# Lab: Database Programmability and Transactions

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

You are provided with the **soft\_uni** database. Use it in the following assignments.

## Count Employees by Town

Write a function ufn\_count\_employees\_by\_town(town\_name) that accepts town\_name as parameter and returns the count of employees who live in that town. Submit your queries using the "**MySQL** **Run Skeleton, run queries and check DB**" strategy.

**CREATE FUNCTION ufn\_count\_employees\_by\_town(town\_name VARCHAR(20))**

**RETURNS INT**

**DETERMINISTIC**

**BEGIN**

**DECLARE e\_count INT;**

**SET e\_count := (SELECT COUNT(employee\_id) FROM employees AS e**

**JOIN addresses AS a ON a.address\_id = e.address\_id**

**JOIN towns AS t ON t.town\_id = a.town\_id**

**WHERE t.name = town\_name);**

**RETURN e\_count;**

**END**

### Example

The following example is given with employees living in **Sofia**.

|  |
| --- |
| **count** |
| 3 |

## Employees Promotion

Write a stored procedure usp\_raise\_salaries(department\_name) to raise the salary of all employees in given department as parameter by 5%. Submit your queries using the "**MySQL** **Run Skeleton, run queries and check DB**" strategy.

### Example

The following example is given with employees in the "**Finance**" department ordered by first\_name, then by salary.

|  |  |
| --- | --- |
| **first\_name** | **salary** |
| Barbara | 27 720.00 |
| Bryan | 19 950.00 |
| Candy | 19 950.00 |
| … | … |

**3. Employees Promotion by ID**

Write a stored procedure usp\_raise\_salary\_by\_id(id) that raises a given employee's salary (by id as parameter) by **5%**. Consider that you cannot promote an employee that **doesn't exist** – if that happens, no changes to the database should be made. Submit your queries using the "**MySQL** **Run Skeleton, run queries and check DB**" strategy.

**CREATE PROCEDURE usp\_raise\_salary\_by\_id(id INT)**

**BEGIN**

**update employees**

**set salary=salary\*1.05 where employee\_id=id;**

**END**

### Example

The following example is given with employee\_id = 17.

|  |
| --- |
| **salary** |
| 14175.0000 |

## 4. Triggered

Create a table deleted\_employees(employee\_id PK, first\_name,last\_name,middle\_name,job\_title,deparment\_id,salary) that will hold information about fired(deleted) employees from the employees table. Add a trigger to employees table that inserts the corresponding information in deleted\_employees. Submit your queries using the "**MySQL** **Run Skeleton, run queries and check DB**" strategy.

**CREATE TABLE deleted\_employees (**

**employee\_id INT PRIMARY KEY AUTO\_INCREMENT,**

**first\_name VARCHAR(50),**

**last\_name VARCHAR(50),**

**middle\_name VARCHAR(50),**

**job\_title VARCHAR(50),**

**department\_id INT,**

**salary DOUBLE**

**);**

**CREATE**

**TRIGGER tr\_deleted\_employees**

**BEFORE DELETE ON employees FOR EACH ROW**

**INSERT INTO deleted\_employees (first\_name , last\_name , middle\_name , job\_title , department\_id , salary) VALUES (OLD.first\_name , OLD.last\_name , OLD.middle\_name , OLD.job\_title , OLD.department\_id , OLD.salary);**