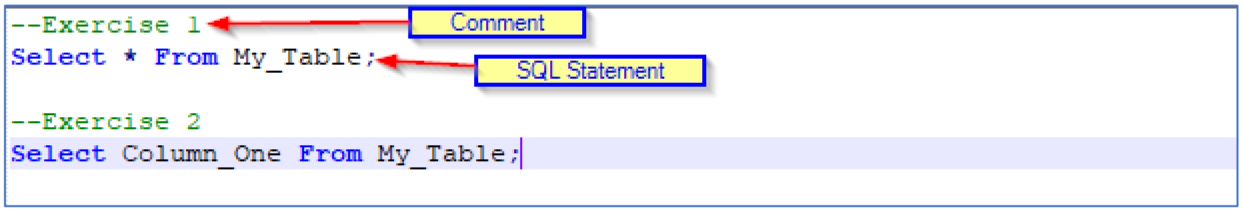
Chapter 3: Lab 2 Part 2

How to retrieve data from a single table

Exercises

Enter and run your own SELECT statements. Save the final result of each exercise to <last name>_<first name>_lab2_part_2.sql (ex: Haley_Shane_lab2_part_2.sql). Use comments before each SQL statement to show which exercise it was for (example below):



```
--Exercise 1
Select * From My_Table;

--Exercise 2
Select Column_One From My_Table;
```

In these exercises, you'll enter and run your own SELECT statements.

1. (**10 points**) Write a SELECT statement that returns these column names and data from the **Addresses** table:

CustomerID	The **CustomerID** column
City	The **City** column
State	The **State** column

   Return only the rows where the customer is from Abilene, TX or Bridgewater, NJ.

   Sort the result set in ascending sequence by the **City**, **State**, and **CustomerID** columns (in that order).

2. (**10 points**) Write a SELECT statement that returns these column names and data from the **Addresses** table:

CustomerID	The **CustomerID** column
City	The **City** column
State	The **State** column

   Return only the rows where the customer is **IN** the states AK, CA, and TX, but **NOT IN** the cities of Fairbanks, San Francisco, and Abilene.

   Sort the result set in ascending sequence by the **State**, **City**, and **CustomerID** columns (in that order).

   Make sure to use the phrases "**IN**" and "**NOT IN**".

3. (**10 points**) Write a SELECT statement that returns these column names and data from the **Addresses** table:

CustomerID	The **CustomerID** column
City	The **City** column
State	The **State** column

Return only the rows where the city starts with ‘C’ and the state ends with letters ranging from ‘A’ through ‘M’. **Hint**: both LIKE phrases will require the ‘%’ wildcard symbol (the second LIKE phrase will require additional wildcard symbols).

Sort the result set in ascending sequence by the **State**, **City**, and **CustomerID** columns (in that order).

4. (**10 points**) Write a SELECT statement that returns these columns from the **Orders** table:

OrderID	The OrderID column
OrderDate	The OrderDate column
ShipDate	The ShipDate column

Return only the rows where the **ShipDate** column contains a null value.

5. (**10 points**) Write a SELECT statement without a FROM clause that creates a row with these columns:

Price	100 (dollars)
TaxRate	.07 (7 percent)
TaxAmount	The price multiplied by the tax rate
Total	The price plus tax

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