# MIS 275 - Lab 7

You will need to complete this lab by using SQL Server Management Studio. Use your textbook, in class activity and lecture as resources for help in completing the lab. Once you have completed the assignment then upload it to Blackboard. Save your document with the following naming convention:

* Course\_Lab Number\_First Initial Last Name (Example: MIS275\_Lab8\_LStewart.docx)

My Guitar Shop Murach Chapter 5 Exercises

In these activities, you’ll use SQL Server Management Studio to enter SQL statements and run them against the MyGuitarShop database. In addition to providing screen prints, create a SQL script file that contains your SQL statements for the 3 activities below. Once you’ve completed this lab, zip this Word document with your answers and your script file and upload the zipped file to the designated link in Blackboard.

1. Write a SELECT statement that returns these columns:

The count of the number of orders in the Orders table

The sum of the TaxAmount columns in the Orders table

Paste a screen print of the SQL statement and the resulting data set below.

1. Write a SELECT statement that returns one row for each category that has products with these columns:

The CategoryName column from the Categories table

The count of the products in the Products table

The list price of the most expensive product in the Products table

Sort the result set so the category with the most products appears first.

Paste a screen print of the SQL statement and the resulting data set below.

1. Write a SELECT statement that returns one row for each customer that has orders with these columns:

The EmailAddress column from the Customers table

The sum of the item price in the OrderItems table multiplied by the quantiy in the OrderItems table

The sum of the discount amount column in the OrderItems table multiplied by the quantiy in the OrderItems table

Sort the result set in descending sequence by the item price total for each customer.

Paste a screen print of the SQL statement and the resulting data set below.

1. Write a SELECT statement that returns one row for each customer that has orders with these columns:

The EmailAddress column from the Customers table

A count of the number of orders

The total amount for each order (*Hint: First, subtract the discount amount from the price. Then, multiply by the quantity.*)

Return only those rows where the customer has more than than 1 order.

Sort the result set in descending sequence by the sum of the line item amounts.

Paste a screen print of the SQL statement and the resulting data set below.

1. Modify the solution to exercise 4 so it only counts and totals line items that have an ItemPrice value that’s greater than 400.

Paste a screen print of the SQL statement and the resulting data set below.

1. Write a SELECT statement that answers this question: What is the total amount ordered for each product? Return these columns:

The product name from the Products table

The total amount for each product in the OrderItems table (*Hint: You can calculate the total amount by subtracting the discount amount from the item price and then multiplying it by the quantity*)

Use the WITH ROLLUP operator to include a row that gives the grand total.

Paste a screen print of the SQL statement and the resulting data set below.

1. Write a SELECT statement that answers this question: Which customers have ordered more than one product? Return these columns:

The email address from the Customers table

The count of distinct products from the customer’s orders

Paste a screen print of the SQL statement and the resulting data set below.

My Guitar Shop Murach Chapter 6 Exercises

In these activities, you’ll use SQL Server Management Studio to enter SQL statements and run them. In addition to providing screen prints, create a SQL script file that contains your SQL statements for the activities below. Once you’ve completed this lab, zip this Word document with your answers and your script file and upload the zipped file to the designated link in Blackboard.

1. Write a SELECT statement that returns the same result set as this SELECT statement, but don’t use a join. Instead, use a subquery in a WHERE clause that uses the IN keyword.

SELECT DISTINCT CategoryName

FROM Categories c JOIN Products p

ON c.CategoryID = p.CategoryID

ORDER BY CategoryName

Paste a screen print of the SQL statement and the resulting data set below.

1. Write a SELECT statement that answers this question: Which products have a list price that’s greater than the average list price for all products?

Return the ProductName and ListPrice columns for each product.

Sort the results by the ListPrice column in descending sequence.

Paste a screen print of the SQL statement and the resulting data set below.

1. Write a SELECT statement that returns the CategoryName column from the Categories table.

Return one row for each category that has never been assigned to any product in the Products table. To do that, use a subquery introduced with the NOT EXISTS operator.

Paste a screen print of the SQL statement and the resulting data set below.

1. Write a SELECT statement that returns three columns: EmailAddress, OrderID, and the order total for each customer. To do this, you can group the result set by the EmailAddress and OrderID columns. In addition, you must calculate the order total from the columns in the OrderItems table.

Write a second SELECT statement that uses the first SELECT statement in its FROM clause. The main query should return two columns: the customer’s email address and the largest order for that customer. To do this, you can group the result set by the EmailAddress column.

Paste a screen print of the SQL statement and the resulting data set below.

1. Write a SELECT statement that returns the name and discount percent of each product that has a unique discount percent. In other words, don’t include products that have the same discount percent as another product.

Sort the results by the ProductName column.

Paste a screen print of the SQL statement and the resulting data set below.

1. Use a correlated subquery to return one row per customer, representing the customer’s oldest order (the one with the earliest date). Each row should include these three columns: EmailAddress, OrderID, and OrderDate.

Paste a screen print of the SQL statement and the resulting data set below.

Complete My Guitar Shop Murach Chapter 7 Exercises

To test whether a table has been modified correctly as you do these exercises, you should write and run an appropriate SELECT statement.

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

CategoryName: Brass

Code the INSERT statement so SQL Server automatically generates the value for the CategoryID column.

Paste a screen print of the SQL statement and the resulting data set below.

1. Write an UPDATE statement that modifies the row you just added to the Categories table. This statement should change the CategoryName column to “Woodwinds”, and it should use the CategoryID column to identify the row.

Paste a screen print of the SQL statement and the resulting data set below.

1. Write a DELETE statement that deletes the row you added to the Categories table in exercise 1. This statement should use the CategoryID column to identify the row.

Paste a screen print of the SQL statement and the resulting data set below.

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

ProductID: The next automatically generated ID   
CategoryID: 4  
ProductCode: dgx\_640  
ProductName: Yamaha DGX 640 88-Key Digital Piano  
Description: Long description to come.  
ListPrice: 799.99  
DiscountPercent: 0  
DateAdded: Today’s date/time.

Use a column list for this statement.

Paste a screen print of the SQL statement and the resulting data set below.

1. Write an UPDATE statement that modifies the product you added in exercise 4. This statement should change the DiscountPercent column from 0% to 35%.

Paste a screen print of the SQL statement and the resulting data set below.

1. Write a DELETE statement that deletes the row in the Categories table that has an ID of 4. 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.

Paste a screen print of the SQL statement and the resulting data set below.

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

EmailAddress: rick@raven.com  
Password: (empty string)  
FirstName: Rick  
LastName: Raven

Use a column list for this statement.

Paste a screen print of the SQL statement and the resulting data set below.

1. 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](mailto:rick@raven.com).

Paste a screen print of the SQL statement and the resulting data set below.

1. Write an UPDATE statement that modifies the Customers table. Change the password column to “reset” for every customer in the table.

Paste a screen print of the SQL statement and the resulting data set below.

1. Open the script named CreateMyGuitarShop.sql that’s in the Exercise Starts directory. Then, run this script. That should restore the data that’s in the database.

Paste a screen print of the SQL statement and the resulting data set below.

When you are finished zip this Word document with your answers AND your script files then zip file then upload to the designated link in Blackboard.