SQL Query Assignment 2

Premiere Products

## Directions

* Write the following queries
* Screenshot the **working SQL and output screen**.
* Submit to Blackboard - be sure your name is in the file

## Queries

If I do not specify a field list, use \* for the field list. For each query, I have specified the number of rows you should see

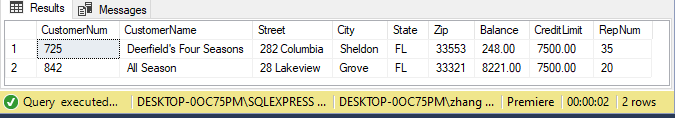
Pattern Matching

Complete these queries using the like operator

1. Find all customers with the word season in their name. (2 rows)

**select \* from Customer**

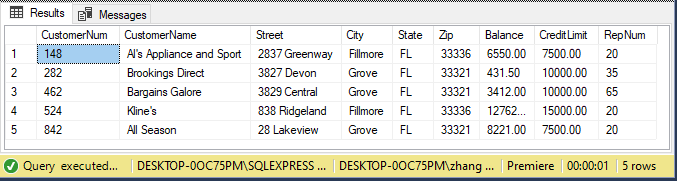
**where CustomerName like '%season%';**

****

1. Find all customers whose zipcode starts with 333 (5 rows)

**select \* from Customer**

**where Zip like '%333%';**

****

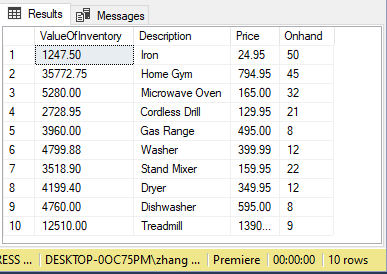
## Calculated Fields

Complete the following tasks by creating a calculated field.

1. Using the part table, display the part description, price, onhand. Calculate the ValueOfInventory with price \* quantity on hand. When you are correct, Iron's ValueOfInventory will be 1247.50

**select price \* Onhand as ValueOfInventory,**

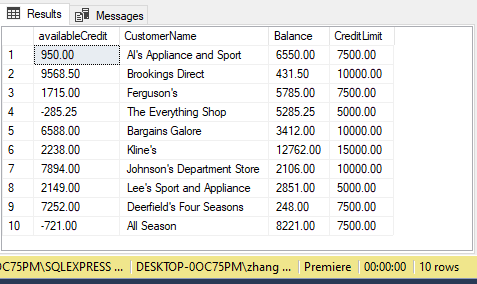
**Description, Price, Onhand from Part;**

****

1. Using the customer table, display the customer name, balance and credit limit. Calculate their availableCredit with credit limit -  balance. When your math is correct, Al's Appliance's availableCredit is 950.

**select CreditLimit - Balance as availableCredit,**

**CustomerName, Balance, CreditLimit from Customer;**

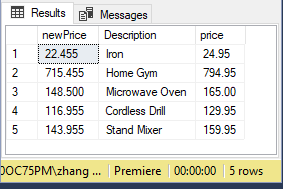
****

1. Using the part table, display the description, cost. Limit the output to only the products with more than 20 on hand. Calculate the new price after a 10% discount. You will be correct when you have 5 rows of output and the discounted price of the iron is 22.455

**select Price \* 0.9 as newPrice,**

**Description, price from Part**

**where OnHand > 20;**

****

## Aggregate Calculations

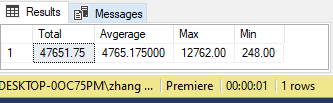
**Be sure to rename all aggregate output fields**

1. Using customers, calculate the total, average, max, and min balances for our customers. You will be correct when there is one line of output and the total is 47651.75

**select sum(Balance) as Total, avg(Balance) as Avgerage,**

**max(Balance) as Max, min(Balance) as Min**

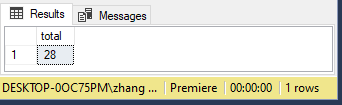
**from Customer;**

****

1. Using orderline, provide the count of orders and the total numOrdered. Again you will have one line, and the total numOrdered is 28.

**select sum(NumOrdered) as total**

**from OrderLine;**

****

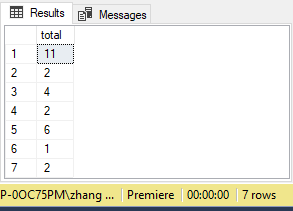
Orders and orderline are related. The orderNum in orderline refers to a particular order. Each row in orderline are items in a particular order. Imagine this is a restaurant. You place an order, with a Sandwich, chips, and a drink. Your one order has three orderlines associated with it.

1. Repeat the query from number 7. This time, group by ordernum. You will have seven rows of output.

**select sum(NumOrdered) as total**

**from OrderLine**

**group by OrderNum;**

****

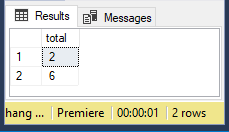
1. Repeat the query from number 8. Now use the having clause to display only those orderlines with 2 or more ordernums. You will be correct when you have only 2 rows of data.

**select sum(NumOrdered) as total**

**from OrderLine**

**group by OrderNum**

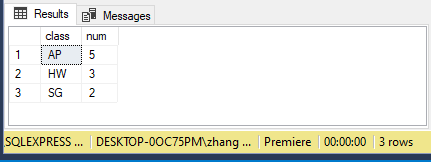
**having count(\*) >= 2;**



1. Provide a count of parts grouped by class. You are correct when you have three rows of output and the count of AP is 5.

**select class, count(\*) as num from part**

**group by class**

****

## Bonus

1. Display the number of customers and avg balance per repnum. Also in the same query show the overall average and total number of customers. You will have 4 rows of output, and the overall average is 4765.175
2. Display the customers who have a balance less than the average balance of all customers. You can do this in 1 or 2 queries. More bonus if you can do this in one query.