**Intermediate SQL Projects**

USE Northwind\_SPP;

--20. Show total number of products in each category. Sort the results by the total number of products

-- in desc order

SELECT \* FROM Categories;

SELECT \* FROM Products;

SELECT

CategoryName, COUNT(Products.CategoryID) as TotalProducts

FROM

Categories

JOIN Products ON Products.CategoryID = Categories.CategoryID

Group by CategoryName

Order by TotalProducts DESC;

--21. In the customers table, show total number of customers by Country and City

SELECT \* FROM Customers;

SELECT

Country, City, COUNT(CustomerID) as TotalCustomers

FROM

Customers

GROUP BY Country, City

Order by TotalCustomers DESC;

--22. Determine which products need reorder (the products where UnitsinStock <= reorderlevel)

--No need to order by ASC, as is the default

SELECT \* FROM Products;

SELECT

ProductID, ProductName, UnitsinStock, Reorderlevel

FROM Products

WHERE UnitsinStock <= ReorderLevel

ORDER BY ProductID ASC;

--23. Determine which product need reordered (the products where UnitsinStock + UnitsOnOrder <= ReorderLevel)

SELECT ProductID, ProductName, UnitsInStock, UnitsOnOrder, ReorderLevel, Discontinued

FROM Products

WHERE UnitsInStock + UnitsOnOrder <= ReorderLevel AND Discontinued = 0;

--24. Sort customers by Region alphabetically, customers within the same region should be sorted by CustomerID,

--with null values in region at the end of the list

SELECT

CustomerID, CompanyName, Region,

Case

when Region is null then 1

else 0

END as RegionExists

FROM Customers

Order By RegionExists, Region, CustomerID;

--25. Select 3 countries with highest average freight charges

SELECT \* FROM Orders;

SELECT

ShipCountry, AverageFreight = AVG(Freight)

FROM

Orders

GROUP BY ShipCountry

ORDER BY AverageFreight DESC

OFFSET 0 ROWS FETCH FIRST 3 ROWS ONLY;

--26. Select 3 countries with the highest freight charges for 2015

SELECT

ShipCountry, AverageFreight = AVG(Freight)

FROM

Orders

WHERE year(OrderDate) = '2015'

GROUP BY ShipCountry

ORDER BY AverageFreight DESC

OFFSET 0 ROWS FETCH FIRST 3 ROWS ONLY;

--27. Run this query: select \* from orders order by OrderDate, and look at rows around December 31, 2015.

--What do you notice about the freight field-there is a $2000 freight charge from France on December, 31, 2015

--When using the between, do the before and the day after you want to query

SELECT TOP 3

ShipCountry

, AverageFreight = avg(freight)

FROM Orders

WHERE

OrderDate between '20141231' AND '20160101'

Group BY ShipCountry

Order By AverageFreight desc;

SELECT \* FROM Orders order by OrderDate;

SELECT

OrderDate, ShipCountry, AverageFreight = AVG(Freight)

FROM

Orders

Group By ShipCountry

ORDER BY OrderDate DESC;

--28. Get three countries with average freight charges, using the last 12 months of order data

--below returns single max order date

SELECT

MAX(OrderDate) as "MaxDate"

FROM

Orders;

SELECT \* FROM Orders;

SELECT

ShipCountry, AverageFreight = AVG(Freight)

FROM

Orders

WHERE OrderDate >

(SELECT Dateadd(yy, -1, (Select Max(OrderDate) from Orders)))

GROUP BY ShipCountry

ORDER BY AverageFreight DESC

OFFSET 0 ROWS FETCH FIRST 3 ROWS ONLY;

--29. Show employee and order Detail information like table and sort by OrderID and ProductID

SELECT \* FROM Employees;

SELECT \* FROM Orders;

SELECT \* FROM Products;

SELECT \* FROM OrderDetails;

SELECT

Employees.EmployeeID, Employees.LastName, Orders.OrderID, Products.ProductName, OrderDetails.Quantity

FROM

Employees

join

Orders ON Employees.EmployeeID = Orders.EmployeeID

join

OrderDetails ON OrderDetails.OrderID=Orders.OrderID

join

Products ON Products.ProductID=OrderDetails.ProductID

ORDER BY Orders.OrderID, Products.ProductID;

--30. Show customers who have never actually placed an order

SELECT

Customers\_CustomerID = Customers.CustomerID, Orders\_CustomerID=Orders.CustomerID

FROM Customers

left join Orders ON Orders.CustomerID=Customers.CustomerID

WHERE Orders.CustomerID IS NULL

Order BY Customers.CustomerID;

--31. Only show orders where customers have never placed an order with Margaret Peacock (EmployeeID is 4)

--Left Join Orders - want orders data to be on right side of dataset

SELECT \* FROM Customers;

SELECT \* FROM Orders;

SELECT

Customers.CustomerID, Orders.CustomerID

FROM Customers

Left JOIN Orders

ON Orders.CustomerID=Customers.CustomerID

and Orders.EmployeeID=4

WHERE Orders.CustomerID IS NULL

ORDER BY Customers.CustomerID;

--Or

SELECT CustomerID

FROM Customers

WHERE CustomerID not in (select CustomerID from Orders where EmployeeID = 4)

--Or

SELECT CustomerID

FROM Customers

WHERE NOT EXISTS

(

SELECT CustomerID

from Orders

WHERE Orders.CustomerID=Customers.CustomerID

and EmployeeID=4

)