Title Page (including all author information)

Database Management Systems – Their Place in Nursing Informatics Education

Karen L. Courtney, RN, PhD
Assistant Professor
School of Nursing
University of Pittsburgh

Linda K. Goodwin, RN, PhD
Dean
College of Human and Health Sciences
California State University - Stanislaus

Jill Aubrecht, RN, MSN, MBA
Project Coordinator
School of Nursing
University of Pittsburgh

Corresponding Author:
Karen Courtney
School of Nursing
University of Pittsburgh
415 Victoria Building
Pittsburgh, PA 15261
Telephone: 412-624-9647
Fax: 412-383-7293
courtk@pitt.edu

- Nursing informatics requires understanding of data structures
- Including database knowledge and skills in nursing informatics curricula
- Models for teaching

Key words: Nursing Informatics; Education; Database Management Systems

Introduction

As noted in the American Nurses Association (ANA) Nursing Informatics Scope and Standards of Practice,[1] principles and concepts of database structures are an inherent component of nursing informatics practice. Many of the standards assume an understanding of data structures, management, and evaluation. While database management systems (DBMS) are not specifically mentioned within the standards, it is hard to imagine accomplishing many of the standards without the direct use of databases or information systems that are built upon databases such as electronic health records.

Even within the ANA Informatics Scope and Standards ethics standard (12) and advocacy standard (15) standards of practice, knowledge of data structures and management is implicit. For example, in order to meet standard 15 (advocacy), the nurse informaticist must understand how the data are structured as well as understand the technical and organizational protocols regarding the sharing of data. This knowledge forms the foundation for assertions regarding the degree of data security and ease of data sharing. The standards that are most likely to require database management knowledge include: 1, assessment; 3, outcomes identification; 4, planning; 5, implementation; 6, evaluation; 9, quality of practice; 12, ethics; 13, research; and 15, advocacy.

Practicing nurse informaticists may be expected to create and maintain databases, develop and revise interfaces, and develop data entry forms. Additionally, some nurse informaticists are responsible for creating complex queries and reports of quality or performance information from existing databases. Coursework in data structures, management, and evaluation along with structured experiences applying theoretical knowledge in "real-world" situations is important in preparing nurses to work competently in a variety of technology environments.

Competency in data structures and management is also recognized in ANA nursing informatics certification. For certification, nurse informaticists are tested on their ability to apply the principles of data structures, management, and evaluation. Databases and data concepts are integrated throughout the American Nurses Credentialing Center (ANCC) test plan.[2] The test plan specifically mentions skills in the logical and physical design of databases as well as underlying knowledge of data representation, data sets, classifications, and data integrity among other things.

Despite an acknowledgment of the importance of database management skills and understanding of data structures, management, and evaluation in nursing informatics practice, there remains little description in the literature of the minimal requirements let alone ideal requirements for DBMS concepts to be incorporated into nursing informatics educational programs. This paper will make recommendations for minimal content necessary to meet the standards of practice and certification for nursing informatics and review two models for including database management systems (DBMS) knowledge into nursing informatics curricula.

Minimal Requirements for Educational Preparation

Based on the ANA Scope and Standards for Nursing Informatics [1] and the ANCC nursing informatics certification test plan,[2] we suggest that at least the following concepts and experiences (Table 1) be presented in nursing informatics educational programs. In this table, we cross-match these concepts with a sample activity. Each of these concept areas is described below.

[insert Table 1 about here]

Logical Design

Logical design can be classified as the planning processes for a database. These processes typically include user needs assessment (requirements analysis), modeling of the proposed systems, and optimization of the proposed database through normalization.

As part of the requirements analysis, students should be able to demonstrate skills in collecting information from users, understanding the database project requirements, classifying and prioritizing the requirements, resolving conflicts in requirements, and validating requirements with users. Gathering the requirements for a database project is similar to other information technology projects but requires students to think about the structure and relationships of the data within the database. If the proposed database is to interact with other systems, students should discuss the challenges of interoperability between information systems.

Once students have an understanding of the requirements, they need to know how to model the required data, the information and control flow, and operational behaviors. There are many different modeling standards. We would recommend students be exposed to some of the more common models (Table 2) which they might encounter in the workplace such as Entity-Relationship diagrams[3] or Unified Modeling Language (UML) models[4] such as the Activity Diagram, the Use Case Model, and Class Diagrams. Students should be able to accurately model the database system based on their requirements analysis.

[Insert revised Table 2 about here]

Each of these steps in the logical design is iterative and requires validation with the users. Students should have the opportunity to interact with users and practice user needs assessment skills. Once students have identified and validated the requirements and the desired system responses, students need to know how to optimize the efficiency of the database through normalization. Using a step-by-step approach to normalization, students should be able to modify the database structure to at least third normal form[5], meaning that all attributes (or columns) in the record (or row) are defined by the primary key, the whole key and nothing but the key. Additionally, students would be expected to describe situations in which de-normalization might increase the efficiency of a specific database application. Special topics in this area might also include how data warehouses can be used for specialized applications.

Physical Design

The physical database design phase is the construction of the database along with development of queries, forms, and reports. This phase is often what students think of first when anticipating a database course. The logical design serves as the foundation of the physical database. By using an iterative process in the logical design, less time and fewer revisions are needed when building the physical database. In addition to creating the table structure, students should also know how to design data entry forms, queries, and reports. Students should be able to apply their knowledge of normalization to enhance the efficiency of the database and create the necessary tables and relations to support this.

One area of contention in the physical design of databases is which database management system (DBMS) students will use. Arguments can be made for many of the common systems such as Microsoft Access, MySQL, or Oracle. However, no matter which system is used, invariably some graduates will work on a different database management system than the one they used in their coursework. Important issues to consider when choosing a DBMS system for coursework are the availability to students, cost, available support resources, and the complexity of the database project being built. Depending on the DBMS used, students may also demonstrate that they can create easy data entry interfaces. Additional topics in this area would include security issues, distributed databases, data sharing, and interoperability among databases.

Standardized Terminologies

Standardized terminology is one area that is substantially different from the content acquired within a computer science database management course. The area speaks to the foundation of nursing informatics practice. Students should understand common nursing and biomedical data sets and terminologies and the relationship between these and facility-based databases such as the electronic health record. If possible, students should have a hands-on experience in querying a nursing data set created using a standardized terminology and presenting their results.

Data Warehouses

Data warehouses address the longitudinal uses of data and represent a special use of databases. Topics in this area may also include data cleaning, data aggregation, and data mining. Patient privacy issues arise even in de-identified databases used for data mining; students should be introduced to procedures for data security, as well as de-identification and protecting personally identifiable health information.

Interoperability

Interoperability stresses the complexity and challenges of the integration of information systems. Topics may include common terminologies, identification of barriers, and tools and application of relevant privacy and security regulations. If possible, students should have the opportunity to discuss these challenges with practicing nursing informatics specialists in the context of their database projects.

Model Curricula

These educational needs could be met in a number of different ways. In this paper, we describe two different methods of incorporating database management content into a nursing informatics curriculum. Both programs use a "real-world" project-centric approach for teaching this content. Alumni report enjoying the opportunity to interact with actual end users and to create projects that are meaningful to an audience outside class. Sample project assessment rubrics from both programs are available upon request.

University of Pittsburgh

In the MSN Nursing Informatics program at the University of Pittsburgh, DBMS content is covered in a dedicated nursing informatics DBMS course. The benefits of this approach are that the content and experiences are tailored for nursing informatics practice, the course can be structured without extensive programming pre-requisites, and students can have a start-to-finish experience with DBMS leading to an understanding of the entire process. The disadvantage to this approach may be a lack of qualified faculty to teach the course and a potential lack of integration of DBMS content and experience with other nursing informatics content.

At the University of Pittsburgh, the DBMS course is taught in an incremental and sequential method. Each new concept builds on the previous content and is simultaneously applied to a semester-long course project (see Table 3 for content). Throughout the course, students progress from experience with the logical design of databases (DBMS designs, requirements analysis, modeling, and normalization) to physical design of the database (building, testing, querying, and generating reports). By the end of the course, students have produced a fully-functional prototype database based on a user's request. Data warehouses are a special topic included within the DBMS course; standardized terminologies and interoperability are currently reviewed in another course in the nursing informatics program.

[insert revised Table 3 about here]

Students are assessed using periodic content quizzes (30%) and a rubric-based evaluation of the course-long database project (70%). The database course includes practice exercises and homework assignments which are reviewed but not graded. Additionally students can provide a rough draft of several of their project components (modeling, normalization, and queries) for ungraded feedback early in the semester. Peer evaluation of the final products is also an important component in the final project assessment. This allows students to see how others approached the same problem and provides an opportunity for discussion about strengths and weaknesses of design choices.

Depending on the database project for the course, students learn how to use additional software such as Microsoft Access, MySQL, Visio, and/or Pendragon Forms. Appropriate software is provided in the School's computer labs for easy access. If home access is desired by the student, software programs can be purchased through the University at a substantially discounted rate. Brief software tutorials are provided within the course and additional software training resources (such as books, websites, or classes) are listed for students within the syllabus.

Duke University

In the Nursing Informatics program at Duke University, DBMS content is taught in a modularized and threaded fashion throughout 3 of the 4 informatics specialty semesters (Table 4). Content builds each semester and emphasizes foundations and issues (semester 1), project management (semester 2), information infrastructure and patient safety (semester 3), and the final (4^{th}) semester is a hands-on synthesis and practicum experience with a qualified mentor. The advantage to this approach is that DBMS concepts are embedded within an "applied" teaching strategy that helps prepare students for the real world of informatics practice. But a disadvantage is that content becomes blurred and students sometimes have difficulty retaining important details when they are merged with so many other important content threads. Like with the standalone DBMS course, by the end of the program, students have produced a fully-functional database project based on a user request.

[insert revised Table 4 about here]

Student progress is assessed using rubrics that are designed to evaluate mastery of DBMS content and hands-on skills, ANA Nursing Informatics standards, [1] and overall course and curricular program objectives, through both individual and team project deliverables.

Other Potential Models

Students could take a DBMS course within another discipline such as computer science to meet most of these needs. However, DBMS courses in other disciplines may not adequately match the practice needs for nursing informaticists and may require additional prerequisites such as programming courses. The level of content within a computer science course for database administrators may not be an ideal match for the practice needs of nursing informatics specialists.

Curricular Recommendations

Based on our alumni experiences, the ANA Scope and Standards of Nursing Informatics [1] and ANCC nursing informatics certification test plan,[2] we recommend at least the following database content be included within the nursing informatics curricula: logical and physical database design, standard terminologies, domain modeling, data warehouses, and interoperability. Although a DBMS course within other disciplines such as computer science may cover much of this content, we believe that content inclusion from within the nursing informatics discipline is preferable when available.

References

1. American Nurses Association. *Scope and Standards of Nursing Informatics Practice*. Washington, DC: American Nurses Publishing; 2008.
2. American Nurses Credentialing Center. Informatics Nurse. 2009; http://www.nursecredentialing.org/NurseSpecialties/Informatics.aspx. Accessed 04/15/2009.
3. Garcia-Molina H, Ullman JD, Widom J. *Database Systems The Complete Book*. 2nd ed. Upper Saddle River, NJ: Pearson Prentice Hall; 2009.
4. Fowler M. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. 3rd ed. Boston, MA: Addison-Wesley Professional; 2004.
5. Kent W. A Simple Guide to Five Normal Forms in Relational Database Theory. *Communications of the ACM.* 1983;26(2):120 - 125. http://www.bkent.net/Doc/simple5.htm. Accessed 05/18/2010.

Legend

Table

Table 1: Recommended DBMS content for nursing informatics programs

Concept	Sample Activity
Logical design – Requirements analysis	Analyze a current clinical domain problem with user(s)
Logical design – Modeling	Use appropriate standard models, such as UML, with user input to model the above domain problem and potential solutions
Logical design – Normalization	Create paper-based normalized tables (to at least third normal form) with faculty guidance for the proposed database
Physical design – Create simple database	Build a simple database that reflects the logical design proposed earlier (PC based and/or PDA based)
Physical design – Build & execute queries	Create and run appropriate queries on database based on user requests (PC based)
Physical design – Design report	Design and present appropriate reports from database based on user needs (PC based)
Standard Terminologies	Identify appropriate standard terminologies for use in the proposed database
Data Warehouses	Discuss with expert guest speaker(s) the design, development, maintenance of existing clinical data warehouse(s)
Interoperability	Discuss with expert guest speaker(s) the design, development, and management of interoperability projects

Table

Table 2: Overview of Common Database Models

Model	Description
Entity-Relationship Diagrams	A graphical representation of the relationships between important concepts (these will later become tables in the database). Each entity can be described by attributes or properties. For example: A student (entity) registers (relationship) for classes (entity). The student entity may contain the attributes of last name, first name, and student identification number. The class entity contains attributes such as class identification number, subject, and class title. These diagrams are useful for capturing how tables will be linked to each other in the database being built.
UML: Activity Diagrams	A graphical representation of the dynamic behavior of a system. These are similar to flow charts. For example, a student initiates the registration process by entering in a user id and password (activity); the database then checks to see if this is a valid user (activity); a message regarding authorization to register is then returned to the student (activity). These diagrams are particularly useful for understanding workflow or how the database will control information flow between tables, processes or even outside systems.
UML: Use Case Models	A graphical and textual representation of the systems behavior from the user's perspective. For example, the Use Case model specifies the types of interactions people will have with the University Registrar's database. Students (user) may check their current schedule (one use) or may add an additional class (another use). Faculty (another type of user) may check their class roster (use). Deans (user) may assign a class to be taught by a faculty member. These models are useful for understanding how different users will interact with the developed systems and can assist with planning security controls and/or user-specific views of the database.
UML: Class Diagrams	A graphical representation of the relationships between important concepts as well as functions. Class diagrams are similar to Entity-Relationship diagrams in that both describe the relationships between tables and include a list of attributes for each entity. The main difference between the two is the class diagrams also include methods or the types of functions that the class can perform. For example, a student (class) registers (association) for courses (class). The student class may contain the attributes of last name, first name and student identification number. The course class contains attributes such as course identification number, subject, and course title. The methods listed for the student class might be to add a course (method) or drop a course (method). These diagrams are useful for capturing how tables will be linked to each other in the database being built.

Table

Table 3: University of Pittsburgh DBMS Content

Week	Topic
1	• Introduction to course software and resources • Introduction to course project • Introduction to databases • History of database design • Flat file vs. Relational databases • Relational vs. Object Oriented Databases • Data dictionaries
2	• Requirements Analysis/ Functional Specifications • Introduction to Access • Introduction to Palm Emulator
3	• Data modeling – Entity-Relationship Diagrams (E-R) • "Ask the Client" (45 minutes)
4	• Data modeling – Unified Modeling Language (UML) • "Ask the Client" (45 minutes)
5	• Normalization • Joins, etc.
6	• Introduction to Pendragon Forms
7	• Advanced Pendragon Features • Algorithms & Scripting • Access – Queries & Reports
8	• Assisted Project Development
9	• Assisted Project Development
10	• Assisted Project Development
11	• Independent Project Development
12	• Introduction to SQL • Data warehouses and data mining
13	• Final Project Presentations

Table 4. Duke University's informatics curriculum model for DBMS content (1996 – 2008)

Concept	1st semester	2nd semester	3rd semester	4th semester
Basic database concepts	Field, record, structure, query, report	User interface	Usability	Final semester is a hands-on practicum experience that synthesizes previous knowledge with new learning
Basic clinical information system concepts	Clinical database applications, informatics standards; health information exchange	Interaction of people, technology, systems	Interfaces	
Project management and systems lifecycles	Systems thinking Teamwork	Socio-political, ethical, legal, and economic considerations	Structures, systems, processes, QI, patient safety	
Logical design – Requirements analysis	Fundamentals of analysis	Systems analysis	Requirements analysis (document)	
Logical design – Modeling	The value of clinical expertise in health IT	The impact of clinical expertise in IT tool development	Domain modeling - UML diagrams	
Logical design – Normalization			Build PC database	
Physical design – Create simple database		Build PDA forms for clinical data collection	Build PC database	
Physical design – Build & execute queries		Define underlying MS Access structures	Build queries	
Physical Design – Design report		Build simple reports	Build complex reports	
Nursing data sets	ANA recognized vocabularies	Tower of Babel exercise	Define dataset for project	
Data warehouses	Data warehouse defined	Tower of Babel exercise	Data mining lecture	
Relevant theories	Change Cybernetics and Systems	Diffusion of Innovation Information Processing	Participative Design Usability	Chaos/Complexity Org Theories

***Copyright Transfer Form**

This piece of the submission is being sent via mail.