric	Weight gain 5% or more in last 30 days; or 10% or more in last 180 days
K4A_COMPLAINS_ABOUT_TASTE	Numeric	Complains about the taste of many foods

K4B_COMPLAINS_OF_HUNGER	Numeric	Regular or repetitive complaints of hunger
K4C_LEAVES_FOOD_UNEATEN	Numeric	Leaves 25% or more of food uneaten at most meals
K5A_PARENTERAL_IV	Numeric	Parenteral/IV
K5B_FEEDING_TUBE	Numeric	Feeding tube
K5C_MECHANIC_ALTERED_DIET	Numeric	Mechanically altered diet
K5D_ORAL_FEEDING	Numeric	Syringe (oral feeding)
K5E_THERAPEUTIC_DIET	Numeric	Therapeutic diet
K5F_DIETARY_SUPPLEMENT	Numeric	Dietary supplement between meals
K5G_PLATE_GUARD	Numeric	Plate guard, stabilized built up utensil, etc.
K5H_PLANNED_WEIGHT_CHANGE_PROG	Numeric	On a planned weight change program
K6A_TOTAL_CALORIES	Numeric	Parenteral or Enteral Intake: Total Calories
K6B_AVERAGE_FLUIDS	Numeric	Parenteral or Interal Intake: Average Fluid Intake
L1A_DEBRIS_IN_MOUTH	Numeric	Debris (soft, easily removable substances) present in mouth prior to going to bed at night
L1B_DENTURES_REMOVE_BRIDGE	Numeric	Has dentures or removable bridge
L1C_NATURAL_TEETH_LOST	Numeric	Some/all natural teeth lost; does not have or does not use dentures (or partial plates)
L1D_BROKEN_LOOSE_TEETH	Numeric	Broken, loose or carious teeth
L1E_INFLAMED_GUMS	Numeric	Inflamed gums (gingiva); swollen or bleeding gums; oral abscesses; ulcers or rashes
L1F_DAILY_CLEANING_TEETH	Numeric	Daily cleaning of teeth/dentures or daily mouth care by resident or staff
M1A_STAGE1_ULCERS	Numeric	Stage 1. A persistent area of skin redness (without a break in the skin) that does not disappear when pressure is relieved.
M1B_STAGE2_ULCERS	Numeric	Stage 2. A partial thickness loss of skin layers that presents clinically as an abrasion, blister, or shallow crater.
M1C_STAGE3_ULCERS	Numeric	Stage 3. A full thickness of skin is lost, exposing the subcutaneous tissues.presents as a deep crater with or without undermining adjacent tissue.
M1D_STAGE4_ULCERS	Numeric	Stage 4. A full thickness of skin and subcutaneous tissues is lost, exposing muscle or bone.
M2A_STAGE_OF_PRESSURE_ULCER	Numeric	Pressure ulcer: Any lesion caused by pressure resulting in damage of underlying tissue
M2B_STAGE_OF_STASIS_ULCER	Numeric	Stasis ulcer: Open lesion caused by poor circulation in the lower extremities
M3_HISTORY_OF_RESOLVED_ULCERS	Numeric	History of Resolved Ulcers
M4A_ABRASIONS_BRUISES	Numeric	Abrasions, bruises
M4B_BURNS	Numeric	Burns (second or third degree)
M4C_OPEN_LESIONS_NOT_ULCERS	Numeric	Open lesions other than ulcers, rashes, cuts, e.g. cancer lesions
M4D_RASHES	Numeric	Rashes. e.g. intertrigo, eczema, drug rash, heat rash, herpes zoster

M4E_SKIN_DESENSITIZED_TO_PAIN	Numeric	Skin desensitized to pain or pressure
M4F_SKIN_TEARS_OR_CUTS	Numeric	Skin tears or cuts (other than surgery)
M4G_SURGICAL_WOUNDS	Numeric	Surgical wounds
M5A_RELIEVING_DEVICE_CHAIR	Numeric	Pressure relieving device(s) for chair
M5B_RELIEVING_DEVICE_BED	Numeric	Pressure relieving device(s) for bed
M5C_TURNING_PROGRAM	Numeric	Turning/repositioning program
M5D_NUTRITION_INTERVENTION	Numeric	Nutrition or hydration intervention to manage skin problems
M5E_ULCER_CARE	Numeric	Ulcer care
M5F_SURGICAL_WOUND_CARE	Numeric	Surgical wound care
M5G_APPLY_DRESSINGS_NOT_FEET	Numeric	Application of dressings (with or without topical medications) other than to feet
M5H_APPLY_OINTMENTS_NOT_FEET	Numeric	Application of ointments/medications (other than to feet)
M5I_OTHER_PREVENT_NOT_FEET	Numeric	Other preventative or protective skin device (other than to feet)
M6A_HAS_FOOT_PROBLEM	Numeric	Resident has one or more foot problems, (e.g. corns, calluses, bunions, hammer toes, overlapping toes, pain, structural problems)
M6B_INFECTION_OF_FOOT	Numeric	Infection of the foot, (e.g. cellulitis, purulent drainage)
M6C_OPEN_LESIONS_ON_FOOT	Numeric	Open lesions on the foot
M6D_NAILS_CALLUSES_TRIMMED	Numeric	Nails/calluses trimmed during last 90 days
M6E_RECEIVED_PREVENT_FOOT_CARE	Numeric	Received preventative or protective foot care (e.g. used special shoes, inserts, pads, toe separators)
M6F_APPLY_DRESSING_FOOT	Numeric	Application of dressings (with or without topical medications)
N1A_TIME_AWAKE_MORNING	Numeric	Morning
N1B_TIME_AWAKE_AFTERNOON	Numeric	Afternoon
N1C_TIME_AWAKE_EVENING	Numeric	Evening
N2_AVERAGE_TIME_ACTIVITIES	Numeric	Average time involved in activities
N3A_PREF_ACT_OWN_ROOM	Numeric	Own room
N3B_PREF_ACT_ACTIVITY_ROOM	Numeric	Day/activity room
N3C_PREF_ACT_INSIDE	Numeric	Inside facility/off unit
N3D_PREF_ACT_OUTSIDE	Numeric	Outside facility
N4A_PREF_ACT_CARDS_GAMES	Numeric	Cards/other games
N4B_PREF_ACT_CRAFTS	Numeric	Crafts/arts
N4C_PREF_ACT_EXERCISE	Numeric	Exercise/sports
N4D_PREF_ACT_MUSIC	Numeric	Music
N4E_PREF_ACT_READING	Numeric	Reading/writing
N4F_PREF_ACT_SPIRITUAL	Numeric	Spiritual/religious activities
N4G_PREF_ACT_TRIPS	Numeric	Trips/shopping
N4H_PREF_ACT_WALKING	Numeric	Walking/wheeling outdoors
N4I_PREF_ACT_WATCH_TV	Numeric	Watching TV
N4J_PREF_ACT_GARDENING	Numeric	Gardening or plants
N4K_PREF_ACT_TALKING	Numeric	Talking or conversing

N4L_PREF_ACT_HELP_OTHERS	Numeric	Helping others
N5A_PREFER_CHANGE_IN_ACTIVITY	Numeric	Resident prefers change in type of activities in which resident is currently involved
N5B_PREFER_CHANGE_IN_INVOLV	Numeric	Resident prefers change in extent of resident involvement in activities
O1_NUM_OF_MEDICATIONS	Numeric	Number of Medications
O2_NEW_MEDICATIONS	Numeric	New Medications during the last 90 days
O3_DAYS_INJECTIONS	Numeric	Days injections: the number of days injections of any type were received in the last seven (7) days. Enter 0 if none used.
O4A_DAYS_ANTIPSYCHOTIC	Numeric	Antipsychotic: the number of days during last seven (7) days
O4B_DAYS_ANTIANXIETY	Numeric	Antianxiety : the number of days during last seven (7) days
O4C_DAYS_ANTIDEPRESSANTS	Numeric	Antidepressant: the number of days during last seven (7) days
O4D_DAYS_HYPNOTIC	Numeric	Hypnotic: the number of days during last seven (7) days
O4E_DAYS_DIURETIC	Numeric	Diuretic: the number of days during last seven (7) days
O4F_DAYS_ANALGESIC	Numeric	Analgesic: the number of days during last seven (7) days
P1AA_CHEMOTHERAPY	Numeric	Chemotherapy
P1AB_DIALYSIS	Numeric	Dialysis
P1AC_IV_MEDICATION	Numeric	IV medication
P1AD_INTAKE_OUTPUT	Numeric	Intake/output
P1AE_MONITOR_MEDICAL_CONDITION	Numeric	Monitoring acute medical condition
P1AF_OSTOMY_CARE	Numeric	Ostomy care
P1AG_OXYGEN_THERAPY	Numeric	Oxygen therapy
P1AH_RADIATION	Numeric	Radiation
P1AI_SUCTIONING	Numeric	Suctioning
P1AJ_TRACHEOSTOMY	Numeric	Tracheostomy care
P1AK_TRANSFUSIONS	Numeric	Transfusions
P1AL_VENTILATOR_OR_RESPIRATOR	Numeric	Ventilator or respirator
P1AM_ALCOHOL_DRUG_PROGRAM	Numeric	Alcohol/drug treatment program
P1AN_ALZHEIMER_CARE_UNIT	Numeric	Alzheimer.s/dementia special care unit
P1AO_HOSPICE_CARE	Numeric	Hospice care
P1AP_PAEDIATRIC_UNIT	Numeric	Pediatric care
P1AQ_RESPITE_CARE	Numeric	Respite care
P1AR_TRAINING_COMMUNITY_SKILLS	Numeric	Training in skills required to return to community
P1BAA_DAYS_SPEECH_THERAPY	Numeric	Record the number of days each of the following therapies was administered (for at least 15 minutes a day) in the last seven (7) calendar days. 0 if none or less than 15 minutes daily.
P1BAB_MINS_SPEECH_THERAPY	Numeric	Record the total minutes each of the following therapies was administered in the last seven (7) calendar days.

P1BBA_DAYS_OCCUPATION_THERAPY	Numeric	Record the number of days each of the following therapies was administered (for at least 15 minutes a day) in the last seven (7) calendar days. 0 if none or less than 15 minutes daily.
P1BBB_MINS_OCCUPATION_THERAPY	Numeric	Record the total minutes each of the following therapies was administered in the last seven (7) calendar days.
P1BCA_DAYS_PHYSICAL_THERAPY	Numeric	Record the number of days each of the following therapies was administered (for at least 15 minutes a day) in the last seven (7) calendar days. 0 if none or less than 15 minutes daily.
P1BCB_MINS_PHYSICAL_THERAPY	Numeric	Record the total minutes each of the following therapies was administered in the last seven (7) calendar days.
P1BDA_DAYS_RESPIRATORY_THERAPY	Numeric	Record the number of days each of the following therapies was administered (for at least 15 minutes a day) in the last seven (7) calendar days. 0 if none or less than 15 minutes daily.
P1BDB_MINS_RESPIRATORY_THERAPY	Numeric	Record the total minutes each of the following therapies was administered in the last seven (7) calendar days.
P1BEA_DAYS_PSYCHO_THERAPY	Numeric	Record the number of days each of the following therapies was administered (for at least 15 minutes a day) in the last seven (7) calendar days. 0 if none or less than 15 minutes daily.
P1BEB_MINS_PSYCHO_THERAPY	Numeric	Record the total minutes each of the following therapies was administered in the last seven (7) calendar days.
P1BFA_DAYS_RECREATION_THERAPY	Numeric	Record the number of days each of the following therapies was administered (for at least 15 minutes a day) in the last seven (7) calendar days. 0 if none or less than 15 minutes daily.
P1BFB_MINS_RECREATION_THERAPY	Numeric	Record the total minutes each of the following therapies was administered in the last seven (7) calendar days.
P2A_SPEC_BEHAVIOR_SYMP_PROGRAM	Numeric	Special behaviour symptom evaluation program
P2B_EVAL_BY_LICENSED_SPECIALST	Numeric	Evaluation by a licensed mental health specialist in last 90 days
P2C_GROUP_THERAPY	Numeric	Group therapy
P2D_RES_SPECIFIC_CHNGE_ENVIRO	Numeric	Resident-specific deliberate changes in the environment to address mood/behaviour/patterns, e.g. providing bureau in which to rummage
P2E_REORIENTATION	Numeric	Reorientation e.g. cueing
P3A_REHAB_DAYS_ROM_PASSIVE	Numeric	Range of motion (passive): Record the number of days each of the rehabilitation or restorative techniques or practices was provided to the resident for more than or

		equal to 15 minutes per day in the last 7 days.
P3B_REHAB_DAYS_ROM_ACTIVE	Numeric	Range of motion (active): Record the number of days each of the rehabilitation or restorative techniques or practices was provided to the resident for more than or equal to 15 minutes per day in the last 7 days.
P3C_REHAB_DAYS_SPLINT_ASSIST	Numeric	Splint or brace assistance: Record the number of days each of the rehabilitation or restorative techniques or practices was provided to the resident for more than or equal to 15 minutes per day in the last 7 days.
P3D_REHAB_DAYS_BED_MOBILITY	Numeric	Bed Mobility : Record the number of days each of the rehabilitation or restorative techniques or practices was provided to the resident for more than or equal to 15 minutes per day in the last 7 days.
P3E_REHAB_DAYS_TRANSFER	Numeric	Transfer: Record the number of days each of the rehabilitation or restorative techniques or practices was provided to the resident for more than or equal to 15 minutes per day in the last 7 days.
P3F_REHAB_DAYS_WALKING	Numeric	Walking: Record the number of days each of the rehabilitation or restorative techniques or practices was provided to the resident for more than or equal to 15 minutes per day in the last 7 days.
P3G_REHAB_DAYS_DRESSING	Numeric	Dressing or grooming: Record the number of days each of the rehabilitation or restorative techniques or practices was provided to the resident for more than or equal to 15 minutes per day in the last 7 days.
P3H_REHAB_DAYS_EATING	Numeric	Eating or swallowing: Record the number of days each of the rehabilitation or restorative techniques or practices was provided to the resident for more than or equal to 15 minutes per day in the last 7 days.
P3I_REHAB_DAYS_AMPUTATION	Numeric	Amputation/prosthesis care: Record the number of days each of the rehabilitation or restorative techniques or practices was provided to the resident for more than or equal to 15 minutes per day in the last 7 days.
P3J_REHAB_DAYS_COMMUNICATION	Numeric	Communication: Record the number of days each of the rehabilitation or restorative techniques or practices was provided to the resident for more than or equal to 15 minutes per day in the last 7 days.
P3K_REHAB_DAYS_OTHER	Numeric	Other: Record the number of days each of the rehabilitation or restorative techniques or practices was provided to the resident for more than or equal to 15 minutes per day in the last 7 days.

P4A_FULL_BED_RAILS	Numeric	Bed-rails.full bed rails on all open sides of bed
P4B_OTHER_TYPES_OF_RAILS	Numeric	Bed rails.other types of side rails used, e.g. half rail, one side
P4C_TRUNK_RESTRAINT	Numeric	Trunk restraint
P4D_LIMB_RESTRAINT	Numeric	Limb restraint
P4E_CHAIR_PREVENTS_RISING	Numeric	Chair prevents rising
P5_HOSPITAL_STAYS	Numeric	number of times resident was admitted to hospital in last 90 days (or since last assessment if less than 90 days).
P6_EMERGENCY_ROOM_VISITS	Numeric	number of times resident visited ER in last 90 days (or since last assessment if less than 90 days).
P7_DAYS_PHYSICIAN_VISITS	Numeric	In the last 14 days (or since admission if less than 14 days in facility), on how many days has the physician examined the resident
P8_DAYS_DOCTOR_ORDERS_CHANGED	Numeric	In the last 14 days (or since admission if less than 14 days in facility), on how many days has the physician changed the resident's orders
P9_ABNORMAL_LAB_VALUES	Numeric	whether the resident had abnormal lab values during the last 90 days (or since admission).
Q1A_WANTS_RETURN_TO_COMMUNITY	Numeric	Resident Expresses/Indicates Preference to Return to the Community
Q1B_SUPPORT_POSITIVE_DISCHARGE	Numeric	Resident Has a Support Person Who is Positive Towards Discharge
Q1C_STAY_SHORT_DURATION	Numeric	Stay Projected to be of Short Duration; Discharge Projected Within 90 Days
Q2_CHANGE_IN_CARE_NEEDS	Numeric	Whether the resident's overall level of self-sufficiency has changed significantly as compared to status of 90 days ago (or since last assessment if less than 90 days).
R1A_RES_PARTICIPATED_ASSESS	Numeric	Resident's Participation in Assessment
R1B_FAMILY_PARTICIPATED_ASSESS	Numeric	Family's Participation in Assessment
R1C_OTHER_PARTICIPATED_ASSESS	Numeric	Significant Other's Participation in Assessment
CPS	Numeric	Score for Cognitive Performance Scale for the resident on current ax
DRS	Numeric	Score for Depression Rating Scale for the resident on current ax
ISE	Numeric	Score for Index for Social Engagement for the resident on current ax
ADL_SHORT_FORM	Numeric	Score for ADL Short Form Scale for the resident on current ax
ADL_LONG_FORM	Numeric	Score for ADL Long Form Scale for the resident on current ax
ADL_HIERARCHY	Numeric	Score for ADL Hierarchy Scale for the resident on current ax
CHESS	Numeric	Score for CHESS for the resident on current ax
PAIN	Numeric	Score for Pain Scale for the resident on current ax

ABS	Numeric	Score for Aggressive Behaviour Scale for the resident on current ax
PURS	Numeric	Score for Pressure Ulcer Risk Scale
ADL_CAP	Numeric	Activities of Daily Living CAP
PHYSICAL_RESTRAINTS_CAP	Numeric	Physical Restraints CAP
COGNITIVE_LOSS_CAP	Numeric	Cognitive Loss CAP
DELIRIUM_CAP	Numeric	Delirium CAP
COMMUNICATION_CAP	Numeric	Communication CAP
MOOD_CAP	Numeric	Mood CAP
BEHAVIOUR_CAP	Numeric	Behaviour CAP
ACTIVITIES_CAP	Numeric	Activities CAP
SOCIAL_RELATIONSHIP_CAP	Numeric	Social Relationship CAP
FALLS_CAP	Numeric	Falls CAP
PAIN_CAP	Numeric	Pain CAP
PRESSURE_ULCER_CAP	Numeric	Pressure Ulcer CAP
CARDIO_RESPIRATORY_CONDITION_CAP	Numeric	Cardio Respiratory Condition CAP
UNDERNUTRITION_CAP	Numeric	Undernutrition CAP
DEHYDRATION_CAP	Numeric	Dehydration CAP
FEEDING_TUBE_CAP	Numeric	Feeding Tube CAP
APPROPRIATE_MEDICATIONS_CAP	Numeric	Appropriate Medications CAP
URINARY_INCONTINENCE_CAP	Numeric	Urinary Incontinence CAP
BOWEL_CONDITIONS_CAP	Numeric	Bowel Conditions CAP
NO_TRIGGERED_CAPS	Numeric	No Triggered CAPs
QI_CAT02_D	Numeric	Percent of residents with indwelling catheters Denominator
QI_CAT02_N	Numeric	Percent of residents with indwelling catheters Numerator
QI_CNT04_D	Numeric	Percent of residents with a urinary tract infection Denominator
QI_CNT04_N	Numeric	Percent of residents with a urinary tract infection Numerator
QI_DRG01_D	Numeric	Percent of residents on antipsychotics without a diagnosis of psychosis Denominator
QI_DRG01_N	Numeric	Percent of residents on antipsychotics without a diagnosis of psychosis Numerator
QI_FAL02_D	Numeric	Percent of residents who fell in the last 30 days Denominator
QI_FAL02_N	Numeric	Percent of residents who fell in the last 30 days Numerator
QI_INF0X_D	Numeric	Percent of residents with one or more infections Denominator
QI_INF0X_N	Numeric	Percent of residents with one or more infections Numerator
QI_NUT01_D	Numeric	Percent of residents with a feeding tube Denominator
QI_NUT01_N	Numeric	Percent of residents with a feeding tube Numerator
QI_PAI0X_D	Numeric	Percent of residents with pain Denominator

QI_PAI0X_N	Numeric	Percent of residents with pain Numerator
QI_PRU05_D	Numeric	Percent of residents who had a stage 2 to 4 pressure ulcer Denominator
QI_PRU05_N	Numeric	Percent of residents who had a stage 2 to 4 pressure ulcer Numerator
QI_RES01_D	Numeric	Percent of residents in daily physical restraints Denominator
QI_RES01_N	Numeric	Percent of residents in daily physical restraints Numerator
QI_WGT01_D	Numeric	Percent of residents who had unexplained weight loss Denominator
QI_WGT01_N	Numeric	Percent of residents who had unexplained weight loss Numerator
QI_ADL01_D	Numeric	Percent of residents whose late-loss ADL functioning (bed mobility, transfer, eating and toilet) worsened Denominator
QI_ADL01_N	Numeric	Percent of residents whose late-loss ADL functioning (bed mobility, transfer, eating and toilet) worsened Numerator
QI_ADL05_D	Numeric	Percent of residents whose mid-loss ADL functioning (transfer and locomotion) improved or who remained completely independent in mid-loss ADLs Denominator
QI_ADL05_N	Numeric	Percent of residents whose mid-loss ADL functioning (transfer and locomotion) improved or who remained completely independent in mid-loss ADLs Numerator
QI_ADL06_D	Numeric	Percent of residents whose early-loss ADL functioning (dressing and personal hygiene) improved or who remained completely independent in early-loss ADLs Denominator
QI_ADL06_N	Numeric	Percent of residents whose early-loss ADL functioning (dressing and personal hygiene) improved or who remained completely independent in early-loss ADLs Numerator
QI_ADL1A_D	Numeric	Percent of residents whose late-loss ADL functioning (bed mobility, transfer, eating and toilet) improved Denominator
QI_ADL1A_N	Numeric	Percent of residents whose late-loss ADL functioning (bed mobility, transfer, eating and toilet) improved Numerator
QI_ADL5A_D	Numeric	Percent of residents whose mid-loss ADL functioning (transfer and locomotion) worsened or who remained completely dependent in mid-loss ADLs Denominator
QI_ADL5A_N	Numeric	Percent of residents whose mid-loss ADL functioning (transfer and locomotion) worsened or who remained completely dependent in mid-loss ADLs Numerator
QI_ADL6A_D	Numeric	Percent of residents whose early-loss ADL functioning (dressing and personal hygiene) worsened or who remained completely dependent in early-loss ADLs Denominator
QI_ADL6A_N	Numeric	Percent of residents whose early-loss ADL functioning (dressing and personal hygiene)

		worsened or who remained completely dependent in early-loss ADLs Numerator
QI_ADLD7_D	Numeric	Percent of residents whose ADL self-performance worsened Denominator
QI_ADLD7_N	Numeric	Percent of residents whose ADL self-performance worsened Numerator
QI_BEHD4_D	Numeric	Percent of residents whose behavioural symptoms worsened Denominator
QI_BEHD4_N	Numeric	Percent of residents whose behavioural symptoms worsened Numerator
QI_BEHI4_D	Numeric	Percent of residents whose behavioural symptoms improved Denominator
QI_BEHI4_N	Numeric	Percent of residents whose behavioural symptoms improved Numerator
QI_CNT02_D	Numeric	Percent of residents whose bowel continence worsened Denominator
QI_CNT02_N	Numeric	Percent of residents whose bowel continence worsened Numerator
QI_CNT03_D	Numeric	Percent of residents whose bladder continence worsened Denominator
QI_CNT03_N	Numeric	Percent of residents whose bladder continence worsened Numerator
QI_CNT2A_D	Numeric	Percent of residents whose bowel continence improved Denominator
QI_CNT2A_N	Numeric	Percent of residents whose bowel continence improved Numerator
QI_CNT3A_D	Numeric	Percent of residents whose bladder continence improved Denominator
QI_CNT3A_N	Numeric	Percent of residents whose bladder continence improved Numerator
QI_COG01_D	Numeric	Percent of residents whose cognitive ability worsened Denominator
QI_COG01_N	Numeric	Percent of residents whose cognitive ability worsened Numerator
QI_COG1A_D	Numeric	Percent of residents whose cognitive ability improved Denominator
QI_COG1A_N	Numeric	Percent of residents whose cognitive ability improved Numerator
QI_COM01_D	Numeric	Percent of residents whose ability to communicate worsened Denominator
QI_COM01_N	Numeric	Percent of residents whose ability to communicate worsened Numerator
QI_COM1A_D	Numeric	Percent of residents whose ability to communicate improved Denominator
QI_COM1A_N	Numeric	Percent of residents whose ability to communicate improved Numerator
QI_DEL0X_D	Numeric	Percent of residents with symptoms of delirium Denominator
QI_DEL0X_N	Numeric	Percent of residents with symptoms of delirium Numerator
QI_MOB01_D	Numeric	Percent of residents whose ability to locomote worsened Denominator
QI_MOB01_N	Numeric	Percent of residents whose ability to locomote worsened Numerator

QI_MOB1A_D	Numeric	Percent of residents whose ability to locomote improved Denominator
QI_MOB1A_N	Numeric	Percent of residents whose ability to locomote improved Numerator
QI_MOD4A_D	Numeric	Percent of residents whose mood from symptoms of depression worsened Denominator
QI_MOD4A_N	Numeric	Percent of residents whose mood from symptoms of depression worsened Numerator
QI_PAN01_D	Numeric	Percent of residents whose pain worsened Denominator
QI_PAN01_N	Numeric	Percent of residents whose pain worsened Numerator
QI_PRU06_D	Numeric	Percent of residents whose stage 2 to 4 pressure ulcer worsened Denominator
QI_PRU06_N	Numeric	Percent of residents whose stage 2 to 4 pressure ulcer worsened Numerator
QI_PRU09_D	Numeric	Percent of residents who had a newly occurring stage 2 to 4 pressure ulcer Denominator
QI_PRU09_N	Numeric	Percent of residents who had a newly occurring stage 2 to 4 pressure ulcer Numerator
QI_RSPX2_D	Numeric	Percent of resident who developed a respiratory condition or have not gotten better Denominator
QI_RSPX2_N	Numeric	Percent of resident who developed a respiratory condition or have not gotten better Numerator

Appendix 5: Constellation Rule Processing Procedures

Our first stored procedure is PR_PROCESS_CONSTELLATION_RULES. This procedures is for processing constellation rules that identify records that meet a given criteria.

```
CREATE procedure [Constellation_Build].[pr_process_Constellation_Rules]
    @TaskExecKeyIn int,          -- Execution ID of the task that called the procedure
    @Database_Name varchar(128), -- database name for table being processed for rules
    @Schema_Name varchar(128),   -- Schema name for table being processed for rules
    @Table_Name varchar(128),    -- Table name being processed for rules
    @Status_code char(10)        -- status of the rule (Dev, Prod, etc.)
as
begin

    declare @Constellation_SQL_Code varchar(max) -- The condition the rule is testing as a sql statement
    Declare @Constellation_ID int                -- Identifier / Key for the rule being tested
/*
dbo.pr_process_Constellation_Rules

Process the SQL statements stored in the Constellation_rules table for the business rules of data
relationships. The rules are processed according to the table, schema, database name, and status of the
rules that are effective (i.e. terminated date is not populated or is dated in the future)

The procedure executes a cursor to return the individual rules from the Constellation_rules table.
This table contains the rules (SQL statements) that are read in one at a time and creates a dynamic SQL
merge statement. This SQL merge statement is then executed to retrieve the records that satisfy the rule.
The merge statement will save results into the Constellation_Join table by inserting new records, deleting
relations that no longer exist in the results, and leaving existing results that are still returned.

The value that is captured is the DW_SEQ_ID of the record that satisfies the rule. This could be a fact
table or dimension table identifier. For a dimension identifier we associate the dimension record to the
rule through a single bridge/cross reference table.

  Parameters:
  @taskexeckeyin        -       The task execution for the calling procedure
  @Database_Name        -       The Database for the target source table
  @Schema_name          -       The database schema
  @Target_Table_name    -       The Target table for the process stage

  Conditions:
        Sql statements are generated dynamically and the condition SQL statement
        needs to return a value. In order to work correctly the value being returned
        must have the proper column names and form.
        SELECT dw_seq_id,... from ... (any table and condition)

        The DW_SEQ_ID must be present as this is the identifier for the record. Additional fields may
exist but these are ignored. Only the returned DW_SEQ_ID is used.
    History:
        Robert Hart     2016-03-02   Original Version

*/

-- Get the Constellation rules cursor. This cursor retrieves the Constellation rules from our rules table
-- based on the metadata schema for the table name, database name, and schema name. We also check the
-- status code so we only process the required rules.

        declare c_getrules cursor for
        select SQL_Code, Constellation_Definition_ID
        from Constellation.Constellation_Build.Constellation_Definition
        where
                Database_Name=@Database_Name
                and Schema_Name=@Schema_Name
                and Table_Name=@Table_Name
```

```
            and status_Code=@Status_Code
            and getdate() between isnull(Rule_Effective_Date,'1900-01-01')
            and isnull(Rule_Terminated_Date,'2100-01-01')
    order by Sequence

--
-- for error handling we have a begin try statement/block
--

-- open our cursor to get the rules and fetch the record

  begin try

    open c_getrules
    fetch c_getrules into @Constellation_SQL_Code, @Constellation_ID

-- while our fetch works process the rules one at a time

    while @@fetch_status = 0
      begin
        Declare @SQL_constellation_statement nvarchar(max)

--
-- Constellation rules retrieved from the cursor are built into a merge SQL statement for execution as
-- dynamic SQL. The rule statement is combined as a with clause and the additional SQL to
-- retrieve the existing errors. Then execute this statement to save results in Constellation_Join.
-- It is a merge statement and we need to make some decisions based on the results. New records
-- are saved, existing records that are still returned are ignored, and existing records that are
-- no longer returned in the result set are deleted.
--

-- ok, so dynamic Statement creation is below.
set @SQL_constellation_statement = 'with constellation_Identifier_Sql_code as (' + @Constellation_SQL_Code
     +' )  merge [Constellation].[Constellation_Build].[Constellation_Join] as target '
     +'using ( '
     +'SELECT distinct ci.dw_seq_id, ' + cast(@Constellation_ID as varchar)
     + ' as Constellation_Definition_ID '
     +'  FROM constellation_Identifier_Sql_code as ci ) as source '
     +'on ( '
     +'source.dw_seq_id=target.dw_seq_id and '
     +'source.Constellation_Definition_ID=target.Constellation_Definition_ID ) '
     +'when not matched by target then '
     +'insert (Constellation_Definition_ID,dw_seq_id) '
     +'values (source.Constellation_Definition_ID,source.dw_seq_id) '
     +'when not matched by source and target.Constellation_Definition_ID= '
     + cast(@Constellation_ID as varchar) + '  then '
     +'delete; '

-- The print statement below is for debuging
-- print @SQL_constellation_statement

-- And we execute below
      Begin try
        exec sp_executesql @SQL_constellation_statement
        -- and we are done
        end try
      Begin Catch

--Standard error handling information captured and raise the error

      DECLARE @ErrorMessage NVARCHAR(4000);
      DECLARE @ErrorSeverity INT;
      DECLARE @ErrorState INT;
      DECLARE @ErrorNumber INT;

      SELECT
        @ErrorMessage = ERROR_MESSAGE(),
        @ErrorSeverity = ERROR_SEVERITY(),
        @ErrorState = ERROR_STATE(),
```

```
        @ErrorNumber = ERROR_NUMBER();

      Set @ErrorMessage = 'Error in Procedure pr_Process_Constellation by Value Rules' + @ErrorMessage;

      INSERT INTO [Constellation_Build].[Constellation_Rule_Execution_Failure]
        ([ErrorMessaqge], [ErrorSeverity], [ErrorState], [ErrorNumber], [Constellation_Definition_ID],
[Execution_date])
      VALUES (@ErrorMessage, @ErrorSeverity, @ErrorState, @ErrorNumber, @Constellation_ID,getdate())

 End Catch

      fetch c_getrules into @Constellation_SQL_Code, @Constellation_ID

End

-- close the cursor
    close c_getrules
    deallocate c_getrules

-- and we are done
end try

-- Now we close the rule

Begin Catch
  DECLARE @ErrorMessage2 NVARCHAR(4000);
  DECLARE @ErrorSeverity2 INT;
  DECLARE @ErrorState2 INT;
  DECLARE @ErrorNumber2 INT;

  SELECT
    @ErrorMessage2 = ERROR_MESSAGE(),
    @ErrorSeverity2 = ERROR_SEVERITY(),
    @ErrorState2 = ERROR_STATE(),
    @ErrorNumber2 = ERROR_NUMBER();

   Set @ErrorMessage = 'Error in Procedure pr_Process_Constellation_Rules related to Cursor and rules
table' + @ErrorMessage;

--  close c_error_cursor
--  deallocate c_error_cursor
    close c_getrules
    deallocate c_getrules

    INSERT INTO [Constellation_Build].[Constellation_Rule_Execution_Failure]
      ([ErrorMessaqge], [ErrorSeverity], [ErrorState], [ErrorNumber],  [Constellation_Definition_ID],
[Execution_date])
    VALUES (@ErrorMessage2, @ErrorSeverity2, @ErrorState2, @ErrorNumber2, NULL, getdate())
 End Catch
end
```

Our second stored procedure is PR_PROCESS_CONSTELLATION_BY_VALUE_RULES. This procedures is for processing constellation rules that identify a record and associate new information to the record.

```
CREATE procedure [Constellation_Build].[pr_process_Constellation_By_Value_Rules]
  @TaskExecKeyIn int,            -- Execution ID of the task
  @Database_Name varchar(128),   -- database name for table being processed
  @Schema_Name varchar(128),     -- Schema name for table being processed
  @Table_Name varchar(128),      -- Table name being processed
  @Status_code char(10)          -- status of the rule (Dev, Prod, etc.)
as
begin

  declare @Constellation_By_Value_SQL_Code varchar(max)  -- The rule being processed SQL statement
  Declare @Constellation_By_Value_ID int                 -- Identifier / Key for the rule being tested

/*
dbo.pr_process_Constellation_By_Value_Rules

Process the SQL statements stored in the Constellation_By_Value_rules table for the business rules of data
relationships. The rules are processed according to the table, schema, database name, and status of the
rules that are effective (i.e. terminated date is not populated or is dated in the future)

The procedure executes a cursor to return the individual rules from the Constellation_By_Value_rules
table. This table contains the rules (SQL statements) that are read in one at a time and creates a dynamic
SQL merge statement. This SQL merge statement is then executed to retrieve the records that satisfy the
rule. The merge statement will save results into the Constellation_By_Value_Join table by inserting new
records, deleting relations that no longer exist in the results, and leaving existing results that are
still returned.

The Information that is captured is the DW_SEQ_ID of the record and the value or information that we want
to associate with the record and satisfies the rule. This could be a fact table or dimension table
identifier. For a dimension identifier we associate the dimension record to the rule through a single
bridge/cross reference table.

Parameters:
  @taskexeckeyin        -       The task execution for the calling procedure
  @Database_Name        -       The Database for the target source table
  @Schema_name          -       The database schema
  @Target_Table_name    -       The Target table for the process stage

  Conditions:
   Sql statements are generated dynamically and the condition SQL statement
        needs to return a value pair (dw_seq_id, Value). In order to work correctly, the value being
        returned must have the proper column names and form.
        SELECT dw_seq_id, value ... from ... (any table and condition)


    The "DW_SEQ_ID" field must be present with this name as this is the identifier for the record.
    The "VALUE" field must exist by this name as that is also required to identify the dimension record.
    Additional fields may exist but these are ignored. Only the DW_SEQ_ID and VALUE are used.
    History:
        Robert Hart     2011-03-02   Orginal Version
        Robert Hart     2016-01-13   Modified to change catch so that processing continues and an error is
logged rather than fail.
*/


-- Get the Constellation rules cursor. This cursor retrieves the Constellation rules from our rules table
-- based on the metadata schema for the table name, database name, and schema name. We also check the
-- status code so we only process the required rules.

        declare c_getrules cursor for
        select SQL_Code, Constellation_By_Value_Definition_ID
        from Constellation.Constellation_Build.Constellation_By_Value_Definition
        where
```

```
                Database_Name=@Database_Name
                and Schema_Name=@Schema_Name
                and Table_Name=@Table_Name
                and status_Code=@Status_Code
        and getdate() between isnull(Rule_Effective_Date,'1900-01-01') and
isnull(Rule_Terminated_Date,'2100-01-01')
    order by Sequence

--
-- for error handling we have a begin try statement/block
--

Begin try

-- open our cursor to get the rules and fetch the record


  open c_getrules
  fetch c_getrules into @Constellation_By_Value_SQL_Code, @Constellation_By_Value_ID

-- while our fetch works process the rules one at a time

  while @@fetch_status = 0
    begin
    Declare @SQL_constellation_By_Value_statement nvarchar(max) -- New variable for SQL rule

--
-- Constellation rules retrieved from the cursor are built into a merge SQL statement for execution as
-- dynamic SQL. The rule statement is combined as a with clause and the additional SQL to
-- retrieve the existing errors. Then execute this statement to save results in
-- Constellation_By_Value_Join. It is a merge statement and we need to make some decisions based on
-- the results. New records or those with a new value are saved, existing records that are still returned
-- and have the same value are ignored, and those no longer returned or with a new value are deleted.
--

-- ok, so dynamic Statement creation is below.
    set @SQL_constellation_By_Value_statement = 'with constellation_By_Value_Identifier_Sql_code as (' +
@Constellation_By_Value_SQL_Code
    +' )  merge [Constellation].[Constellation_Build].[Constellation_By_Value_Join] as target '
    +'using ( '
    +'SELECT distinct ci.dw_seq_id, ci.value,'
    + cast(@Constellation_By_Value_ID as varchar) + ' as Constellation_By_Value_Definition_ID '
    +'  FROM constellation_By_Value_Identifier_Sql_code as ci ) as source '
    +'on ( '
    +'source.dw_seq_id=target.dw_seq_id and source.value=target.value and '
    +'source.Constellation_By_Value_Definition_ID=target.Constellation_By_Value_Definition_ID ) '
    +'when not matched by target then '
    +'insert (Constellation_By_Value_Definition_ID,dw_seq_id,value) '
    +'values (source.Constellation_By_Value_Definition_ID,source.dw_seq_id,source.value) '
    +'when not matched by source and target.Constellation_By_Value_Definition_ID= '
    + cast(@Constellation_By_Value_ID as varchar) + ' then delete; '

-- The print statement is for debuging
--                                       print @SQL_constellation_statement

-- And we execute below
    begin try
      exec sp_executesql @SQL_constellation_By_Value_statement
    end try

    Begin Catch

--Standard error handling information captured and raise the error

    DECLARE @ErrorMessage NVARCHAR(4000);
    DECLARE @ErrorSeverity INT;
    DECLARE @ErrorState INT;
    DECLARE @ErrorNumber INT;
```

```
    SELECT
      @ErrorMessage = ERROR_MESSAGE(),
      @ErrorSeverity = ERROR_SEVERITY(),
      @ErrorState = ERROR_STATE(),
      @ErrorNumber = ERROR_NUMBER();

    Set @ErrorMessage = 'Error in Procedure pr_Process_Constellation_by_Value_Rules' + @ErrorMessage;

    INSERT INTO [Constellation_Build].[Constellation_Value_Rule_Execution_Failure]
      ([ErrorMessaqge], [ErrorSeverity], [ErrorState], [ErrorNumber],
[Constellation_By_Value_Definition_ID], [Execution_date])
 VALUES (@ErrorMessage  ,@ErrorSeverity ,@ErrorState ,@ErrorNumber ,@Constellation_By_Value_ID,getdate())

  End Catch

  fetch c_getrules into @Constellation_By_Value_SQL_Code, @Constellation_By_Value_ID

End

-- close the cursor
  close c_getrules
  deallocate c_getrules

end try

  Begin Catch
    DECLARE @ErrorMessage2 NVARCHAR(4000);
    DECLARE @ErrorSeverity2 INT;
    DECLARE @ErrorState2 INT;
    DECLARE @ErrorNumber2 INT;

  SELECT
    @ErrorMessage2 = ERROR_MESSAGE(),
    @ErrorSeverity2 = ERROR_SEVERITY(),
    @ErrorState2 = ERROR_STATE(),
    @ErrorNumber2 = ERROR_NUMBER();

  Set @ErrorMessage = 'Error in Procedure pr_Process_constellation_by_value_rules related to Cursor and
rules table' + @ErrorMessage;

  close c_getrules
  deallocate c_getrules

  INSERT INTO [Constellation_Build].[Constellation_Value_Rule_Execution_Failure]
    ([ErrorMessaqge], [ErrorSeverity], [ErrorState], [ErrorNumber],
[Constellation_By_Value_Definition_ID], [Execution_date])
  VALUES(@ErrorMessage2, @ErrorSeverity2, @ErrorState2, @ErrorNumber2, NULL, getdate())

 End Catch

end
```

Our third stored procedure is PR_PROCESS_CONSTELLATION_RELATION_RULES. This procedures is for processing constellation rules that associates two records together by capturing each records unique identifier.

```
CREATE procedure [Constellation_Build].[pr_process_Constellation_Relation_Rules]
  @TaskExecKeyIn int,                     -- Execution ID of the task
  @Child_Database_Name varchar(128),      -- database name for child table being processed for rule
  @Child_Schema_Name varchar(128),        -- Schema name for child table being processed for rule
  @Child_Table_Name varchar(128),         -- Table name being processed
  @Status_code char(10)                   -- status of the rule (Dev, Prod, etc.)
as
begin

  declare @Constellation_SQL_Code varchar(max)  -- The Constellation sql statement
  Declare @Constellation_ID int                 -- Identifier / Key for the rule
/*
dbo.pr_process_Constellation_Relation_Rules

Process the SQL statements stored in the Constellation_relation_rules table for the business rules of data
relationships. The rules are processed according to the child table, schema, database name, and status of
the rules that are effective (i.e. terminated date is not populated or is dated in the future)

The procedure executes a cursor to return the individual rules from the Constellation_relation_rules
table. This table contains the rules (SQL statements) that are read in one at a time and creates a dynamic
SQL merge statement. This SQL merge statement is then executed to retrieve the records that satisfy the
rule. The merge statement will save results into the Constellation_relation_Join table by inserting new
records, deleting relations that no longer exist in the results, and leaving existing results that are
still returned.

The values that are captured are the DW_SEQ_ID of the parent record and the DW_SEQ_ID of the child record
that satisfies the rule. This could be a fact table or dimension table identifier but is normally two
facts. Dimension tables would normally be used with a value association as they contain information values
and do not normally contain foreign keys so are better suited to value association. Associations are
accomplished through bridge/cross reference table structures but are dependent on tools used as many do
not support complex table structure

  Parameters:
  @taskexeckeyin        -       The task execution for the calling procedure
  @Database_Name        -       The Database for the target source table
  @Schema_name          -       The database schema
  @Target_Table_name    -       The Target table for the process stage

  Conditions:
   Sql statements are generated dynamically and the condition SQL statement
        needs to return a value pair (child_dw_seq_id, parent_dw_seq_id). In order to work correctly,
        the value being returned must have the proper column names and form.
        SELECT child_dw_seq_id, parent_dw_seq_id... from ... (any table and condition)

        The two DW_SEQ_ID values must be present as these are the identifiers for the records. Additional
fields may exist but are ignored. Only the returned child_dw_seq_id, parent_dw_seq_id are used.
    History:
        Robert Hart     2016-03-02   Original Version

*/

-- Get the Constellation rules cursor. This cursor retrieves the Constellation rules from our rules table
-- based on the metadata schema for the child table name, database name, and schema name. We also check
-- the status code so we only process the required rules.

  declare c_getrules cursor for
  select SQL_Code, Constellation_By_Relation_Definition_ID
  from Constellation.Constellation_Build.Constellation_By_Relation_Definition
  where
```

```
    CHILD_Database_Name=@Child_Database_Name
    and CHILD_Schema_Name=@Child_Schema_Name
    and CHILD_Table_Name=@Child_Table_Name
    and status_Code=@Status_Code
    and getdate() between isnull(Rule_Effective_Date,'1900-01-01')
    and isnull(Rule_Terminated_Date,'2100-01-01')
    order by Sequence

--
-- for error handling we have a begin try statement/block
--

-- open our cursor to get the rules and fetch the record

    begin try
      open c_getrules
      fetch c_getrules into @Constellation_SQL_Code, @Constellation_ID

-- while our fetch works process the rules one at a time

      while @@fetch_status = 0
        begin
        Declare @SQL_constellation_statement nvarchar(max) -- New cursor for the actual rules

--
-- Constellation rules retrieved from the cursor are built into a merge SQL statement for execution as
-- dynamic SQL. The rule statement is combined as a with clause and the additional SQL to
-- retrieve the existing errors. Then execute this statement to save results in
-- Constellation_relation_Join. It is a merge statement and we need to make some decisions based on
-- the results. New records are saved, existing records that are still returned are ignored, and existing
-- records that are no longer returned in the result set are deleted.
--

-- Dynamic Statement creation is below.
   set @SQL_constellation_statement = 'with constellation_Identifier_Sql_code as (' +
@Constellation_SQL_Code +' )  merge
[Constellation].[Constellation_Build].[Constellation_By_Relation_Join] as target '
   +'using ( '
   +'SELECT distinct ci.child_dw_seq_id,ci.parent_dw_seq_id, '+ cast(@Constellation_ID as varchar)
   + ' as Constellation_By_Relation_Definition_ID '
   +'  FROM constellation_Identifier_Sql_code as ci ) as source '
   +'on ( '
   +'source.Child_dw_seq_id=target.Child_dw_seq_id and '
   +'source.Parent_dw_seq_id=target.Parent_dw_seq_id and '
   +'source.Constellation_By_Relation_Definition_ID=target.Constellation_By_Relation_Definition_ID ) '
   +'when not matched by target then '
   +'insert (Constellation_By_Relation_Definition_ID,Child_dw_seq_id,Parent_Dw_Seq_ID) '
   +'values (source.Constellation_By_Relation_Definition_ID, source.Child_dw_seq_id,
source.Parent_Dw_Seq_ID) '
   +'when not matched by source and target.Constellation_By_Relation_Definition_ID= '     +
cast(@Constellation_ID as varchar)
   + '  then delete; '

-- The print statement is for debuging
--   print @SQL_constellation_statement

-- And we execute below
  Begin try
    exec sp_executesql @SQL_constellation_statement
        -- and we are done
  end try
  Begin Catch

--Standard error handling information captured and raise the error

    DECLARE @ErrorMessage NVARCHAR(4000);
    DECLARE @ErrorSeverity INT;
    DECLARE @ErrorState INT;
    DECLARE @ErrorNumber INT;
```

```sql
    SELECT
      @ErrorMessage = ERROR_MESSAGE(),
      @ErrorSeverity = ERROR_SEVERITY(),
      @ErrorState = ERROR_STATE(),
      @ErrorNumber = ERROR_NUMBER();

   Set @ErrorMessage = 'Error in Procedure pr_Process_Constellation_Relation_Rules' + @ErrorMessage;

   INSERT INTO [Constellation_Build].[Constellation_Rule_Execution_Failure]
     ([ErrorMessaqge], [ErrorSeverity], [ErrorState], [ErrorNumber], [Constellation_Definition_ID], 
[Execution_date])
     VALUES (@ErrorMessage, @ErrorSeverity, @ErrorState, @ErrorNumber, @Constellation_ID,getdate())

 End Catch

fetch c_getrules into @Constellation_SQL_Code, @Constellation_ID

End
-- close the cursor
   close c_getrules
   deallocate c_getrules

end try

-- Now we close the rule

Begin Catch
  DECLARE @ErrorMessage2 NVARCHAR(4000);
  DECLARE @ErrorSeverity2 INT;
  DECLARE @ErrorState2 INT;
  DECLARE @ErrorNumber2 INT;

SELECT
  @ErrorMessage2 = ERROR_MESSAGE(),
  @ErrorSeverity2 = ERROR_SEVERITY(),
  @ErrorState2 = ERROR_STATE(),
  @ErrorNumber2 = ERROR_NUMBER();

  Set @ErrorMessage = 'Error in Procedure pr_Process_Constellation_Relation_Rules related to Cursor and 
rules table' + @ErrorMessage;

  close c_getrules
  deallocate c_getrules

  INSERT INTO [Constellation_Build].[Constellation_By_Relation_Rule_Execution_Failure]
    ([ErrorMessaqge], [ErrorSeverity], [ErrorState], [ErrorNumber], 
[Constellation_By_Relation_Definition_ID], [Execution_date])
     VALUES (@ErrorMessage2, @ErrorSeverity2, @ErrorState2, @ErrorNumber2, NULL, getdate())

 End Catch
end
```

Appendix 6: Sort Concatenate Database Aggregate String Function

The Sort concatenate database function below is a custom C# program that is used to create a concatenated string based on the values passed to it. This program creates an array where values are passed in one at a time. When finished, the program will sort the array and concatenate the values together in a comma separated string.

```
using System;
using System.Data;
using Microsoft.SqlServer.Server;
using System.Data.SqlTypes;
using System.Data.SqlClient;
using System.IO;
using System.Collections;
using System.Text;

[Serializable]
[SqlUserDefinedAggregate(
    Format.UserDefined, //use clr serialization to serialize the intermediate result
    IsInvariantToNulls = true, //optimizer property
    IsInvariantToDuplicates = false, //optimizer property
    IsInvariantToOrder = false, //optimizer property
    MaxByteSize = -1) //maximum size in bytes of persisted value
]
public class SortConcatenate : IBinarySerialize
{
    /// <summary>
    /// The variable that holds the intermediate result of the concatenation
    /// </summary>
    private ArrayList valuelist;

    /// <summary>
    /// Initialize the internal data structures
    /// </summary>
    public void Init()
    {
        this.valuelist = new ArrayList();
    }

    /// <summary>
    /// Accumulate the next value, but ignore if the value is null
    /// </summary>
    /// <param name="value"></param>
    public void Accumulate(SqlString value)
    {
        if (value.IsNull)
        {
            return;
        }
        this.valuelist.Add(value);
    }

    /// <summary>
    /// Merge the partially computed aggregate with this aggregate.
    /// </summary>
    /// <param name="other"></param>
```

```
    public void Merge(SortConcatenate group)
    {
        this.valuelist.AddRange(group.valuelist);
    }

    /// <summary>
    /// Called at the end of aggregation, to return the results of the aggregation.
    /// </summary>
    /// <returns></returns>
    public SqlString Terminate()
    {
        string output = string.Empty;
        //delete the trailing comma, if any
        this.valuelist.Sort();

        if (this.valuelist.Count > 0)
        {
            output = Convert.ToString ( this.valuelist[0] );
            for (int i = 1; i < valuelist.Count; i++)
                output = output + "," + Convert.ToString (this.valuelist[i]);
        }

        return new SqlString(output);
    }

    public void Read(BinaryReader r)
    {
        valuelist = new ArrayList();
        string[] tmpList = r.ReadString().Split('|');

        foreach (string entry in tmpList)
        {
            valuelist.Add(entry);
        }

    }

    public void Write(BinaryWriter w)
    {
        string[] tmpList = new string[valuelist.Count];
        for (int i = 0; i < valuelist.Count; i++)
        {
            tmpList[i] = Convert.ToString( valuelist[i] );
        }
        w.Write(String.Join("|", tmpList));

    }
}
```

Appendix 7: Seniors Advocate Study SQL Constellation Rules

The SQL statements below were provided by the Vancouver Island Health Authority and the Province of British Columbia. They were adapted to the data structures created as part of this thesis. They were utilized in the analysis study of appropriate placement of seniors in residential care.

1) Light Care patients in CCRS

```
select dw_seq_id from (SELECT
        f.dw_seq_id
        ,case when d2.CPS in (0,1) and d3.ADL_HIERARCHY in (0,1) and d4.CHESS in (0,1,2)
                and d5.E4AA_WANDERING_FREQ=0 then 'Light Care Needs'
        else Null end as value
FROM  star.dbo.F_CCRS_ASSESSMENT AS f
        INNER JOIN star.dbo.D_CCRS_ASSESSMENT_FLAGS AS d1 ON
        f.CRS_ASSESSMENT_FLAGS_Dim_Key = d1.CRS_ASSESSMENT_FLAGS_Dim_Key
        INNER JOIN  star.dbo.D_Scales_Cognitive_Depression_Social_CCRS AS d2 ON
        f.Scales_Cognitive_Depression_Social_Dim_Key = d2.Scales_Cognitive_Depression_Social_Dim_Key
         INNER JOIN  star.dbo.D_Scales_ADL AS d3 on
        f.Scores_ADL_Dim_Key = d3.Scores_ADL_Dim_Key
        INNER JOIN star.dbo.D_H1a_To_H3b_CCRS as d6 on
        d6.H1a_To_H3b_Dim_Key=f.H1a_To_H3b_Dim_Key
        inner join  star.dbo.D_P1aa_P1bfa_CCRS as d8 on
        d8.P1aa_P1bfa_Dim_Key=f.P1aa_P1bfa_Dim_Key
        inner join star.dbo.D_Scales_Chess_Pain_PURS_ABS_CCRS AS d4 ON
        f.Scales_Chess_Pain_PURS_ABS_Dim_Key = d4.Scales_Chess_Pain_PURS_ABS_Dim_Key
        inner join star.dbo.D_E4ca_To_E5_CCRS as d7 on
        d7.E4ca_To_E5_Dim_Key=f.E4ca_To_E5_Dim_Key
        inner join star.dbo.D_E2_To_E4bb_CCRS as d5 on
        d5.E2_To_E4bb_Dim_Key=f.E2_To_E4bb_Dim_Key
        left outer join
 (select * from (select Disease_Group_Dim_Key,ccrs_observation_value,CCRS_OBSERVATION_FIELD from
        star.dbo.B_DISEASE_DIAGNOSIS_BRIDGE as BDIS left outer join
         star.dbo.D_Disease_Diagnosis_CCRS as ddis on
        ddis.DISEASE_DIAGNOSIS_DIM_KEY=BDIS.DISEASE_DIAGNOSIS_DIM_KEY) as source
         pivot (max(ccrs_observation_value) for CCRS_OBSERVATION_FIELD in
         ([i1a],[i1b],[i1c],[i1d],[i1e],[i1f],[i1g],[i1h] ,[i1i] ,[i1j] ,[i1k],[i1l] ,[i1m],[i1n] ,[i1o] ,[i1p] ,[i1q] ,[i1r] ,[i1s] ,[i1t] ,
        [i1u] ,[i1v] ,[i1w] ,[i1x] ,[i1y] ,[i1z],[i1aa],[i1bb],[i1cc],[i1dd],[i1ee],[i1ff],[i1gg],[i1hh],[i1ii],[i1jj],[i1kk],[i1ll],[i1
        mm],[i1nn],[i1oo],[i1pp],[i1qq],[i1rr],[i1ss],[i1tt],[i1uu])) as pivottable) as ddis on
        ddis.Disease_Group_Dim_Key=f.Disease_Group_Dim_Key
where d1.AA8_ASSESSMENT_TYPE in (1,2,5) ) as a where a.value is not null
```

2) Assisted Living Plus patients in CCRS

```
select dw_seq_id from (SELECT
        f.dw_seq_id
        ,case when d2.CPS in (0,1) and d3.ADL_LONG_FORM in (0,1,2,3,4,5,6)
```

```
        and ddis.i1ff is null and ddis.i1gg is null and ddis.i1hh is null and ddis.i1ii is null and
        E4AA_WANDERING_FREQ=0 and E4EA_RESISTS_CARE_FREQ=0 and E4DA_DISRUPTIVE_FREQ=0 and
        E4CA_PHYSICAL_ABUSE_FREQ=0 and E4BA_VERBAL_ABUSE_FREQ=0 and P1AG_OXYGEN_THERAPY=0
        then 'Assisted Living Plus' else null end as Value
FROM     star.dbo.F_CCRS_ASSESSMENT AS f
        INNER JOIN  star.dbo.D_CCRS_ASSESSMENT_FLAGS AS d1 ON
        f.CRS_ASSESSMENT_FLAGS_Dim_Key = d1.CRS_ASSESSMENT_FLAGS_Dim_Key
        INNER JOIN star.dbo.D_Scales_Cognitive_Depression_Social_CCRS AS d2 ON
        f.Scales_Cognitive_Depression_Social_Dim_Key = d2.Scales_Cognitive_Depression_Social_Dim_Key
        INNER JOIN star.dbo.D_Scales_ADL AS d3 ON
        f.Scores_ADL_Dim_Key = d3.Scores_ADL_Dim_Key
        INNER JOIN  star.dbo.D_H1a_To_H3b_CCRS as d6 on
        d6.H1a_To_H3b_Dim_Key=f.H1a_To_H3b_Dim_Key
        inner join star.dbo.D_P1aa_P1bfa_CCRS as d8 on
        d8.P1aa_P1bfa_Dim_Key=f.P1aa_P1bfa_Dim_Key
        inner join star.dbo.D_Scales_Chess_Pain_PURS_ABS_CCRS AS d4 ON
        f.Scales_Chess_Pain_PURS_ABS_Dim_Key = d4.Scales_Chess_Pain_PURS_ABS_Dim_Key
        inner join star.dbo.D_E4ca_To_E5_CCRS as d7 on
        d7.E4ca_To_E5_Dim_Key=f.E4ca_To_E5_Dim_Key
        inner join star.dbo.D_E2_To_E4bb_CCRS as d5 on
        d5.E2_To_E4bb_Dim_Key=f.E2_To_E4bb_Dim_Key
        left outer join
 (select * from (select Disease_Group_Dim_Key,ccrs_observation_value,CCRS_OBSERVATION_FIELD from
        star.dbo.B_DISEASE_DIAGNOSIS_BRIDGE as BDIS left outer join
        star.dbo.D_Disease_Diagnosis_CCRS as ddis on
        ddis.DISEASE_DIAGNOSIS_DIM_KEY=BDIS.DISEASE_DIAGNOSIS_DIM_KEY) as source
        pivot (max(ccrs_observation_value) for CCRS_OBSERVATION_FIELD in
        ([i1a],[i1b],[i1c],[i1d],[i1e],[i1f],[i1g],[i1h] ,[i1i] ,[i1j] ,[i1k],[i1l] ,[i1m],[i1n] ,[i1o] ,[i1p] ,[i1q] ,[i1r] ,[i1s] ,[i1t] ,[
        i1u] ,[i1v] ,[i1w] ,[i1x] ,[i1y] ,[i1z],[i1aa],[i1bb],[i1cc],[i1dd],[i1ee],[i1ff],[i1gg],[i1hh],[i1ii],[i1jj],[i1kk],[i1ll],[i1
        mm],[i1nn],[i1oo],[i1pp],[i1qq],[i1rr],[i1ss],[i1tt],[i1uu])) as pivottable) as ddis on
        ddis.Disease_Group_Dim_Key=f.Disease_Group_Dim_Key
        where d1.AA8_ASSESSMENT_TYPE in (1,2,5)) as a where a.value is not null
```

3) Dementia Care Needs patients in CCRS

```
select dw_seq_id from (SELECT
        f.dw_seq_id
        ,case when d2.CPS in (0,1,2,3) and d3.ADL_LONG_FORM in (0,1,2,3,4) and
        d6.H1B_BLADDER_CONTINENCE_SELF in (0,1,2,3) and
        (ddis.i1r=1 or ddis.i1v=1) and ddis.i1ff is null and ddis.i1gg is null and ddis.i1hh is null and ddis.i1ii is null
        and E4EA_RESISTS_CARE_FREQ=0 and E4DA_DISRUPTIVE_FREQ=0 and E4CA_PHYSICAL_ABUSE_FREQ=0 and
        E4BA_VERBAL_ABUSE_FREQ=0 and P1AG_OXYGEN_THERAPY=0
        then 'Dementia Care Needs' else null end as Value
FROM     star.dbo.F_CCRS_ASSESSMENT AS f
        INNER JOIN  star.dbo.D_CCRS_ASSESSMENT_FLAGS AS d1 ON
        f.CRS_ASSESSMENT_FLAGS_Dim_Key = d1.CRS_ASSESSMENT_FLAGS_Dim_Key
        INNER JOIN star.dbo.D_Scales_Cognitive_Depression_Social_CCRS AS d2 ON
         f.Scales_Cognitive_Depression_Social_Dim_Key = d2.Scales_Cognitive_Depression_Social_Dim_Key
        INNER JOIN star.dbo.D_Scales_ADL AS d3 ON
        f.Scores_ADL_Dim_Key = d3.Scores_ADL_Dim_Key
        INNER JOIN star.dbo.D_H1a_To_H3b_CCRS as d6 on
        d6.H1a_To_H3b_Dim_Key=f.H1a_To_H3b_Dim_Key
        inner join star.dbo.D_P1aa_P1bfa_CCRS as d8 on
        d8.P1aa_P1bfa_Dim_Key=f.P1aa_P1bfa_Dim_Key
        inner join star.dbo.D_Scales_Chess_Pain_PURS_ABS_CCRS AS d4 ON
         f.Scales_Chess_Pain_PURS_ABS_Dim_Key = d4.Scales_Chess_Pain_PURS_ABS_Dim_Key
        inner join star.dbo.D_E4ca_To_E5_CCRS as d7 on
```

```
d7.E4ca_To_E5_Dim_Key=f.E4ca_To_E5_Dim_Key
inner join star.dbo.D_E2_To_E4bb_CCRS as d5 on
d5.E2_To_E4bb_Dim_Key=f.E2_To_E4bb_Dim_Key
left outer join
(select * from (select Disease_Group_Dim_Key,ccrs_observation_value,CCRS_OBSERVATION_FIELD from
star.dbo.B_DISEASE_DIAGNOSIS_BRIDGE as BDIS left outer join
 star.dbo.D_Disease_Diagnosis_CCRS as ddis on
ddis.DISEASE_DIAGNOSIS_DIM_KEY=BDIS.DISEASE_DIAGNOSIS_DIM_KEY) as source
pivot (max(ccrs_observation_value) for CCRS_OBSERVATION_FIELD in
([i1a],[i1b],[i1c],[i1d],[i1e],[i1f],[i1g],[i1h] ,[i1i] ,[i1j] ,[i1k],[i1l] ,[i1m],[i1n] ,[i1o] ,[i1p] ,[i1q] ,[i1r] ,[i1s] ,[i1t] ,
        [i1u] ,[i1v] ,[i1w] ,[i1x] ,[i1y] ,[i1z],[i1aa],[i1bb],[i1cc],[i1dd],[i1ee],[i1ff],[i1gg],[i1hh],[i1ii],[i1jj],[i1k
k],[i1ll],    [i1mm],[i1nn],[i1oo],[i1pp],[i1qq],[i1rr],[i1ss],[i1tt],[i1uu])) as pivottable) as ddis on
ddis.Disease_Group_Dim_Key=f.Disease_Group_Dim_Key
where d1.AA8_ASSESSMENT_TYPE in (1,2,5)) as a where a.value is not null
```

4) Prior Home Care Assessment before Continuing Care Assessment

```
select distinct dw_seq_id as child_dw_seq_id,
        isnull((select top 1 dw_seq_id from star.dbo.F_HCRS_ASSESSMENT as fd where
        fd.patient_dim_key=fca.Patient_DIM_KEY and
        fd.Assessment_Reference_Date_Dim_Key<fca.Assessment_Date_Dim_Key order by
        fd.Assessment_Reference_Date_Dim_Key desc),-1) as parent_dw_seq_id
        from Star.dbo.F_CCRS_ASSESSMENT as fca
```

5) Prior Discharge Abstract record before Continuing Care Assessment where Home Care Assessment exists

```
select child_dw_seq_id,
        case when parent_HCRS_dw_seq_id!=-1 then parent_dw_seq_id else -1 end as parent_dw_seq_id
        from
        (  select distinct dw_seq_id as child_dw_seq_id,
        isnull((select top 1 dw_seq_id from star.dbo.F_DAD as fd
          where fd.patient_dim_key=fca.Patient_DIM_KEY and
        fd.Discharge_Date_Dim_Key<fca.Assessment_Date_Dim_Key
        order by fd.Discharge_Date_Dim_Key desc),-1) as parent_dw_seq_id
        , isnull((select top 1 dw_seq_id from star.dbo.F_HCRS_ASSESSMENT as fd where
        fd.patient_dim_key=fca.Patient_DIM_KEY and
        fd.Assessment_Reference_Date_Dim_Key<fca.Assessment_Date_Dim_Key order by
        fd.Assessment_Reference_Date_Dim_Key desc),-1) as parent_HCRS_dw_seq_id
 from Star.dbo.F_CCRS_ASSESSMENT as fca ) as a
```

6) MAPLE score from last Home Care Assessment

```
select * from
        (select dw_seq_id,
        (select top 1 dm.maple_hc_Name
        from star.dbo.F_HCRS_ASSESSMENT as fh
        inner join star.dbo.D_CHESS_MAPLE_IADL_HCRS as dm on
        dm.CHESS_MAPLE_IADL_HCRS_Dim_Key=fh.CHESS_MAPLE_IADL_HCRS_Dim_Key
        where fh.Patient_DIM_KEY=fc.Patient_DIM_KEY and
        fh.Assessment_Reference_Date_Dim_Key<fc.Assessment_Date_Dim_Key order by
        fh.Assessment_Reference_Date_Dim_Key desc ) as value
```

```
from  star.dbo.F_CCRS_ASSESSMENT as fc) as a where a.value is not null
```

7) ALC stay on last Discharge Abstract record

```
select DW_SEQ_ID, value from
        (select dw_seq_id,
        (select top 1 case when isnull([ALC_LOS_DAYS],0)>isnull([ACUTE_LOS_DAYS],0) then 'ALC Stay' else 'Acute
        Stay' end as Value
        from star.dbo.F_DAD as fh
        where fh.Patient_DIM_KEY=fc.Patient_DIM_KEY and
        fh.[Discharge_Date_Dim_Key]<fc.Assessment_Date_Dim_Key order by fh.[Discharge_Date_Dim_Key] desc
        ) as value
from  star.dbo.F_CCRS_ASSESSMENT as fc) as a where value is not null
```

Appendix 8: Ethics Approval

University of Victoria | Office of Research Services | Human Research Ethics Board
Administrative Services Building Rm B202 PO Box 1700 STN CSC Victoria BC V8W 2Y2 Canada
T 250-472-4545 | F 250-721-8960 | uvic.ca/research | ethics@uvic.ca

Certificate of Approval

PRINCIPAL INVESTIGATOR:	Robert Hart	ETHICS PROTOCOL NUMBER Minimal Risk - Chair/Vice-chair	15-326
UVic STATUS:	Master's Student	ORIGINAL APPROVAL DATE:	16-Sep-15
UVic DEPARTMENT:	HEIS	APPROVED ON:	16-Sep-15
SUPERVISOR:	Dr. Alex Kuo	APPROVAL EXPIRY DATE:	15-Sep-16

PROJECT TITLE: **Extending Dimensional Modeling through the abstraction of data relationships and development of the Associative Dimension**

RESEARCH TEAM MEMBER Alex Kuo (Supervisor, UVic)

DECLARED PROJECT FUNDING: None

CONDITIONS OF APPROVAL

This Certificate of Approval is valid for the above term provided there is no change in the protocol.

Modifications
To make any changes to the approved research procedures in your study, please submit a "Request for Modification" form. You must receive ethics approval before proceeding with your modified protocol.

Renewals
Your ethics approval must be current for the period during which you are recruiting participants or collecting data. To renew your protocol, please submit a "Request for Renewal" form before the expiry date on your certificate. You will be sent an emailed reminder prompting you to renew your protocol about six weeks before your expiry date.

Project Closures
When you have completed all data collection activities and will have no further contact with participants, please notify the Human Research Ethics Board by submitting a "Notice of Project Completion" form.

Certification

This certifies that the UVic Human Research Ethics Board has examined this research protocol and concluded that, in all respects, the proposed research meets the appropriate standards of ethics as outlined by the University of Victoria Research Regulations Involving Human Participants.

Dr. Rachael Scarth
Acting Associate Vice-President, Research

15-326 Hart, Robert

Certificate Issued On: 16-Sep-15

University of Victoria

Office of Research Services | Human Research Ethics Board
Administrative Services Building Rm B202 PO Box 1700 STN CSC Victoria BC V8W 2Y2 Canada
T 250-472-4545 | F 250-721-8960 | uvic.ca/research | ethics@uvic.ca

Certificate of Renewed Approval

PRINCIPAL INVESTIGATOR:	Robert Hart	ETHICS PROTOCOL NUMBER Minimal Risk - Chair/Vice-chair	15-326
UVic STATUS:	Master's Student	ORIGINAL APPROVAL DATE:	16-Sep-15
UVic DEPARTMENT:	HEIS	RENEWED ON:	02-Sep-16
SUPERVISOR:	Dr. Alex Kuo	APPROVAL EXPIRY DATE:	15-Sep-17

PROJECT TITLE: **Extending Dimensional Modeling through the abstraction of data relationships and development of the Associative Dimension**

RESEARCH TEAM MEMBER Alex Kuo (Supervisor, UVic)

DECLARED PROJECT FUNDING: None

CONDITIONS OF APPROVAL

This Certificate of Approval is valid for the above term provided there is no change in the protocol.

Modifications
To make any changes to the approved research procedures in your study, please submit a "Request for Modification" form. You must receive ethics approval before proceeding with your modified protocol.

Renewals
Your ethics approval must be current for the period during which you are recruiting participants or collecting data. To renew your protocol, please submit a "Request for Renewal" form before the expiry date on your certificate. You will be sent an emailed reminder prompting you to renew your protocol about six weeks before your expiry date.

Project Closures
When you have completed all data collection activities and will have no further contact with participants, please notify the Human Research Ethics Board by submitting a "Notice of Project Completion" form.

Certification

This certifies that the UVic Human Research Ethics Board has examined this research protocol and concluded that, in all respects, the proposed research meets the appropriate standards of ethics as outlined by the University of Victoria Research Regulations Involving Human Participants.

Dr. Rachael Scarth
Associate Vice-President Research Operations

15-326 Hart, Robert

Certificate Issued On: 06-Sep-16

University of Victoria
Office of Research Services | Human Research Ethics Board
Michael Williams Building Rm B202 PO Box 1700 STN CSC Victoria BC V8W 2Y2 Canada
T 250-472-4545 | F 250-721-8960 | ethics@uvic.ca | uvic.ca/research |

Certificate of Renewed Approval

PRINCIPAL INVESTIGATOR:	**Robert Hart**	**ETHICS PROTOCOL NUMBER:** Minimal Risk - Chair/Vice-chair	**15-326**
UVic STATUS:	**Master's Student**	ORIGINAL APPROVAL DATE:	**16-Sep-15**
UVic DEPARTMENT:	**HEIS**	RENEWED ON:	**24-Aug-17**
SUPERVISOR:	**Dr. Alex Kuo**	APPROVAL EXPIRY DATE:	**15-Sep-18**

PROJECT TITLE: **Extending Dimensional Modeling through the abstraction of data relationships and development of the Associative Dimension**

RESEARCH TEAM MEMBER **None**

DECLARED PROJECT FUNDING: **None**

CONDITIONS OF APPROVAL

This Certificate of Approval is valid for the above term provided there is no change in the protocol.

Modifications
To make any changes to the approved research procedures in your study, please submit a "Request for Modification" form. You must receive ethics approval before proceeding with your modified protocol.

Renewals
Your ethics approval must be current for the period during which you are recruiting participants or collecting data. To renew your protocol, please submit a "Request for Renewal" form before the expiry date on your certificate. You will be sent an emailed reminder prompting you to renew your protocol about six weeks before your expiry date.

Project Closures
When you have completed all data collection activities and will have no further contact with participants, please notify the Human Research Ethics Board by submitting a "Notice of Project Completion" form.

Certification

This certifies that the UVic Human Research Ethics Board has examined this research protocol and concluded that, in all respects, the proposed research meets the appropriate standards of ethics as outlined by the University of Victoria Research Regulations Involving Human Participants.

Dr. Rachael Scarth
Associate Vice-President Research Operations

Certificate Issued On: 24-Aug-17

15-326 Hart, Robert

References

1. Kimball, R. (2007). An architecture for data quality. *DM Review, 17*(10), 21.
2. Kimball, R. (2007). Resist the urge to start coding ; DM review welcomes ralph kimball as a new columnist. the kimball perspectives series will systematically describe classic best practices as well as new trends in technologies. *DM Review, 17*(11), 36.
3. Kimball, R. (2007). Set your boundaries. *DM Review, 17*(12), 29.
4. Kimball, R. (2008). Data wrangling. *DM Review, 18*(1), 8.
5. Kimball, R. (2008). Myth busters. *DM Review, 18*(2), 24.
6. Kimball, R. (2008). Dividing the world. *DM Review, 18*(3), 6.
7. Kimball, R. (2008). Essential steps for the integrated enterprise data warehouse, part 1. *DM Review, 18*(4), 15.
8. Kimball, R. (2008). Essential steps for the integrated enterprise data warehouse, part 2: If you plan to combine data across subject areas, these personality types will do the job. *DM Review, 18*(5), 15.
9. Kimball, R. (2008). Drill down to ask why, part 1. *DM Review, 18*(7), 6.
10. Kimball, R. (2008). Drill down to ask why, part 2. *DM Review, 18*(8), 8.
11. Kimball, R. (2008). Slowly changing dimensions. *DM Review, 18*(9), 29.
12. Kimball, R. (2008). Slowly changing dimensions, types 2 and 3. *DM Review, 18*(10), 19.
13. Kimball, R. (2008). Judge your BI tool through your dimensions. *DM Review, 18*(11), 16.
14. Kimball, R. (2008). Fact tables. *DM Review, 18*(12), 10.
15. Kimball, R. (2004). The 38 subsystems of ETL: To create a successful data warehouse, rely on best practices, not intuition United Business Media LLC
16. Becker, B. (2007). Subsystems of ETL Revisited - Kimball Group. (2007). Kimball Group. Retrieved 21 May 2017, from http://www.kimballgroup.com/2007/10/subsystems-of-etl-revisited/
17. Pavlov, I. (2013). A QoX model for ETL subsystems: Theoretical and industry perspectives. Paper presented at the 15-21. doi:10.1145/2516775.2516778
18. Kimball, R. (2009). Exploit your fact tables: Clean fact table designs have application in the front room and back room. *Information Management, 19*(1), 44.
19. Kimball, R. (2003). TCO starts with the end user: The conventional view of data warehouses total cost of ownership is myopic and wrong. (data warehouse designer) United Business Media LLC

20. Davenport R. J. (2008). ETL vs ELT White Paper | Business Intelligence | Data Warehouse. (2017). Scribd. Retrieved 23 May 2017, from https://www.scribd.com/document/91717639/ETL-vs-ELT-White-Paper
21. Sanders D. The Late Binding Data Warehouse Technical Overview -- By Dale Sanders. (2017). Health Catalyst. Retrieved 23 May 2017, from https://www.healthcatalyst.com/late-binding-data-warehouse-explained/
22. Sanders D. et al (2014). It all starts with a data warehouse, Why the Healthcare data warehouse is becoming the critical foundation platform for analytics success in the upcoming healthcare transformation environment. (2017). Healthcatalyst.com. Retrieved 23 May 2017, from https://www.healthcatalyst.com/wp-content/uploads/2014/02/Healthcare-Data-Warehouse.pdf
23. Sanders D. (2017). Late binding in Data Warehouses: Designing for Analytic Agility. Healthcatalyst.com. Retrieved 23 May 2017, from https://www.healthcatalyst.com/wp-content/uploads/2014/06/Late-Binding-in-Data-Warehouses-Designing-for-Analytic-Agility.pdf
24. Barlow S. (2014). What is the best Healthcare data warehouse Model, Comparing Enterprise Data Models, Independent Data Marts, and Late Binding Solutions. Healthcatalyst.com. Retrieved 23 May 2017, from https://www.healthcatalyst.com/wp-content/uploads/2014/07/Best-Healthcare-Data-Warehouse-Model.pdf
25. White Paper, Health Catalyst, https://www.healthcatalyst.com/
26. Kimball, R. (2008). Myth busters. *DM Review, 18*(2), 24.
27. http://www.kimballgroup.com/2008/01/dimensional-perspectives-myth-busters/
28. Thalhammer, T., Schrefl, M., & Mohania, M. (2001). Active data warehouses: Complementing OLAP with analysis rules. *Data & Knowledge Engineering, 39*(3), 241-269. doi:10.1016/S0169-023X(01)00042-8
29. Özsu, M. T., 1951, Liu, L., & SpringerLink Ebook Collection. (2009). *Encyclopedia of database systems*. New York;London;: Springer. doi:10.1007/978-0-387-39940-9
30. Bukhari, S. (2013). Real Time Data Warehouse. Arxiv.org. Retrieved 29 May 2017, from https://arxiv.org/abs/1310.5254
31. Santos, R., & Bernardino, J. (2008). Real-time data warehouse loading methodology. Proceedings of the 2008 international symposium on database engineering & applications 49-58. doi:10.1145/1451940.1451949

32. Costa, J. P., Cecílio, J., Martins, P., & Furtado, P. (2012). Overcoming the scalability limitations of parallel star schema data warehouses. (pp. 473-486). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-33078-0_34
33. Song, I., Khare, R., & Dai, B. (2007). SAMSTAR: A semi-automated lexical method for generating star schemas from an entity-relationship diagram. Paper presented at the 9-16. doi:10.1145/1317331.1317334
34. Kimball, R., Ross, M. Dimensional Modelling in Depth. Lecture Materials, Kimball University.
35. Kimball, R., & Ross, M. (2013). *The data warehouse toolkit: The definitive guide to dimensional modeling, 3rd edition* John Wiley & Sons.
36. Kimball, R. (2008;2011;). The data warehouse lifecycle toolkit (2nd;2; ed.) Practical Techniques for Building Data Warehouse and Business Intelligence Systems. US: John Wiley & Sons Inc
37. Kimball, R., & Caserta, J. (2004). *The data WarehouseETL toolkit : Practical techniques for extracting, cleaning, conforming, and delivering data* Wiley.
38. Blechner, M., Saripalle, R. K., & Demurjian, S. A. (2012). A proposed star schema and extraction process to enhance the collection of contextual & semantic information for clinical research data warehouses. Paper presented at the 798-805. doi:10.1109/BIBMW.2012.6470242
39. Darmont, J., & Olivier, E. (2008). Biomedical Data Warehouses. In N. Wickramasinghe, & E. Geisler (Eds.) Encyclopedia of Healthcare Information Systems (pp. 149-156). Hershey, PA: . doi:10.4018/978-1-59904-889-5.ch02
40. Murphy, S. N., Morgan, M. M., Barnett, G. O., & Chueh, H. C. (1999). Optimizing healthcare research data warehouse design through past COSTAR query analysis. Proceedings / AMIA . Annual Symposium. AMIA Symposium, , 892.
41. Abelló, A., Samos, J., & Saltor, F. (2003). Implementing operations to navigate semantic star schemas. Paper presented at the 56-62. doi:10.1145/956060.956071
42. Kimball, R. (2003). The soul of the data warehouse, part one: Drilling down; drilling down just means "show me more detail". (data warehouse designer) United Business Media LLC
43. Kimball, R. (2003). *Fact tables and dimension tables: The logical foundation of dimensional modeling. (data warehouse designer)* United Business Media LLC.
44. Kimball, R. (1998). Pipelining your surrogates. *Dbms, 11*(7), 18.
45. http://www.kimballgroup.com/1998/06/pipelining-your-surrogates/
46. Ross, M., & Kimball, R. (2003). *No detail too small: Although there's no substitute for atomic details, look into complementary consolidations* United Business Media LLC.

47. Riazati, D., & Thom, J. A. (2011). Matching star schemas. (pp. 428-438). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-23091-2_36
48. Mohania, M., Schrefl, M., Nambiar, U., & Vincent, M. (2009). Active and real-time data warehousing, Encyclopedia of Database Systems. 2009:21-26
49. Kimball, R., Ross, M., Thornthwaite, W., Mundy, J., & Becker, B. (2016). The kimball group reader: Relentlessly practical tools for data warehousing and business intelligence remastered collection (Second ed.). US: John Wiley & Sons Inc
50. Olson, J. E., & Books24x7, I. (2003). Data quality: The accuracy dimension (1st ed.). US: Morgan Kaufmann Publishers Inc
51. Maydanchik, A., & Books24x7, I. (2007). *Data quality assessment* (1st ed.). Bradley Beach, N.J: Technics Publications.
52. Enterprise, N., & Kimball, R. (2017). New Directions For ETL | Articles | Big Data. Channels.theinnovationenterprise.com. Retrieved 23 May 2017, from https://channels.theinnovationenterprise.com/articles/new-directions-for-etl
53. Chisholm, M. (2007). The twin towers of BI babel. DM Review, 17(12), 24.
54. Haughey, T. (2004). Is dimensional modeling one of the great con jobs in data management history? part 1. DM Review, 14(3), 56
55. Haughey, T. (2004). Is dimensional modeling one of the great con jobs in data management history? ; part 2. *DM Review, 14*(4), 52.
56. Canadian Institute for Health Information (2017). Cihi.ca. NACRS Data Elements 2015 – 2016 Retrieved 29 May 2017, from https://www.cihi.ca/sites/default/files/nacrs_data_element_table_09_en_0.pdf
57. National Ambulatory Care Reporting System Metadata (NACRS) | CIHI. (2017). Cihi.ca. Retrieved 29 May 2017, from https://www.cihi.ca/en/national-ambulatory-care-reporting-system-metadata
58. Canadian Institute for Health Information (2017). Secure.cihi.ca. Canadian Coding Standards for Version 2015 ICD-10-CA and CCI Retrieved 29 May 2017, from https://secure.cihi.ca/free_products/Coding
59. Canadian Institute for Health Information (2009) International statistical classification of diseases and related health problems, 10th revision, [canada] (2009). . Ottawa, Ont: Canadian Institute for Health Information

60. Canadian Institute for Health Information (2017). Secure.cihi.ca. Home Care Reporting System Data Submission Specifications Manual 2017-2018 Retrieved 29 May 2017, from https://secure.cihi.ca/free_products/HCRS-Data-Submission-Specs-2017-2018-EN.pdf
61. Canadian Institute for Health Information (2017). Secure.cihi.ca. Home Care Reporting System RAI-HC Output Specifications, 2017-2018 Retrieved 29 May 2017, from https://secure.cihi.ca/free_products/HCRS-RAI-HC-Output-Specifications-2017-2018-EN.pdf
62. Canadian Institute for Health Information Continuing Care Reporting System Specifications Manual 2011-2012. ISBN 978-1-55465-860-2
63. Canadian Institute for Health Information, InterRai (2017). Secure.cihi.ca Continuing Care Reporting System RAI-MDS 2.0 Output Specifications, 2017-2018. Retrieved 29 May 2017. From https://secure.cihi.ca/free_product/CCRS-RAI-MDS-OutputSpecsManual-2017-2018-EN.pdf
64. Discharge Abstract Database Metadata (DAD) | CIHI. (2017). Cihi.ca. Retrieved 29 May 2017, from https://www.cihi.ca/en/discharge-abstract-database-metadata
65. Canadian Institute for Health Information (2017). CIHI.ca. DAD Data Elements 2015–2016. Retrieved 29 May 2017, from http://www.cihi.ca/CIHI-ext-portal/pdf/internet/DAD_DATA_ELEMENTS_2015_2016_EN
66. Simsion G., Witt G. (2004). Data Modelling Essentials. Morgan Kaufman, Dec 3, 2004
67. Ross, M., Thornthwaite W. The Data Warehouse and Business Intelligence lifecycle in depth. Lecture Materials, Kimball University. (2017). Kimballgroup.com. Description Retrieved 29 May 2017, from http://www.kimballgroup.com/wp-content/uploads/2012/08/Kimball-University-Data-Warehouse-Business-Intelligence-Lifecycle-in-Depth-Course-Description.pdf
68. Kimball, R., Becker, B. Extract Transform and Load Architecture in depth. Lecture Materials, Kimball University.
69. Knoblock, C. A., & Szekely, P. (2015). Exploiting semantics for big data integration. *AI Magazine, 36*(1), 25.
70. Bizer, C., Heath, T., & Berners-Lee, T. (2011). Linked Data: The Story so Far. In A. Sheth (Ed.), Semantic Services, Interoperability and Web Applications: Emerging Concepts (pp. 205-227). Hershey, PA: IGI Global. doi:10.4018/978-1-60960-593-3.ch008
71. Report – Seniors' Housing in B.C.: Affordable, Appropriate, Available – Seniors Advocate. (2015). Seniorsadvocatebc.ca. Retrieved 23 May 2017, from https://www.seniorsadvocatebc.ca/osa-reports/seniors-housing-in-b-c-affordable-appropriate-available/

72. Report – Placement, Drugs and Therapy…We Can Do Better – Seniors Advocate. (2015). Seniorsadvocatebc.ca. Retrieved 23 May 2017, from https://www.seniorsadvocatebc.ca/osa-reports/placement-drugs-and-therapy-we-can-do-better/
73. Hirdes, J. P. et al., An evaluation of data quality in Canada's Continuing Care Reporting System (CCRS): secondary analyses of Ontario data submitted between 1996 and 2011. BMC Medical Informatics and Decision Making 13 (2013), 27-27.
74. Hirdes, J. P., Poss, J. W., & Curtin-Telegdi, N. (2008). The method for assigning priority levels (MAPLe): A new decision-support system for allocating home care resources. BMC Medicine, 6(1), 9-9. doi:10.1186/1741-7015-6-9
75. Morris, J. N., Carpenter, I., Berg, K., & Jones, R. N. (2000). Outcome measures for use with home care clients. Canadian Journal on Aging/Revue Canadienne Du Vieillissement, 19(S2), 87-105. doi:10.1017/S071498080001391X
76. RDF - Semantic Web Standards. (2017). W3.org. Retrieved 21 May 2017, from https://www.w3.org/RDF/
77. Tkachuk R. (2005) Many-to-Many Dimensions in Analysis Services 2005. (2017). Technet.microsoft.com. Retrieved 28 May 2017, from https://technet.microsoft.com/en-us/library/ms345139(v=sql.90).aspx
78. (2017). *Sqlbi.com*. The Many to Many Revolution (Advanced Dimensional Modeling with Microsoft SQL Server Analysis Server). Retrieved 17 July 2017, from http://www.sqlbi.com/wp-content/uploads/The_Many-to-Many_Revolution_2.0.pdf
79. *Bridge Tables and IBM Cognos 8*. (2017). *Ibm.com*. Retrieved 17 July 2017, from https://www.ibm.com/developerworks/data/library/cognos/page350.html
80. Hart, R., Kuo, A. M. (2016). Meeting Health Care Research Needs in a Kimball Integrated Data Warehouse, *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Montreal, QC, 2016, pp. 697-705. doi: 10.1109/DSAA.2016.91
81. Hart, R., Kuo, A. M. (2017). *IOS Press Ebooks - Better Data Quality for Better Healthcare Research Results – A Case Study*. (2017). *Ebooks.iospress.nl*. Retrieved 18 July 2017, from http://ebooks.iospress.nl/volumearticle/46158