PART 1 & 2 – Chapter 7 NORMALIZATION (Beginning Database Design Solutions)

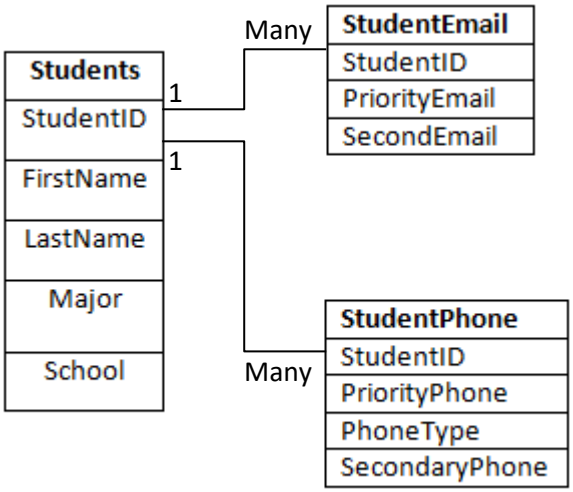
1. Suppose a student contact list contains the fields Name, Email, Email, Phone1, PhoneType1, Phone2, PhoneType2, and MajorOrSchool. The student's preferred email address is listed in the first Email field. Similarly the preferred phone number is in the Phone1 field. The MajorOrSchool field stores the student's major if he or she has picked one and the student's school (School of Engineering, School of Liberal Arts, School of Metaphysics, and so forth) otherwise.

   a. Explain why this list isn't in 1NF.

   These are not in 1NF, as they infringe on the 1NF rules:

   - Every column in the table must be unique. Since there are two columns with headings 'Email' is not a form of 1NF.
   - One column must only contain one type of data. In this contact list the major and the school are listed in one column.
   - One column must only contain one variable. The data type 'name' refers to first name and last name. These are two variables that must be in two separate columns with heading FirstName and LastName respectively.
   - Row and column order is not significant. Having two columns with preferred and secondaries respectively makes no difference to the database. Both emails will be listed without bias.

   b. Convert it into 1NF. Draw a relational diagram for it.



2. Consider the following table that lists errands that you need to run. The list shows the most important items at the top.

Location	Items
Grocery store	milk, eggs, bananas
Office supply store	paper, pencils, divining rod
Post Office	stamps
Computer store	flash drive, 8'' floppy disks

   a. Explain why this list isn't in 1NF.

- There are multiple variables in the one column 'items' and the priority is not given is not significant, values must be given to show level of priority

b. Convert this list into a single 1NF table. Be sure to define a primary key.

The primary key for the table below is the combination of Location and Items together.

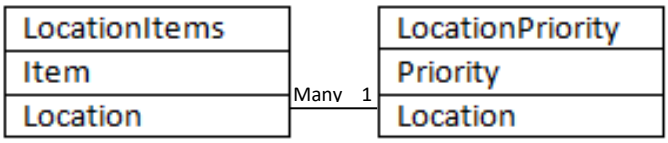
Location	Items	Priority
Grocery store	Milk	1
Grocery store	Eggs	1
Grocery store	Banana	1
Office supply store	Paper	2
Office supply store	Pencils	2
Office supply store	Driving rod	2
Post Office	Stamps	3
Computer store	Flash drive	4
Computer store	8" floppy disks	4

3. For the table you built for Exercise 2:

a. Explain why the table isn't in 2NF.

All partial dependencies must be removed. All of the non-key field depend on some of the key fields. An example is the Priority fields are dependant of the location and not the Items. Separating the Items and the Priority with the common field of location would make them 2NF.

b. Convert the table into 2NF.



LocationItems	
Location	**Item**
Grocery store	milk
Grocery store	eggs
Grocery store	bananas
Office supply store	paper
Office supply store	pencils
Office supply store	diving rod
Post Office	stamps
Computer Store	flash drive
Computer Store	8" floppy disk

LocationProrities	
Location	**Priorities**
Grocery Store	1
Office supply store	2
Post Office	3
Computer Store	4

4. Consider the following employee assignments table, which uses Employee as its primary key.

Employee	Project	Department
Alice Most	Work Assignment	Network Lab
Bill Michaels	Network Routing	Network Lab
Deanna Fole	Survey Design	Human Factors
Josh Farfar	Work Assignment	Network Lab

Employee	Project	Department
Julie Wish	Survey Design	Human Factors
Mandy Ponem	Network Routing	Network Lab
Mike Mix	New Services Analysis	Human Factors

a. Explain why the table isn't in 3NF

- It has transitive dependencies that must be removed

b. Convert the table into 3NF.

Employee	Project
Bill Michaels	Network Routing
Mandy Ponem	Network Routing
Mike Mix	Net Service Analysis
Deanna Fole	Survey Design
Julie Wish	Survey Design
Alice Most	Work Assignment
John Farfar	Work Assignment

Project	Department
Network Routing	Network Lab
Net Service Analysis	Human Factors
Survey Design	Human Factors
Work Assignment	Network Lab

5. One of your friends has decided to start a breakfast club. What each member can cook depends on his or her skills and equipment. Your friend built the following table to record all of the combinations.

Person	Food	Tool
Alice	Muffins	Muffin tin
Alice	Muffins	Omelet pan
Alice	Muffins	Pancake griddle
Alice	Omelets	Muffin tin
Alice	Omelets	Omelet pan
Alice	Omelets	Pancake griddle
Alice	Pancakes	Muffin tin
Alice	Pancakes	Omelet pan
Alice	Pancakes	Pancake griddle
Bob	Muffins	Omelet pan
Bob	Omelets	Omelet pan
Bob	Pancakes	Omelet pan
Cyndi	Omelets	Muffin tin
Cyndi	Omelets	Pancake griddle

Fortunately you know all about normalization, so help your friend by:

a. Explaining why the table isn't in 5NF.

It does not contain related multi-value dependencies

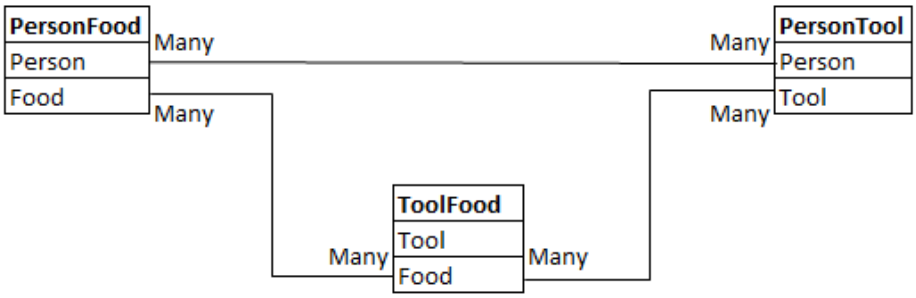


Figure above is an ER diagram for this model

b. Converting the table into 5NF.

PersonFood	
Food	**Person**
Muffins	Alice
Omelets	Alice
Pancakes	Alice
Muffins	Bob
Omelets	Bob
Pancakes	Bob
Omelets	Cyndi

PersonTool	
Person	**Tool**
Alice	Muffin Tin
Alice	Omelet Pan
Alice	Pancake Griddle
Bob	Omelet Pan
Cyndi	Muffin Tin
Cyndi	Pancake Griddle

ToolFood	
Tool	**Food**
Muffin Tin	Muffin
Omelet Pan	Omelet
Pancake Griddle	Pancake

6. In Figure 7-30, match the normal forms on the left with their corresponding rules on the right.

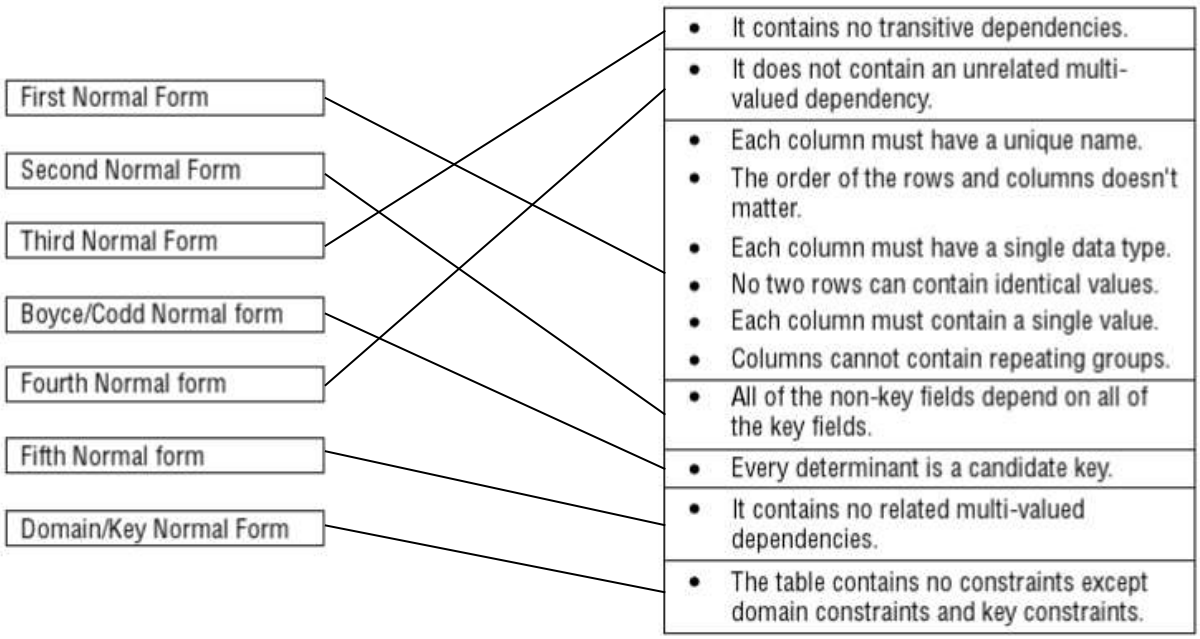


Figure 7-30

PART 3 & 4 – Chapter 4 DESIGNING A RELATIONAL DATABASE (Beginning MySQL)

The following questions are provided as a way for you to better acquaint yourself with the material covered in this chapter. Be sure to work through each exercise carefully. To view the answers to these questions, see Appendix A.

1. What are the components that make up a table in the relational model?

A Primary key, columns and rows

2. What requirements must a relation meet to be in compliance to the first normal form?

To be in compliance with the first normal form, a relation must meet the following requirements:

- Each column in a row must be atomic. This means that a column can contain only one value for any row in the database
- All rows in a database must be unique. At times a database variable may have similar values as another, however these variables must not be alike
- Each row in a database must have the same number of columns. If the tables have varying number of columns they will not conform to the rules of the first normal form.

3. How does a one-to-many relationship differ from a many-to-many relationship?

In a one-to-many relationship, a row in the first table can be related to one or more rows in the second table, with that in mind a row in the second table cannot be related to more than one

row from the first table. In a many-to-many relationship, a row in the first table can be related to one or more rows in the second table and a row in the second table can be related to one or more rows in the first table.

4. You are creating a data model for a MySQL database. You identify the entities that you found in the business rules. You then group those entities into categories of related data. What step should you take next?

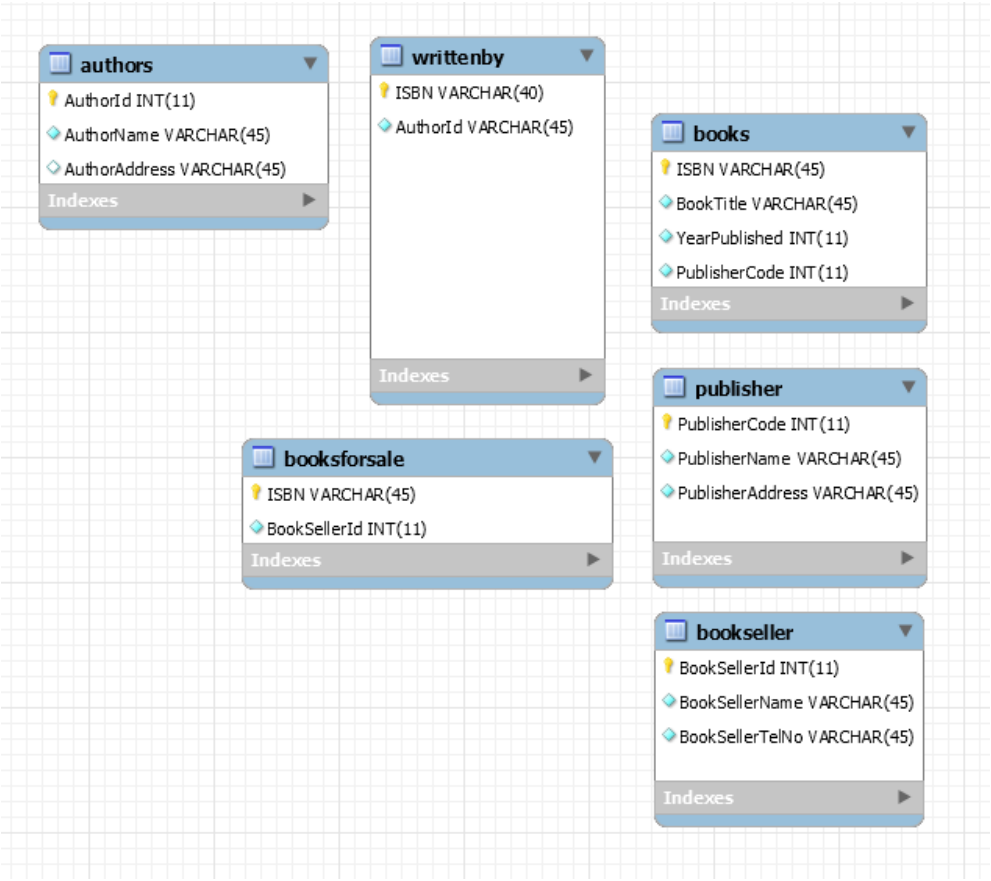
Normalization rules must be applied to the data.

5. How are many-to-many relationships implemented in MySQL?

Many-to-many relationships are implemented in MySQL by adding a junction table between the two tables and creating two one-to-many relationships, which creates one many-to-many relationship between the junction table and the first original table.

PART 5 & 6 – Chapter 4 ADVANCED DATABASE DESIGN (Beginning SQL)

1. Your friend runs a bookstore and uses a simple database to store details of books, authors, and booksellers. She the database might need a bit of a tune-up. Figure 4-2 shows its current structure, which consists of just one table containing two columns. Column Name and Data Type. Using what you learned about normalization, see if you can improve the current structure.



2. When improving the Film Club database, you added various constraints as well as primary and foreign keys. You didn't however add a FOREIGN KEY constraint to the FavCategory table. Create and execute the SQL required to enforce all its relationships with the other tables.

```
alter table FavCategory
add constraint favcat_cat_fk
foreign key (CategoryId)
references Category(CategoryId);

alter table FavCategory
add constraint favcat_member_fk
foreign key (MemberId)
references MemberDetails(MemberId);
```

PART 7 & 8 – Chapter 6 GROUPING AND AGGREGATING DATA (Beginning SQL)

1. The film club's chairperson decides that she'd like a list showing all the film costs broken down by film category and detailing how many films the club would have to buy in each category. She also wants you to factor in a 10% sales tax into the cost of each DVD. Change the results to list only those categories where one DVD is listed.

```
SELECT Category, COUNT(DVDPrice), SUM(DVDPrice * 1.1)
FROM Films
INNER JOIN Category
ON Films.CategoryId = Category.CategoryId
WHERE AvailableOnDVD = 'Y'
GROUP BY Category;
```

The SQL Code creates the following table

Category	COUNT(DVDPrice)	SUM(DVDPrice *1.1)
Historical	3	38.423
Horror	2	20.834
Romance	1	14.289
Sci-fi	1	14.289
Thriller	2	17.578
War	1	14.289

The final SQL results are in the table below

Category	COUNT(DVDPrice)	SUM(DVDPrice *1.1)
Romance	1	14.289
Sci-fi	1	14.289
War	1	14.289

2. Write a SQL query that finds out the highest and lowest rating for each film category.

```
SELECT Category, MAX(Rating) AS "Highest Rated", MIN(Rating) AS "Lowest Rated"
FROM Films INNER JOIN Category
ON Films.CategoryId = Category.CategoryId
GROUP BY Catagory;
```

Category	Highest Rated	Lowest Rated
Historical	5	3
Horror	2	1
Romance	4	4
Sci-fi	5	1
Thriller	4	1
War	5	2

The data comes from the Films and Category tables, which have been joined by INNER JOIN on the CategoryId Field. The results are grouped by category column and the highest & lowest values for each group are got using MAX() and MIN() functions

--END--