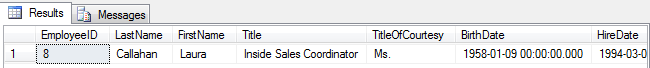
**Examples of tasks on SQL:**

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

SELECT EmployeeID, LastName, FirstName, Title, TitleOfCourtesy, BirthDate, HireDate, Address, City, Region, PostalCode, Country, HomePhone, Extension, Photo, Notes, ReportsTo, PhotoPath FROM employees

WHERE EmployeeID = 8;

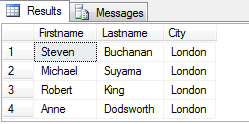


1. **Show the list of first and last names of the employees from London.**

SELECT Firstname, Lastname, City

FROM employees

WHERE city = 'London';

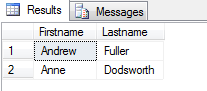


1. **Show the list of first and last names of the employees whose first name begins with letter A.**

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.**

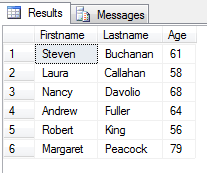
SELECT Firstname, Lastname,

DATEDIFF (YYYY,BirthDate,GETDATE()) AS Age

FROM employees

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

ORDER BY Lastname;

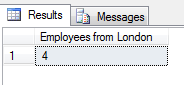


1. **Calculate the count of employees from London.**

SELECT COUNT (\*) AS 'Employees from London'

FROM employees

WHERE city = 'London';



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

SELECT City,

MAX (DATEDIFF(YYYY,BirthDate,GETDATE())) AS MaxAge,

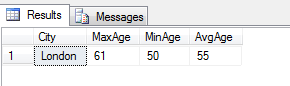
MIN (DATEDIFF(YYYY,BirthDate,GETDATE())) AS MinAge,

AVG (DATEDIFF(YYYY,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,

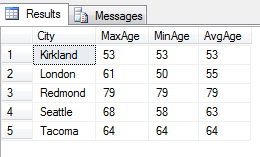
MAX (DATEDIFF(YYYY,BirthDate,GETDATE())) AS MaxAge,

MIN (DATEDIFF(YYYY,BirthDate,GETDATE())) AS MinAge,

AVG (DATEDIFF(YYYY,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)**

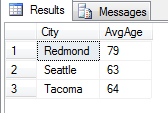
SELECT City,

AVG (DATEDIFF(YYYY,BirthDate,GETDATE())) AS AvgAge

FROM Employees

GROUP BY City

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



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

SELECT Firstname, Lastname

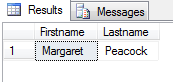
FROM employees

WHERE Birthdate = (

SELECT MIN (Birthdate)

FROM employees

);



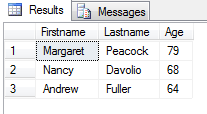
1. **Show first, last names and ages of 3 eldest employees.**

SELECT TOP 3 Firstname, Lastname,

DATEDIFF(YYYY,BirthDate,GETDATE()) AS Age

FROM employees

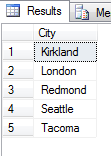
ORDER BY Birthdate ASC;



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

SELECT DISTINCT City

FROM Employees;

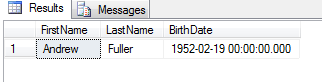


1. **Show first, last names and dates of birth of the employees who celebrate their birthdays this month.**

SELECT FirstName, LastName, BirthDate

FROM Employees

WHERE DATEPART(MM,BirthDate)= DATEPART(MM, GETDATE ());



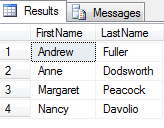
1. **Show first and last names of the employees who used to serve orders shipped to Madrid.**

SELECT DISTINCT FirstName, LastName

FROM Employees JOIN Orders

ON employees.EmployeeID = Orders.employeeID

WHERE 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).**

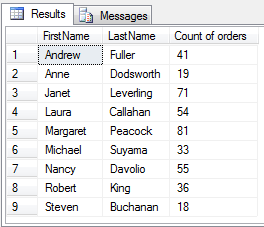
SELECT FirstName, LastName, COUNT(OrderID)AS'Count of orders'

FROM Employees

LEFT JOIN Orders ON employees.EmployeeID = Orders.employeeID

WHERE DATEPART(YYYY,OrderDate) = '1997'

GROUP BY LastName, FirstName;



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 a subquery).**

SELECT Employees.FirstName, Employees.LastName,

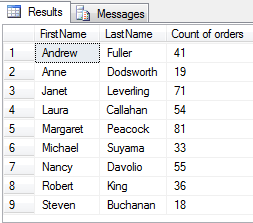
(SELECT COUNT(Orders.OrderID)

FROM Orders WHERE DATEPART(YYYY,OrderDate) = '1997'

AND employees.EmployeeID = Orders.employeeID)AS'Count of orders'

FROM Employees

ORDER BY Employees.FirstName, Employees.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).**

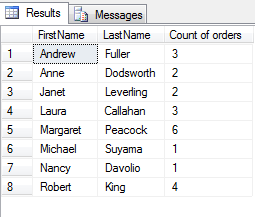
SELECT FirstName, LastName, COUNT(OrderID)AS'Count of orders'

FROM Employees LEFT JOIN Orders

ON employees.EmployeeID = Orders.employeeID

WHERE DATEPART(YYYY,OrderDate) = '1997' AND ShippedDate > RequiredDate

GROUP BY LastName, FirstName;



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

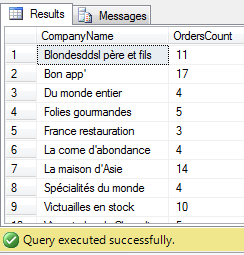
SELECT CompanyName, COUNT(OrderID) AS OrdersCount

FROM Customers, Orders

WHERE Customers.CustomerID=Orders.CustomerID

AND Country = 'France'

GROUP BY CompanyName



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

SELECT ContactName, COUNT(OrderID) AS OrdersCount

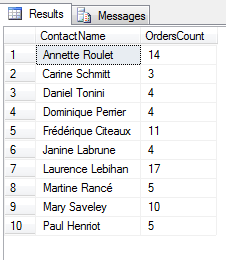
FROM Customers, Orders

WHERE Customers.CustomerID=Orders.CustomerID

AND Country = 'France'

GROUP BY ContactName

HAVING COUNT(OrderID) > 1



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

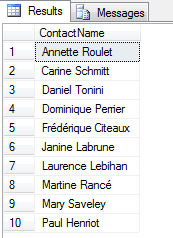
SELECT ContactName, (SELECT COUNT(OrderID)FROM Orders WHERE Customers.CustomerID=Orders.CustomerID) AS 'Orders'

FROM Customers

WHERE Country = 'France' AND

(SELECT COUNT(OrderID)FROM Orders

WHERE Customers.CustomerID=Orders.CustomerID) > 1



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

SELECT ContactName

FROM Customers

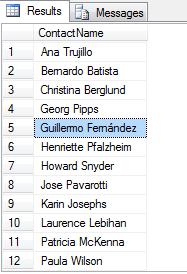
WHERE CustomerID IN

(SELECT CustomerID FROM [Orders] WHERE OrderID IN

(SELECT OrderID FROM [Order Details] WHERE ProductID IN

(SELECT ProductID FROM Products WHERE ProductName = 'Tofu')))

GROUP 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 customers.ContactName, SUM([Order Details].quantity\*[Order Details].UnitPrice) AS TotalSum,

SUM([Order Details].quantity) AS TotalAmount

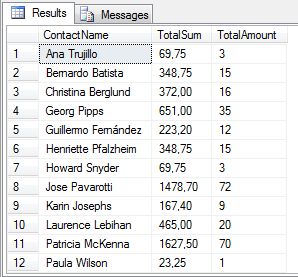
FROM customers JOIN orders ON customers.customerid=orders.CustomerID

JOIN [Order Details] ON Orders.orderid=[Order Details].orderid

JOIN products ON products.ProductID=[Order Details].ProductID

WHERE products.productname='tofu'

GROUP BY customers.ContactName;



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

SELECT DISTINCT Customers.ContactName

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 Suppliers ON Products.SupplierID = Suppliers.SupplierID

WHERE Customers.Country = 'France' AND Suppliers.Country <> 'France';



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

SELECT DISTINCT Customers.ContactName

FROM Customers

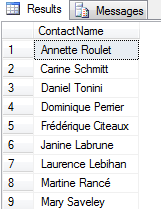
WHERE Country = 'France' AND CustomerID IN

(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 DISTINCT Customers.ContactName

FROM Customers

WHERE Country = 'France' AND 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' ))))

SELECT DISTINCT Customers.ContactName

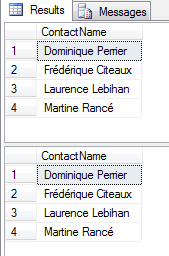
FROM Customers LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID

LEFT JOIN [Order details] ON Orders.OrderID = [Order details].OrderID

LEFT JOIN Products ON [Order details].ProductID = Products.ProductID

LEFT JOIN Suppliers ON Products.SupplierID = Suppliers.SupplierID

WHERE Customers.Country = 'France' AND Suppliers.Country = 'France';



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

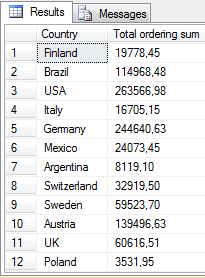
SELECT Customers.Country, SUM ([Order Details].UnitPrice\*[Order Details].Quantity) AS 'Total ordering sum'

FROM Customers

JOIN Orders ON Customers.CustomerID = Orders.CustomerID

JOIN [Order details] ON Orders.OrderID = [Order details].OrderID

GROUP BY Country;



1. \*Show the total ordering sums calculated for each customer’s country for domestic and non-domestic products separately (e.g.: “France – French products ordered – Non-french products ordered” and so on for each country).

SELECT nd.Country, d.DomesticSum, nd.NonDomesticSum

FROM (SELECT Customers.Country, SUM([Order Details].UnitPrice\*[Order Details].Quantity) AS DomesticSum

FROM Customers JOIN Orders

ON Customers.CustomerID=Orders.CustomerID JOIN [Order Details]

ON Orders.Orderid=[Order details].Orderid JOIN Products

ON [Order details].ProductID=Products.ProductID JOIN Suppliers

ON Products.SupplierID=Suppliers.SupplierID

WHERE Customers.Country=Suppliers.Country

GROUP BY Customers.Country) d

FULL JOIN (SELECT Customers.Country, SUM([Order Details].UnitPrice\*[Order Details].Quantity) AS NonDomesticSum

FROM Customers JOIN Orders

ON Customers.CustomerID=Orders.CustomerID JOIN [Order Details]

ON Orders.Orderid=[Order details].Orderid JOIN Products

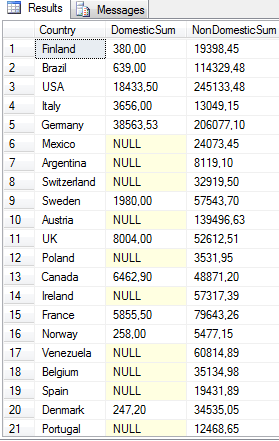
ON [Order details].ProductID=Products.ProductID JOIN Suppliers

ON Products.SupplierID=Suppliers.SupplierID

WHERE Customers.Country<>Suppliers.Country

GROUP BY Customers.Country) nd

ON d.Country=nd.Country;



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(ROUND([Order Details].unitprice \* [order details].quantity \* (1-[Order Details].Discount),2))

AS 'Total Ord\_Sum'

FROM Categories JOIN Products

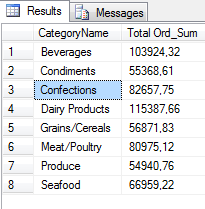
ON Categories.CategoryID=Products.CategoryID JOIN [Order Details]

ON Products.ProductID=[Order Details].ProductID JOIN Orders

ON [Order Details].OrderID=Orders.OrderID

WHERE Year(Orders.OrderDate) = 1997

GROUP BY CategoryName;



1. \*Show the list of product names along with unit prices and the history of unit prices taken from the orders (show ‘Product name – Unit price – Historical price’). The duplicate records should be eliminated. If no orders were made for a certain product, then the result for this product should look like ‘Product name – Unit price – NULL’. Sort the list by the product name.

SELECT DISTINCT ProductName, Products.UnitPrice,

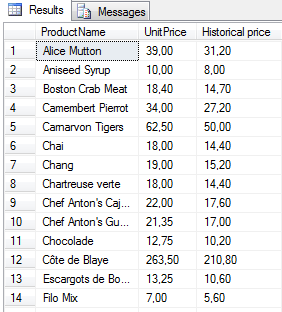
[Order Details].UnitPrice AS 'Historical price'

FROM Products LEFT JOIN [Order Details]

ON Products.ProductID = [Order Details].ProductID

WHERE Products.UnitPrice <> [Order Details].UnitPrice

ORDER BY ProductName;



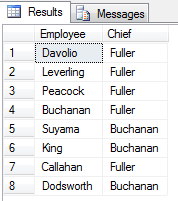
1. \*Show the list of employees’ names along with names of their chiefs (use left join with the same table).

SELECT e1. LastName AS Employee, e2. LastName AS Chief

FROM Employees e2

LEFT JOIN Employees e1 ON e2.EmployeeID=e1.ReportsTo

WHERE e1.LastName 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)

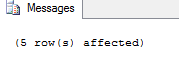
VALUES ('Ivantsiv','Nazar','1988-03-22','2016-12-28','Urozhajna','IF','Ukraine', 'Ivantsiv'),

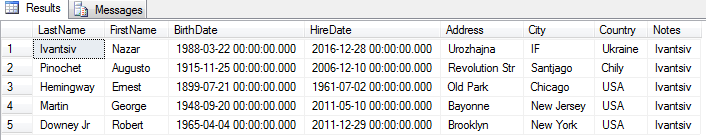
('Pinochet','Augusto','1915-11-25','2006-12-10','Revolution Str','Santjago', 'Chily','Ivantsiv'),

('Hemingway','Ernest','1899-07-21','1961-07-02','Old Park','Chicago','USA', 'Ivantsiv'),

('Martin','George','1948-09-20','2011-05-10','Bayonne','New Jersey','USA', 'Ivantsiv'),

('Downey Jr','Robert','1965-04-04','2011-12-29','Brooklyn','New York','USA', 'Ivantsiv');



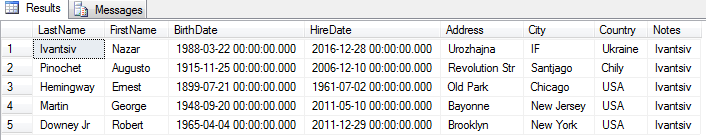


1. \*Fetch the records you have inserted by the SELECT statement

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

FROM Employees

WHERE Notes LIKE 'Ivantsiv';

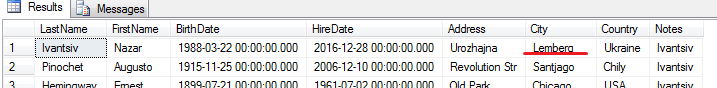


1. \*Change the City field in one of your records using the UPDATE statement (first run the SELECT statement to check whether you are updating the appropriate records!).

UPDATE Employees

SET City = 'Lemberg'

WHERE EmployeeID = 14;

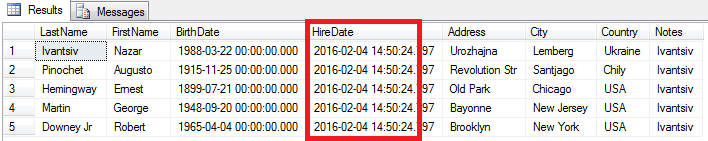


1. \*Change the HireDate field in all your records to current date (first run the SELECT statement to check whether you are updating the appropriate records!).

UPDATE Employees

SET HireDate = GETDATE()

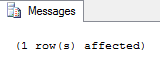
WHERE Notes LIKE 'Ivantsiv';

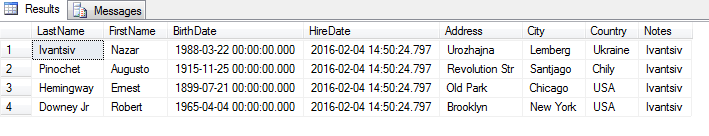


1. \*Delete one of your records (first run the SELECT statement to check whether you are deleting the appropriate record!).

DELETE FROM Employees

WHERE LastName = 'Martin' AND Notes LIKE 'Ivantsiv';





# Home works

1. Write SQL statements for the examples marked with (\*) above.
2. Write you own 20 examples covering all types of queries mentioned in the lecture that are cases: where, like, order by, group by, count(\*), count(<field>), sum, having, order by, inner join, left (right) join, subquery; delete, insert (+results of select), update

**Cautions**:

1. When using INSERT commands, fill some text field with your own name.
2. When using UPDATE or DELETE commands, only your records should be affected.

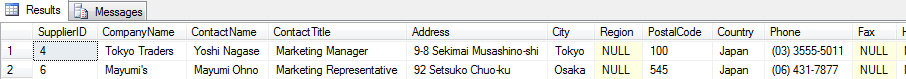
**20 examples**

1. Show all information about the suppliers from Japan (**WHERE**).

SELECT \*

FROM Suppliers

WHERE Country LIKE 'Japan';

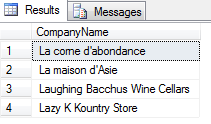


1. Show all customers’ companies with starts with letters “la” (**LIKE**).

SELECT CompanyName

FROM Customers

WHERE CompanyName LIKE 'la%';



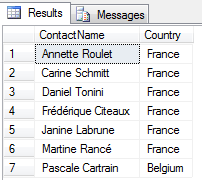
1. Show the list of contact names, countries from customers who have postal code with ‘000’. The result should be sorted by names (**ORDER BY**).

SELECT ContactName, Country

FROM Customers

WHERE PostalCode LIKE '%000%'

ORDER BY ContactName;



1. Show the list of countries (from where the products are supplied ) with amount of products. Group (**GROUP BY**).

SELECT Country, COUNT (\*) AS 'Products amount'

FROM Products, Suppliers

WHERE Products.SupplierID=Suppliers.SupplierID

GROUP BY Country



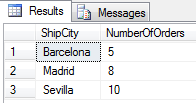
1. Show the amount of orders from Madrid or all Spain. (**COUNT(\*)**)

SELECT ShipCity, COUNT(\*) AS NumberOfOrders

FROM Orders

WHERE ShipCity='Madrid' OR ShipCountry = 'Spain'

GROUP BY ShipCity



1. Show first and last names of the employees as well as the count of orders each of them have received during the year 1996 (**LEFT JOIN and COUNT (field)**).

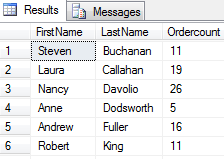
SELECT FirstName, LastName, COUNT (OrderID) AS Ordercount

FROM Employees LEFT JOIN Orders

ON Orders.EmployeeID=Employees.EmployeeID

WHERE year(OrderDate)=1996

GROUP BY FirstName, LastName



1. Show the total freight of ship ‘Tortuga Restaurante’ shipped to Mexico from Orders table.

SELECT SUM(Freight) AS Total, ShipName

FROM Orders

WHERE ShipName LIKE 'tor%' AND ShipCountry LIKE 'Mexico'

GROUP BY ShipName

