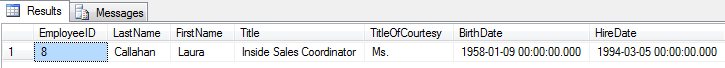
1. **Show all info about the employee with ID 8.**

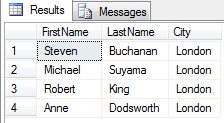
SELECT \*

FROM Employees

WHERE EmployeeID=8

****

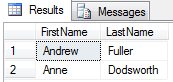
1. **Show the list of first and last names of the employees from London.**

SELECT FirstName,LastName,City

FROM Employees

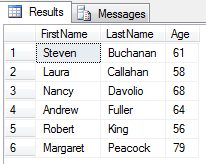
WHERE City='London'

1. **Show the list of first and last names of the employees whose first name begins with letter A.**

SELECT FirstName,LastName

FROM Employees

WHERE FirstName LIKE 'A%'

1. **Show the list of first, last names and ages of the employees whose age is greater than 55. The result should be sorted by last name.**

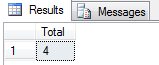
SELECT FirstName, LastName, DATEDIFF(YEAR,BirthDate,GETDATE()) AS Age

FROM Employees

WHERE DATEDIFF(YEAR,BirthDate,GETDATE())>'55'

ORDER BY LastName

1. **Calculate the count of employees from London.**

SELECT COUNT(EmployeeID) AS Total

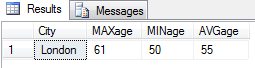
FROM Employees

WHERE City = 'London'

1. **Calculate the greatest, the smallest and the average age among the employees from London.**

SELECT City, MAX(DATEDIFF(YEAR,BirthDate,GETDATE()))AS MAXage,

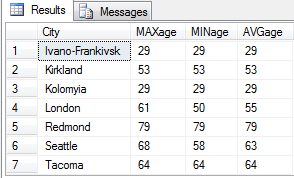
MIN(DATEDIFF(YEAR,BirthDate,GETDATE()))AS MINage,

AVG(DATEDIFF(YEAR,BirthDate,GETDATE()))AS AVGage

FROM Employees

WHERE City='London'

GROUP BY City

1. **Calculate the greatest, the smallest and the average age of the employees for each city.**

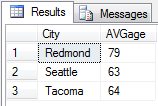
SELECT City, MAX(DATEDIFF(YEAR,BirthDate,GETDATE()))AS MAXage,

MIN(DATEDIFF(YEAR,BirthDate,GETDATE()))AS MINage,

AVG(DATEDIFF(YEAR,BirthDate,GETDATE()))AS AVGage

FROM Employees

GROUP BY City

1. **Show the list of cities in which the average age of employees is greater than 60 (the average age is also to be shown)**

SELECT City, AVG(DATEDIFF(YEAR,BirthDate,GETDATE()))AS AVGage

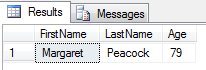
from Employees

GROUP BY City

HAVING AVG(DATEDIFF(YEAR,BirthDate,GETDATE()))>'60'

1. **Show the first and last name(s) of the eldest employee(s). Use a subquery.**

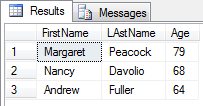
SELECT TOP 1 FirstName, LastName, A.Age

FROM

(SELECT FirstName, LastName, DATEDIFF(YEAR,BirthDate,GETDATE())AS Age

FROM Employees) AS A

ORDER BY A.Age DESC

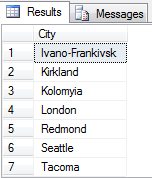
1. **Show first, last names and ages of 3 eldest employees.**

SELECT TOP 3 FirstName, LAstName, DATEDIFF(YEAR,BirthDate,GETDATE()) AS Age

FROM Employees

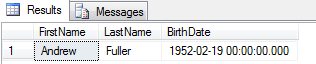
ORDER BY BirthDate ASC

1. **Show the list of all cities where the employees are from.**

SELECT DISTINCT City

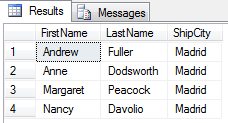
FROM Employees

1. **Show first, last names and dates of birth of the employees who celebrate their birthdays this month.**

SELECT FirstName, LastName, BirthDate

FROM Employees

WHERE MONTH(BirthDate)=MONTH(GETDATE())

1. **Show first and last names of the employees who used to serve orders shipped to Madrid.**

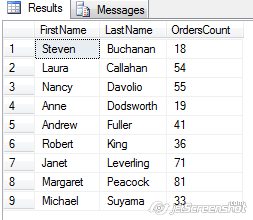
SELECT DISTINCT FirstName, LastName,Orders.ShipCity

FROM Employees

LEFT JOIN Orders

ON Employees.EmployeeID=Orders.EmployeeID

WHERE Orders.ShipCity='Madrid'

1. **Show first and last names of the employees as well as the count of orders each of them have received during the year 1997 (use left join).**

SELECT FirstName, LastName, COUNT(OrderID) AS OrdersCount

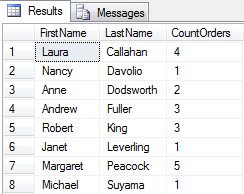
FROM Employees

LEFT JOIN Orders

ON Employees.EmployeeID=Orders.EmployeeID

WHERE YEAR(OrderDate)='1997'

GROUP BY FirstName,LastName

1. **Show first and last names of the employees as well as the count of their orders shipped after required date during the year 1997 (use left join).**

SELECT FirstName, LastName, COUNT(OrderID) AS CountOrders

FROM Employees

LEFT JOIN Orders

ON Employees.EmployeeID=Orders.EmployeeID

WHERE ShippedDate>RequiredDate

AND YEAR(ShippedDate)='1997'

GROUP BY FirstName, LastName

1. **Show the count of orders made by each customer from France.**

SELECT CompanyName, COUNT(OrderID) AS CountOrders

FROM Customers

LEFT JOIN Orders

ON Customers.CustomerID=Orders.CustomerID

WHERE Country='France'

GROUP BY CompanyName

ORDER BY CompanyName

1. **Show the list of french customers’ names who have made more than one order (use grouping).**

SELECT Customers.ContactName, COUNT(OrderID) AS CountOrders

FROM Customers

LEFT JOIN Orders

ON Customers.CustomerID=Orders.CustomerID

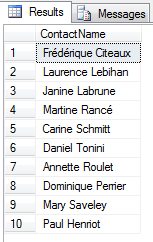
WHERE Customers.Country='France'

GROUP BY Customers.ContactName

HAVING COUNT(OrderID)>'1'

ORDER BY Customers.ContactName

1. **Show the list of french customers’ names who have made more than one order (use a subquery).**

SELECT ContactName

FROM Customers

WHERE Country='France'

AND CustomerID IN

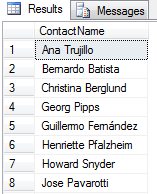
(SELECT CustomerID

FROM Orders

GROUP BY CustomerID

HAVING COUNT(OrderID)>'1')

1. **Show the list of customers’ names who used to order the ‘Tofu’ product (use a subquery).**

SELECT ContactName

FROM Customers

WHERE CustomerID IN

(SELECT CustomerID

FROM Orders

WHERE OrderID IN

(SELECT OrderID

FROM [Order Details]

WHERE ProductID IN

(SELECT ProductID

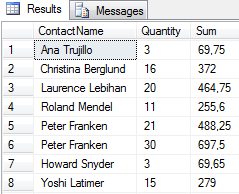
FROM Products

WHERE ProductName='Tofu')))

ORDER BY ContactName

1. **\*Show the list of customers’ names who used to order the ‘Tofu’ product, along with the total amount of the product they have ordered and with the total sum for ordered product calculated.**

SELECT ContactName, [Order Details].Quantity, ([Order Details].UnitPrice \*[Order Details].Quantity-[Order Details].Discount) AS [Sum]

FROM Customers

LEFT JOIN Orders

ON Customers.CustomerID=Orders.CustomerID

LEFT JOIN [Order Details]

ON orders.OrderID=[Order Details].OrderID

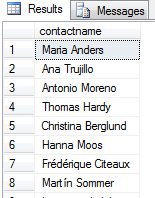
WHERE productId IN

(SELECT ProductID

FROM Products

WHERE ProductName='Tofu')

1. .
2. **\*Show the list of french customers’ names who used to order non-french products (use a subquery).**

SELECT contactname

FROM customers

WHERE customerID IN

(SELECT customerID

FROM Orders

WHERE orderId IN

(SELECT orderId

FROM [Order Details]

WHERE productId IN

(SELECT productId

FROM Products

WHERE supplierId IN

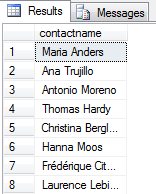
(SELECT SupplierID

FROM Suppliers

WHERE Country !='France'))))

1. **\*Show the list of french customers’ names who used to order french products.**

SELECT contactname

FROM customers

WHERE customerID IN

(SELECT customerID

FROM Orders

WHERE orderId IN

(SELECT orderId

FROM [Order Details]

WHERE productId IN

(SELECT productId

FROM Products

WHERE supplierId IN

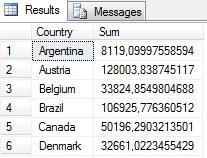
(SELECT SupplierID

FROM Suppliers

WHERE Country ='France'))))

1. **\*Show the total ordering sum calculated for each country of customer.**

SELECT Customers.Country, SUM ([Order Details].UnitPrice \*[Order Details].Quantity\*(1-[Order Details].Discount))AS [Sum]

FROM Customers

LEFT JOIN Orders

ON Customers.CustomerID=Orders.CustomerID

LEFT JOIN [Order Details]

ON orders.OrderID=[Order Details].OrderID

GROUP BY Customers.Country

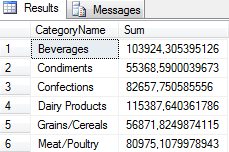
ORDER BY Customers.Country ASC

1. **\*Show the list of product categories along with total ordering sums calculated for the orders made for the products of each category, during the year 1997.**

SELECT CategoryName, SUM ([Order Details].UnitPrice \*[Order Details].Quantity\*(1-[Order Details].Discount))AS [Sum]

FROM Customers

LEFT JOIN Orders

ON Customers.CustomerID=Orders.CustomerID

LEFT JOIN [Order Details]

ON orders.OrderID=[Order Details].OrderID

LEFT JOIN Products

ON [Order Details].ProductID=Products.ProductID

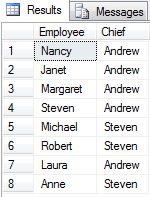
LEFT JOIN Categories

ON Categories.CategoryID=Products.CategoryID

WHERE YEAR(OrderDate)='1997'

GROUP BY Categories.CategoryName

ORDER BY Categories.CategoryName ASC

1. **\*Show the list of employees’ names along with names of their chiefs (use left join with the same table).**

SELECT EmployeeName.FirstName AS Employee, Chief.FirstName AS Chief

FROM Employees AS EmployeeName

LEFT JOIN Employees AS Chief

ON EmployeeName.ReportsTo=Chief.EmployeeID

WHERE Chief.FirstName IS NOT NULL

1. **\*Show the list of cities where employees and customers are from and where orders have been made to. Duplicates should be eliminated.**

SELECT DISTINCT City

FROM Employees

UNION

SELECT DISTINCT City

FROM Customers

UNION

SELECT DISTINCT ShipCity

FROM Orders

1. **\*Insert 5 new records into Employees table. Fill in the following fields: LastName, FirstName, BirthDate, HireDate, Address, City, Country, Notes. The Notes field should contain your own name (to distinguish your records from the ones inserted by other trainees).**

Insert into Employees

(LastName, FirstName, BirthDate, HireDate, [Address], City, Country, Notes)

values

('Oliver', 'Andrey', 1987-03-25 00:00:00.000, 2015-03-25 00:00:00.000, '55 Franko str', 'Ivano-Frankivsk', 'UKR','Nataliia Sheludiakova')

Insert into Employees

(LastName, FirstName, BirthDate, HireDate, [Address], City, Country, Notes)

values

('Garashuk', 'Oleg', 1987-04-26 00:00:00.000, 2015-04-26 00:00:00.000, '55 Stussa str', 'Ivano Frankivsk', 'UKR','Nataliia Sheludiakova');

Insert into Employees

(LastName, FirstName, BirthDate, HireDate, [Address], City, Country, Notes)

values

('Gavrylko', 'Anna', 1987-05-27 00:00:00.000, 2015-05-27 00:00:00.000, '55 Era str', 'Ivano-Frankivsk', 'UKR','Nataliia Sheludiakova');

Insert into Employees

(LastName, FirstName, BirthDate, HireDate, [Address], City, Country, Notes)

values

('Ivanyshyn', 'Oleg', 1987-06-27 00:00:00.000, 2015-06-28 00:00:00.000, '55 Kostenko str', 'Ivano-Frankivsk', 'UKR','Nataliia Sheludiakova');

Insert into Employees

(LastName, FirstName, BirthDate, HireDate, [Address], City, Country, Notes)

values

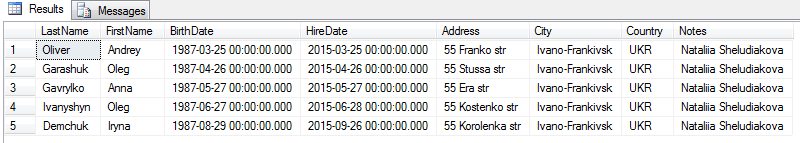
('Demchuk', 'Iryna', 1987-08-29 00:00:00.000, 2015-09-26 00:00:00.000, '55 Korolenka str', 'Ivano-Frankivsk', 'UKR','Nataliia Sheludiakova');

1. **\*Fetch the records you have inserted by the SELECT statement**

SELECT LastName, FirstName, BirthDate, HireDate, [Address], City, Country, Notes

FROM Employees

WHERE Notes Like 'Nataliia Sheludiakova'



1. **\*Change the City field in one of your records using the UPDATE statement (first run the SELECT statement to check whether you are updating the appropriate records!).**

update Employees

set City='Kolomyia'

where LastName='Demchuk'

C:\Users\Mike\Desktop\SoftServe\Attachments\2016-02-04_13-00_Microsoft SQL Server.jpg

1. **\*Change the HireDate field in all your records to current date (first run the SELECT statement to check whether you are updating the appropriate records!).**

UPDATE Employees

SET HireDate=GETDATE()

WHERE EmployeeID=10;

UPDATE Employees

SET HireDate=GETDATE()

WHERE EmployeeID=11;

UPDATE Employees

SET HireDate=GETDATE()

WHERE EmployeeID=12;

UPDATE Employees

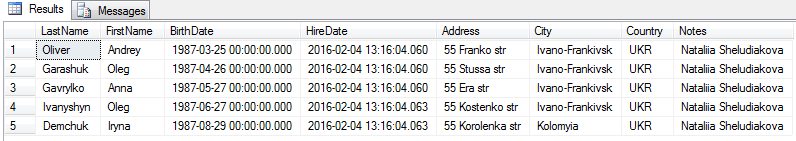
SET HireDate=GETDATE()

WHERE EmployeeID=13;

UPDATE Employees

SET HireDate=GETDATE()

WHERE EmployeeID=14;



1. **\*Delete one of your records (first run the SELECT statement to check whether you are deleting the appropriate record!).**

SELECT LastName, FirstName, BirthDate, HireDate, [Address], City, Country, Notes

FROM Employees

DELETE FROM Employees

WHERE LastName='Oliver'

