## Indian Institute of Information Technology Surat

****

# Lab Report on

# Advanced Database Management (CS 604) Practical

**Submitted by**

### [RAHUL KUMAR SINGH] (UI21CS44)

**Course Faculty**

### Mr. Rishi Sharma

## Department of Computer Science and Engineering

## Indian Institute of Information Technology Surat

## Gujarat-394190, India

**Jan-2024**

## Lab No: 3

**Aim: Write a PL/SQL (MySQL Procedure) code block to perform specific tasks on tables Teacher, Class and Pay\_scale.**

**Description:** Consider the following table to write PL/SQL code as specified under

* Teacher (t\_no, f\_name, l\_name, salary, supervisor, joining\_date, birth\_date, title)
* Class (class\_no, t\_no, room\_no)
* Pay\_scale (Min\_limit, Max\_limit, grade)

1. Accept a range of salary and print the details of teachers from the teacher table.

2. By using cursor - Calculate the bonus amount to be given to a teacher depending on the following conditions:

a) if salary < 10000 then bonus is 10% of the salary.

b) if the salary is between 10000 and 20000 then bonus is 20% of the salary.

c) if the salary is between 20000 and 25000 then bonus is 25% of the salary.

d) if the salary exceeds 25000 then the bonus is 30% of the salary.

3. Using a simple LOOP structure, list the first 10 records of the ‘teachers’ table.

4. Accept the room number and display the teacher details like t\_no, f\_name, l\_name, birth\_date, title from table Teacher.

## Source Code:

**Teacher Table:**

CREATE TABLE Teacher (

t\_no INT PRIMARY KEY,

f\_name VARCHAR(50) NOT NULL,

l\_name VARCHAR(50) NOT NULL,

salary DECIMAL(10, 2) NOT NULL,

supervisor INT,

joining\_date DATE NOT NULL,

birth\_date DATE NOT NULL,

title VARCHAR(50) NOT NULL

);

**Class Table:**

CREATE TABLE Class (

class\_no INT PRIMARY KEY,

t\_no INT,

room\_no INT,

FOREIGN KEY (t\_no) REFERENCES Teacher(t\_no),

UNIQUE KEY unique\_teacher\_class (t\_no, room\_no)

);

**Pay\_scale Table:**

CREATE TABLE Pay\_scale (

Min\_limit DECIMAL(10, 2) NOT NULL,

Max\_limit DECIMAL(10, 2) NOT NULL,

grade VARCHAR(10) PRIMARY KEY

);

**Task 1:**

DELIMITER //

CREATE PROCEDURE GetTeachersBySalaryRange(

IN minSalary DECIMAL(10, 2),

IN maxSalary DECIMAL(10, 2)

)

BEGIN

SELECT \*

FROM Teacher

WHERE salary BETWEEN minSalary AND maxSalary;

END //

DELIMITER ;

**Task 2:**

DELIMITER //

CREATE PROCEDURE CalculateTeacherBonus()

BEGIN

DECLARE done INT DEFAULT FALSE;

DECLARE t\_no\_var INT;

DECLARE salary\_var DECIMAL(10, 2);

DECLARE bonus\_var DECIMAL(10, 2);

DECLARE teacher\_cursor CURSOR FOR

SELECT t\_no, salary

FROM Teacher;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

OPEN teacher\_cursor;

teacher\_loop: LOOP

FETCH teacher\_cursor INTO t\_no\_var, salary\_var;

IF done THEN

LEAVE teacher\_loop;

END IF;

IF salary\_var < 10000 THEN

SET bonus\_var = salary\_var \* 0.10;

ELSEIF salary\_var BETWEEN 10000 AND 20000 THEN

SET bonus\_var = salary\_var \* 0.20;

ELSEIF salary\_var BETWEEN 20000 AND 25000 THEN

SET bonus\_var = salary\_var \* 0.25;

ELSE

SET bonus\_var = salary\_var \* 0.30;

END IF;

SELECT t\_no\_var AS Teacher\_ID, salary\_var AS Salary, bonus\_var AS Bonus;

END LOOP;

CLOSE teacher\_cursor;

END //

DELIMITER ;

**Task 3:**

DELIMITER //

CREATE PROCEDURE ListFirst10Teachers()

BEGIN

DECLARE counter INT DEFAULT 0;

DECLARE t\_no\_var INT;

DECLARE f\_name\_var VARCHAR(50);

DECLARE l\_name\_var VARCHAR(50);

DECLARE salary\_var DECIMAL(10, 2);

DECLARE supervisor\_var INT;

DECLARE joining\_date\_var DATE;

DECLARE birth\_date\_var DATE;

DECLARE title\_var VARCHAR(50);

DECLARE teacher\_cursor CURSOR FOR

SELECT t\_no, f\_name, l\_name, salary, supervisor, joining\_date, birth\_date, title

FROM Teacher;

OPEN teacher\_cursor;

teacher\_loop: LOOP

FETCH teacher\_cursor INTO t\_no\_var, f\_name\_var, l\_name\_var, salary\_var, supervisor\_var, joining\_date\_var, birth\_date\_var, title\_var;

IF counter >= 10 OR t\_no\_var IS NULL THEN

LEAVE teacher\_loop;

END IF;

SET counter = counter + 1;

SELECT t\_no\_var AS Teacher\_ID, f\_name\_var AS First\_Name, l\_name\_var AS Last\_Name, salary\_var AS Salary,

supervisor\_var AS Supervisor, joining\_date\_var AS Joining\_Date, birth\_date\_var AS Birth\_Date, title\_var AS Title;

END LOOP;

CLOSE teacher\_cursor;

END //

DELIMITER ;

**Task 4:**

DELIMITER //

CREATE PROCEDURE GetTeachersByRoomNumber(IN roomNumber INT)

BEGIN

SELECT t.t\_no, t.f\_name, t.l\_name, t.birth\_date, t.title

FROM Teacher t

JOIN Class c ON t.t\_no = c.t\_no

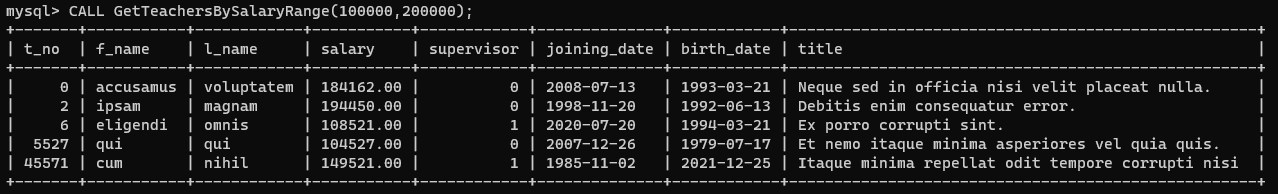
WHERE c.room\_no = roomNumber;

END //

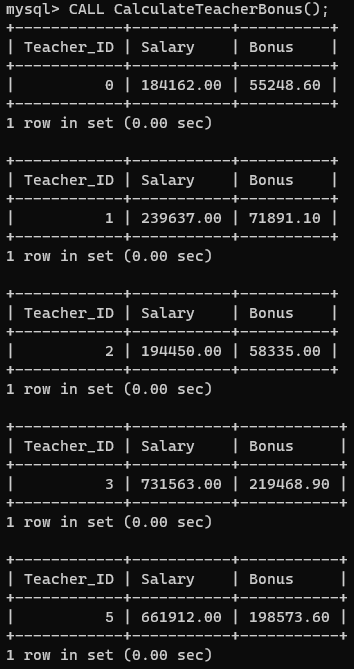
DELIMITER ;

## Output:

**Task 1:**

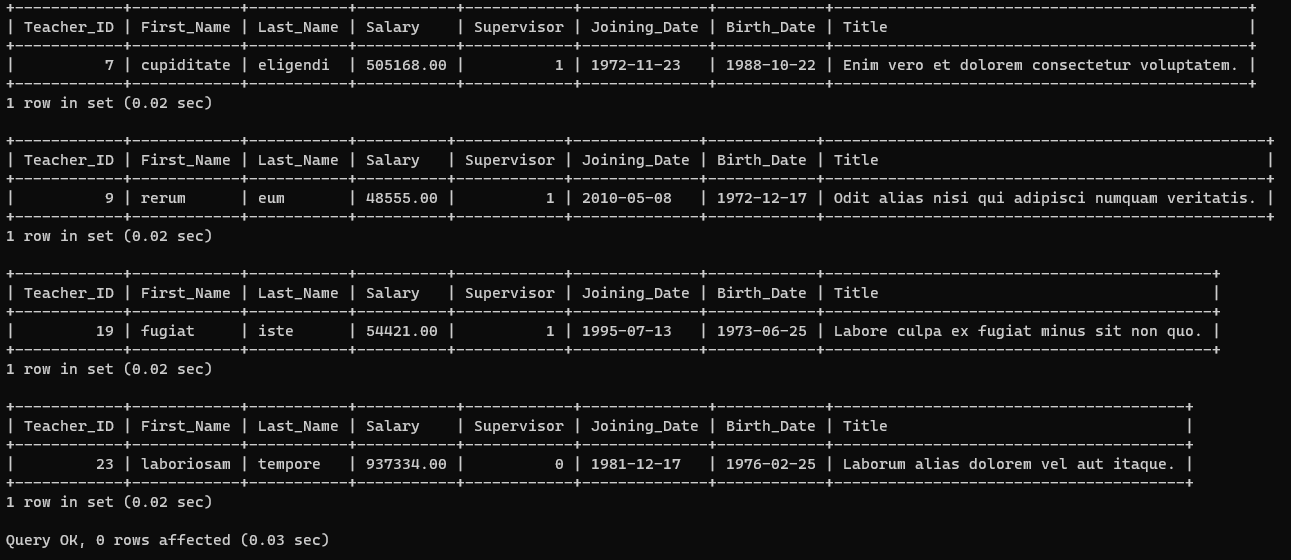
****

**Task 2:**

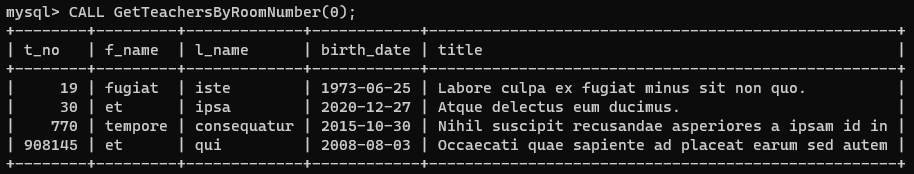
****

**Task 3:**

****

****

**Task 4:**

****

## Conclusion:

* GetTeachersBySalaryRange: Accepts a salary range and retrieves details of teachers from the Teacher table within that range.
* CalculateTeacherBonus: Uses a cursor to calculate the bonus amount for teachers based on salary conditions, considering different bonus percentages for specific salary ranges.
* ListFirst10Teachers: Utilizes a simple LOOP structure to list the first 10 records from the Teacher table.
* GetTeachersByRoomNumber: Accepts a room number and displays specific details (t\_no, f\_name, l\_name, birth\_date, title) of teachers associated with that room from the Teacher table.
* The code is structured in a modular manner using a MySQL Procedure block for better understanding.
* The code is designed for execution in interactive environments.
* Utilized DECLARE and BEGIN sections to define variables and execute procedural logic.
* Applied the DBMS\_OUTPUT.PUT\_LINE function for displaying output for all the procedures.

.