**Assignment Chapter 3**

**Note**

* You will need to complete “Exercise Chapter 2” before you will be able to start this “Assignment Chapter 3”. All questions use the my\_guitar\_shop schema.
* Please return answers to questions in a single SQL script file.
* Add your name in a comment in the SQL script file.
* Run the  utility to “Beautify/Reformat the SQL Script”. This will format and capitalize your SQL script file.
* The total number of points for the assignment is 50. There are 7 questions and each question is worth 50/7 = 7 points. The remainder 50%7 = 1 points will be awarded for completing all questions in the assignment.

**Example Question**

* Write a SELECT statement that returns four columns from the Products table:
  + product\_code
  + product\_name
  + list\_price
  + discount\_percent.

Then, run this statement to make sure it works correctly.

Add an ORDER BY clause to this statement that sorts the result set by list price in descending sequence. Then, run this statement again to make sure it works correctly. This is a good way to build and test a statement, one clause at a time.

**Example Answer**

/\* Andrew Chapman \*/

USE my\_guitar\_shop;

/\* Example Answer \*/

SELECT

product\_code, product\_name, list\_price, discount\_percent

FROM

products

ORDER BY list\_price DESC;

**Assignment Questions**

1. Write a SELECT statement that returns one column from the Customers table named full\_name that joins the last\_name and first\_name columns.

Format this column with the last name, a comma, a space, and the first name like this:

Doe, John

Sort the result set by last name in ascending sequence.

Return only the customers whose last name begins with letters from M to Z.

NOTE: When comparing strings of characters, ‘M’ comes before any string of characters that begins with ‘M’. For example, ‘M’ comes before ‘Murach’.

1. Write a SELECT statement that returns these columns from the Products table:

product\_name The product\_name column

list\_price The list\_price column

date\_added The date\_added column

Return only the rows with a list price that’s greater than 500 and less than 2000.

Sort the result set in descending sequence by the date\_added column.

1. Write a SELECT statement that returns these column names and data from the Products table:

product\_name The product\_name column

list\_price The list\_price column

discount\_percent The discount\_percent column

discount\_amount A column that’s calculated from the previous two columns

discount\_price A column that’s calculated from the previous three columns

Round the discount\_amount and discount\_price columns to 2 decimal places.

Sort the result set by discount price in descending sequence.

Use the LIMIT clause so the result set contains only the first 5 rows.

1. Write a SELECT statement that returns these column names and data from the Order\_Items table:

item\_id The item\_id column

item\_price The item\_price column

discount\_amount The discount\_amount column

quantity The quantity column

price\_total A column that’s calculated by multiplying the item price by the quantity

discount\_total A column that’s calculated by multiplying the discount amount by the quantity

item\_total A column that’s calculated by subtracting the discount amount from the item price and then multiplying by the quantity

Only return rows where the item\_total is greater than 500.

Sort the result set by item total in descending sequence.

1. Write a SELECT statement that returns these columns from the Orders table:

order\_id The order\_id column

order\_date The order\_date column

ship\_date The ship\_date column

Return only the rows where the ship\_date column contains a null value.

1. Write a SELECT statement without a FROM clause that uses the NOW function to create a row with these columns:

today\_unformatted The NOW function unformatted

today\_formatted The NOW function in this format:   
DD-Mon-YYYY

This displays a number for the day, an abbreviation for the month, and a four-digit year.

1. Write a SELECT statement without a FROM clause that creates a row with these columns:

price 100 (dollars)

tax\_rate .07 (7 percent)

tax\_amount The price multiplied by the tax

total The price plus the tax

To calculate the fourth column, add the expressions you used for the first and third