# Lab: Subqueries and JOINs

This document defines the lab assignments for [MySQL Course @ Software University](https://softuni.bg/opencourses/databases-basics-mysql).

You will use the **soft\_uni** **database** to write queries for the following exercises.

## Managers

Write a query to retrieve information about the **managers** – id, full\_name, deparment\_id and department\_name. Select the **first 5** departments ordered by employee\_id. Submit your queries using the "**MySQL prepare DB and Run Queries**" strategy.

**SELECT**

**e.employee\_id,**

**CONCAT(e.first\_name, ' ', e.last\_name) AS full\_name,**

**d.department\_id,**

**d.`name` AS department\_name**

**FROM**

**employees AS e**

**JOIN**

**departments AS d ON d.manager\_id = e.employee\_id**

**ORDER BY e.employee\_id**

**LIMIT 5;**

### Example

|  |  |  |  |
| --- | --- | --- | --- |
| **employee\_id** | **full\_name** | **department\_id** | **department\_name** |
| 3 | Roberto Tamburello | 10 | Finance |
| 4 | Rob Walters | 2 | Tool Design |
| … | … | … | … |

## Towns Addresses

Write a query to get information about the **addresses** in the database, which are in **San Francisco**, **Sofia** or **Carnation**. Retrieve town\_id, town\_name, address\_text. Order the result by town\_id, then by address\_id. Submit your queries using the "**MySQL prepare DB and Run Queries**" strategy.

**SELECT**

**t.town\_id, t.`name`, a.address\_text**

**FROM**

**towns AS t**

**JOIN**

**addresses AS a ON a.town\_id = t.town\_id**

**WHERE**

**a.town\_id IN (9 , 15, 32)**

**ORDER BY t.town\_id , a.address\_id;**

### Example

|  |  |  |
| --- | --- | --- |
| **town\_id** | **town\_name** | **address\_text** |
| 9 | San Fransisco | 1234 Seaside Way |
| 9 | San Fransisco | 5725 Glaze Drive |
| 15 | Carnation | 1411 Ranch Drive |
| … | … | … |

## Employees Without Managers

Write a query to get information about employee\_id, first\_name, last\_name, department\_id and salary for all employees who **don't have** a manager. Submit your queries using the "**MySQL prepare DB and Run Queries**" strategy.

**SELECT**

**employee\_id,first\_name,last\_name,department\_id,salary**

**FROM**

**employees**

**WHERE**

**manager\_id is null;**

### Example

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **employee\_id** | **first\_name** | **last\_name** | **department\_id** | **salary** |
| 109 | Ken | Sanchez | 16 | 125 500 |
| 291 | Svetlin | Nakov | 6 | 48 000 |
| 292 | Martin | Kulov | 6 | 48 000 |
| 293 | George | Denchev | 6 | 48 000 |

## Higher Salary

Write a query to count the number of employees who receive salary higher than the **average**. Submit your queries using the "**MySQL prepare DB and Run Queries**" strategy.

**SELECT**

**COUNT(e.employee\_id) AS `count`**

**FROM**

**employees AS e**

**WHERE**

**e.salary > (SELECT**

**AVG(a.salary)**

**FROM**

**employees AS a);**

### Example

|  |
| --- |
| **Count** |
| 88 |