Create queries to answer following questions using the link http://www.w3schools.com/sql/trysql.asp?filename=trysql\_select\_all

Hint: For top/bottom types of queries (e.g. top 3 orders) you can use ORDER BY along with LIMIT x,y where x = offset (optional) and y=limit.

Example 1) Display Top 5 orders displaying Order ID, Customer Name, Order Date and Total Price for each order sorted in descending order (biggest order on top)

SELECT O.OrderId, C.CustomerName, O.OrderDate, ROUND(SUM(P.Price \* OD.Quantity),2) as TotalPrice FROM [Orders] AS O

INNER JOIN [OrderDetails] as OD ON O.OrderID = OD.OrderID

INNER JOIN [Products] as P ON OD.ProductID = P.ProductID

INNER JOIN [Customers] as C ON O.CustomerID= C.CustomerID

GROUP BY OD.OrderId

ORDER BY TotalPrice DESC

LIMIT 5;

Note: I have used LIMIT 5 which is same as saying LIMIT 0,5 because offset of 0 is default.

Example 2) Display next top 5 orders (rank 6 - 10) displaying Order ID, Customer Name, Order Date and Total Price for each order sorted in descending order (biggest order on top)

SELECT O.OrderId, C.CustomerName, O.OrderDate, ROUND(SUM(P.Price \* OD.Quantity),2) as TotalPrice FROM [Orders] AS O

INNER JOIN [OrderDetails] as OD ON O.OrderID = OD.OrderID

INNER JOIN [Products] as P ON OD.ProductID = P.ProductID

INNER JOIN [Customers] as C ON O.CustomerID= C.CustomerID

GROUP BY OD.OrderId

ORDER BY TotalPrice DESC

LIMIT 5,5;

Note: I am specifying the offset 5 so that the query will return results starting from record 6 and give me 5 records.

Example 3) Display Bottom 5 orders displaying Order ID, Customer Name, Order Date and Total Price for each order sorted in ascending order (smallest order on top)

SELECT O.OrderId, C.CustomerName, O.OrderDate, ROUND(SUM(P.Price \* OD.Quantity),2) as TotalPrice FROM [Orders] AS O

INNER JOIN [OrderDetails] as OD ON O.OrderID = OD.OrderID

INNER JOIN [Products] as P ON OD.ProductID = P.ProductID

INNER JOIN [Customers] as C ON O.CustomerID= C.CustomerID

GROUP BY OD.OrderId

ORDER BY TotalPrice ASC

LIMIT 5;

Note: there is no change in the limit part of the query, I just flipped the ORDER BY to ASC.

==============================================================================

For questions below I have provided the output for you to check your answers

==============================================================================

Q1: List the top shipper (in terms of total number of orders shipped) and total number of order shipped.

ShipperName TotalShipping

United Package 74

SELECT count(OrderID) as countOrder, Orders.ShipperID, Shippers.ShipperName FROM [Orders]

inner join Shippers on Orders.ShipperID= Shippers.ShipperID

group by Orders.ShipperID

order by countOrder desc;

Q2: Show top 5 employees, excluding the top employee (show employees ranked 2-5) in terms of total sales done by those employees. In the query display employee's first name, last name and TotalSales sorted in descending order.

FirstName LastName TotalSales

Nancy Davolio 57690.39

Janet Leverling 42838.35

Robert King 39772.3

Laura Callahan 39309.38

SELECT e.FirstName, e.LastName, Round(sum(od.Quantity \* p.Price),2) as TotalSales FROM [Orders] as o

inner join OrderDetails as od on od.OrderID = o.OrderID

inner join Products as p on p.ProductID = od.ProductID

inner join Employees as e on e.EmployeeID = o.EmployeeID

group by o.EmployeeID

order by TotalSales desc

limit 1,4

Q3: Now modify the query in Q2 to filter the data for only orders done in year 1996.

Hint: In SQL usually the dates are written in YYYY-MM-DD format.

FirstName LastName TotalSales

Nancy Davolio 48523.78

Laura Callahan 28967.78

Andrew Fuller 28560.7

Steven Buchanan 27480.8

SELECT e.FirstName, e.LastName, Round(sum(od.Quantity \* p.Price),2) as TotalSales FROM [Orders] as o

inner join OrderDetails as od on od.OrderID = o.OrderID

inner join Products as p on p.ProductID = od.ProductID

inner join Employees as e on e.EmployeeID = o.EmployeeID

where o.OrderDate between '1996-01-01' and '1996-12-31'

group by o.EmployeeID

order by TotalSales desc

limit 1,4;

Q4: List top 5 products (in terms of quantity sold) and total quantity sold.

ProductName TotalQuantity

Gorgonzola Telino 458

Camembert Pierrot 430

Steeleye Stout 369

Raclette Courdavault 346

Chang 341

SELECT p.ProductName, round(sum(OrderDetails.Quantity),2) as TotalQuantity FROM [OrderDetails]

inner join Products as p on OrderDetails.ProductID = p.ProductID

group by p.ProductName

order by TotalQuantity desc

limit 5;

Q5: List SupplierID, SupplierName and ItemSupplied (count of number of items supplied by a supplier), sort the list first by number of items supplied (descending) and then by supplier name (ascending)

SupplierID SupplierName ItemSupplied

7 Pavlova, Ltd. 5

12 Plutzer Lebensmittelgroßmärkte AG 5

2 New Orleans Cajun Delights 4

8 Specialty Biscuits, Ltd. 4

... and more

SELECT p.SupplierID, s.SupplierName,count(p.ProductID) as TotalItems FROM [Products] as p

inner join Suppliers as s on s.SupplierID = p.SupplierID

group by p.SupplierID

order by TotalItems desc, s.SupplierName asc;

Q6: List top 10 suppliers (with supplierID and name) along with TotalOrders (total of price \* quantity of the products sold, supplied by these suppliers) for the year 1997 in descending order.

SupplierID SupplierName TotalOrders

18 Aux joyeux ecclésiastiques 26086.5

12 Plutzer Lebensmittelgroßmärkte AG 10233.7

7 Pavlova, Ltd. 9793.7

... and 7 more

SELECT p.SupplierID, s.SupplierName, round(sum(od.Quantity \* p.Price),2) as TotalSales FROM [Products] as p

inner join Suppliers as s on s.SupplierID = p.SupplierID

inner join OrderDetails as od on od.ProductID = p.ProductID

inner join Orders as o on o.OrderID = od.OrderID

where o.OrderDate between '1997-01-01' and '1997-12-31'

group by p.SupplierID

order by TotalSales desc

limit 10;

Q7: [Note this question is more challenging than others] List customers for each category and the total of order placed by that customer in a given category. In the query show three column: CategoryName, CustomerName, and TotalOrders (which is price \* quantity for orders for a given customer in a given category). Sort this data in descending order by TotalOrders.

CategoryName CustomerName TotalOrders

Beverages Mère Paillarde 13991.5

Beverages Piccolo und mehr 13175

Beverages Simons bistro 13175

Beverages Queen Cozinha 10720

Condiments Ernst Handel 7652.6

Dairy Products Frankenversand 7440.8

...and more

SELECT cat.CategoryName, cus.CustomerName, round(sum(od.Quantity \* p.Price),2) as TotalOrders FROM [Orders] as o

inner join Customers as cus on cus.CustomerID = o.CustomerID

inner join OrderDetails as od on od.OrderID = o.OrderID

inner join Products as p on p.ProductID = od.ProductID

inner join Categories as cat on cat.CategoryID = p.CategoryID

group by cus.CustomerID, cat.CategoryName

order by TotalOrders desc;