## SELECT m.FirstName + ' ' + m.LastName AS [Full name],

# m.Salary

# FROM Employees As m

# WHERE m.Salary = (SELECT MIN(Salary) FROM Employees)

## SELECT m.FirstName + ' ' + m.LastName AS [Full name],

# m.Salary

# FROM Employees As m

# WHERE m.Salary < (

# SELECT MIN(Salary)\*1.1 FROM Employees)

# ORDER BY m.Salary

# Order by added by me for better reading of result

## SELECT m.FirstName + ' ' + m.LastName AS [Full name],

# m.Salary,

# DepartmentID

# FROM Employees As m

# WHERE m.Salary = (

# SELECT MIN(Salary)

# FROM Employees

# WHERE DepartmentID = m.DepartmentID)

## SELECT AVG(Salary) AS [Average salary in Department #1]

# FROM Employees As e

# WHERE e.DepartmentID = 1

## SELECT AVG(e.Salary) AS [Average salary]

# FROM Employees As e

# INNER JOIN Departments As d

# ON e.DepartmentID = d.DepartmentID

# WHERE d.Name = 'Sales'

## SELECT COUNT(\*) AS [Salesmans]

# FROM Employees As e

# INNER JOIN Departments As d

# ON e.DepartmentID = d.DepartmentID

# WHERE d.Name = 'Sales'

## SELECT COUNT(\*) AS [Workers]

# FROM Employees As e

# INNER JOIN Employees As m

# ON e.ManagerID = m.EmployeeID

## SELECT COUNT(\*) AS [Salesmans]

# FROM Employees As e

# WHERE e.ManagerID IS NULL

## SELECT AVG(Salary) AS [Average salary],

# m.DepartmentID AS [Department]

# FROM Employees AS m

# GROUP BY DepartmentID

# ORDER BY [Average salary]

## SELECT COUNT(\*) AS [Workers], a.TownID, e.DepartmentID

# FROM Employees AS e

# INNER JOIN Addresses AS a

# ON a.AddressID = e.AddressID

# GROUP BY a.TownID, e.DepartmentID

## SELECT COUNT(\*) AS [Sherpas], m.FirstName + ' ' + m.LastName AS [Boss full name]

# FROM Employees AS e

# INNER JOIN Employees AS m

# ON m.EmployeeID = e.ManagerID

# GROUP BY m.FirstName, m.LastName

# HAVING COUNT(\*) = 5

## SELECT e.FirstName AS [Employee],

# COALESCE(m.FirstName, 'no manager') AS [Manager]

# FROM Employees AS e

# LEFT JOIN Employees AS m

# ON e.ManagerID = m.EmployeeID

# **Version with ISNULL**

# SELECT e.FirstName AS [Employee],

# ISNULL(m.FirstName, 'no manager') AS [Manager]

# FROM Employees AS e

# LEFT JOIN Employees AS m

# ON e.ManagerID = m.EmployeeID

## SELECT e.FirstName + ' ' + e.LastName AS [Full name]

# FROM Employees AS e

# WHERE LEN(e.LastName) = 5

## SELECT CONVERT(VARCHAR(24),m.HireDate,113) AS [Strnage date/time]

# FROM Employees AS m

## CREATE TABLE Users(

# UserID int IDENTITY,

# UserName nvarchar(20) NOT NULL UNIQUE,

# UserPassword nvarchar(20) NOT NULL CHECK(LEN(UserPassword) >= 5),

# FullName nvarchar(50),

# LastLogin date,

# CONSTRAINT PK\_Users PRIMARY KEY (UserID),

# )

# GO

## CREATE VIEW [User logged today] AS

# SELECT \* FROM Users AS u

# WHERE u.LastLogin = { fn CURDATE() }

## CREATE TABLE Groups(

# GroupID int IDENTITY,

# GroupName nvarchar(20) NOT NULL UNIQUE,

# CONSTRAINT PK\_Groups PRIMARY KEY (GroupID),

# )

# GO

## ALTER TABLE Users

# ADD GroupID int

# ALTER TABLE Users

# ADD CONSTRAINT FK\_Users\_Groups

# FOREIGN KEY (GroupID)

# REFERENCES Groups(GroupID)

## INSERT INTO Groups

# Values ('Specials')

# INSERT INTO Users (UserName, UserPassword, FullName, LastLogin, GroupID)

# VALUES('novUser','staraParola','Moeto Ime',NULL,4)

## UPDATE Users

# SET UserName='Gosho'

# WHERE UserName='pesho'

## DELETE FROM Users

# WHERE UserName='Gosho'

## INSERT INTO Users (UserName, UserPassword, FullName, LastLogin, GroupID)

# SELECT LOWER(LEFT(e.FirstName,1)+e.LastName) AS [UserName],

# LOWER(LEFT(e.FirstName,1)+e.LastName) AS [UserPassword],

# e.FirstName + ' ' + e.LastName AS [FullName],

# Null AS LastLogin,

# Null AS GroupID

# FROM Employees AS e

# WHERE LEN(e.LastName) > 5

## UPDATE Users

# SET LastLogin = NULL

# WHERE LastLogin < '20100310'

## DELETE FROM Users

# WHERE UserPassword IS NULL

## SELECT AVG(Salary) AS [Average salary], e.DepartmentID, e.JobTitle

# FROM Employees AS e

# GROUP BY e.DepartmentID, e.JobTitle

# ORDER BY e.DepartmentID

## SELECT MIN(e.Salary) AS [Minimal salary], e.DepartmentID, e.JobTitle--, e.FirstName

# FROM Employees AS e

# WHERE EXISTS

# (SELECT EmployeeID

# FROM Employees As m

# WHERE m.DepartmentID = e.DepartmentID

# AND m.JobTitle = e.JobTitle)

# GROUP BY e.DepartmentID, e.JobTitle

# ORDER BY e.DepartmentID

## SELECT COUNT(e.EmployeeID), t.Name AS [Town]

# FROM Employees As e

# JOIN Addresses AS a

# ON e.AddressID = a.AddressID

# JOIN Towns AS t

# ON a.TownID = t.TownID

# GROUP BY t.Name

# HAVING COUNT(e.EmployeeID) =

# (SELECT MAX(c)

# FROM (SELECT t.Name, COUNT(e.EmployeeID) AS c

# FROM Employees AS e

# JOIN Addresses AS a

# ON e.AddressID = a.AddressID

# JOIN Towns AS t

# ON a.TownID = t.TownID

# GROUP BY t.Name) AS t)

## SELECT COUNT(m.FirstName) AS [Managers], t.Name [Town]

# FROM Employees AS m

# JOIN Addresses AS a

# ON m.AddressID = a.AddressID

# JOIN Towns AS t

# ON a.TownID = t.TownID

# GROUP BY t.Name

## CREATE TABLE WorkHours(

# WorkHourID int IDENTITY,

# EmployeeID int NOT NULL,

# WorkHoursDate date,

# Task nvarchar(30),

# Hours time,

# Comments text,

# CONSTRAINT PK\_WorkHour PRIMARY KEY (WorkHourID),

# CONSTRAINT FK\_WorkHours\_Employees

# FOREIGN KEY (EmployeeID)

# REFERENCES Employees(EmployeeID)

# )

# GO

## BEGIN TRAN

# DELETE FROM

# Employees AS m

# INNER JOIN Departments As d

# ON e.DepartmentID = d.DepartmentID

# WHERE d.Name = 'Sales'

# ROLLBACK TRAN

# GO