

Normalization

Lab 12 and Take-home Exam II

Objective

The goal of this lab is to give you more practice on normalizing a database. You will analyze an existing database to determine the changes needed to convert it to Third Normal Form. You will then use SQL Server to refactor the database without losing any existing data.

Resources

The class slides, your notes, and the sample problem that we worked in class may be helpful for reference.

Turn In Instructions

For this lab you will:

- enter answers in an Angel quiz and
- make changes to a database on the server.

The Angel portion of the lab will count towards your Lab 12 grade. The database portion of the lab will count towards your Exam II grade.

Tasks

Problem Setup

The marketing manager for a computer products store has been using a spreadsheet to keep track of special sales programs. This worked fine when the store was small. However, the store has been growing and has recently added some kiosks at local malls. The spreadsheet has also grown to include lots of redundancy. The columns of the spreadsheet are as follows:

- **Store**: the name of the store or kiosk, like Main or North Grand Mall
- **Address (Addr)**: the address of the store
- **Manager (Mgr)**: the manager of the store
- **Manager Phone (MgrPh)**: the cell phone number of the manager
- **Promotion Name (PName)**: the name of the promotional plan, like January White Sale
- **Promotion Terms (PTerms)**: the terms of the promotion, like “buy a new computer, get a free soldering iron”.
- **Total Sales For Promotion (PTotal)**: the volume, in dollars, of products sold under a particular promotion at a particular store

The terms in parentheses in the list above are abbreviations that you may choose to use in your solution to the problem.

We'll treat this table as a single relation:

Promotions(Store, Addr, Mgr, MgrPh, PName, PTerms, PTotal)

The following assumptions hold:

- A1: Each store has just one manager, but one person may manage multiple stores.
- A2: Promotion names are not repeated
- A3: A promotion only has one set of terms

Algorithmically Normalize the Relation

Below are four functional dependencies for the Promotions relation. For each functional dependency answer the two associated questions on the **Angel Quiz for Lab 12**.

FD1: Store \rightarrow Address, Manager

- **Question 1:** Why is FD1 a functional dependency for the Promotions relation?
- **Question 2:** FD1 violates third normal form (3NF) for the Promotions relation. What is it about FD1 that violates 3NF?

FD2: Manager \rightarrow ManagerPh

- **Question 3:** Why is FD2 a functional dependency for the Promotions relation?
- **Question 4:** FD2 violates 3NF for the Promotions relation. What is it about FD2 that violates 3NF?

FD3: Promotion Name \rightarrow Promotion Terms

- **Question 5:** Why is FD3 a functional dependency for the Promotions relation?
- **Question 6:** FD3 violates 3NF for the Promotions relation. What is it about FD3 that violates 3NF?

FD4: Store, Promotion Name \rightarrow Total Sales For Promotion

- **Question 7:** Why is FD4 a functional dependency for the Promotions relation?
- **Question 8:** FD4 actually satisfies 3NF for the Promotions relation. Why does FD4 satisfy 3NF?

Here again are the functional dependencies for the Promotions relation. The functional dependencies that violate third normal form are indicated.

- *Violation!* FD1: Store \rightarrow Address, Manager
- *Violation!* FD2: Manager \rightarrow Manager Phone
- *Violation!* FD3: Promotion Name \rightarrow Promotion Terms
- FD4: Store, Promotion Name \rightarrow Total Sales For Promotion

Question 9: Calculate the closure $\{\text{Store}\}^+$ of FD1.

Question 10: Decompose the Promotions relation into **two** new relations based on FD1. Give the relations and attributes in Relation(Attribute1, Attribute2, ...) format. **Note:** Do not skip directly to the final solution. We want to see that you can use the algorithmic technique discussed in class so that you're prepared to deal with very large databases.

Question 11: Continue the process of decomposing the relations that still violate third normal form until all the relations are in third normal form. Give the final relations and attributes in Relation(Attribute1, Attribute2, ...) format. (Hint: Your final decomposition

should have four relations, two of which have two attributes and two of which have three attributes.)

Refactor the Actual Database

You should have a new database on the **dyknow.cs** server named **Promos<Username>**. This database has one table that matches the original Promotions relation given above. Your final task will be refactoring this database to match your new design.

Step 1:

Verify that you can access your **Promos** database. Look at the contents of the Promotions table.

Step 2:

Create new, empty tables matching your normalized relations from Question 11 above. Be sure to make the column types match those from the corresponding attributes in the existing Promotions table. Set appropriate primary and foreign key constraints on your new tables.

Step 3:

The biggest step in refactoring your database is to populate the new tables using data from the existing tables. You can do this using SQL commands like:

```
INSERT INTO dbo.Manager(Manager , ManagerPhone)
SELECT DISTINCT Manager , ManagerPhone
FROM Promotions
```

At the end of this step you'll copy and paste your SQL commands into the Angel quiz.

Note that you'll need to start with a table that has no foreign keys and work "backwards" up the foreign key constraints.

Question 12: Copy and paste into the Angel quiz your four SQL commands for populating your new tables.

Step 4:

Confirm that all the data from the original Promotions table is now present in your new tables. Delete the original Promotions table from your database.

Feedback

Please complete the anonymous feedback for Lab 12 on Angel.

Turn In Instructions

Enter your answers to the quiz on Angel. We will also check your database on the server to see that the appropriate changes were made.

Revision History

Feb. 1, 2007 Written by Curt Clifton