***ID 204502926***

***OFIR NESHER***

**Assignment 1: SQL**

All of the questions below refer to AdventureWorksLT2017 database (see below) that serves as the tutorial database for Microsoft’s SQL Server and can be downloaded as part of its installation.

[Here](https://sqlzoo.net/w/images/2/28/AdventureWorks.png) you can find the ERD schema of the database.

You should write a SQL statement that answers each of the following questions. Note which columns should be returned and whether duplicate answers should be canceled. Submit a Word file solution containing all the answers. Check your answers. To help you, I have listed for each question how many answers should be returned according to the data.

You can install your own SQL Server + AdventureWorksLT database. For your convenience, I have created a SQL Server instance in the cloud that includes the AdventureWorksLT database. You can connect to this server using the SQL Server Management Studio (SSMS), which can be downloaded from [here](https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15), or any other database client (DBeaver, DataGrip, TablePlus, etc).

**In the SSMS login window, choose “SQL Server authentication” and write:**

**Server name: 194.153.101.138**

**Login: student**

**Password: BigData2020**

**Questions**:

1. Show the Company Name for James D. Kramer (1 row)

* SELECT CompanyName FROM SalesLT.Customer WHERE FirstName='James' AND MiddleName='D.' AND LastName='Kramer'

1. Show the Company Name for all customers located in a City whose name starts with D. (11 rows)

* SELECT DISTINCT CompanyName FROM (SalesLT.Customer JOIN SalesLT.CustomerAddress ON SalesLT.Customer.CustomerID=SalesLT.CustomerAddress.CustomerID) JOIN SalesLT.Address ON SalesLT.CustomerAddress.AddressID=SalesLT.Address.AddressID WHERE City like 'D%'

1. Show all companies that have a shipping address, with that address. Use the explicit join notation (10 rows)

* SELECT DISTINCT CompanyName, AddressLine1 FROM (SalesLT.Customer JOIN SalesLT.CustomerAddress ON SalesLT.Customer.CustomerID=SalesLT.CustomerAddress.CustomerID) JOIN SalesLT.Address ON SalesLT.CustomerAddress.AddressID=SalesLT.Address.AddressID WHERE SalesLT.CustomerAddress.AddressType='Shipping'

1. Print the CompanyName of those customers with orders over $50000 and the order amount, including the subtotal, tax, and freight, from highest value to lowest (8 rows)

* SELECT CompanyName, TotalDue, SubTotal, TaxAmt, Freight FROM (SalesLT.Customer JOIN SalesLT.SalesOrderHeader ON SalesLT.Customer.CustomerID=SalesLT.SalesOrderHeader.CustomerID) WHERE TotalDue >= 50000 ORDER BY TotalDue DESC

1. How many products in ProductCategory 'Cranksets' have been sold to an address (shipping or billing) in 'London' (1 row)?

* SELECT COUNT (ProductCategory.ProductCategoryID) AS 'ProductAmount' FROM ((((SalesLT.SalesOrderHeader JOIN SalesLT.SalesOrderDetail ON SalesOrderHeader.SalesOrderID = SalesOrderDetail.SalesOrderID) JOIN SalesLT.Product ON SalesOrderDetail.ProductID = Product.ProductID) JOIN SalesLT.ProductCategory ON Product.ProductCategoryID = ProductCategory.ProductCategoryID) JOIN SalesLT.Address ON Address.AddressID=SalesOrderHeader.ShipToAddressID OR Address.AddressID=SalesOrderHeader.BillToAddressID) WHERE ProductCategory.Name = 'Cranksets' AND City = 'London'

1. Show pairs of products with the same list price, weight, and color, w/o duplicates (2 rows)

* SELECT DISTINCT Product.Name, DUPLICATE\_PRODUCT.Name, Product.ListPrice, Product.Weight, Product.Color FROM SalesLT.Product JOIN SalesLT.Product AS "DUPLICATE\_PRODUCT" ON Product.Weight = DUPLICATE\_PRODUCT.Weight AND Product.Color = DUPLICATE\_PRODUCT.Color AND Product.ListPrice = DUPLICATE\_PRODUCT.ListPrice AND Product.ProductID < DUPLICATE\_PRODUCT.ProductID

1. How many items with ListPrice more than $1000 have been sold, and what was their average price? (1 row)

* SELECT Count(OrderQty) AS 'Order Quantity', AVG(UnitPrice - UnitPriceDiscount) AS 'Average Price' FROM SalesLT.SalesOrderDetail JOIN SalesLT.Product ON SalesOrderDetail.ProductID=Product.ProductID WHERE ListPrice > 1000

1. Show the product categories (names) and the total weight of products in each category, if the product weight is not NULL (12 rows)

* SELECT ProductCategory.Name, SUM(Product.Weight) AS 'Total Weight' FROM SalesLT.Product JOIN SalesLT.ProductCategory ON Product.ProductCategoryID = ProductCategory.ProductCategoryID WHERE Weight is not NULL GROUP BY Product.ProductCategoryID, ProductCategory.Name

1. Show product model names and the number of products of each model, if there are at least 10 such products (8 rows)

* SELECT ProductModel.Name, COUNT(Product.ProductID) AS 'Number of Products' FROM SalesLT.ProductModel JOIN SalesLT.Product ON ProductModel.ProductModelID = Product.ProductModelID GROUP BY ProductModel.ProductModelID, ProductModel.Name HAVING COUNT(Product.ProductID)>=10

1. Show the 5 best selling items AND their total selling price in ascending order in a column called Total (5 rows)

* SELECT \* FROM (SELECT TOP 5 Product.Name, SUM(UnitPrice - UnitPriceDiscount) AS 'Total' FROM SalesLT.SalesOrderDetail JOIN SalesLT.Product ON SalesOrderDetail.ProductID = Product.ProductID GROUP BY Product.ProductID, Product.Name, OrderQty \* UnitPrice ORDER BY OrderQty \* UnitPrice DESC) AS Temp ORDER BY Temp.Total

1. Show the names of product models that have products in both size XL and S (3 rows). Note the Size field may be called Sze.

* SELECT TEMP.Name FROM (SELECT ProductModel.Name AS 'Name', Size FROM SalesLT.Product JOIN SalesLT.ProductModel ON Product.ProductModelID = ProductModel.ProductModelID GROUP BY ProductModel.Name, Size HAVING Size IN ('S','XL')) AS Temp GROUP BY TEMP.Name HAVING COUNT(\*) > 1

1. Show the details of the product with the lowest list price (1 row)

* SELECT TOP 1 \* FROM SalesLT.Product ORDER BY ListPrice