# Lab: Basic CRUD

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

Download and get familiar with the **hotel** database schemas and tables. You will use it in the following exercises to write queries.

# Problem 1: Select Employee Information

Write a query to select all employees and retrieve information about their **id**, **first\_name, last\_name** and **job\_title** **ordered by id.**

**select `id`, `first\_name`, `last\_name`, `job\_title` from employees**

**order by id;**

### Example

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **first\_name** | **last\_name** | **job\_title** |
| 1 | John | Smith | Manager |
| 2 | John | Johnson | Customer Service |
| 3 | Smith | Johnson | Porter |
| … | … | … | … |

# Problem 2: Select Employees with Filter

Write a query to select all employees (**id, first\_name and last\_name (as full\_name), job\_title, salary**) whose salaries are **higher than 1000.00**, **ordered by id.** Concatenate fields **first\_name** and **last\_name** into '**full\_name**'.

**select `id`, CONCAT(`first\_name`,' ',`last\_name`),`job\_title`, `salary` from employees WHERE `salary` > 1000.00**

**order by id;**

### Example

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **full\_name** | **job\_title** | **salary** |
| 3 | Smith Johnson | Porter | 1100 |
| 4 | Peter Petrov | Front Desk Clerk | 1100 |
| 5 | Peter Ivanov | Sales | 1500.23 |
| … | … | … | … |

# Problem 3: Update Employees Salary

Update all employees' salaries whose **job\_title** is "**Manager**" by **adding 100**.

**Retrieve** information about **salaries** from table **employees**.

**UPDATE `employees`**

**SET `salary` = `salary`+100 WHERE `job\_title`='Manager';**

**SELECT `salary` FROM `employees`;**

# Problem 4: Top Paid Employee

Write a query to create a view that selects all information about the top paid employee from the "**employees**" table in the **hotel** database.

**SELECT \* from employees**

**order by `salary` DESC**

**limit 1;**

### Example

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **id** | **first\_name** | **last\_name** | **job\_title** | **department\_id** | **salary** |
| 8 | Pedro | Petrov | Front Desk Supervisor | 1 | 2100 |

# Problem 5: Select Employees by Multiple Filters

Write a query to retrieve information about employees, who are in **department 4** and has a salary **higher or equal to 1000**. Order the information by **id**.

**SELECT \* from employees WHERE `department\_id`=4 and `salary`>=1000**

**order by `id`;**

### Example

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **id** | **first\_name** | **last\_name** | **job\_title** | **department\_id** | **salary** |
| 3 | Smith | Johnson | Porter | 4 | 1100 |
| 9 | Nikolay | Ivanov | Housekeeping | 4 | 1600 |

# Problem 6: Delete from Table

Write a query to delete all employees from the "**employees**" table who are in department **2 or 1**. Then select all from table `employees` and order the information by id.

**DELETE from employees WHERE `department\_id`=2 or `department\_id`=1;**

**select \* from employees**

**order by `id`;**

### Example

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **id** | **First\_name** | **Last\_name** | **Job\_title** | **Department\_id** | **salary** |
| 3 | Smith | Johnson | Porter | 4 | 1100 |
| 6 | Ivan | Petrov | Waiter | 3 | 990 |
| 7 | Jack | Jackson | Executive Chef | 3 | 1800 |
| 9 | Nikolay | Ivanov | Housekeeping | 4 | 1600 |