Homework 3

Due Sep 26 by 11:59pm **Points** 50 **Submitting** a file upload **File Types** sql and txt
Available Sep 19 at 12:20pm - Sep 26 at 11:59pm 7 days

This assignment was locked Sep 26 at 11:59pm.

Submit a query for each of the following questions. All queries will be solutions against the **WideWorldImporters** database. The questions make specific request regarding implementation requirements, but you may need additional constructs as well. For example, if the question is asking for a correlated subquery to calculate a required field, you may also need joins in your solution.

Question 1

10 points - Write a query to return all orders from January 2016 placed by customers in the “Computer Store” category. Your solution should use a self-contained subquery to identify the customers belonging to the “Computer Store” category.

Required Tables

- Sales.CustomerCategories
- Sales.Customers
- Sales.Orders
- Sales.OrderLines

Result Columns

- **OrderID** – Identifier of the order as it exists in *Sales.Orders*.
- **OrderDate** – The date of the order as it exists in *Sales.Orders*.
- **CustomerID** – The identifier of the customer placing the order.
- **OrderTotal** – The total for the order calculated using the *Quantity* and *UnitPrice* from *Sales.OrderLines*.

The results should be sorted with the highest order total first, then by *OrderID* in ascending order to make the results deterministic.

Partial Results

With the correct solution, the first two rows should be as follows:

OrderID	OrderDate	CustomerID	OrderTotal
64519	2016-01-08	1036	18405.00
65309	2016-01-21	1055	12643.00

Question 2

10 points - Write a query to return suppliers from the category of “Novelty Goods Supplier” for which no stock items are kept. Your solution should use a correlated subquery to indicate whether a given supplier has stock items.

Required Tables

- Purchasing.SupplierCategories
- Purchasing.Suppliers
- Warehouse.StockItems
- Application.Cities
- Application.StateProvinces

Result Columns

- **SupplierID** – Identifier of the supplier as it exists in *Purchasing.Suppliers*.
- **SupplierName** – Name of the supplier as it exists in *Purchasing.Suppliers.*
- **City** – Name of the postal city of the supplier as it appears in *CityName* of *Application.Cities*.
- **State** – Postal state of the supplier as it appears in *StateProviceCode* of *Application.StateProvinces* for the postal city.
- **PostalCode** – As it appears in the *PostalPostalCode* of *Purchasing.Suppliers*.

The results should be sorted by supplier name in ascending order.

Partial Results

With the correct solution, the first row should be as follows:

SupplierID	SupplierName	City	State	PostalCode
8	Lucerne Publishing	Jonesborough	TN	37659

Question 3

10 points - Write a query that returns all orders for customer with a CustomerID of **90**, along with the order total and number of days since the customer’s previous order.

You should use a correlated subquery to calculate the number of days since the previous order.

Required Tables

- Sales.Orders
- Sales.OrderLines

Result Columns

- **OrderID** – Identifier of the order as it exists in *Sales.Orders*.
- **OrderDate** – The date of the order as it exists in *Sales.Orders*.

- **OrderTotal** – The total for the order calculated using the *Quantity* and *UnitPrice* from *Sales.OrderLines*.
- **DaysSincePreviousOrder** – The days since the previous order for the customer. In addition to using a correlated subquery, you should use the **DATEDIFF (https://docs.microsoft.com/en-us/sql/t-sql/functions/datediff-transact-sql)** function. Its first argument should be *DAY*. The second argument would be the earliest of the two dates compared, which in our case is the date of the previous order. The third argument would be the row's order date, which would be *OrderDate* from *Sales.Orders*.
- The results should be sorted by *OrderID* in ascending order.

Partial Results

With the correct solution, the first two rows should be as follows:

OrderID	OrderDate	OrderTotal	DaysSincePreviousOrder
1455	2013-01-29	365.00	NULL
1890	2013-02-06	915.00	8

Question 4

10 points - Write a query to return sales information for each customer who placed an order in 2015.

Your solution should use a derived table to calculate the *OrderCount* and *Sales* for each *CustomerID*.

Required Tables

- Sales.Customers
- Sales.Orders
- Sales.OrderLines

Result Columns

- **CustomerID** – Identifier of the customer as it exists in *Sales.Customers*.
- **CustomerName** – The name of the customer as it exists in *Sales.Customers*.
- **OrderCount** – The number of orders place in 2015 by each customer.
- **Sales** – The total sales for each customer in 2015. Sales are calculated using the *Quantity* and *UnitPrice* from *Sales.OrderLines*.

The results should be sorted with the customer having the highest sales first, then by *CustomerID* in ascending order to make the results deterministic.

Partial Results

With the correct solution, the first two rows should be as follows:

CustomerID	CustomerName	OrderCount	Sales
820	Knut Svensson	55	150701.50
996	Laszlo Gardenier	38	150506.70

Question 5

10 points - Use the same requirements as with Question 4 and write another query using a common table expression rather than a derived table to calculate the *OrderCount* and *Sales* for each *CustomerID*. The results should be equivalent to those in Question 4.

Submission

Please submit your solution to each question in a single SQL file clearly marking each question with a comment line, or each in a separate file with a file name clearly indicating the question it solves. Include a comment line above each solution indicating which question it answers. Please do not submit your results, only the SQL solutions.

Five Questions - 10 Each

Criteria	Ratings		Pts
Question 1	**10.0 pts** **Full Marks**	**0.0 pts** **No Marks**	10.0 pts
Question 2	**10.0 pts** **Full Marks**	**0.0 pts** **No Marks**	10.0 pts
Question 3	**10.0 pts** **Full Marks**	**0.0 pts** **No Marks**	10.0 pts
Question 4	**10.0 pts** **Full Marks**	**0.0 pts** **No Marks**	10.0 pts
Question 5	**10.0 pts** **Full Marks**	**0.0 pts** **No Marks**	10.0 pts
			Total Points: 50.0