

Integrating Relational Database Technology into the Construction Management Curriculum

Charles S Duvel and Klaus Schmidt
Illinois State University
Normal, Illinois

Students in construction management programs often lack a solid knowledge about databases and database design. In addition to reasons why students should learn about database applications, the authors provide an introduction to relational database nomenclature and examples on how database technology could be incorporated into construction management course work. A road map through some of the issues related to integrating database applications and design in construction management will be provided including basic concepts of Entity Relationship Modeling (ERM) and Data Flow Modeling (DFM).

Key words: Relational Databases, Database design, Teamwork

Introduction

Understanding the application and design of databases is important for the modern construction manager. The current curricula of construction management (CM) programs provide little instruction about databases. During the past year, the researchers reviewed the college catalogs of 38 different accredited four-year programs in construction management. Information about construction management programs was obtained from the college catalogs of programs accredited by the American Council for Construction Education (ACCE) that were posted on the World Wide Web. The ACCE publishes an annual list of all accredited two-year and four-year construction management programs in the United States. For purposes of this investigation, only accredited four-year programs that posted their college catalogs to the World Wide Web were examined. There are 47 accredited programs out of the approximately 100 four-year colleges that offer construction management and related programs in the United States. Programs listed as being in accreditation candidacy were not included in the survey. A total of 23 out of 38 of the curricula reviewed did not contain any courses that include database management. The researchers concluded that databases are not taught as a regular part of these programs. This causes a hardship to CM students because they miss out on the significant benefits advanced database software applications provide.

The benefits of understanding modern database management features include extending the use of scheduling, document management and estimating software programs commonly used in industry. Learning about database design and how to manipulate databases is becoming an integral part of construction companies' overall management needs (Construction Financial Management Association, 2000). However, current programs do not capture these benefits because they fail to teach database technology and how they relate to company management.

Why Study Databases

First, most software programs used in CM include databases as an integral part of their system. Examples include scheduling software (e.g., Primavera Project Planner) and document management software (e.g., Primavera Expedition). Estimating software such as Timberline Precision Estimating, HeavyBid, and WinEst also rely on database technology.

Second, the significant increase in construction volume coupled with sophisticated building products has forced construction firms to embrace new technologies for estimating, scheduling, and project management (Construction Financial Management Association, 2000). Construction firms that do not have specialty programs for estimating or scheduling often rely on generic relational database management system software such as Microsoft Access or FoxPro or try to use the limited database capabilities of electronic spreadsheet programs. These companies are seeking CM graduates that have appropriate software knowledge.

Third, students apply management principles regarding quality planning and analysis they have learned in other classes by using a problem-centered, activity-based approach to solve real world problems. Students learn the importance and roles that business rules play in the company organization by documenting work processes.

Fourth, CM students sharpen their analytical skills by applying data modeling techniques to relational database system design and implementation. Students learn the importance of careful, advance planning and project analysis by considering factors such as: What are the attributes of the data to be collected and stored; how is it related; and how will it be entered, verified, and reported?

One way to communicate these factors to students is to introduce relational database concepts into multiple courses across the construction management curriculum. The following section will introduce relational database terminology to provide a baseline of knowledge on which to build course modules.

Relational Database Terminology

A database is a collection of tables, queries, forms, and reports. The fundamental element of a relational database is a table, or *data file*, similar to a spreadsheet (Thierauf, 1989). A *table* is composed of *columns* and *rows*. The intersection of a column and a row is called a database *field*. Multiple horizontal fields make up a record as shown in Figure 1. Records in one table have similar characteristics or properties (e.g., automobile has color, Vehicle Identification Number, model year, and assembly date).

tblEmployees : Table					
	EmployeeID	Employee Name	Address	City	State
▶	A100	John Smith	200 South Blvd	Chicago	IL
	A101	Mary Miller	100 East Blvd	Chicago	IL
	A102	Jeff Johnson	302 South Drive	South Bend	IN
	A103	Linda Jefferson	340 East Blvd	South Bend	IN
	A104	Jason Kline	200 South Park	Chicago	IL
*					

Figure 1: Sample database table for employees

There are various types of database systems of which the relational database has become the most popular. A relational database is made up of a number of tables that contain raw data and share a commonality. For example, as shown in Figure 2, a construction firm's relational database could consist of two tables. The Employee Information Table (Table 1) contains employee information such as Social Security number, Address, Name, Phone Number etc. The Project Information Table (Table 2) contains records about the company's projects such as Project Identification number, Location, Type of Work, Size of Job, and Employees Assigned (by Social Security number). Both tables contain a field called Social Security number. The Social Security number in Table 1 is a unique *identifier* for each employee. This identifier, called the *primary key*, is used to link tables within the same database or to tables within other databases (Hernandez, 1997). The Social Security numbers in Table 2 are used to identify which projects an employee is assigned to. Since employees can be assigned to multiple projects, the Social Security number cannot serve as the primary key for this table. A meaningful primary key for Table 2 would be the Project Identification Number since each project will have a unique ID number.

There are several ways of linking tables in a database management system, depending upon the type of relationship that has been established. In the example above, the Social Security number field of Table 1 can be linked to the Social Security number field in Table 2. Since the Social Security number field in Table 1 is a primary key, but can occur multiple times in table 2, we speak of a *one-to-many* relationship.

Another common type of relationship is the *one-to-one* relationship where a primary key field in one table is related to a primary key field in a second table. An example of this type of relationship would be a table with Social Security number and a second table containing Drivers License Numbers associated with Social Security numbers. Since only one Social Security number can be linked with one Drivers License number, we speak of a one-to-one relationship.

The third type of relationship is known as a *many-to-many* relationship. In this type of relationship one or more of one type of data can be associated with one or more of another data types. An example for a many-to-many relationship is a firm's database that contains 3 tables. Table 1 contains employee information using the Social Security number as primary key. Table 2 contains the firm's Construction equipment with the Vehicle Identification number (VIN) as the primary key. Table 1 and 2 do not contain a common field. Table 3 is the Work Orders Table (e.g., oil change). As construction equipment requires maintenance, work order information about the employee (Social Security number) and the vehicle (VIN) is entered into the Work

Orders Table. Therefore the orders table must be linked to tables 1 and 2. Both Tables 1 and 2 will have a one-to-many relationship with the Orders Table, creating a many-to-many relationship between the Employees and Equipment Tables.

The diagram shows two tables. The top table, 'tblEmployees : Table', has columns: EmployeeID, Employee Name, Address, City, and State. The bottom table, 'tblProjects : Table', has columns: ProjectID, EmployeeID Assignment, Project Start Date, and Anticipated End Date. An arrow labeled 'Primary Key' points from the 'EmployeeID' column in the first table to the 'EmployeeID Assignment' column in the second table.

tblEmployees : Table					
	EmployeeID	Employee Name	Address	City	State
▶	A100	John Smith	200 South Blvd	Chicago	IL
	A101	Mary Miller	100 East Blvd	Chicago	IL
	A102	Jeff Johnson	302 South Drive	South Bend	IN
	A103	Linda Jefferson	340 East Blvd	South Bend	IN
	A104	Jason Kline	200 South Park	Chicago	IL
*					

tblProjects : Table				
	ProjectID	EmployeeID Assignment	Project Start Date	Anticipated End Date
▶	2000-1-12	A100	1/15/2000	4/15/2000
	2000-2-18	A100	2/15/2000	4/25/2000
	2000-2-19	A101	2/20/2000	6/30/2000
	2000-3-10	A101	3/1/2000	6/15/2000
	2000-6-01	A102	6/1/2000	9/15/2000
	2000-9-04	A102	9/20/2000	1/15/2001
	2001-10-10	A103	10/1/2001	1/15/2002
	2001-1-12	A103	1/1/2001	6/30/2001
	2001-12-1	A104	12/1/2001	3/1/2002
	2001-5-14	A103	5/15/2001	9/20/2001
	2001-9-10	A100	9/1/2001	3/1/2002
	2002-1-1	A104	1/1/2002	1/30/2002

Figure 2: Sample database Table for Employees

Queries

Once the tables have been established, the raw data stored in those tables can be selected and manipulated using a wide array of queries. The most common type of query is the Select query (Figure 3). The Select query filters information from one or more linked tables to produce a particular set of records. Depending on the particular Relational Database Management Systems (RDBMS) program used, queries can produce reports, create graphs, choose records, sort records or do calculations (Prague and Irwin, 1997). In addition to select queries there are Action Queries and Cross-Tab queries.

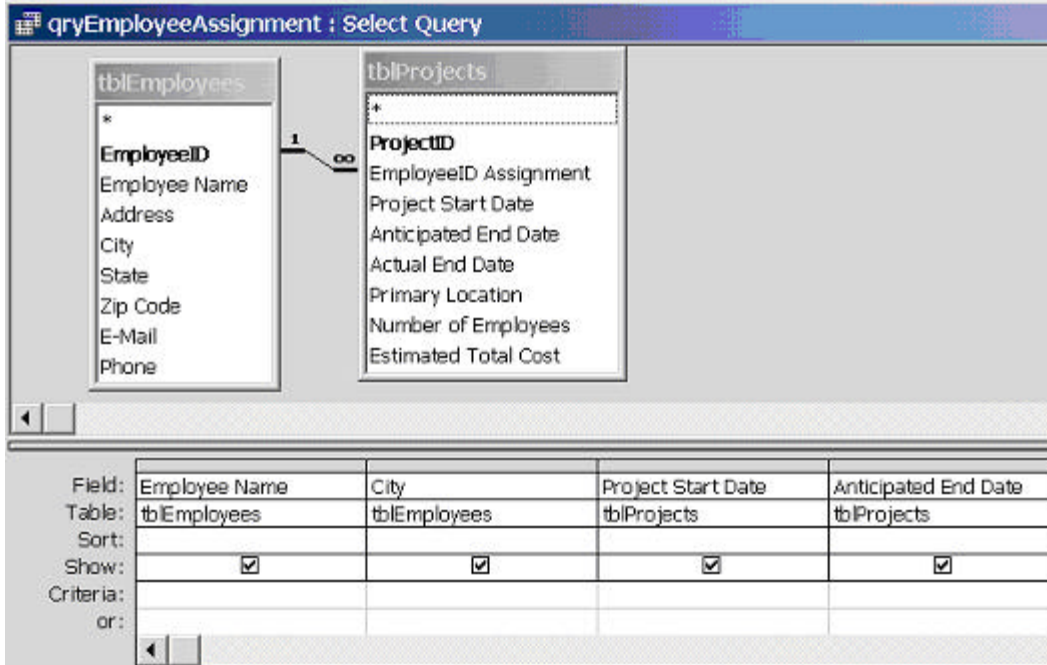


Figure 3: Query Design View for a Select Query

Action Queries are not used to simply display a particular record set like is the case with Select queries. Rather, they are used to manipulate data in an existing table. For example, if you would like to change the wage rate in your employees table, you can use an Update query to update the wages in that table. Other types of action queries include Append queries that append records from one table to the records of another, and the Delete query where records in a table are deleted according to specific criteria.

If you want to find information about how many deliveries a particular employee of your company made to a particular jobsite, a cross-tab query would be used. The cross-tab query calculates (e.g., sums, multiplies) information at the intersection of a row and a column. In the above example, jobsites would be displayed as column headings while the employees would be displayed as row headings. At the intersection of each row and column, the volume of deliveries person A made to projects A, B, C, and D would appear.

Forms and Reports

For a nice display of whatever has been queried from the tables, a variety of reporting tools are available (Figure 4). Reports themselves can be customized to the extent that calculations, summaries, and sort features can be included. Print features of the reports can vary from printer per customer, per country, in alphabetical order or by sales volume.

<i>Turnover by Employee</i>					
<i>Employee Name</i>	<i>Start Date</i>	<i>Anticipated End</i>	<i>Actual End</i>	<i>Location</i>	<i>Total Cost</i>
<i>Jason Kline</i>					
	1/1/2002	1/30/2002	2/15/2002	Detroit	\$12,000.00
	12/1/2001	3/1/2002	3/15/2002	Chicago	\$34,000.00
<i>Summary for 'Employee Name' = Jason Kline (2 detail records)</i>					
Sum					\$46,000.00
<i>Jeff Johnson</i>					
	9/29/2000	1/15/2001	1/15/2001	Chicago	\$48,000.00
	9/1/2000	9/15/2000	9/15/2000	Chicago	\$24,000.00
<i>Summary for 'Employee Name' = Jeff Johnson (2 detail records)</i>					
Sum					\$72,000.00
<i>John Smith</i>					
	9/1/2001	3/1/2002	3/12/2002	Chicago	\$34,000.00
	2/15/2000	4/25/2000	4/30/2000	South Bend	\$20,000.00
	1/15/2000	4/15/2000	4/15/2000	Chicago	\$40,000.00
<i>Summary for 'Employee Name' = John Smith (3 detail records)</i>					
Sum					\$94,000.00

Figure 4: Part of a Report displaying Turnover by Employee

Forms are the important front-end of the database (Figure 5). Forms commonly are equipped with Filtering features to display selected records or groups of records in a particular format. In addition, data entry clerks without any knowledge of database design may use forms.

Figure 5: Data Entry Form for frmEmployees

Database Exercises in Construction Management Courses

Students learn most effectively when they see immediate application of their classroom experience to real world problems. Integrating database applications into the curriculum provides these experiences. Students must connect their classroom experience with database concepts in order to understand and meaningfully use database software (Hill, 1999). For example, a course

in estimating is a natural application for the use of databases. In an introductory estimating course Sophomore and Junior students set up their own Microsoft Access database tables and store employee and information (e.g., employee name, address, phone) and project data for the purposes of bidding. This type of exercise provides an excellent opportunity for teaching database concepts. Tables can be used to teach students the powerful functionality of databases. The students can develop queries based on relationships (e.g., Employee ID) and link the tables.

Project Management courses is another area that provides multiple opportunities for teaching and integrating relational database fundamentals into the curriculum. General jobsite records like requests for information, labor administration, and purchasing are areas that can be more effectively managed using database technology (Flowers, 1996). In fact, many of the more sophisticated Project Management specific software programs available on the market (e.g., Prolog, Expedition) are specialized database programs. These types of programs have been created with pre-designed forms for easier data entry and analysis. Students learn to create forms for data entry based on paper forms they are likely to see in their professional career. For example, the request for payment, requisition forms, and employee hire data.

Scheduling courses also provide opportunities for teaching database creation and usage. Students can learn about the different types of existing database models. Most scheduling programs, like Primavera Project Planner or Microsoft Project, allow the scheduler to develop composite construction crews, set labor rates and other operational data within an internal database. Particular scheduling software programs help the instructor demonstrate how databases are used, and how they are structured. In addition, the instructor can emphasize the role databases play in the implementation and administration of scheduling programs. Students learn how to use relational database programs for labor allocation and control, labor cost data, embedding objects and to produce customized reports based upon queried data.

Instructional Methods and Strategies

In an introductory class ‘Cost Estimating and Project Planning’ students are introduced to database technology by lecture. These lectures are followed by several detailed step-by-step homework assignments (see Appendix A). Each homework assignment is designed to get the students familiar with the software package (e.g., Microsoft Access) via “action learning” (Fryer, 1997). Many students, particularly those with a limited experience in computing, depend on detailed instructions about the software before they can begin to see its potential applications. Students with low confidence in their computing skills are initially hesitant to try out computer applications on their own (Lambrecht, 1999). Lambrecht (1999) encourages systematic computing instruction by providing straightforward, applied assignments. Based on these assignments, students learn how to create tables, manipulate data, run queries and export the results of a query to another program.

The real test of the student’s abilities comes from being able to apply their new skills to situations that require critical thinking (Lundgren, Lundgren, and Mundrake, 1995). Then, students can demonstrate that they have moved beyond the basic mechanics of software program tutorials and are able to solve real-world problems. Students cannot become mired in lower level

concept and skill development. To overcome the push-button mentality of the introductory tutorials, students enrolled in the more advanced 'Computerized Estimating and Scheduling' course are paired to analyze a construction firm and produce a database that can be used to track subcontractors and vendors for a construction project. Students in this class are primarily seniors and are expected to synthesize earlier learning concepts. Developing data models and creating databases encourages teamwork and collaborative learning which occurs as each student plays a different key role in a team. One student is assigned the role of database administrator and the other student assumes the role of a system user. Students that served as the database administrator in one pairing are assigned the role of program user in a second team. Design failures, problems of data integrity or project inadequacies can be determined almost immediately by the team. Student "action learning" is maximized, because immediate feedback is provided regarding their database's design strengths and weaknesses (Skinner, 1968, Fryer, 1997). Solid design of relational databases is important if the database system is to function correctly. The student teams are actively engaged, developing their own business rules, testing the integrity of the data and table schema, and ensuring the validity of the output. Information about design failures in one group is shared with other groups so that such errors might be prevented in the future.

Additional team projects have included the use of formalized database modeling techniques such as Entity Relationship Modeling (ERM) and Data Flow Modeling (DFM). These team projects help develop an understanding of the relationship between employees and how data will be handled. Simsion (1994) defines ERM as the processing of designing an appropriate set of entities and relationships to meet a business problem. Using ERM, students develop a flowchart model of the relationship between entities such as field operations staff and the accounting department within a construction firm. The ERM project requires that students ask questions about structure of the construction firm and find out exactly what activities employees, particularly project managers, are responsible for.

Data Flow Modeling, on the other hand, has the main objective to define the functional structure of an organization by means of the combined consideration of functions and data. The data flows are the interfaces between these functions. In DFM, information collected by a construction firm is graphically represented. Students studying DFM are forced to think through the business processes and seek to understand the differences between the types of data that is collected (e.g., material costs, purchase orders, labor records, financial statements or job cost reports) and the way data should be reported.

Peer evaluations of each team's finished design is used to assess the quality of the database produced. Faculty evaluates students based on (1) their performance on completing the introductory individual assignments (e.g., creation of a database and tables, filtered queries, report creation and exporting data to another software program) and (2) the collaborative effort and project performance of the team's database to track subcontractor and vendor performance.

Closing Thoughts

Relational database concepts and database design remain a missing component in the technical background of construction management students. However, each course taught in the CM core curriculum provides opportunities to include concepts of databases and their design. A firm understanding is necessary for a technically proficient construction manager to succeed in the construction industry (Paulson, 1995). Databases are the basis for many other software programs in this industry. Zhang and Espinoza (1997) observed that, “instructors should provide activities that will assist students to learn the importance of computers in their future careers.” Hands-on activities will help students overcome fear and gain a feeling of confidence when using computers and new software packages. Additionally, developing good database design requires students to draw on their prior education in business management theory and quality control. The repeated use of database software throughout a student’s course of study will reinforce instruction and ensure that students enhance skills and knowledge they have acquired in previous classes.

References

- Construction Financial Management Association (CFMA) (2000) *Computerization Survey for the construction Industry*. New York: CFMA.
- Flowers, R. (1996). *Computing for Site Managers: database techniques*. Oxford: Blackwell Science.
- Fryer, B. (1997). *The Practice of Construction Management* (3rd ed.). Oxford: Blackwell Science.
- Hernandez, M. J. (1997). *Database Design for Mere Mortals: a hands on guide to relational database design*. Reading, MA: Addison-Wesley.
- Hill, J. R., (1999). Teaching Technology: Implementing a Problem-Centered, Activity-Based Approach. *Journal of Research on Computing in Education*, 31(3), 261-279.
- Lambrecht, J. J. (1999). Teaching Technology-related skills. *Journal of Education for Business*, 74(3), 144-151.
- Lundgren, C. A., Lundgren, T. D., & Mundrake G. A. (1995). *Teaching Computer Applications*. Little Rock, AR: Delta Pi Epsilon.
- Paulson, B. C., (1995). *Computer Applications in Construction*. New York: McGraw-Hill.
- Prague, C. N., and Irwin, M. R., (1997). *Access 97 Bible*. Foster City, CA: IDG Books.
- Simsion, G. C. (1994). *Data Modeling Essentials: Analysis, Design, and Innovation*. 2nd Edition, Scottsdale, AZ.

Skinner, B. F. (1968). *The Technology of Teaching*. New York: Apelton-Century-Crofts.

Thierauf, R. J. (1989). *Group Decisions Support Systems for Effective Decision Making*. New York: Quorum.

Zhang, Y. and Espinoza, S. (1997). Affiliations of Computer Self-efficacy and attitudes with need for learning computer skills. *Journal of Educational Computing Research*, 17(4) 371-383.

Appendix A Introductory Exercise (Example)

Scenario: You are to create an Employee Allocation Database that details all the needed information about your company's employees and how they are assigned to various projects of your construction management company. Goal of this project is to produce a report that displays specifically how many projects an employee is assigned to and what the total 'volume' of the projects are for each employee.

1. Create an Access database and name it EmployeeAllocation.mdb
2. Create a first table called tblEmployees containing the following information:

Field Name	Data Type	Properties
EmployeeID	Number	Input Mask: 5 digit numbers only
EmployeeName	Text	Field Size 40
Address	Text	Field Size 60
City	Text	Lookup Wizard, add 4 cities of your choice
State	Text	Default Value 'IL'
E-Mail	Hyperlink	Required: Yes
PostalCode	Text	Input Mask (use wizard)
Phone	Text	Input Mask (use wizard)

3. Make EmployeeID the primary key field for tblEmployees.
4. Enter 5 complete records into this table. Use a wizard to create a form that helps you facilitate data entry. Your first employee should have the EmployeeID A100; the second of A101; and so on!
5. Create a table called tblProjects with the following fields:

Field Name	Data Type	Properties
ProjectID	Text	
EmployeeID	Number	5 digit numbers only
StartDate	Date/Time	Format 'Medium Date'
AnticipatedEndDate	Date/Time	Format 'Medium Date'
ActualEndDate	Date/Time	Format 'Medium Date'
Location	Text	Required: No
Number of Staff	Number	Default Value '1'
RatePerHour	Currency	Required 'Yes'
TotalPrice	Currency	Required 'Yes'

6. Make ProjectID the primary key field for tblProjects.
7. Add 10 complete records to this table. One employee can be assigned to multiple projects. Enter data as follows: ProjectID 2000-1-12, EmployeeID A100; (this means Employee A100 was assigned to Project 2000-1-12); ProjectID 2000-2-15, EmployeeID A100; ProjectID 2000-3-01, EmployeeD A101; and so forth.
8. Open the relationship window. Add both tables to the window.
9. Create a one-to-many relationship. Make sure you check the referential integrity box! Use an Outer Join Type that displays all records from your Customer table!

Scenario: You want to create a **query** to combine information about the employees and the projects they have been assigned to. This query will be based on information from both tables! You will add a **report** based on this query to display and summarize the total value of the projects by employee.

1. Create a query called qryEmployeeAssignment. Include the following fields from tblEmployees: EmployeeName, City, State. Include the following fields from tblProjects: ProjectID, StartDate, AnticipatedEndDate, ActualEndDate, TotalPrice.

2. Run the query! You should see each employee's information for as how many many projects they have been assigned to. E.g., if an employee has been assigned to two projects, you should see the information about this particular employee twice.
3. Using the report wizard, create a report based on above query in which you summarize the total value of the project. Display each project an employee is assigned to, the value of the project, and a summary of the totals for all projects.