

## Assignment1\_OJAS\_PHANSEKAR\_001826636

2)

a) Total sales

a. Using SalesOrderHeader

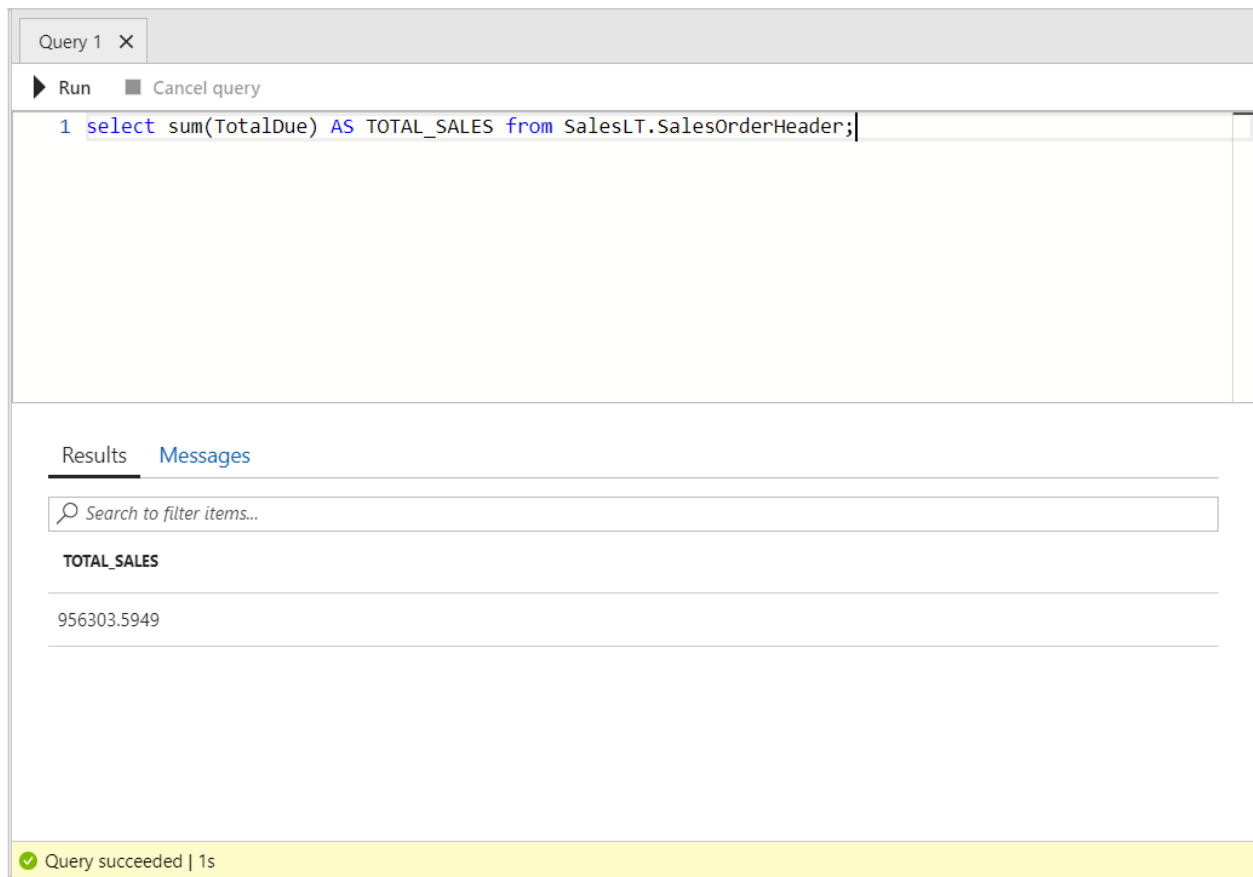
**SQL Query:** `select sum(TotalDue) AS TOTAL_SALES from SalesLT.SalesOrderHeader;`

**Input Tables:** SalesLT.SalesOrderHeader

**Input Columns :** TotalDue

**Output Columns :** TOTAL\_SALES

**Total Number of Rows in results :** 1



The screenshot displays the SQL Server Enterprise Manager interface. At the top, a tab labeled 'Query 1' is active. Below it, a toolbar contains 'Run' and 'Cancel query' buttons. The query editor shows the following SQL query: `1 select sum(TotalDue) AS TOTAL_SALES from SalesLT.SalesOrderHeader;`. Below the query editor, the 'Results' tab is selected, showing a single column named 'TOTAL\_SALES' with a value of '956303.5949'. A search bar is present above the results. At the bottom, a status bar indicates 'Query succeeded | 1s'.

TOTAL_SALES
956303.5949

a)b. Using SalesOrderDetail

**SQL Query:** `select sum(((UnitPrice-UnitPriceDiscount)*OrderQty)) AS TOTAL_SALES from SalesLT.SalesOrderDetail;`

**Input Tables:** SalesLT.SalesOrderDetail

**Input Columns :** UnitPrice, UnitPriceDiscount, OrderQty

**Output Columns :** TOTAL\_SALES

**Total Number of Rows in results :** 1

Query 1 X

▶ Run ■ Cancel query

1 select sum(((UnitPrice-UnitPriceDiscount)\*OrderQty)) AS TOTAL\_SALES from SalesLT.SalesOrderDetail;

Results Messages

🔍 Search to filter items...

TOTAL\_SALES

713969.7636

b) Total sales by country – ranked/sorted (highest to lowest)

**SQL Query:** select Address.CountryRegion,sum(SalesLT.SalesOrderHeader.TotalDue) as Total\_Sales from SalesLT.SalesOrderHeader INNER JOIN SalesLT.Customer on SalesLT.SalesOrderHeader.CustomerID = SalesLT.Customer.CustomerID INNER JOIN SalesLT.CustomerAddress ON SalesLT.Customer.CustomerID = SalesLT.CustomerAddress.CustomerID INNER JOIN SalesLT.Address on SalesLT.CustomerAddress.AddressID = SalesLT.Address.AddressID GROUP BY SalesLT.Address.CountryRegion ORDER BY Total\_Sales Desc;

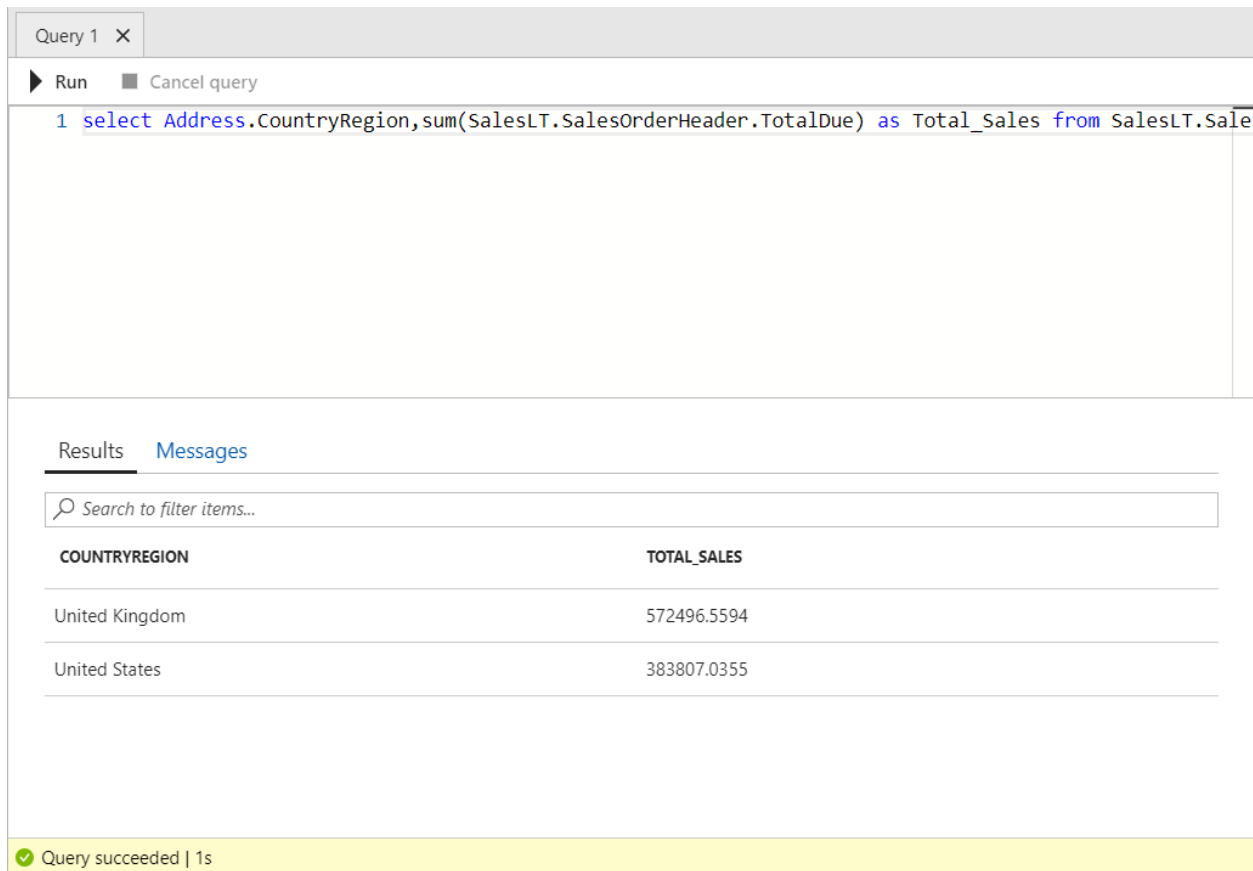
**Input Tables:** SalesLT.SalesOrderHeader, SalesLT.Customer, SalesLT.CustomerAddress, SalesLT.Address,

**Input Columns :** Address.CountryRegion, SalesLT.SalesOrderHeader.TotalDue, SalesLT.SalesOrderHeader.CustomerID, SalesLT.Customer.CustomerID,

SalesLT.CustomerAddress.CustomerID, SalesLT.CustomerAddress.AddressID,  
SalesLT.Address.AddressID

**Output Columns :** COUNTRYREGION, TOTAL\_SALES

**Total Number of Rows in results :** 2



The screenshot shows a SQL query execution window. At the top, there's a tab labeled 'Query 1' with a close button. Below it are 'Run' and 'Cancel query' buttons. The query text area contains the following SQL statement:

```
1 select Address.CountryRegion,sum(SalesLT.SalesOrderHeader.TotalDue) as Total_Sales from SalesLT.Sale
```

Below the query area, there are two tabs: 'Results' (selected) and 'Messages'. Under the 'Results' tab, there's a search bar with the placeholder text 'Search to filter items...'. Below the search bar is a table with two columns: 'COUNTRYREGION' and 'TOTAL\_SALES'. The table contains two rows of data:

COUNTRYREGION	TOTAL_SALES
United Kingdom	572496.5594
United States	383807.0355

At the bottom of the window, a status bar indicates 'Query succeeded | 1s'.

c) Total sales by city & country – ranked/sorted (highest to lowest)

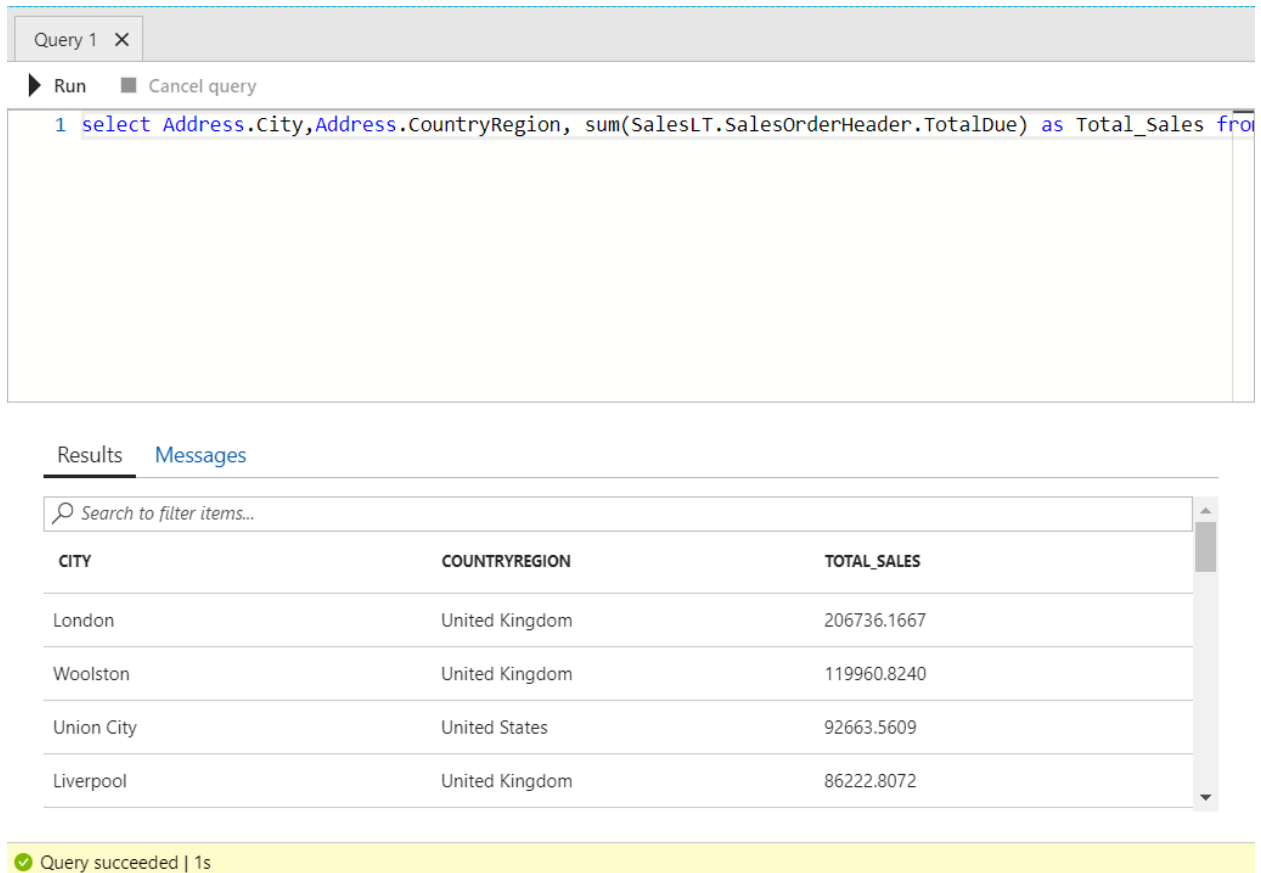
**SQL Query:** `select Address.City,Address.CountryRegion,  
sum(SalesLT.SalesOrderHeader.TotalDue) as Total_Sales from  
SalesLT.SalesOrderHeader INNER JOIN SalesLT.Customer on  
SalesLT.SalesOrderHeader.CustomerID = SalesLT.Customer.CustomerID INNER JOIN  
SalesLT.CustomerAddress ON SalesLT.Customer.CustomerID =  
SalesLT.CustomerAddress.CustomerID INNER JOIN SalesLT.Address on  
SalesLT.CustomerAddress.AddressID = SalesLT.Address.AddressID GROUP BY  
SalesLT.Address.CountryRegion,SalesLT.Address.City ORDER BY Total_Sales Desc;`

**Input Tables:** SalesLT.SalesOrderHeader, SalesLT.Customer, SalesLT.CustomerAddress,  
SalesLT.Address

**Input Columns :** Address.City ,Address.CountryRegion,  
SalesLT.SalesOrderHeader.TotalDue, SalesLT.SalesOrderHeader.CustomerID,  
SalesLT.Customer.CustomerID, SalesLT.CustomerAddress.CustomerID,  
SalesLT.CustomerAddress.AddressID, SalesLT.Address.AddressID

**Output Columns :** CITY,COUNTRYREGION, TOTAL\_SALES

**Total Number of Rows in results :** 29



Query 1 X

Run Cancel query

```
1 select Address.City,Address.CountryRegion, sum(SalesLT.SalesOrderHeader.TotalDue) as Total_Sales from
```

Results Messages

Search to filter items...

CITY	COUNTRYREGION	TOTAL_SALES
London	United Kingdom	206736.1667
Woolston	United Kingdom	119960.8240
Union City	United States	92663.5609
Liverpool	United Kingdom	86222.8072

Query succeeded | 1s

d) Total sales by customer (person) – ranked/sorted (highest to lowest)

**SQL Query:** select (Customer.FirstName + SPACE(1)+ Customer.LastName) as Full\_Name,  
sum(SalesLT.SalesOrderHeader.TotalDue) as Total\_Sales from  
SalesLT.SalesOrderHeader inner join SalesLT.Customer on  
SalesLT.SalesOrderHeader.CustomerID = SalesLT.Customer.CustomerID Group by  
(Customer.FirstName + SPACE(1) +Customer.LastName) order by Total\_Sales desc;

**Input Tables:** SalesLT.SalesOrderHeader, SalesLT.Customer

**Input Columns :** Customer.FirstName, Customer.MiddleName, Customer.LastName,  
SalesLT.SalesOrderHeader.TotalDue, SalesLT.SalesOrderHeader.CustomerID,  
SalesLT.Customer.CustomerID

**Output Columns :** FULL\_NAME, TOTAL\_SALES

**Total Number of Rows in results : 32**

The screenshot shows a SQL query execution window. At the top, there's a tab labeled 'Query 1'. Below it, a toolbar contains a 'Run' button (a right-pointing triangle) and a 'Cancel query' button (a square). The query text area contains the following SQL statement:

```
1 select (Customer.FirstName + SPACE(1)+ Customer.LastName) as Full_Name, sum(SalesLT.SalesOrderHeader
```

Below the query text, there's a 'Results' section. It includes a timestamp '6/1/2008 12:00:00 AM' and a search bar with the placeholder text 'Search to filter items...'. The results are displayed in a table with two columns: 'FULL\_NAME' and 'TOTAL\_SALES'. The table contains five rows of data:

FULL_NAME	TOTAL_SALES
Terry Eminhizer	119960.8240
Krishna Sunkammurali	108597.9536
Christopher Beck	98138.2131
Kevin Liu	92663.5609

At the bottom of the window, a yellow status bar indicates 'Query succeeded | 1s'.

e) Total sales by customer (company) – ranked/sorted (highest to lowest)

**SQL Query:** `select Customer.CompanyName, sum(SalesLT.SalesOrderHeader.TotalDue) as Total_Sales from SalesLT.SalesOrderHeader inner join SalesLT.Customer on SalesLT.SalesOrderHeader.CustomerID = SalesLT.Customer.CustomerID Group by Customer.CompanyName order by Total_Sales desc;`

**Input Tables:** SalesLT.SalesOrderHeader, SalesLT.Customer

**Input Columns :** Customer.CompanyName, SalesLT.SalesOrderHeader.TotalDue, SalesLT.SalesOrderHeader.CustomerID, SalesLT.Customer.CustomerID

**Output Columns :** COMPANYNAME, TOTAL\_SALES

**Total Number of Rows in results : 32**

Query 1 X

▶ Run ■ Cancel query

```
1 CustomerID = SalesLT.Customer.CustomerID Group by Customer.CompanyName order by Total_Sales desc;
```

Results Messages

COMPANYNAME	TOTAL_SALES
Action Bicycle Specialists	119960.8240
Metropolitan Bicycle Supply	108597.9536
Bulk Discount Store	98138.2131
Eastside Department Store	92663.5609

✔ Query succeeded | 2s

f) Sales by product category – ranked/sorted (highest to lowest)

**SQL Query:** `select productCat.name , sum(salesOrder.orderqty*(salesOrder.unitprice- salesOrder.unitpricediscount)) as Total_Sales from saleslt.productcategory prod join saleslt.productcategory productCat on prod.parentproductcategoryid= productCat.productcategoryid join saleslt.product salesProd on prod.productcategoryid= salesProd.productcategoryid join saleslt.salesorderdetail salesOrder on salesProd.productid= salesOrder.productid group by productCat.name order by Total_Sales desc;`

**Input Tables:** SalesLT.SalesOrderDetail, SalesLT.Product,SalesLT.ProductCategory

**Input Columns :**

**Output Columns :** NAME,TOTAL\_SALES

**Total Number of Rows in results :** 4

Query 1

Run
Cancel query

```
1 select productCat.name , sum(salesOrder.orderqty*( salesOrder.unitprice- salesOrder.unitpricediscoun
```

ResultsMessages

Search to filter items...

NAME	TOTAL_SALES
Bikes	579657.6318
Components	111075.7320
Clothing	16290.3664
Accessories	6946.0334

Query succeeded | 1s

g) Sales by product name – ranked/sorted (highest to lowest)

**SQL Query:** `select SalesLT.Product.ProductID,SalesLT.Product.Name, sum((SalesLT.SalesOrderDetail.UnitPrice - SalesLT.SalesOrderDetail.UnitPriceDiscount)*SalesLT.SalesOrderDetail.OrderQty) as Total_Sales from SalesLT.SalesOrderDetail inner join SalesLT.Product on SalesLT.SalesOrderDetail.ProductID = SalesLT.Product.ProductID Group by SalesLT.Product.ProductID,SalesLT.Product.Name order by Total_Sales Desc;`

**Input Tables:** SalesLT.SalesOrderDetail, SalesLT.Product

**Input Columns :**

**Output Columns :** PRODUCTID,NAME,TOTAL\_SALES

**Total Number of Rows in results :** 142

Query 1 X

▶ Run

■ Cancel query

1

select SalesLT.Product.ProductID,SalesLT.Product.Name, sum((SalesLT.SalesOrderDetail.UnitPrice - Sal

2

3

|

Results

Messages

🔍 Search to filter items...

PRODUCTID	NAME	TOTAL_SALES
976	Road-350-W Yellow, 48	38609.9730
969	Touring-1000 Blue, 60	37191.4920
783	Mountain-200 Black, 42	37178.8380
782	Mountain-200 Black, 38	35801.8440

✔ Query succeeded | 1s