# Lab: Database Programmability and Transactions

**Submit your solutions** to the SoftUni [Judge System](https://alpha.judge.softuni.org/Contests/Database-Programmability-Lab/4112).

You are provided with the **softuni\_db.sql** file. Create a database **soft\_uni** and import the sql file into its **Query Tab**.Execute all queries. Use this database in the following assignments.

## Count Employees by Town

Create a function fn\_count\_employees\_by\_town(town\_name) that accepts town\_name **VARCHAR(20)** as a parameter and returns the **count of employees** living there.

Submit your query about **function creation only**.

### Example

The following example is for employees living in "**Sofia"**.

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

## Employees Promotion

Create a stored procedure sp\_increase\_salaries(department\_name) to increase the salary of all employees working at a given department (provided as a parameter). Increase salaries by **5%**.

Submit your query about **stored procedure creation only**.

### Example

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

|  |  |
| --- | --- |
| **first\_name** | **salary** |
| Barbara | 27720.0000 |
| Bryan | 19950.0000 |
| Candy | 19950.0000 |
| … | … |

1. Employees Promotion by **ID**

Create a stored procedure(transaction) sp\_increase\_salary\_by\_id(id) that increases a given employee's salary (by id as a parameter) by **5%**. Consider that you cannot promote an employee who **doesn't exist** – if that happens**, no changes** to the database shall be made.

Submit your query about **transaction creation only**.

### Example

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

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

## Triggered

Create a table deleted\_employees(employee\_id PK, first\_name, last\_name, middle\_name, job\_title, department\_id, salary) that will hold information about fired (deleted) employees from the **employees** table. Add a **trigger** to the **employees** table that inserts the corresponding information into deleted\_employees table.

* **deleted\_employees** table:
  + **employee\_id** – primary key
  + **first\_name** - max-length 20
  + **last\_name** - max-length 20
  + **middle\_name** - max-length 20
  + **job\_title** – max-length 50
  + **department\_id** – integer
  + **salary** – numeric (19,4)

Submit your queries about **table creation, trigger function, and the trigger itself**.