

An Effective Methodology for an Upper-level Fundamentals of Database Systems Course

Charles L. McDonald, Jr., Ph.D.
Texas A & M University – Texarkana
Charles.McDonald@tamut.edu

Theresa McDonald, Ph.D.
Texarkana College
tmcdonald@texarkanacollege.edu

Abstract

Most college graduates have a vague concept of what a database is and very few would be able to design a database. This may be because normalization, the set of rules needed to create effective databases, is not effectively addressed in most undergraduate database courses. This at least correlates with the lack of applying correct methodologies in database design as being a major cause of failure in the development of database systems (Connolly & Begg).

Introduction

For about 20 years, this author has offered a junior-level fundamentals of database systems course to support Management Information Systems (MIS) majors. This course introduces fundamental concepts necessary for the design, implementation, and use of database systems to support today's information-based workplace. Specifically, the rules of normalization, query processing, structured query language (SQL), keys, relationships, report designs, and performance issues are addressed.

Organizations report that the lack of applying correct methodologies in database design is a major cause of failure in the development of database systems (Connolly & Begg). Apparently, most students graduate without an accurate concept of what a database is nor how it can be utilized to benefit an organization. Most students that have completed an undergraduate database course have not developed the skills necessary to convert a set of unnormalized data into third normal form (3NF). Only a token of graduates have the knowledge and technical skills to develop or guide the development of a database to meet the needs of an organization.

Background

Normalization is a bottom-up process based on the relationships between attributes to establish a set of tables that efficiently support the data requirements of an organization, and this process can be used as an effective technique to validate database schemas (Connolly & Begg).

Normalization is usually the most challenging concept for students to grasp in an undergraduate database class.

The relational database model has become the dominate data-processing methodology. SQL, a data manipulation language, was initially conceptualized by IBM to support this model (Connolly & Begg). “SQL is the first, and so far, the only standard database language to gain wide acceptance” (Connolly & Begg, p. 137). SQL is frequently referred to as a fourth generation or non-procedural language that provides users a high-level view to query “what they want” without building “how to get it” procedural coding. Most students enjoy developing queries with a menu-driven database management system (DBMS) using existing databases. Students perceive query development as “game-like” with immediate feedback concerning errors or instant gratification for successful efforts.

Undergraduate database courses should present two significant topic areas for students, rules of data normalization and development of queries via a structured query language (SQL). There is some debate among educators as to the order that these topics should be addressed in an undergraduate database course; although, most undergraduate database educators follow the selected textbook’s order of topics. The course’s content sequencing generally follows one of three scenarios.

Approaches

In the normalization-first approach, terms, keys, and rules of data normalization are addressed before using SQL via a tool (e.g., Microsoft Access) to perform queries and develop reports. In this approach, students need multiple examples and exercises as the process to move unnormalized data through first, second, and third normal forms presents a logically challenging set of tasks for most students (Vician). Although about one-half of a semester is needed for students to master data normalization techniques, these students perform well when addressing SQL queries and reports in the latter weeks of the semester. This approach is most effective for the more technically-oriented students.

In the SQL-first approach, the relational model is usually selected for study. Terms and relationships are quickly addressed before moving students into performing queries with existing databases. This group of students may be better prepared to understand normalization concepts after a hands-on SQL experience; although, some students prefer to utilize a DBMS with existing databases and show limited interest in reverting to database design concepts. This approach works well for nontechnical students that prefer a higher-level view of database concepts. Although this methodology is popular, it frequently enables the rules of normalization to be an elective topic near the end of the semester.

In the third approach, students are presented terms, keys, SQL (using a DBMS), and rules of data normalization in an integrated fashion. In this approach, students may find it easier to build their knowledge by combining hands-on SQL skills with modeling. A possible disadvantage to

this approach is that it may be difficult for students to separate the logical and physical database designs. In addition, these students may find it difficult to develop a database without a DBMS. “This learning approach is akin to the prototyping systems development methodology.” (Vician, p. 152). Some professors feel that this is the most difficult approach to teach.

After reviewing syllabi from several institutions, it appears that many undergraduate database courses emphasize proficiency in the operations of a DBMS utilizing existing databases; whereas, database design via the rules of data normalization is frequently not a listed lecture topic. Although this scenario would make the course easier to teach, produce higher student grades, and, perhaps, higher instructor evaluations, these students may be less attractive when applying to an organization interested in designing a database.

Proven Course Methodology

As commercial software developers with 20 years of experience teaching undergraduate database courses, these authors have developed a course methodology that has proven to be successful for our students. Note that student involvement during lectures is crucial for success.

Lecture topics

Discuss file processing methodology: terms, advantages, and limitations. Provide scenarios that reveal limitations of file cabinet processing.

Discuss database concepts. Address terms, requirements, components, functions, advantages, and disadvantages.

Provide a printout of a two-table database example to identify and discuss tables, attributes, tuples, domains, and keys to include superkeys, candidate keys, composite keys, primary keys, and foreign keys.

Discuss DBMS components. Include schemas and sub-schemas, database instance, data definition language (DDL), data manipulation language (DML), and data dictionary

Use a top-down approach to develop (on the whiteboard) a rudimentary database structure to support sales. Be flexible to encourage student involvement suggesting attributes, identifying keys, and establishing relationships. Utilize student input to design customer, salesperson, and inventory tables. Usually, students realize that a transaction table is needed to establish relationships between these tables. Students usually agree that these four tables make a database as shown in Table 1. After some example transactions, students realize that the design would not allow a customer to purchase multiple items in one transaction. Explain that a second transaction table is needed to track multiple inventory purchases on one transaction as demonstrated in Table 2. Add a vendor table with a requirement that part numbers identify vendors as provided in Table 3. Discuss concepts of nulls, entity integrity, and referential integrity; although, these terms become more meaningful after a hands-on DBMS experience. The completed example would appear similar to the structure depicted in Table 3 with primary keys in red and foreign keys in green.

Table 1

Customer	Salesperson	Inventory	Transact
Cust ID	Sales ID	Part ID	Transact Num
Cust Name	Sales Name	Part Description	Cust ID
Cust Address	Sales Phone	Qty on Hand	Sales ID
Cust Phone			Part ID
			Qty Sold
			Date of Sale

Table 2

Customer	Salesperson	Inventory	Transact	Transact2
Cust ID	Sales ID	Part ID	Transact Num	Transact Num
Cust Name	Sales Name	Part Description	Cust ID	Part ID
Cust Address	Sales Phone	Qty on Hand	Sales ID	Qty Sold
Cust Phone			Date of Sale	

Table 3

Customer	Salesperson	Vendor	Inventory	Transact	Transact2
Cust ID	Sales ID	Vend ID	Part ID	Transact Num	Transact Num
Cust Name	Sales Name	Vend Name	Part Description	Cust ID	Part ID
Cust Address	Sales Phone	Vend Phone	Qty on Hand	Sales ID	Qty Sold
Cust Phone			Vend ID	Date of Sale	

At this point, students enter a study of the rules of data normalization. Be patient with students as normalization combines new terms, complex conceptualizations, and logical relationships to manipulate unnormalized data into a database. Historically, it takes about four 3-hour classes for students to grasp the necessary steps to normalize data. Numerous examples and walk-through exercises are crucial for students to develop proficiency.

Explain what normalization is and why it is needed. Walk-through a simple bottom-up approach using data similar to the sales structure that created the database in Table 1. Include data that represents about six transactions with at least two transactions depicting multiple items purchased. Walk students through the following steps.

1. Identify the repeating groups.
2. Display 1NF after removing repeating groups.
3. Identify the primary key.
4. Identify the functional dependencies.
5. Display 2NF based on functional dependencies.
6. Identify primary keys for clarity.
7. Identify any transitive dependencies.

8. Display 3NF after removing transitive dependencies and building additional tables.
9. Identify primary and foreign keys.
10. Ask students, "Is this a database?"
11. Have students walk through adding a transaction to purchase multiple items. Students usually discover that the existing structure limits customers to one purchase for each transaction.
12. To allow multiple items to be purchased using one transaction number, a second transaction table is needed.
13. Display the database with changes and walk-through a multi-item purchase to verify the design.

Over the next several class meetings, students will need to work through a series of examples before they can successfully transform small sets of unnormalized data into 3NF.

By mid semester, students are introduced to Microsoft Access. Access is an excellent DBMS teaching tool for importing data, building tables, identifying data types, assigning keys, establishing relationships, learning SQL, performing queries, and creating reports. Although Access offers a multitude of other features, they are outside the scope for this course.

After a walkthrough orientation of selected menu items in Access, students receive an assignment to import a delimited text file into a table. The file contains a set of unnormalized data. Students should copy and paste as needed to create tables in 3NF, assign primary keys, identify relationships, and establish referential integrity. Students continue by performing a series of simple queries (SQL code and expected output provided on the first two) to verify that their results match the expected outputs. Additional assignments of increasingly complex SQL query requests and reports complete the semester.

Conclusion

Certainly, each of these teaching methodologies; building queries using SQL before studying the rules of normalization, studying the rules of normalization before building queries using SQL, and integrating SQL and normalization have proven to be successful for the faculty and students. As discussed, our chosen course methodology was refined through years of preparing students to present perceived value to business and industry in the area of database development.

References

Connolly, Thomas M. and Carolyn E. Begg. *Database Systems A Practical Approach to Design, Implementation, and Management* (5th ed.). Addison Wesley, 2010.

Vician, Chelley, Monica Garfield, Jeffrey A. Hoffer, Mary Prescott, Bruce Rollier, Diane M. Strong and Kevin L. Elder. "The AMCIS 2003 Panels on IS Education-II: The Chicken and the

Egg Debate: Positioning Database Content in the Information Systems Curriculum.”
Communications of the Association for Information Systems, Vol. 2004, No. 14, 2004, p. 147-157.