Kennesaw State University

Department of Computer Science

CS 3310 - Introduction to Database Systems

Assignment 5 : Database Design

Benjamin Farmer, Michael Hug, Ronald Sullivan, Jovannie Sweeting

bfarme10[@students.kennesaw.edu](mailto:hmichae4@students.kennesaw.edu)

[hmichae4@students.kennesaw.edu](mailto:hmichae4@students.kennesaw.edu)

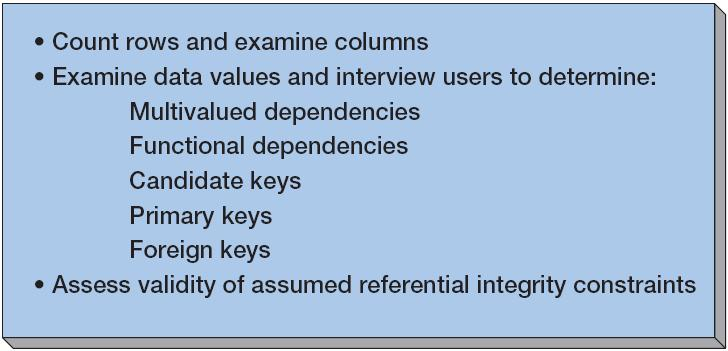
rsulli22[@students.kennesaw.edu](mailto:hmichae4@students.kennesaw.edu)

jsweeti1[@students.kennesaw.edu](mailto:hmichae4@students.kennesaw.edu)

17 October 2013

**Initial problem statement**

Work the Morgan Importing Problem described as described below. No database development is required for this assignment, just the table design. The assignment involves concepts covered in chapter 4 and its slides. Pay attention to the following guidelines for Assessing Table Structure from Figure 4-1 of Chapter 4.



Assessing Table Structure

Morgan Importing Problem:

*Phillip Morgan, the owner of Morgan Importing, makes periodic buying trips to various countries. During the trips, he keeps notes about the items he purchases and basic data about their shipments. He hired a college student as an intern, and she transformed his notes into the spreadsheets in Figure 4-9. This is just sample data. Phillip has purchased hundreds of items over the years, and they have been shipped in dozens of different shipments.*

*Phillip wants to enter the information age, thus he has decided to develop a database of his inventory. He wants to keep track of the items he has purchased, their shipments, and eventually customers and sales. To get started, he has asked you to create a database for the data in Figure 4-9.*

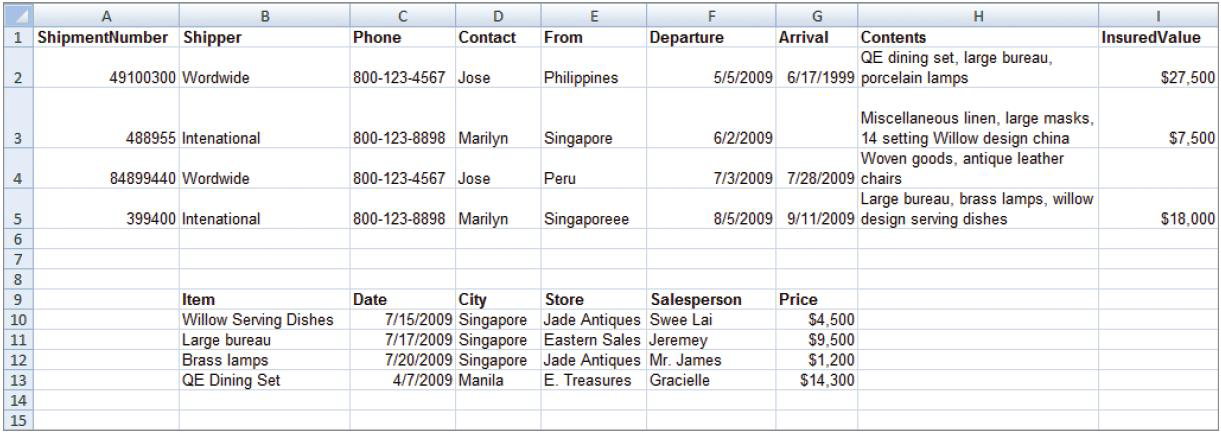


Figure 4-9 – Spreadsheets from Morgan Importing

**Summary and purpose of the assignment activity**

The goal is to practice database design using normalization. Eleven sets of instructions are given and lettered A-K. Each instruction requires analyzation, modification, or feedback from the given data structure.

**Completion of the assignment activity**

1. *Follow the “Assessing Table Structure” procedure shown. List multivalued dependencies, functional dependencies, candidate keys, primary keys, and foreign keys. State your assumptions.*

Multivalued Dependencies:

**From, Contact, Phone, Shipper, City**

Functional Dependencies:

**ShipmentNumber →** ( Shipper, Phone, Contact, From, Departure, Arrival, Contents, InsuredValue, Item, Date, City, Store, Salesperson, Price)

**Contact→** (Phone, Item, ShipmentNumber, City)

**Item→** (Contents, ShipmentNumber, InsuredValue, Price, Store, Salesperson, From, Arrival, Date)

Candidate Keys(s):

**Item →** (ShipmentNumber, Contents)

**Contacts →** ( ShipmentNumber)

Primary Key(s):

**ShipmentNumber, Contact, Item,**

Foreign Key(s):

**ShipmentNumber, Contact**

Based on the data given by , we surmised with three possible tables. For multivalued dependencies, attributes like From, Contact, and Phone carried the same value since Morgan bought more than one item and bought from different places. The functional dependencies are what we proposed by mingling them into three tables based on the kind of information given. The names of three tables that would be created are SALESPERSON, ITEM, and ITEM\_SHIPMENT. The primary keys would be ShipmentNumber, Contact, and Item since they serve primarily to identify the relations in the rows because it would easily identifiable to the buyer/seller. Likewise, ShipmentNumber and Contact would become foreign keys since they must relate have a relation to keep track on what the Morgan bought and from whom.

1. *List questions you would ask Phillip to verify your assumptions.*

“Mr. Morgan, I have several questions concerning with the data you provided…”

1. By City and From, is city is where you are getting the item and From is where the item was purchased or vice versa?
2. Why is the attribute **Shipper** seems to have the same exact information for every column but in a different name? Wordwide(or Worldwide) and Intenational(or International) are the same thing, Morgan.
3. Is it possible to include a last name to the contacts rather than just the first name to ensure that there are no confusion for contacts with the same names?
4. *Create tables as necessary to eliminate multivalued dependencies, if any.*

SHIPMENT\_DATA (ShipmentNumber, Shipper, From, Departure, Arrival, Contents, InsuredValue)

SHIPPER\_DATA (Shipper, Phone, Contact)

ITEM\_DATA (ItemNumber, Item)

SALESPERSON\_DATA (Salesperson, City, Store)

PRICE\_DATA (ItemNumber, Salesperson, Date, Price)

CONTENTS\_DATA (ShipmentNumber, ItemNumber)

1. *The relationship between shipment and item data could be inferred by matching values in the From cells to values in the City cells. Describe two problems with that strategy.*

There is no column named shipment or item data, so assume shipment means ShipmentNumber and item data means item. The From column seems to indicate the originating county of the shipment, and the city column seems to indicate the city of purchase. The problem is that something could be bought in one city and shipped from another.

1. *Describe a change to this spreadsheet that does express the shipment/item relationship.*

A change that would express the relationship between shipment and item would be to use the actual city in both the ‘City’ column for item and the ‘From’ column for the shipment. Currently in the ‘From’ column it has things such as Singapore listed which is a city-state-country not just a city, as well as the Philippines and Peru which are countries. So, there is an issue there. Also, in the ‘From’ column you will see that Singapore is actually misspelled as Singaporeee, which is a problem. I would use a drop down box for the ‘From’ and ‘City’ columns that has entries based on where there are store locations. This change would eliminate inconsistencies in spelling and using countries instead of cities.

1. *Assume that Phillip wishes to create an updateable database from this data. Design tables you think are appropriate. State all referential integrity constraints.*

Create the following tables : SHIPMENT\_DATA, SHIPPER\_DATA, ITEM\_DATA, SALESPERSON\_DATA, PRICE\_DATA , CONTENTS\_DATA

SHIPMENT\_DATA

ShipmentNumber→ (*Shipper*, Departure, Arrival, From, Item, InsuredValue) \*one to one,one,one,one,many,one

SHIPPER\_DATA

Shipper → (Phone, Contact) \*one to one,one

ITEM\_DATA

ShipmentNumber →(*Item*) \*one to many

SALESPERSON\_DATA

Salesperson → (City, Store) \*one to one,one

PRICE\_DATA

Item, Salesperson, Date → (Price) \*one,one,one to one

CONTENTS\_DATA

ShipmentNumber → (ItemNumber) \*one to many

1. *Assume that Phillip wishes to create a read-only database from this data. Design tables you think are appropriate. State all referential integrity constraints.*

In a read-only database, you can just have three tables with one primary key(ShipmentNumber), the second table as the item table, and the third as a salesperson table. This would maximize retrial speed.

1. *Do these data have the multivalue, multicolumn problem? If so, how will you deal with it?*

Yes, if one considers Item and contents to be equal. A solution would be to keep items in a desperate table, then reference that table every time a contents column or a sales row need queried.

1. *Do these data have the inconsistent data problem? If so, how will you deal with it?*

Yes.

In the Contents column, an example would be “willow design serving dishes” compared with Item ''Willow serving dishes”. Solution is detection and modification. If one was to compare item with contents is a where clause and the comparison of the text ‘Willow Serving Dishes’ would yield no result. In general you can count the columns for a specific value in this case ‘Willow serving dishes’ for each of these columns and you would see that one column in this case “Item” would give you a count of one, while Contents would give you 0. So then they have to be made the same, because for normalization to take place of course inconsistent data problems must be taken care of. This example uses group by and count to find number of rows for contents and that for Item as well. Once that is done one knows which column has these inconsistencies and then it can be modified to be the same. More important if such columns are designed to be the primary key and foreign keys because checking integrity would always fail.

Last example this uses two made up tables Employee and CollegeStudent. Take a column called EmployeeName in some table called Employee and it is a foreign key in CollegeStudent called Employee-Student-Name. If the values are different even just slightly then integrity fails. Example if there is a name “John Smith” in Employee but the name is spelled “Jon Smith” in the Employee-Student-name table then if we are trying to find information about John smith from both of these tables then we are in trouble. Like salary which is in Employee table and GPA which is in CollegeStudent table. To solve such a made up problem, change the name to John Smith in CollegeStudent table.

1. *Do these data have a null value data problem? If so, how will you deal with it?*

Yes, at arrival column. Not much can be done with null values because the data was never entered. In this case the null value is in Arrival. Would of had to of arrived at some point or it could not depart and it does. Finally more general null values can be found with the where “column” is null statement, and of course a Count “column” as “numberOfNulls” but again this just finds null values. Not too much can be done with null values and normalization does not solve the null value problem but rather the inconsistent data problem. Most likely the data for Arrival time is appropriate but it is unknown or was just not put in. No value can be added here because the format of the column is date and although the date would have to be before departure date the exact date is still unknown.

1. *Do these data have the general-purpose remarks problem? If so, how will you deal with it*

No, although ‘Contents’ is verbose it is consistent and is not a general-purpose remarks problem. If such a problem existed to fix it is complicated and it requires the creation of more columns most of the time to store different items in their own columns.

**Comments and Conclusion**

Assignment 5 is an exercise in database design using normalization. The given data was not in in any type of data structure. Data was displayed on a spreadsheet. The spreadsheet was only used to line up column visually, not to organize the data. The spreadsheet would have become exponentially more difficult to manage as the remaining Morgan shipping data from previous years was added to the spreadsheet. This database design outlined in this document, will allow for scalability of the Morgan shipping data . The database will simplify organization, storage, and retrieval of Morgan shipping data.

**References**

[1] David M. Kroenke and DavidAuer, *Database Processing: Fundamentals, Design, and Implementation,* 13th Edition ed. , Prentice Hall, 2012.