**🡸---------------OFFICE STAFF MANAGEMENT SYSTEM-----------🡺**

**NAME : CHIRAG PATEL**

**ROLL NO : 173**

**BRANCH : CE**

**DIVISION : D6**

**ENROLLMENT NO : 22002170110107**

1. **GETTING EMPLOYEE PERSONAL DETAILS AND DESIGNATION DETAILS BY HIS/HER ID :**

CREATE OR REPLACE PROCEDURE GET\_EMP\_DEATILS\_DESIGNATION1(IN\_ID INT)AS

$$

DECLARE

CHECK\_S INT;

O\_ID EMPLOYEE.E\_ID%TYPE;

FIRST EMPLOYEE.E\_FIRSTNAME%TYPE;

LAST EMPLOYEE.E\_LASTNAME%TYPE;

AGE EMPLOYEE.E\_AGE%TYPE;

EMAIL EMPLOYEE.E\_MAIL%TYPE;

NUMBER EMPLOYEE.E\_NUMBER%TYPE;

DOB EMPLOYEE.E\_DOB%TYPE;

DESIGN EMP\_DESIGNATION.E\_DESIGNATION%TYPE;

SALARY EMP\_DESIGNATION.E\_SALARY%TYPE;

JOINING EMP\_DESIGNATION.E\_JOININGDATE%TYPE;

PRECOM EMP\_DESIGNATION.E\_PREVIOUSCOMPANY%TYPE;

BEGIN

SELECT COUNT(E\_ID) INTO CHECK\_S FROM EMPLOYEE WHERE E\_ID=IN\_ID;

IF(CHECK\_S>0) THEN

SELECT E\_ID,E\_FIRSTNAME,E\_LASTNAME,E\_AGE,E\_MAIL,E\_NUMBER,E\_DOB INTO O\_ID,FIRST,LAST,AGE,EMAIL,NUMBER,DOB FROM EMPLOYEE WHERE E\_ID=IN\_ID;

SELECT E\_DESIGNATION,E\_SALARY,E\_JOININGDATE,E\_PREVIOUSCOMPANY INTO DESIGN,SALARY,JOINING,PRECOM FROM EMP\_DESIGNATION WHERE E\_ID=IN\_ID;

RAISE NOTICE 'EMPLOYEE ID : % EMPLOYEE NAME : % % ',O\_ID,FIRST,LAST;

RAISE NOTICE 'EMPLOYEE AGE : % EPMPLOYEE EMAIL : % EMPLOYEE NUMBER : %',AGE,EMAIL,NUMBER;

RAISE NOTICE 'EMPLOYEE DOB : % EMPLOYEE DESIGANTION : % EMPLOYEE SALARY : % ',DOB,DESIGN,SALARY;

RAISE NOTICE 'EMPLOYEE JOIN DATE : % EMPLOYEE PREVIOUS COMPANY : % ',JOINING,PRECOM;

ELSE

RAISE NOTICE 'EMPLOYEE DOESNOT EXIST !';

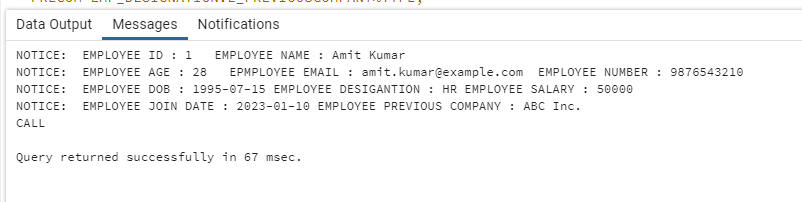
END IF;

END;

$$

LANGUAGE PLPGSQL;

CALL GET\_EMP\_DEATILS\_DESIGNATION1(1);



1. **GETTING ADMINS PERSONAL DETAILS AND DESIGNATION DETAILS BY HIS/HER ID :**

CREATE OR REPLACE PROCEDURE GET\_ADM\_DEATILS\_DESIGNATION2(IN\_ID INT)AS

$$

DECLARE

CHECK\_S INT;

O\_ID ADMINISTRATION.A\_ID%TYPE;

FIRST ADMINISTRATION.A\_FIRSTNAME%TYPE;

LAST ADMINISTRATION.A\_LASTNAME%TYPE;

AGE ADMINISTRATION.A\_AGE%TYPE;

EMAIL ADMINISTRATION.A\_EMAIL%TYPE;

NUMBER ADMINISTRATION.A\_NUMBER%TYPE;

DOB ADMINISTRATION.A\_DOB%TYPE;

DESIGN ADMIN\_DESIGNATION.A\_DESIGNATION%TYPE;

SALARY ADMIN\_DESIGNATION.A\_SALARY%TYPE;

PREPOS ADMIN\_DESIGNATION.A\_PREVIOUSPOST%TYPE;

PRECOM ADMIN\_DESIGNATION.A\_PREVIOUSCOMPANY%TYPE;

BEGIN

SELECT COUNT(A\_ID) INTO CHECK\_S FROM ADMINISTRATION WHERE A\_ID=IN\_ID;

IF(CHECK\_S>0) THEN

SELECT A\_ID,A\_FIRSTNAME,A\_LASTNAME,A\_AGE,A\_EMAIL,A\_NUMBER,A\_DOB INTO O\_ID,FIRST,LAST,AGE,EMAIL,NUMBER,DOB FROM ADMINISTRATION WHERE A\_ID=IN\_ID;

SELECT A\_DESIGNATION,A\_SALARY,A\_PREVIOUSCOMPANY,A\_PREVIOUSPOST INTO DESIGN,SALARY,PRECOM,PREPOS FROM ADMIN\_DESIGNATION WHERE A\_ID=IN\_ID;

RAISE NOTICE 'ADMIN ID : % ADMIN NAME : % % ',O\_ID,FIRST,LAST;

RAISE NOTICE 'ADMIN AGE : % ADMIN EMAIL : % ADMIN NUMBER : %',AGE,EMAIL,NUMBER;

RAISE NOTICE 'ADMIN DOB : % ADMIN DESIGANTION : % ADMIN SALARY : % ',DOB,DESIGN,SALARY;

RAISE NOTICE ' ADMIN PREVIOUS COMPANY : % ADMIN PREVIOUS POST : % ',PRECOM,PREPOS;

ELSE

RAISE NOTICE 'ADMIN DOESNOT EXIST !';

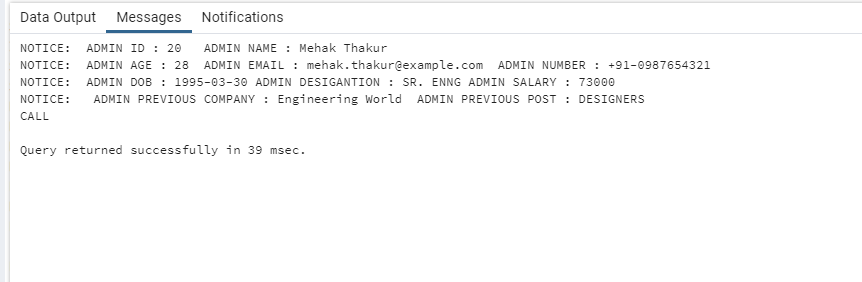
END IF;

END;

$$

LANGUAGE PLPGSQL;

CALL GET\_ADM\_DEATILS\_DESIGNATION2(20);



1. **GETTING MAX SALARY OF EMPLOYEE/ADMINISTARTION BY ENTERING DESIGNATION NAME :**

CREATE OR REPLACE FUNCTION GET\_MAX\_SALARY\_BY\_DESIGN(DESIGNATION VARCHAR(20)) RETURNS INT AS

$$

DECLARE

MAX\_SALARY INT;

BEGIN

SELECT MAX(E\_SALARY) INTO MAX\_SALARY FROM EMP\_DESIGNATION WHERE E\_DESIGNATION=DESIGNATION;

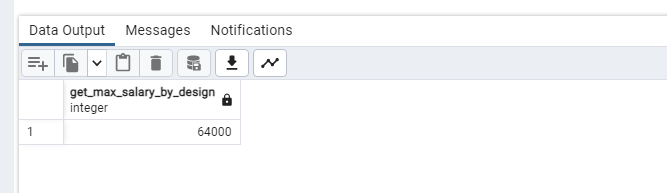
RETURN MAX\_SALARY;

END;

$$

LANGUAGE PLPGSQL;

SELECT GET\_MAX\_SALARY\_BY\_DESIGN('MARKETING');



1. **GETTING ADMINISTRATIONS LIST BY DESIGNATION:**

CREATE OR REPLACE PROCEDURE GET\_ADMININTRATIONS\_BY\_DESIGNATION(IN\_D VARCHAR(30)) AS

$$

DECLARE

FIRST ADMINISTRATION.A\_FIRSTNAME%TYPE;

LAST ADMINISTRATION.A\_LASTNAME%TYPE;

DESIGN ADMIN\_DESIGNATION.A\_DESIGNATION%TYPE;

SALARY ADMIN\_DESIGNATION.A\_SALARY%TYPE;

C1 CURSOR FOR SELECT ADMINISTRATION.A\_FIRSTNAME,ADMINISTRATION.A\_LASTNAME,ADMIN\_DESIGNATION.A\_DESIGNATION,

ADMIN\_DESIGNATION.A\_SALARY FROM ADMIN\_DESIGNATION INNER JOIN ADMINISTRATION

ON ADMINISTRATION.A\_ID=ADMIN\_DESIGNATION.A\_ID WHERE A\_DESIGNATION=IN\_D;

BEGIN

OPEN C1;

LOOP

FETCH C1 INTO FIRST,LAST,DESIGN,SALARY;

EXIT WHEN NOT FOUND;

RAISE NOTICE 'ADMINISTRATION NAME: % %

ADMINISTRAION DESIGNATION : %

ADMINISTRATION SALARY : % ',FIRST,LAST,DESIGN,SALARY;

END LOOP;

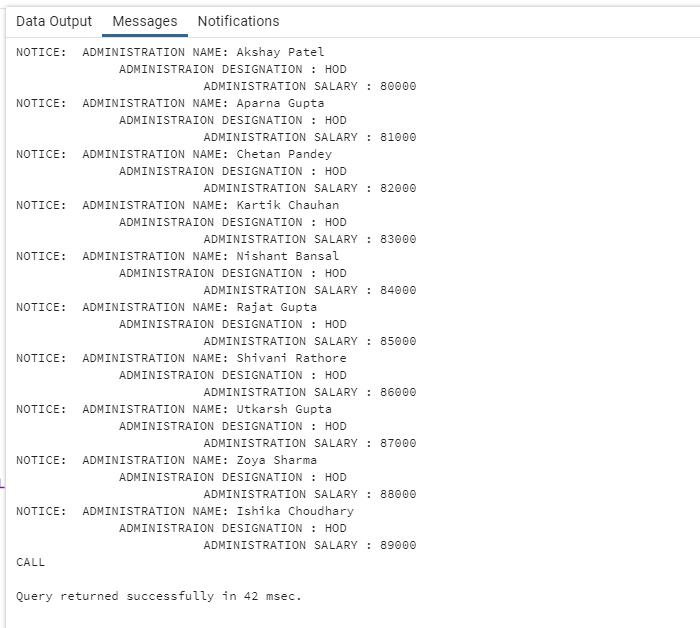
CLOSE C1;

END;

$$

LANGUAGE PLPGSQL;

-- CALL GET\_ADMININTRATIONS\_BY\_DESIGNATION('HOD');



1. **TRIIGERS ON INSERTION WHEN OLD ID IS INSETRED AGAIN OR NEGATIVE ID IS INSERTED IN TABLE :**

CREATE OR REPLACE FUNCTION CHECK\_E\_ID()returns trigger as

$$

DECLARE

CHECK\_E INT;

BEGIN

SELECT COUNT(\*) INTO CHECK\_E FROM EMPLOYEE WHERE E\_ID=NEW.E\_ID;

IF NEW.E\_ID <0 THEN

RAISE 'LESSS THAN ZERO VALUE NOT ALLOWED';

ELSE

IF CHECK\_E >0 THEN

RAISE 'THAT ID ALREAFDY EXIST !';

END IF;

END IF;

RETURN NEW;

END;

$$

LANGUAGE PLPGSQL;

CREATE TRIGGER CHECK\_E\_ID

BEFORE INSERT ON EMPLOYEE

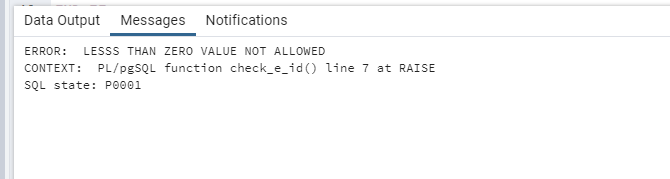
FOR EACH ROW

EXECUTE PROCEDURE CHECK\_E\_ID();

INSERT INTO employee (E\_ID, E\_FIRSTNAME, E\_LASTNAME, E\_AGE, E\_MAIL, E\_NUMBER, E\_DOB)

VALUES

(-1, 'Amit', 'Kumar', 28, 'amit.kumar@example.com', '9876543210', '1995-07-15');



1. **GETTING EMPLOYEES AND ADMINS BY ENTERING DEPARTAMENT NAME :**

CREATE OR REPLACE PROCEDURE GET\_EMP\_ADM\_BY\_DEPT(DEPART VARCHAR(20)) AS

$$

DECLARE

FIRST\_E EMPLOYEE.E\_FIRSTNAME%TYPE;

LAST\_E EMPLOYEE.E\_LASTNAME%TYPE;

FIRST\_A ADMINISTRATION.A\_FIRSTNAME%TYPE;

LAST\_A ADMINISTRATION.A\_LASTNAME%TYPE;

C1 CURSOR FOR SELECT EMPLOYEE.E\_FIRSTNAME,EMPLOYEE.E\_LASTNAME FROM EMPLOYEE INNER JOIN DEPARTMENT

ON EMPLOYEE.E\_ID=DEPARTMENT.E\_ID WHERE D\_NAME=DEPART;

C2 CURSOR FOR SELECT ADMINISTRATION.A\_FIRSTNAME,ADMINISTRATION.A\_LASTNAME FROM ADMINISTRATION INNER JOIN DEPARTMENT

ON ADMINISTRATION.A\_ID=DEPARTMENT.A\_ID WHERE D\_NAME=DEPART;

BEGIN

OPEN C1;

OPEN C2;

LOOP

FETCH C1 INTO FIRST\_E,LAST\_E;

EXIT WHEN NOT FOUND;

RAISE NOTICE 'EMPLOYEEE LIST : % % ',FIRST\_E,LAST\_E;

END LOOP;

LOOP

FETCH C2 INTO FIRST\_A,LAST\_A;

EXIT WHEN NOT FOUND;

RAISE NOTICE 'ADMIN LIST : % % ',FIRST\_A,LAST\_A;

END LOOP;

CLOSE C2;

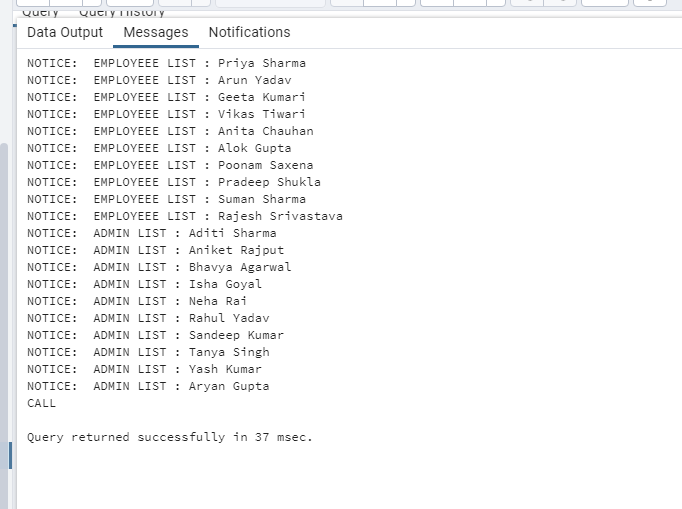
CLOSE C1;

END;

$$

LANGUAGE PLPGSQL;

-- CALL GET\_EMP\_ADM\_BY\_DEPT('ACCOUNTS');



1. **ENTER THE SALARY AND INCREMENT PERECENATGE AND QUERY WILL INCREMENTS ALL EMPLOYEE'S SALARY WHO IS LESS THAN ENTERED SLARY :**

CREATE OR REPLACE PROCEDURE UPDATE\_SALARY1(I\_SALARY INT, INCR INT) AS

$$

DECLARE

INCREMENTED INT;

NEW\_SAL INT;

FIRST ADMINISTRATION.A\_FIRSTNAME%TYPE;

LAST ADMINISTRATION.A\_LASTNAME%TYPE;

DESIGN ADMIN\_DESIGNATION.A\_DESIGNATION%TYPE;

SALARY ADMIN\_DESIGNATION.A\_SALARY%TYPE;

C1 CURSOR FOR SELECT ADMINISTRATION.A\_FIRSTNAME, ADMINISTRATION.A\_LASTNAME, ADMIN\_DESIGNATION.A\_DESIGNATION, ADMIN\_DESIGNATION.A\_SALARY

FROM ADMIN\_DESIGNATION

INNER JOIN ADMINISTRATION ON ADMINISTRATION.A\_ID = ADMIN\_DESIGNATION.A\_ID

WHERE A\_SALARY > I\_SALARY;

BEGIN

INCREMENTED := INCR / 100;

OPEN C1;

LOOP

FETCH C1 INTO FIRST, LAST, DESIGN, SALARY;

EXIT WHEN NOT FOUND;

RAISE NOTICE 'ADMIN NAME : % %', FIRST, LAST;

RAISE NOTICE 'ADMIN OLD SALARY : %', SALARY;

NEW\_SAL := SALARY + (INCREMENTED \* SALARY);

RAISE NOTICE 'ADMIN NEW SALARY : %', NEW\_SAL;

-- Update the salary in the ADMIN\_DESIGNATION table

-- UPDATE ADMIN\_DESIGNATION

-- SET A\_SALARY = NEW\_SAL

-- WHERE CURRENT OF C1;

END LOOP;

CLOSE C1;

-- Commit the changes to the database

COMMIT;

END;

$$

LANGUAGE PLPGSQL;

-- CALL UPDATE\_SALARY1(50000,10);



1. **GET HOW MANY NUMBERS OF EMPLOYEES :**

CREATE OR REPLACE FUNCTION GET\_EMPLOYEE\_COUNT\_BY\_DESIGNATION(DESIGN VARCHAR(30)) RETURNS INT AS

$$

DECLARE

COUNTER INT;

BEGIN

SELECT COUNT(E\_ID) INTO COUNTER FROM EMP\_DESIGNATION WHERE E\_DESIGNATION=DESIGN;

RETURN COUNTER;

END;

$$

LANGUAGE PLPGSQL;

SELECT GET\_EMPLOYEE\_COUNT\_BY\_DESIGNATION('MARKETING');

