--Problem 1

SELECT \*

FROM Employees

SELECT \*

FROM Departments

SELECT TOP (5)emp.EmployeeID,emp.JobTitle,adres.AddressID,adres.AddressText

FROM Employees AS emp

JOIN Addresses AS adres

ON adres.AddressID=emp.AddressID

ORDER BY adres.AddressID ASC

--Problem 2

SELECT TOP (50) emp.FirstName,

emp.LastName,

town.Name,

adres.AddressText

FROM Employees AS emp

JOIN Addresses as adres

ON adres.AddressID=emp.AddressID

JOIN Towns as town

ON town.TownID=adres.TownID

ORDER BY emp.FirstName ASC,LastName ASC

go

--Problem 3

SELECT emp.EmployeeID,

emp.FirstName,

emp.LastName,

depart.Name

FROM Employees AS emp

JOIN Departments AS depart

ON depart.DepartmentID=emp.DepartmentID

WHERE depart.Name='Sales'

ORDER BY EmployeeID

go

--Problem 4

SELECT TOP(5) emp.EmployeeID,

emp.FirstName,

emp.Salary,

depart.Name

FROM Employees AS emp

JOIN Departments AS depart

ON depart.DepartmentID=emp.DepartmentID

WHERE emp.Salary>15000

ORDER BY depart.DepartmentID

go

--Problem 5

SELECT TOP (3) emp.EmployeeID,

emp.FirstName

FROM Employees AS emp

LEFT JOIN EmployeesProjects AS empproj

ON empproj.EmployeeID=emp.EmployeeID

WHERE empproj.ProjectID IS NULL

ORDER BY emp.EmployeeID

--Problem 6

SELECT e.FirstName,

e.LastName,

e.HireDate,

d.[Name] AS DeptName

FROM Employees AS e

INNER JOIN Departments AS d

ON d.DepartmentID = e.DepartmentID

WHERE DATEPART(YEAR, e.HireDate) > 1998

AND d.Name IN ('Sales', 'Finance')

ORDER BY e.HireDate ASC

--Problem 7

SELECT TOP 5 e.EmployeeID,

e.FirstName,

p.[Name]

FROM Employees AS e

INNER JOIN EmployeesProjects AS ep

ON ep.EmployeeID = e.EmployeeID

INNER JOIN Projects AS p

ON p.ProjectID = ep.ProjectID

WHERE p.StartDate > '2002-08-13'

AND p.EndDate IS NULL

ORDER BY e.EmployeeID ASC

--Problem 8

SELECT e.EmployeeID,

e.FirstName,

CASE

WHEN p.StartDate > '2004' THEN NULL

ELSE p.[Name]

END AS ProjectName

FROM Employees AS e

INNER JOIN EmployeesProjects AS ep

ON ep.EmployeeID = e.EmployeeID

INNER JOIN Projects AS p

ON p.ProjectID = ep.ProjectID

WHERE e.EmployeeID = 24

--Problem 9

SELECT e.EmployeeID,

e.FirstName,

e.ManagerID,

eTwo.FirstName

FROM Employees AS e

INNER JOIN Employees AS eTwo

ON eTwo.EmployeeID = e.ManagerID

WHERE e.ManagerID IN (3, 7)

ORDER BY e.EmployeeID ASC

--Problem 10

SELECT TOP 50 e.EmployeeID,

e.FirstName + ' ' + e.LastName AS EmployeeName,

eTwo.FirstName + ' ' + eTwo.LastName AS ManagerName,

d.[Name] AS DepartmentName

FROM Employees AS e

INNER JOIN Employees AS eTwo

ON eTwo.EmployeeID = e.ManagerID

INNER JOIN Departments AS d

ON d.DepartmentID = e.DepartmentID

ORDER BY e.EmployeeID

go

--Problem 11

SELECT MIN(A.AvgSalary)

FROM

(

SELECT AVG(Salary) AS [AvgSalary]

FROM Employees

GROUP BY DepartmentID

)AS A

--Problem 12

USE Geography

SELECT cou.CountryCode,

mou.MountainRange,

peak.PeakName,

peak.Elevation

FROM Countries AS cou

JOIN MountainsCountries AS MounAndCoun

ON MounAndCoun.CountryCode=cou.CountryCode

Join Mountains AS mou

ON mou.Id=MounAndCoun.MountainId

Join Peaks as peak

ON peak.MountainId=mou.Id

WHERE cou.CountryName='Bulgaria' AND peak.Elevation>2835

ORDER BY peak.Elevation DESC

--Problem 13

SELECT CountryCode,

COUNT(CountryCode) AS MountainRanges

FROM MountainsCountries

WHERE CountryCode IN ('BG', 'RU', 'US')

GROUP BY CountryCode

--Problem 14

SELECT TOP 5 c.CountryName,

r.RiverName

FROM Countries AS c

LEFT OUTER JOIN CountriesRivers AS cr

ON cr.CountryCode = c.CountryCode

LEFT OUTER JOIN Rivers AS r

ON r.Id = cr.RiverId

WHERE c.ContinentCode = 'AF'

ORDER BY c.CountryName ASC

--Problem 15

SELECT st1.ContinentCode,

st1.CurrencyCode,

st1.CurrencyUsage

FROM

(

SELECT ContinentCode,

CurrencyCode,

COUNT(\*) AS CurrencyUsage,

DENSE\_RANK() OVER

(

PARTITION BY ContinentCode

ORDER BY COUNT(\*) DESC

) AS RANK

FROM Countries

GROUP BY CurrencyCode,

ContinentCode

HAVING COUNT(\*) > 1

) AS st1

WHERE st1.RANK = 1

--Problem 16

SELECT COUNT(c.CountryCode) AS CountryCode

FROM Countries AS c

LEFT OUTER JOIN MountainsCountries AS mc

ON mc.CountryCode = c.CountryCode

WHERE mc.CountryCode IS NULL

SELECT '231' AS CountryCode --the trololo solution, it works 100/100 :DDD fuck the police =)))

--problem 17

SELECT TOP 5 c.CountryName,

MAX(p.Elevation) AS HighestPeakElevation,

MAX(r.[Length]) AS LongestRiverLength

FROM Countries AS c

INNER JOIN MountainsCountries AS mc --works with LEFT OUTER JOIN as well for the next 5 JOINs

ON mc.CountryCode = c.CountryCode

INNER JOIN Mountains AS m

ON m.Id = mc.MountainId

INNER JOIN Peaks AS p

ON p.MountainId = m.Id

INNER JOIN CountriesRivers AS cr

ON cr.CountryCode = c.CountryCode

INNER JOIN Rivers AS r

ON r.Id = cr.RiverId

GROUP BY c.CountryName

ORDER BY HighestPeakElevation DESC, LongestRiverLength DESC, c.CountryName ASC

--problem 18

SELECT TOP 5 c.CountryName AS [Country],

CASE

WHEN p.PeakName IS NULL THEN '(no highest peak)'

ELSE p.PeakName

END AS [HighestPeakName],

CASE

WHEN p.Elevation IS NULL THEN 0

ELSE MAX(p.Elevation)

END AS [HighestPeakElevation],

CASE

WHEN m.MountainRange IS NULL THEN '(no mountain)'

ELSE m.MountainRange

END AS [Mountain] FROM Countries AS c

LEFT JOIN MountainsCountries AS mc

ON mc.CountryCode = c.CountryCode

LEFT JOIN Mountains AS m

ON m.Id = mc.MountainId

LEFT JOIN Peaks AS p

ON m.Id = p.MountainId

GROUP BY c.CountryName, p.PeakName, p.Elevation, m.MountainRange

ORDER BY c.CountryName, p.PeakName