** 🡸---------------AIRLINE RESERVATION SYSTEM-----------🡺**

**NAME : YASVI VAGHASIYA**

**ROLL NO : 166**

**BRANCH : CE**

**DIVISION : D6**

**ENROLLMENT NO : 22002170110199**

1. **CREATE A TRIGGER TO UPDATE FLIGHT STATUS :**

CREATE OR REPLACE FUNCTION UPDATE\_FLIGHT\_STATUS()

RETURNS TRIGGER AS $$

BEGIN

UPDATE FLIGHTS

SET STATUS = CASE

WHEN FIRSTCLASSSEATS > 0 OR BUSINESSCLASSSEATS > 0 OR ECONOMICALCLASSSEATS > 0 THEN 1

ELSE 0

END;

RETURN NEW;

END;

$$ LANGUAGE PLPGSQL;

CREATE TRIGGER FLIGHT\_STATUS\_UPDATE

AFTER INSERT OR UPDATE ON RESERVATION

FOR EACH ROW

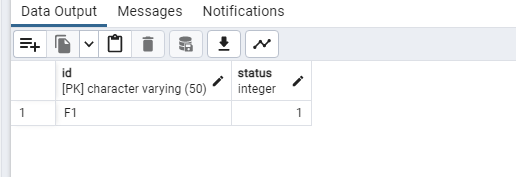
EXECUTE FUNCTION UPDATE\_FLIGHT\_STATUS();

-- INSERT INTO RESERVATION (ID, CLASSNAME, USERID, FLIGHTID)

VALUES

('R1', 'FIRST CLASS', '1', 'F1');

SELECT ID,STATUS FROM FLIGHTS WHERE ID='F1';



1. **FUNCTION TO CALCULATE TOTAL SEATS FOR A FLIGHT :**

CREATE OR REPLACE FUNCTION CALCULATE\_TOTAL\_SEATS(FLIGHT\_ID VARCHAR)

RETURNS INT AS $$

DECLARE

TOTAL\_SEATS INT;

BEGIN

SELECT FIRSTCLASSSEATS + BUSINESSCLASSSEATS + ECONOMICALCLASSSEATS

INTO TOTAL\_SEATS

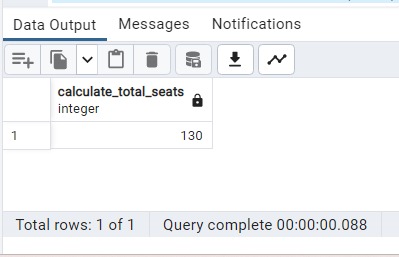
FROM FLIGHTS

WHERE ID = FLIGHT\_ID;

RETURN TOTAL\_SEATS;

END;

$$ LANGUAGE PLPGSQL;



1. **LOOP TO UPDATE FLIGHT PRICES :**

DO $$

DECLARE

FLIGHT\_ID VARCHAR;

BEGIN

FOR FLIGHT\_ID IN (SELECT ID FROM FLIGHTS) LOOP

UPDATE FLIGHTS

SET FIRSTCLASSPRICE = FIRSTCLASSPRICE \* 1.1,

BUSINESSCLASSPRICE = BUSINESSCLASSPRICE \* 1.1,

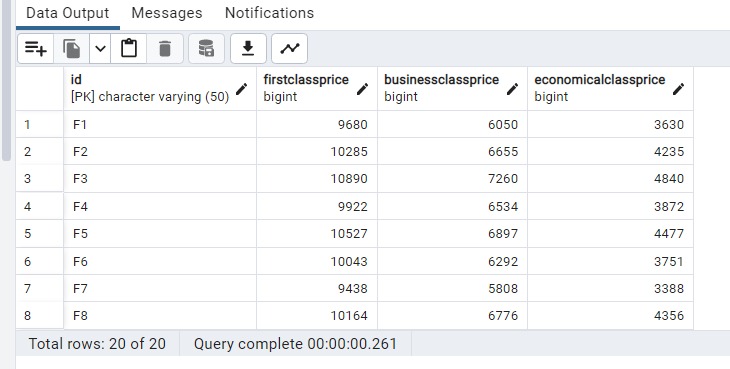
ECONOMICALCLASSPRICE = ECONOMICALCLASSPRICE \* 1.1

WHERE ID = FLIGHT\_ID;

END LOOP;

END;

$$;



1. **TRIGGER TO ENFORCE EMAIL UNIQUENESS FOR USERS :**

CREATE OR REPLACE FUNCTION ENFORCE\_UNIQUE\_EMAIL()

RETURNS TRIGGER AS $$

BEGIN

IF EXISTS (SELECT 1 FROM USERS WHERE EMAIL = NEW.EMAIL) THEN

RAISE EXCEPTION 'EMAIL MUST BE UNIQUE.';

END IF;

RETURN NEW;

END;

$$ LANGUAGE PLPGSQL;

CREATE TRIGGER UNIQUE\_EMAIL

BEFORE INSERT OR UPDATE ON USERS

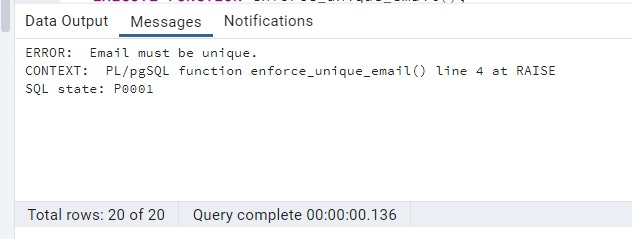
FOR EACH ROW

EXECUTE FUNCTION ENFORCE\_UNIQUE\_EMAIL();

--INSERT INTO ADMINS (A\_ID, A\_NAME, A\_EMAIL, A\_PASSWORD)

VALUES

(1, 'RAJESH ADMIN', 'RAJESH.ADMIN@EXAMPLE.COM', 'ADMIN\_PASSWORD\_1');



1. **FUNCTION TO CALCULATE RESERVATION COST :**

CREATE OR REPLACE FUNCTION CALCULATE\_RESERVATION\_COST(RESERVATION\_ID VARCHAR)

RETURNS INT AS $$

DECLARE

TOTAL\_COST INT;

BEGIN

SELECT CASE

WHEN CLASSNAME = 'FIRST CLASS' THEN FIRSTCLASSPRICE

WHEN CLASSNAME = 'BUSINESS CLASS' THEN BUSINESSCLASSPRICE

WHEN CLASSNAME = 'ECONOMICAL CLASS' THEN ECONOMICALCLASSPRICE

ELSE 0

END

INTO TOTAL\_COST

FROM RESERVATION INNER JOIN FLIGHTS ON FLIGHTS.ID=RESERVATION.FLIGHTID

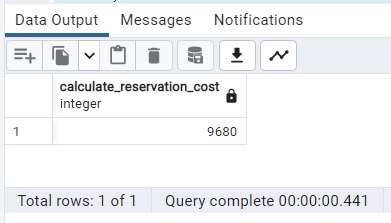
WHERE R\_ID = RESERVATION\_ID;

RETURN TOTAL\_COST;

END;

$$ LANGUAGE PLPGSQL;

SELECT CALCULATE\_RESERVATION\_COST('R1');



1. **PROCEDURE TO CANCEL RESERVATION :**

CREATE OR REPLACE PROCEDURE CANCEL\_RESERVATION(RESERVATION\_ID VARCHAR)

LANGUAGE PLPGSQL AS $$

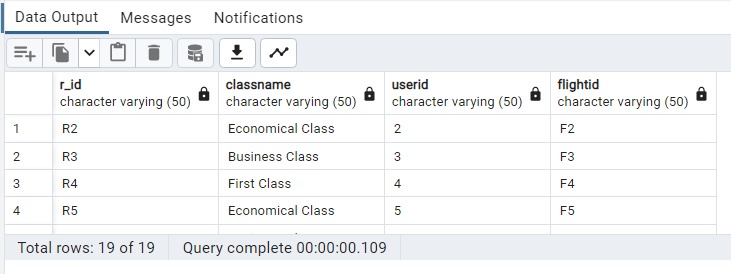
BEGIN

DELETE FROM RESERVATION WHERE R\_ID = RESERVATION\_ID;

END;

$$;

CALL CANCEL\_RESERVATION('R1');



1. **TRIGGER FUNCTION FOR INSERTION A FLIGHTS THAT DOESN’T ALLOW REPEATED FLIGHT ID :**

CREATE OR REPLACE FUNCTION FLIGHT\_CHECK() RETURNS TRIGGER AS

$$

DECLARE

CHECK\_ID INT;

BEGIN

SELECT COUNT(\*) INTO CHECK\_ID FROM FLIGHTS WHERE ID=NEW.ID;

IF CHECK\_ID >0 THEN

RAISE 'THAT ALREADY EXIST';

END IF;

RETURN NEW;

END;

$$

LANGUAGE PLPGSQL;

CREATE TRIGGER FLIGHT\_CHECK

BEFORE INSERT ON FLIGHTS

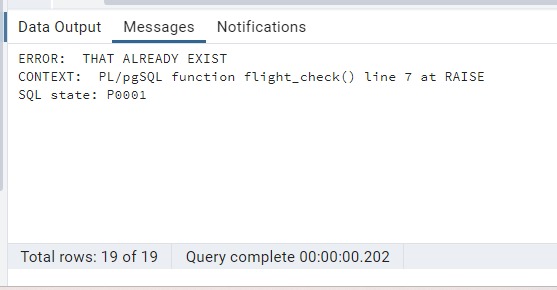
FOR EACH ROW

EXECUTE PROCEDURE FLIGHT\_CHECK();

INSERT INTO FLIGHTS (ID, FROMLOCATION, TOLOCATION, DURATION, DEPARTURETIME, ARRIVALTIME, FIRSTCLASSSEATS, BUSINESSCLASSSEATS, ECONOMICALCLASSSEATS, FIRSTCLASSPRICE, BUSINESSCLASSPRICE, ECONOMICALCLASSPRICE, STATUS)

VALUES

('F1', 'MUMBAI', 'DELHI', '2 HOURS', '2023-10-02 08:00:00', '2023-10-02 10:00:00', 10, 20, 100, 8000, 5000, 3000, 1);



1. **GET FLIGHT DETAILS BY DURATION :**

CREATE OR REPLACE PROCEDURE GET\_FLIGHT\_DETAILS\_BY\_DURATION (DUR VARCHAR(30)) AS

$$

DECLARE

F\_ID FLIGHTS.ID%TYPE;

F\_FROM FLIGHTS.FROMLOCATION%TYPE;

F\_TO FLIGHTS.TOLOCATION%TYPE;

F\_DUR FLIGHTS.DURATION%TYPE;

C1 CURSOR FOR SELECT ID, FROMLOCATION, TOLOCATION, DURATION FROM FLIGHTS WHERE DURATION = DUR;

BEGIN

OPEN C1;

LOOP

FETCH C1 INTO F\_ID, F\_FROM, F\_TO, F\_DUR;

EXIT WHEN NOT FOUND;

RAISE NOTICE 'FLIGHT ID : %

FLIGHT FROM : %

FLIGHT TO : %

FLIGHT DUR : %', F\_ID, F\_FROM, F\_TO, F\_DUR;

END LOOP;

CLOSE C1;

END;

$$

LANGUAGE PLPGSQL;

CALL GET\_FLIGHT\_DETAILS\_BY\_DURATION('2 HOURS');

