

## AM05 Data Mgmt – Assignment 2

---

This assignment has two parts. For the first part you will use the my\_guitar\_shop database you used to complete assignment 1.

### Part 1. SQL Update, Insert, and Delete Challenges

---

There are nine SQL challenges. **When you think you have the query working correctly copy the code to a Word document that will be your assignment submission along with before and after cropped screen shots that shows the effects of the query.**

Note:

- You will have to craft SELECT queries to display the data for the appropriate before and after screen shots but you do not have to submit the code of those queries.
- If you think you have messed up your database just rerun the create\_my\_guitar\_shop.sql script again. That should restore the data that's in the database.
- To test whether a table has been modified correctly as you do these exercises, you can write and run an appropriate SELECT statement.

#### Challenges

1. Write an INSERT statement that adds this row to the Categories table:

category\_name: Brass

Code the INSERT statement so MySQL automatically generates the category\_id column.

2. Write an UPDATE statement that modifies the row you just added to the Categories table. This statement should change the category\_name column to "Woodwinds", and it should use the category\_id column to identify the row.
3. Write a DELETE statement that deletes the row you added to the Categories table in exercise 1. This statement should use the category\_id column to identify the row.
4. Write an INSERT statement that adds this row to the Products table:

product\_id: The next automatically generated ID  
category\_id: 4  
product\_code: dgx\_640  
product\_name: Yamaha DGX 640 88-Key Digital Piano  
description: Long description to come.  
list\_price: 799.99  
discount\_percent: 0  
date\_added: Today's date/time.

Use a column list for this statement.

5. Write an UPDATE statement that modifies the product you added in exercise 4. This statement should change the discount\_percent column from 0% to 35%.
6. Write a DELETE statement that deletes the Keyboards category. When you execute this statement, it will produce an error since the category has related rows in the Products table. To fix that, precede the DELETE statement with another DELETE statement that deletes all products in this category. (Remember that to code two or more statements in a script, you must end each statement with a semicolon.)

7. Write an INSERT statement that adds this row to the Customers table:

email\_address:           rick@raven.com  
password:               (empty string)  
first\_name:             Rick  
last\_name:              Raven

Use a column list for this statement.

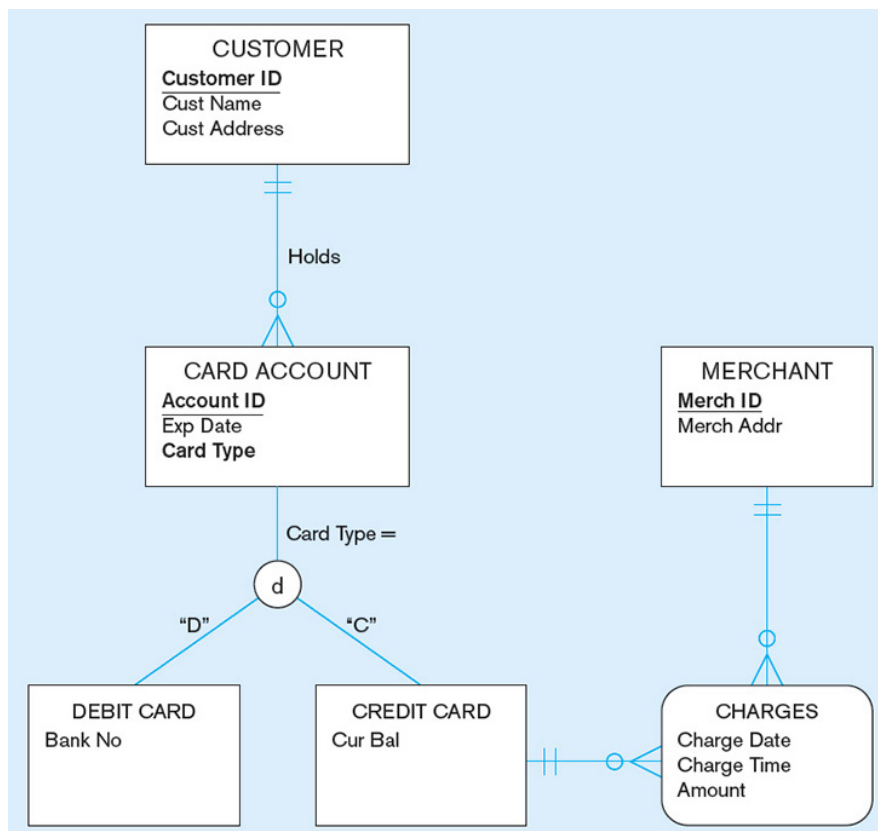
8. Write an UPDATE statement that modifies the Customers table. Change the password column to “secret” for the customer with an email address of rick@raven.com.
9. Write an UPDATE statement that modifies the Customers table. Change the password column to “reset” for every customer in the table. If you get an error due to safe-update mode, you can add a LIMIT clause to update the first 100 rows of the table. (This should update all rows in the table.)

Rerun the create\_my\_guitar\_shop.sql script again. That should restore the data that’s in the database.

## Part 2. Logical Database Design

In this part you can demonstrate your understanding of logical database design. Submit your answers in the same Word document you are using for Part 1 of this assignment.

Figure below shows an ERD for a simplified credit card environment. There are two types of card accounts: debit cards and credit cards. Credit card accounts accumulate charges with merchants. Each charge is identified by the date and time of the charge as well as the primary key of merchant and credit card.



1. Develop a relational schema. Use this format (you can use italics to show foreign keys):

<u>OrderID</u>	OrderDate	<i>CustomerID</i>
----------------	-----------	-------------------

Use arrows to show the links between **Primary Keys** and *Foreign Keys*, but there is no need to show the functional dependencies for this part

2. Show the functional dependencies using the following style:

**EmpID** → **Name, DeptName, Salary**

3. Is the relational schema you developed in question 1 in Third Normal Form (3NF)? Explain your reasoning. If it is not in 3NF develop a schema so that it is in 3NF.

The table below contains sample data for parts and for vendors who supply those parts. In discussing these data with users, we find that part numbers (but not descriptions) uniquely identify parts and that vendor names uniquely identify vendors.

Part No	Description	Vendor Name	Address	Unit Cost
1234	Logic chip	Fast Chips	Cupertino	10.00
		Smart Chips	Phoenix	8.00
5678	Memory chip	Fast Chips	Cupertino	3.00
		Quality Chips	Austin	2.00
		Smart Chips	Phoenix	5.00

4. Convert this table to a relation (named PART SUPPLIER) in first normal form. Illustrate the relation with the sample data in the table.
5. List the functional dependencies in PART SUPPLIER and identify a candidate key.
6. For the relation PART SUPPLIER identify each of the following:
  - an insert anomaly
  - a delete anomaly
  - a modification anomaly
7. Draw a relational schema for PART SUPPLIER relation and show the functional dependencies using arrows on the schema.
8. In what normal form is this relation? Explain your reasoning.
9. Develop a set of 3NF relations from PART SUPPLIER.
10. Show the 3NF relations as an ERD using *app.diagrams.net* or *MySQL Workbench*
11. Create the SQL DDL Statements that will create a new database, these relations (tables), the primary keys, foreign keys and any other necessary constraints. Submit
  - Your DDL code
  - Cropped screenshots of the metadata for each table (use **DESCRIBE tblname;**)