A Problem-Solving Based Approach to Teaching Database Design

Emergent Research Forum (ERF)

Anil Singh  
University of Texas at Rio Grande Valley  
anil.singh@utrgv.edu

Vikram Bhadauria  
Texas A&M University Texarkana  
vbhadauria@tamut.edu

Anil Gurung  
Marshall University  
gurung@marshall.edu

Abstract

We use a problem-solving approach to learning where students work on a partially completed database with defects. To fix and complete the database design, students enforce entity, referential and column level constraints. After fixing the database, students develop data macros to update changes into other tables. By successfully fixing problems and creating macros, students gain confidence in working with databases. Informal feedback from students revealed evidence of learning among students. Students also learned valuable concepts in database design.

Keywords

MS Access; database integrity; pedagogy; data-macros.

Introduction

Databases instructors have two main challenges: what to cover; and how to teach databases effectively (Al-Dmour, 2010). One objective is to develop problem-solving skills that will mimic real world scenarios (Connolly & Begg, 2006). Past modules in MS Access, have focused on developing the database along with subsequent queries, forms, and reports. Many concentrate on developing abstract level concepts while others skip concepts and teach queries. Often students end up creating tables and doing queries on predefined databases. They miss the experience with fixing errors. While theoretical understanding is good, implementation level wisdom is required too. This paper fills this gap by providing an implementation level activity, which not only helps them to understand database concepts, but also fix problems (intentionally placed by the instructor). This paper, instead of expecting students to develop a database, expects students to fix and complete an incomplete and defective database. Students get the benefit of both building and fixing problems in database. This develops problem-solving skills among students. Thus, in line with papers that used problem based learning approach (Connolly & Begg, 2006; Zanzig & Tsay, 2004; Singh, Mangalaraj, & Taneja 2014; Steinberg, 2009; Philip, 2007), we develop a teaching resource involving fixing errors in a database. This emulation of real world situation gives the student’s confidence in practical aspects of database design.
Teaching Module

In this problem-solving approach to learning, students work on a partially completed database with defects, fix it, work on it further and complete the database design. To complete the assignment students enforce entity, referential and column level constraints. In doing so, they receive implementation level wisdom that is often lacking in the first time database students. During debriefing, the instructor can use the assignment to explain database principles.

Background

This teaching module emphasizes providing a challenge, give just enough information through helpful links and some directions through hints and let the students look for help on the internet/help files and find a solution. Before this assignment, cover Primary Key, Foreign Key, Data Macros, Master-Detail, Bridge Tables, Input Mask, and Normalization.

Student handout

The organization that you have just joined is developing and testing a database and asks you to fix any errors in the test database and improve it. Use the database provided to you by the instructor. Following are details about the database.

- All data in the tables is test data and hence are modifiable.
- Create new tables if required.
- Break Tables into two if required.

Enforcing entity integrity

Enforce relevant primary key constraints wherever required.

Enforcing Referential integrity

Identify all primary key-foreign key relationships and enforce referential integrity in the database. For example, each Item belongs to a Category; Employees can only be from department that exist in the DEPARTMENT table, and allow only states in STATES table in states columns in EMPLOYEE, CUSTOMER and SUPPLIER tables.

Enforcing Column level constraints

To improve the database, column level constraints need to be enforced. Enforce column level integrity through field properties, input mask, and lookup. Enforce the following rules:

1. FName and LName columns of EMPLOYEES and CUSTOMERS tables are required (Hint: Field properties).
2. HomePhone and Mobilephone columns of EMPLOYEES and CUSTOMERS number should be in the proper format: (Hint: input mask).
3. The Department column in the EMPLOYEE table should contain a drop-down menu showing all departments available.
4. The State column in the EMPLOYEE, CUSTOMER and SUPPLIER table should contain a drop-down menu showing all states available.

Creating Bridge Tables

Many suppliers can supply each item and each supplier can supply many items. In addition, each employee can be in one or many projects. Each project can be associated with one of many employees. To address these, create bridge tables and form relevant relationships.
**Order Table normalization**

ORDER table does not seem normalized and may have to be broken into two parts: ORDERMASTER and ORDERDETAIL.

**Save past addresses using Data Macros**

When employees change addresses, your organization wants to store the past addresses of employees, customers and suppliers. For this purpose whenever there is a change in the tables, the changes in columns (customerID, employeeID or supplierID), Address, City, State, and Zip can be transferred into a new history table. Also, include a time stamp to capture the time of change.

**Conclusion and Future Directions**

Students were asked to fix and enhance an incomplete database. Completing the assignment gives students more efficacy in their work with databases. We found evidence of learning from surveying the students. Students who successfully completed the assignment found that they learnt the concepts of database design. You can obtain a starter database and completed database by emailing the author at anil.singh@utrgv.edu.

**REFERENCES**


