

Lab 07

Topic: Stored Procedures and Triggers

General Instructions

- Use the provided [generate_tables.sql](#) file to create and populate the tables.
 - An outline of the procedures to be written is given with each question to avoid confusion about the number and types of parameters and their datatypes.
 - To test the working of the trigger supposed to be written for Q4, an additional [trigger_data.sql](#) file is uploaded with table data. After creating the trigger, use this data to check if it is working as intended.
 - Upload your SQL queries in the link provided on Quanta.
-

QUESTION 1 - 2M (NO PARTIAL)

Write a stored procedure to select the number of employees in a given department. Take the department name as input, store the output in a variable.

Procedure snippet

```
delimiter //
create procedure numemp(in dept varchar(30), out num_emp int)
begin
    /* code goes here */
end //
delimiter ;
```

QUESTION 2 - 3M

Write a stored procedure to clean up data in the employees table, which does the following:

- Deletes all rows with NULL values (in any column) - 1M
- Displays the rows that are deleted - 1M
- Displays the total number of rows deleted - 1M

Hint: Only certain columns are allowed to contain NULL values, use that information to write your procedure. Make sure you call this procedure before proceeding to the following questions.

Procedure snippet

```
delimiter //
create procedure cleanup()
begin

    /* code goes here */

end //
delimiter ;
```

QUESTION 3 - 2M

Create a procedure to increment the salary of all employees with a given job. Take the job name and percentage to increase as input and do the following -

- Update the rows corresponding to the given job name - 1M
- Display the name, job id, old salary and new salary of the modified rows only - 1M

Procedure snippet

```
delimiter //
create procedure sal_increment(job_name varchar(30), increment int)
begin

    /* code goes here */

end //
delimiter ;
```

QUESTION 4 - 2M (NO PARTIAL)

The departments table has 2 columns, namely employee_count and avg_salary, with all values as 0.

Create a trigger to populate these columns - whenever a new row is inserted in the employee table, the employee count and average salary of the corresponding department is updated accordingly.

After creating the trigger, use data from the [trigger_data.sql](#) file to populate the employees table. Don't use data from the [generate_tables.sql](#) file.

Hint: If your trigger populated the departments table with incorrect values, do the following -

- Reset the employee_count and avg_salary columns to 0.
`update departments set employee_count = 0;`
`update departments set avg_salary = 0;`
- Delete all data from the employees table, you won't be able to re-insert data otherwise.
`delete from employees;`