# Exercises: Subqueries and Joins

This document defines the **exercise** assignments for the ["Databases Basics - MSSQL" course @ Software University](https://softuni.bg/modules/22/csharp-db-may-2024/1473).   
You can check your solutions in the [Judge system](https://judge.softuni.org/Contests/393/Subqueries-and-Joins).

# Part I – Queries for SoftUni Database

## Employee Address

Create a query that selects:

* **EmployeeId**
* **JobTitle**
* **AddressId**
* **AddressText**

Return the **first 5** rows **sorted** by **AddressId** in **ascending** order.

### Example

|  |  |  |  |
| --- | --- | --- | --- |
| **EmployeeId** | **JobTitle** | **AddressId** | **AddressText** |
| 142 | Production Technician | 1 | 108 Lakeside Court |
| 30 | Human Resources Manager | 2 | 1341 Prospect St |
| … | … | … | … |

## Addresses with Towns

Write a query that selects:

* **FirstName**
* **LastName**
* **Town**
* **AddressText**

**Sort them** by **FirstName** in **ascending** order, then by **LastName**. Select the **first 50** employees.

### Example

|  |  |  |  |
| --- | --- | --- | --- |
| **FirstName** | **LastName** | **Town** | **AddressText** |
| A.Scott | Wright | Newport Hills | 1400 Gate Drive |
| Alan | Brewer | Kenmore | 8192 Seagull Court |
| … | … | … | … |

## Sales Employee

Create a query that selects:

* **EmployeeID**
* **FirstName**
* **LastName**
* **DepartmentName**

**Sort them** by **EmployeeID** in **ascending** order. Select only **employees** from the "**Sales**" department.

### Example

|  |  |  |  |
| --- | --- | --- | --- |
| **EmployeeID** | **FirstName** | **LastName** | **DepartmentName** |
| 268 | Stephen | Jiang | Sales |
| 273 | Brian | Welcker | Sales |
| … | … | … | … |

## Employee Departments

Create a query that selects:

* **EmployeeID**
* **FirstName**
* **Salary**
* **DepartmentName**

Filter only **employees** with a **salary higher than 15000**. Return the **first 5** rows, **sorted** by **DepartmentID** in **ascending** order.

### Example

|  |  |  |  |
| --- | --- | --- | --- |
| **EmployeeID** | **FirstName** | **Salary** | **DepartmentName** |
| 3 | Roberto | 43300.00 | Engineering |
| 9 | Gail | 32700.00 | Engineering |
| … | … | … | … |

## Employees Without Project

Create a query that selects:

* **EmployeeID**
* **FirstName**

Filter only **employees** **without** a **project**. Return the **first 3** rows, **sorted** by **EmployeeID** in **ascending** order.

### Example

|  |  |
| --- | --- |
| **EmployeeID** | **FirstName** |
| 2 | Kevin |
| 6 | David |
| … | … |

## Employees Hired After

Create a query that selects:

* **FirstName**
* **LastName**
* **HireDate**
* **DeptName**

Filter only **employees** **hired after 1.1.1999** and are from either **"Sales"** or **"Finance"** department. **Sort them** by **HireDate** (**ascending**).

### Example

|  |  |  |  |
| --- | --- | --- | --- |
| **FirstName** | **LastName** | **HireDate** | **DeptName** |
| Debora | Poe | 2001-01-19 00:00:00 | Finance |
| Wendy | Kahn | 2001-01-26 00:00:00 | Finance |
| … | … | … | … |

## Employees with Project

Create a query that selects:

* **EmployeeID**
* **FirstName**
* **ProjectName**

Filter only **employees** **with** a **project** which has **started after 13.08.2002** and it is still **ongoing** (no end date). Return the **first 5** rows **sorted** by **EmployeeID** in **ascending** order.

### Example

|  |  |  |
| --- | --- | --- |
| **EmployeeID** | **FirstName** | **ProjectName** |
| 1 | Guy | Racing Socks |
| 1 | Guy | Road Bottle Cage |
| … | … | … |

## Employee 24

Create a query that selects:

* **EmployeeID**
* **FirstName**
* **ProjectName**

Filter all the **projects** of **employee** with **Id 24**. If the project has **started during or** **after** **2005** the **returned** value should be **NULL**.

### Example

|  |  |  |
| --- | --- | --- |
| **EmployeeID** | **FirstName** | **ProjectName** |
| 24 | David | NULL |
| 24 | David | Road-650 |
| … | … | … |

## Employee Manager

Create a query that selects:

* **EmployeeID**
* **FirstName**
* **ManagerID**
* **ManagerName**

Filter all **employees** with a **manager** who has **ID** equals to **3 or 7**. Return all the rows, **sorted** by **EmployeeID** in **ascending** order.

### Example

|  |  |  |  |
| --- | --- | --- | --- |
| **EmployeeID** | **FirstName** | **ManagerID** | **ManagerName** |
| 4 | Rob | 3 | Roberto |
| 9 | Gail | 3 | Roberto |
| … | … | … | … |

## Employees Summary

Create a query that selects:

* **EmployeeID**
* **EmployeeName**
* **ManagerName**
* **DepartmentName**

Show the **first 50 employees** with their **managers** and the **departments** they are in (show the departments of the employees). **Order** them by **EmployeeID**.

### Example

|  |  |  |  |
| --- | --- | --- | --- |
| **EmployeeID** | **EmployeeName** | **ManagerName** | **DepartmentName** |
| 1 | Guy Gilbert | Jo Brown | Production |
| 2 | Kevin Brown | David Bradley | Marketing |
| 3 | Roberto Tamburello | Terri Duffy | Engineering |
| … | … | … | … |

## Min Average Salary

Create a query that **returns** the value of the **lowest** **average** **salary** of all **departments**.

### Example

|  |
| --- |
| **MinAverageSalary** |
| 10866.6666 |

# Part II – Queries for Geography Database

## Highest Peaks in Bulgaria

Create a query that selects:

* **CountryCode**
* **MountainRange**
* **PeakName**
* **Elevation**

Filter all the **peaks** in **Bulgaria,** which have **elevation** **over** **2835**. **Return** all the rows, **sorted** by **elevation** in **descending** order.

### Example

|  |  |  |  |
| --- | --- | --- | --- |
| **CountryCode** | **MountainRange** | **PeakName** | **Elevation** |
| BG | Rila | Musala | 2925 |
| BG | Pirin | Vihren | 2914 |
| … | … | … | … |

## Count Mountain Ranges

Create a query that selects:

* **CountryCode**
* **MountainRanges**

Filter the **count** of the **mountain** **ranges** in the **United** **States**, **Russia** and **Bulgaria**.

### Example

|  |  |
| --- | --- |
| **CountryCode** | **MountainRanges** |
| BG | 6 |
| RU | 1 |
| … | … |

## Countries With or Without Rivers

Create a query that selects:

* **CountryName**
* **RiverName**

Find the **first** **5** **countries** with or without **rivers** in **Africa**. **Sort** them by **CountryName** in **ascending** order.

### Example

|  |  |
| --- | --- |
| **CountryName** | **RiverName** |
| Algeria | Niger |
| Angola | Congo |
| Benin | Niger |
| Botswana | NULL |
| Burkina Faso | Niger |

## \*Continents and Currencies

Create a query that selects:

* **ContinentCode**
* **CurrencyCode**
* **CurrencyUsage**

Find all **continents** and their **most** **used** **currency**. Filter any **currency,** which is used in **only** **one** **country**. **Sort** your results by **ContinentCode**.

### Example

|  |  |  |
| --- | --- | --- |
| **ContinentCode** | **CurrencyCode** | **CurrencyUsage** |
| AF | XOF | 8 |
| AS | AUD | 2 |
| AS | ILS | 2 |
| EU | EUR | 26 |
| NA | XCD | 8 |
| OC | USD | 8 |

## Countries Without Any Mountains

Create a query that returns the **count** of all **countries,** which **don’t** **have** a **mountain**.

### Example

|  |
| --- |
| **Count** |
| 231 |

## Highest Peak and Longest River by Country

For each country, find the elevation of **the highest peak** and **the length of the longest river**, **sorted** by the **highest peak elevation** (from highest to lowest), then by the **longest river length** (from longest to smallest), then by **country name** (alphabetically). Display **NULL** when no data is available in some of the columns. Limit only the **first 5** rows.

### Example

|  |  |  |
| --- | --- | --- |
| **CountryName** | **HighestPeakElevation** | **LongestRiverLength** |
| China | 8848 | 6300 |
| India | 8848 | 3180 |
| Nepal | 8848 | 2948 |
| Pakistan | 8611 | 3180 |
| Argentina | 6962 | 4880 |

## Highest Peak Name and Elevation by Country

For each country, find the **name**and **elevation** of **the highest peak**, along with its **mountain**. When no peaks are available in some countries, display elevation**0**, "**(no highest peak)**" as **peak name** and "**(no mountain)**" as **a mountain name**. When **multiple peaks** in some countries have the **same elevation**, display **all of them**. **Sort** the results by **country name alphabetically**, then by **highest peak name alphabetically**. Limit only the **first 5** rows.

### Example

|  |  |  |  |
| --- | --- | --- | --- |
| **Country** | **Highest Peak Name** | **Highest Peak Elevation** | **Mountain** |
| Afghanistan | (no highest peak) | 0 | (no mountain) |
| … | … | … | … |
| Argentina | Aconcagua | 6962 | Andes |
| … | … | … | … |
| Bulgaria | Musala | 2925 | Rila |
| Burkina Faso | (no highest peak) | 0 | (no mountain) |
| … | … | … | … |
| United States | Mount McKinley | 6194 | Alaska Range |