CS F212 – DATABASE SYSTEMS

Airline reservation system

MiniProject 2

Name	ID Number	Contribution
ANISETTI KARTHIK	2021AAPS2145H	33.3%
MANISHA JAKKULA	2021A4PS3064H	33.3%
HASINI VEMULA	2021A8PS1960H	33.3%

GROUP ID - 32

A)Creation Of Tables and inserting tuples.

- We create a table of 9 attributes for the Airline Reservation system. And we inserted six tuples into the table.
- **Code and Output :**

SQL Worksheet

```
CREATE TABLE reservations (
    flight_id NUMBER ,
    passenger_name VARCHAR2(50),
    departure_date DATE,
    boarding_station VARCHAR2(100),
    ticket_price NUMBER,
    number_of_seats NUMBER,
    present_cost NUMBER,
    percent_increase NUMBER,
    present_cost_after_hike NUMBER
)
```

```
Table created.
```

SQL Worksheet

Clear Find

```
INSERT INTO reservations (flight_id, passenger_name, departure_date, boarding_station, ticket_price, number_of_seats, present_cost, percent_increase ,present_cost_after_hike )
VALUES (101, 'Karthik', TO_DATE('2023-12-01', 'YYYY-MM-DD'), 'Station A', 100.0, 2, 200.0, 0.0, 0.0);
INSERT INTO reservations (flight_id, passenger_name, departure_date, boarding_station, ticket_price, number_of_seats, present_cost, percent_increase ,present_cost_after_hike)
VALUES (202, 'Manisha', TO_DATE('2023-12-02', 'YYYY-MM-DD'), 'Station B', 150.0, 3, 450.0, 0.0, 0.0 );
INSERT INTO reservations (flight_id, passenger_name, departure_date, boarding_station, ticket_price, number_of_seats, present_cost, percent_increase ,present_cost_after_hike)
VALUES (303, 'Hasini', TO_DATE('2023-12-03', 'YYYY-MM-DD'), 'Station C', 120.0, 4, 480.0, 0.0, 0.0);
INSERT INTO reservations (flight_id, passenger_name, departure_date, boarding_station, ticket_price, number_of_seats, present_cost, percent_increase ,present_cost_after_hike)
VALUES (404, 'Meghana', TO_DATE('2023-12-04', 'YYYY-MM-DD'), 'Station C', 120.0, 4, 480.0, 0.0, 0.0);
INSERT INTO reservations (flight_id, passenger_name, departure_date, boarding_station, ticket_price, number_of_seats, present_cost, percent_increase ,present_cost_after_hike)
VALUES (202, 'Deva', TO_DATE('2023-12-02', 'YYYY-MM-DD'), 'Station B', 150.0, 2, 300.0, 0.0, 0.0);
INSERT INTO reservations (flight_id, passenger_name, departure_date, boarding_station, ticket_price, number_of_seats, present_cost, percent_increase ,present_cost_after_hike)
VALUES (101, 'Rocky', TO_DATE('2023-12-01', 'YYYY-MM-DD'), 'Station A', 100.0, 1, 100.0, 0.0, 0.0);
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

B)DML Operation procedures and Package.

1. **Package Creation:** We've created a PL/SQL package called **ReservationPackage** to handle various aspects of flight reservations. This package consists of three procedures aimed at managing reservations. The first procedure for booking seats allows individuals to reserve tickets. The second procedure is for those who want to cancel reservations. The third procedure is an update procedure, which comes in handy when there's a need to modify existing reservations, such as adjusting the date or time of the flight.

```
VALUES (101,  'Rocky' , TO_DATE( 2023-12-01 ,  'YYYY-MM-DD
CREATE OR REPLACE PACKAGE ReservationPackage AS
    -- Procedure 1
    PROCEDURE InsertReservations (
        reservations_data SYS.ODCIVARCHAR2LIST
    );

    -- Procedure 2
    PROCEDURE DeleteReservations (
        reservations_data SYS.ODCIVARCHAR2LIST
    );

    -- Procedure 3
    PROCEDURE UpdateReservationValues  (
        flight_ids IN SYS.ODCINUMBERLIST,
        new_ticket_prices IN SYS.ODCINUMBERLIST,
        new_number_of_seats IN SYS.ODCINUMBERLIST
    );
END ReservationPackage;
```

Procedure 1:(Insert Reservations Procedure) This procedure facilitates adding new reservations to the system. It accepts relevant details, such as flight ID, passenger name, departure date, boarding station, ticket price, number of seats, present cost, percent increase, and present cost after a potential price hike.

SQL Worksheet

```
    );
END ReservationPackage;
/
CREATE OR REPLACE PACKAGE BODY ReservationPackage AS
    -- Procedure 1
    PROCEDURE InsertReservations (
        reservations_data SYS.ODCIVARCHAR2LIST
    ) IS
    BEGIN
        FOR i IN 1..reservations_data.COUNT / 9
        LOOP
            INSERT INTO reservations (
                flight_id,
                passenger_name,
                departure_date,
                boarding_station,
                ticket_price,
                number_of_seats,
                present_cost,
                percent_increase,
                present_cost_after_hike
            )
            VALUES (
                reservations_data(i * 9 - 8),
                reservations_data(i * 9 - 7),
                reservations_data(i * 9 - 6),
                reservations_data(i * 9 - 5),
                reservations_data(i * 9 - 4),
                reservations_data(i * 9 - 3),
                reservations_data(i * 9 - 2),
                reservations_data(i * 9 - 1),
                reservations_data(i * 9)
            );
        END LOOP;
        COMMIT;
    EXCEPTION
        WHEN OTHERS THEN
            ROLLBACK;
            DBMS_OUTPUT.PUT_LINE('Error inserting reservations: ' || SQLERRM);
    END InsertReservations;

```

```
Table created.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

Package created.

Package Body created.
```

Procedure 2 : (Delete Reservations Procedure)

Designed to handle reservation cancellations, this procedure takes passenger names as input and removes the corresponding reservations from the system.

```
    -- Procedure 2
    PROCEDURE DeleteReservations (
        reservations_data SYS.ODCIVARCHAR2LIST
    ) IS
    BEGIN
        FOR i IN 1..reservations_data.COUNT
        LOOP
            DELETE FROM reservations
            WHERE passenger_name = reservations_data(i);
        END LOOP;

        COMMIT;
    EXCEPTION
        WHEN OTHERS THEN
            ROLLBACK;
            DBMS_OUTPUT.PUT_LINE('Error deleting reservations: ' || SQLERRM);
    END DeleteReservations;
END ReservationPackage;
/
```

Procedure 3: (Update Reservations Procedure)

This allows users to modify reservations with updated information. It considers various parameters, including flight ID, passenger name, departure date, boarding station, ticket price, number of seats, present cost, percent increase, and present cost after a potential price hike.

```
    PROCEDURE UpdateReservationValues (
        flight_ids IN SYS.ODCINUMBERLIST,
        new_ticket_prices IN SYS.ODCINUMBERLIST,
        new_number_of_seats IN SYS.ODCINUMBERLIST
    ) IS
    BEGIN
        IF flight_ids.COUNT <> new_ticket_prices.COUNT OR flight_ids.COUNT <> new_number_of_seats.COUNT THEN
            RAISE_APPLICATION_ERROR(-20001, 'Input arrays must have the same number of elements');
        END IF;

        FOR i IN 1..flight_ids.COUNT
        LOOP
            UPDATE reservations
            SET
                ticket_price = new_ticket_prices(i),
                number_of_seats = new_number_of_seats(i)
            WHERE flight_id = flight_ids(i);
        END LOOP;

        COMMIT;
    EXCEPTION
        WHEN OTHERS THEN
            ROLLBACK;
            DBMS_OUTPUT.PUT_LINE('Error updating reservation values: ' || SQLERRM);
    END UpdateReservationValues;
END ReservationPackage;
/
```

C)Before and After Triggers for inserting, updating, and deleting.

For Procedure 1 : BEFORE and AFTER TRIGGERS For InsertReservations.

```
31
32  CREATE OR REPLACE TRIGGER before_insert_reservations
33  BEFORE INSERT ON reservations
34  FOR EACH ROW
35  BEGIN
36    DBMS_OUTPUT.PUT_LINE('About to insert reservations');
37  END before_insert_reservations;
38  /
39
40  CREATE OR REPLACE TRIGGER after_insert_reservations
41  AFTER INSERT ON reservations
42  FOR EACH ROW
43  BEGIN
44    DBMS_OUTPUT.PUT_LINE('Success!! Inserted some reservations');
45  END after_insert_reservations;
46  /
```

For Procedure 2 : BEFORE and AFTER TRIGGERS FOR DeleteReservations.

```
147
148 CREATE OR REPLACE TRIGGER before_delete_reservations
149 BEFORE DELETE ON reservations
150 FOR EACH ROW
151 BEGIN
152   DBMS_OUTPUT.PUT_LINE('About to delete reservations under the passenger name: ' || :OLD.passenger_name);
153 END before_delete_reservations;
154 /
155
156 CREATE OR REPLACE TRIGGER after_delete_reservations
157 AFTER DELETE ON reservations
158 FOR EACH ROW
159 BEGIN
160   DBMS_OUTPUT.PUT_LINE('Reservation with ' || :OLD.passenger_name || ' deleted successfully.');
161 END after_delete_reservations;
162 /
163
```

For Procedure 3 : BEFORE and AFTER TRIGGERS FOR UpdateReservationValues.

```
CREATE OR REPLACE TRIGGER before_update_reservation_values
BEFORE UPDATE ON reservations
FOR EACH ROW
BEGIN
  DBMS_OUTPUT.PUT_LINE('About to update reservation values for flight_id: ' || :OLD.flight_id);
END before_update_reservation_values;
/

CREATE OR REPLACE TRIGGER after_update_reservation_values
AFTER UPDATE ON reservations
FOR EACH ROW
BEGIN
  DBMS_OUTPUT.PUT_LINE('Reservation values for flight_id ' || :OLD.flight_id || ' updated successfully.');
END after_update_reservation_values;
/
```

D) Test codes and Outputs for 3 procedures.

Test Code For Procedure -1 :

```
SELECT * FROM reservations;
DECLARE
    reservations_data SYS.ODCIVARCHAR2LIST := SYS.ODCIVARCHAR2LIST(1010, 'Karthik', TO_DATE('2023-12-01', 'YYYY-MM-DD'), 'Station A', 100.0, 2, 200.0, 0.0, 0.0,
BEGIN
    ReservationPackage.InsertReservations(reservations_data);
END;
/
SELECT * FROM reservations;
```

Before Procedure -1 :
Initial Table :

FLIGHT_ID	PASSENGER_NAME	DEPARTURE_DATE	BOARDING_STATION	TICKET_PRICE	NUMBER_OF_SEATS	PRESENT_COST	PERCENT_INCREASE	PRESENT_COST_AFTER_HIKE
101	Karthik	01-DEC-23	Station A	100	2	200	0	0
202	Manisha	02-DEC-23	Station B	150	3	450	0	0
303	Hasini	03-DEC-23	Station C	120	4	480	0	0
404	Meghana	04-DEC-23	Station C	120	4	480	0	0
202	Deva	02-DEC-23	Station B	150	2	300	0	0
101	Rocky	01-DEC-23	Station A	100	1	100	0	0

Download CSV

After Procedure -1 : Inserting reservations :

FLIGHT_ID	PASSENGER_NAME	DEPARTURE_DATE	BOARDING_STATION	TICKET_PRICE	NUMBER_OF_SEATS	PRESENT_COST	PERCENT_INCREASE	PRESENT_COST_AFTER_HIKE
101	Karthik	01-DEC-23	Station A	100	2	200	0	0
202	Manisha	02-DEC-23	Station B	150	3	450	0	0
303	Hasini	03-DEC-23	Station C	120	4	480	0	0
404	Meghana	04-DEC-23	Station C	120	4	480	0	0
202	Deva	02-DEC-23	Station B	150	2	300	0	0
101	Rocky	01-DEC-23	Station A	100	1	100	0	0
1010	Karthik	01-DEC-23	Station A	100	2	200	0	0
2023	Manisha	02-DEC-23	Station B	150	3	450	0	0

After Procedure -1: Triggers output :

```
6 rows selected.


Statement processed.
About to insert reservations
Success!! Inserted some reservations
About to insert reservations
Success!! Inserted some reservations
```

Test Code For Procedure -2 :

SQL Worksheet

```
1  SELECT * FROM RESERVATIONS;
2  DECLARE
3      reservations_data SYS.ODCIVARCHAR2LIST := SYS.ODCIVARCHAR2LIST('Hasini', 'Meghana');
4  BEGIN
5      ReservationPackage.DeleteReservations(reservations_data);
6  END;
7  /
8  SELECT * FROM RESERVATIONS;
9
```

Before Procedure -2 :
Initial Table :

FLIGHT_ID	PASSENGER_NAME	DEPARTURE_DATE	BOARDING_STATION	TICKET_PRICE	NUMBER_OF_SEATS	PRESENT_COST	PERCENT_INCREASE	PRESENT_COST_AFTER_HIKE
101	Karthik	01-DEC-23	Station A	100	2	200	0	0
202	Manisha	02-DEC-23	Station B	150	3	450	0	0
303	Hasini	03-DEC-23	Station C	120	4	480	0	0
404	Meghana	04-DEC-23	Station C	120	4	480	0	0
202	Deva	02-DEC-23	Station B	150	2	300	0	0
101	Rocky	01-DEC-23	Station A	100	1	100	0	0
1010	Karthik	01-DEC-23	Station A	100	2	200	0	0
2023	Manisha	02-DEC-23	Station B	150	3	450	0	0

Download CSV

After Procedure -2: Deleting reservations of Hasini And Meghana:

FLIGHT_ID	PASSENGER_NAME	DEPARTURE_DATE	BOARDING_STATION	TICKET_PRICE	NUMBER_OF_SEATS	PRESENT_COST	PERCENT_INCREASE	PRESENT_COST_AFTER_HIKE
101	Karthik	01-DEC-23	Station A	100	2	200	0	0
202	Manisha	02-DEC-23	Station B	150	3	450	0	0
202	Deva	02-DEC-23	Station B	150	2	300	0	0
101	Rocky	01-DEC-23	Station A	100	1	100	0	0
1010	Karthik	01-DEC-23	Station A	100	2	200	0	0
2023	Manisha	02-DEC-23	Station B	150	3	450	0	0

After Procedure -2: Triggers output :

Download CSV

```
8 rows selected.


Statement processed.
About to delete reservations under the passenger name: Hasini
Reservation with Hasini deleted successfully.
About to delete reservations under the passenger name: Meghana
Reservation with Meghana deleted successfully.
```

Test Code For Procedure -3 : UpdateReservationValues:

```
SELECT * FROM RESERVATIONS;
DECLARE
    flight_ids SYS.ODCINUMBERLIST := SYS.ODCINUMBERLIST(101, 202, 303);
    new_ticket_prices SYS.ODCINUMBERLIST := SYS.ODCINUMBERLIST(120.0, 180.0, 150.0);
    new_number_of_seats SYS.ODCINUMBERLIST := SYS.ODCINUMBERLIST(3, 4, 2);
BEGIN
    ReservationPackage.UpdateReservationValues(flight_ids, new_ticket_prices, new_number_of_seats);
END;
/
SELECT * FROM RESERVATIONS;
```

Before Procedure -3 :
Initial Table :

FLIGHT_ID	PASSENGER_NAME	DEPARTURE_DATE	BOARDING_STATION	TICKET_PRICE	NUMBER_OF_SEATS	PRESENT_COST	PERCENT_INCREASE	PRESENT_COST_AFTER_HIKE
101	Karthik	01-DEC-23	Station A	100	2	200	0	0
202	Manisha	02-DEC-23	Station B	150	3	450	0	0
303	Hasini	03-DEC-23	Station C	120	4	480	0	0
404	Meghana	04-DEC-23	Station C	120	4	480	0	0
202	Deva	02-DEC-23	Station B	150	2	300	0	0
101	Rocky	01-DEC-23	Station A	100	1	100	0	0

Download CSV

After Procedure -3 Updating ticket price of 101 ,202 and 303 and new no of seats :

FLIGHT_ID	PASSENGER_NAME	DEPARTURE_DATE	BOARDING_STATION	TICKET_PRICE	NUMBER_OF_SEATS	PRESENT_COST	PERCENT_INCREASE	PRESENT_COST_AFTER_HIKE
101	Karthik	01-DEC-23	Station A	120	3	200	0	0
202	Manisha	02-DEC-23	Station B	180	4	450	0	0
303	Hasini	03-DEC-23	Station C	150	2	480	0	0
404	Meghana	04-DEC-23	Station C	120	4	480	0	0
202	Deva	02-DEC-23	Station B	180	4	300	0	0
101	Rocky	01-DEC-23	Station A	120	3	100	0	0

Download CSV

After Procedure -3: Triggers output :

```
Statement processed.
About to update reservation values for flight_id: 101
Reservation values for flight_id 101 updated successfully.
About to update reservation values for flight_id: 101
Reservation values for flight_id 101 updated successfully.
About to update reservation values for flight_id: 202
Reservation values for flight_id 202 updated successfully.
About to update reservation values for flight_id: 202
Reservation values for flight_id 202 updated successfully.
About to update reservation values for flight_id: 303
Reservation values for flight_id 303 updated successfully.
```

Conclusions :

Hence, our code for the Airline Reservation system sets up a table named `reservations` to keep track of flight reservations. It stores details like `flight_id`, `passenger_name,` `departure_date`, `boarding_station`, `ticket_price`, `number_of_seats`, `present_cost`, `percent_increase`, and `present_cost_after_hike`.

A package called `ReservationPackage` is created, housing procedures for adding (`InsertReservations`), removing (`DeleteReservations`), and adjusting (`UpdateReservationValues`) reservation information. These procedures take in data using PL/SQL collection types like `SYS.ODCIVARCHAR2LIST` and `SYS.ODCINUMBERLIST`.

Triggers are defined to respond to events like inserting, deleting, and updating reservations. These triggers print messages through `DBMS_OUTPUT` to inform about these events.

There's code to showcase how to use the procedures and triggers for testing. It covers adding initial reservation data, making updates to reservation values for multiple entries, and displaying the resulting reservations.