06. DB-Basics-Subqueries-and-JOINs-Exercises

/\* 01EmployeeAddress\*/

SELECT TOP 5 e.EmployeeID, e.JobTitle, a.AddressID, a.AddressText

FROM Employees AS e

INNER JOIN Addresses AS a

ON e.AddressID = a.AddressID

ORDER BY AddressID

/\* 02AddressesWithTowns\*/

SELECT TOP(50) e.FirstName, e.LastName, t.[Name] AS Town, a.AddressText

FROM Employees AS e

INNER JOIN Addresses AS a ON e.AddressID = a.AddressID

INNER JOIN Towns AS t ON t.TownID = a.TownID

ORDER BY e.FirstName, e.LastName

/\* 03SalesEmployee\*/

SELECT e.EmployeeID, e.FirstName, e.LastName, d.[Name] AS DepartmentName

FROM Employees AS e

JOIN Departments AS d

ON d.DepartmentID = e.DepartmentID

WHERE d.[Name] = 'Sales'

ORDER BY e.EmployeeID

--Only one query must be paste in Judge

SELECT e.EmployeeID, e.FirstName, e.LastName, d.[Name] AS DepartmentName

FROM Employees AS e

JOIN Departments AS d

ON (d.DepartmentID = e.DepartmentID

AND d.[Name] = 'Sales')

ORDER BY e.EmployeeID

/\* 04EmployeeDepartments\*/

SELECT TOP(5) e.EmployeeID, e.FirstName, e.Salary, d.[Name]

FROM Employees AS e

INNER JOIN Departments AS d

ON d.DepartmentID = e.DepartmentID

WHERE e.Salary > 15000

ORDER BY e.DepartmentID ASC

/\* 05EmployeesWithoutProject\*/

SELECT TOP(3) e.EmployeeID, e.FirstName

FROM Employees AS e

LEFT JOIN EmployeesProjects AS e\_p ON e\_p.EmployeeID = e.EmployeeID

WHERE e\_p.ProjectID IS NULL

ORDER BY e.EmployeeID

/\* 06EmployeesHiredAfter\*/

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 e.HireDate > '1/1/1999' AND d.[Name] IN ('Sales', 'Finance')

ORDER BY e.HireDate

--Only one query must be paste in Judge

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

AND e.HireDate > '1/1/1999' AND d.[Name] IN ('Sales', 'Finance'))

ORDER BY e.HireDate

/\* 07EmployeesWithProject\*/

SELECT TOP(5) e.EmployeeID, e.FirstName, p.[Name] AS ProjectName

FROM Employees AS e

INNER JOIN EmployeesProjects AS e\_p ON e\_p.EmployeeID = e.EmployeeID

INNER JOIN Projects AS p ON p.ProjectID = e\_p.ProjectID

WHERE p.StartDate > '08/13/2002' AND p.EndDate IS NULL

ORDER BY e.EmployeeID

/\* 08Employee24\*/

SELECT e.EmployeeID, e.FirstName,

CASE

WHEN p.StartDate >= '01/01/2005' THEN NULL

ELSE p.[Name]

END AS ProjectName

FROM Employees AS e

JOIN EmployeesProjects AS e\_p ON e\_p.EmployeeID = e.EmployeeID

JOIN Projects AS p ON p.ProjectID = e\_p.ProjectID

WHERE e.EmployeeID = 24

/\* 09EmployeeManager\*/

SELECT e.EmployeeID, e.FirstName, e.ManagerID, m.FirstName AS ManagerName

FROM Employees AS e

JOIN Employees AS m ON m.EmployeeID = e.ManagerID

WHERE e.ManagerID IN(3, 7)

ORDER BY EmployeeID

/\* 10EmployeeSumary\*/

SELECT TOP(50) e.EmployeeID,

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

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

d.[Name] AS DepartmentName

FROM Employees AS e

LEFT JOIN Employees AS m ON m.EmployeeID = e.ManagerID

LEFT JOIN Departments AS d ON d.DepartmentID = e.DepartmentID

ORDER BY e.EmployeeID

/\* 11.MinAverageSalary\*/

SELECT MIN(avgs.AverageSalary) AS MinAverageSalary

FROM

(

SELECT e.DepartmentID, AVG(e.Salary) AS AverageSalary

FROM Employees AS e

GROUP BY e.DepartmentID

) AS avgs

/\* 12HighestPeaksInBulgaria\*/

SELECT mc.CountryCode, m.MountainRange, p.PeakName, p.Elevation

FROM Mountains AS m

JOIN MountainsCountries AS mc ON (mc.MountainId = m.Id AND mc.CountryCode = 'BG')

JOIN Peaks AS p ON (p.MountainId = m.Id AND p.Elevation > 2835)

ORDER BY p.Elevation DESC

/\* 13CountMountainRanges\*/

SELECT c.CountryCode, COUNT(mc.MountainId) AS MountainRanges

FROM Countries AS c

JOIN MountainsCountries AS mc ON mc.CountryCode = c.CountryCode

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

GROUP BY c.CountryCode

--Only one query must be paste in Judge

SELECT mc.CountryCode, COUNT(m.MountainRange) AS MountainRanges

FROM Mountains AS m

JOIN MountainsCountries AS mc ON(mc.MountainId = m.Id AND mc.CountryCode IN('BG', 'RU', 'US'))

GROUP BY mc.CountryCode

--Only one query must be paste in Judge

SELECT CountryCode, COUNT(MountainId) AS MountainRanges

FROM MountainsCountries

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

GROUP BY CountryCode

/\* 14CountriesWithOrWithoutRivers\*/

SELECT TOP(5) c.CountryName, r.RiverName

FROM Countries AS c

LEFT JOIN CountriesRivers AS cr ON cr.CountryCode = c.CountryCode

LEFT JOIN Rivers AS r ON r.Id = cr.RiverId

WHERE c.ContinentCode = 'AF'

ORDER BY c.CountryName

--Only one query must be paste in Judge

SELECT TOP(5) c.CountryName, r.RiverName

FROM Countries AS c

LEFT JOIN CountriesRivers AS cr ON cr.CountryCode = c.CountryCode

LEFT JOIN Rivers AS r ON r.Id = cr.RiverId

WHERE c.ContinentCode = (SELECT ContinentCode FROM Continents WHERE ContinentName = 'Africa')

ORDER BY c.CountryName

/\* 15ContinentsandCurrencies\*/

WITH CTE\_CurrenciesInfo(ContinentCode, CurrencyCode, CurrencyUsage) AS

(

SELECT ContinentCode, CurrencyCode, COUNT(CurrencyCode) AS CurrencyUsage

FROM Countries

GROUP BY ContinentCode, CurrencyCode

HAVING COUNT(CurrencyCode) > 1

)

SELECT e.ContinentCode, cci.CurrencyCode, e.MaxCurrency AS CurrencyUsage

FROM

(

SELECT ContinentCode, MAX(CurrencyUsage) AS MaxCurrency

FROM CTE\_CurrenciesInfo

GROUP BY ContinentCode

) AS e

JOIN CTE\_CurrenciesInfo AS cci ON cci.ContinentCode = e.ContinentCode AND

cci.CurrencyUsage = e.MaxCurrency

/\* 16CountriesWithoutAnyMountains\*/

SELECT COUNT(\*) AS CountryCode

FROM Countries AS c

LEFT JOIN MountainsCountries AS mc ON mc.CountryCode = c.CountryCode

WHERE mc.MountainId IS NULL

--Only one query must be paste in Judge

SELECT COUNT(\*)

FROM Countries

WHERE CountryCode NOT IN (SELECT CountryCode FROM MountainsCountries)

/\* 17HighestPeakAndLongestRiverByCountry\*/

SELECT TOP(5) c.CountryName, MAX(p.Elevation) AS HighestPeakElevation,

MAX(r.[Length]) AS LongestRiverLength

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 p.MountainId = m.Id

LEFT JOIN CountriesRivers AS cr ON cr.CountryCode = c.CountryCode

LEFT JOIN Rivers AS r ON r.Id = cr.RiverId

GROUP BY c.CountryName

ORDER BY HighestPeakElevation DESC, LongestRiverLength DESC, c.CountryName

/\* 18HighestPeakNameAndElevationByCountry\*/

WITH CTE\_CountriesInfo(CountryName, PeakName, Elevation, Mountain) AS

(

SELECT c.CountryName, p.PeakName, MAX(p.Elevation), m.MountainRange

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 p.MountainId = mc.MountainId

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

)

SELECT TOP(5) e.CountryName AS Country,

ISNULL(cci.PeakName, '(no highest peak)') AS [Highest peak Name],

ISNULL(cci.Elevation, 0) AS [Highest Peak Elevation],

ISNULL(cci.Mountain, '(no mountain)')

FROM

(

SELECT CountryName, MAX(Elevation) AS MaxElevation

FROM CTE\_CountriesInfo

GROUP BY CountryName

) AS e

LEFT JOIN CTE\_CountriesInfo AS cci ON cci.CountryName = e.CountryName AND

cci.Elevation = e.MaxElevation

ORDER BY e.CountryName, cci.PeakName