A Project Report on

Flight Management System

for Database Management Systems (UCS310)

by

Akshat Singhvi	**102303248**
Aishani Shreya	**102303250**
Hardik Tandon	**102303252**
Zorawar Singh	**102303238**

Sub-Group: 2C21 (2C2A)

Submitted to

Dr. Geeta Kasana

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY

(A DEEMED TO BE UNIVERSITY)

PATIALA, PUNJAB

INDIA

Jan-May 2025

INDEX

Problem Statement

Design a robust Oracle SQL/PLSQL database system to address the complexity of managing airline operations, streamlining your flight scheduling, passenger bookings, crew assignments, aeroplane capacity tracking, and airport infrastructure coordination, ensuring data integrity, automated business rule validation, and comprehensive reporting to enhance your operational efficiency and decision-making. Your system must centralize data for all relevant entities and their relationships, support efficient data insertion, retrieval, and deletion, and enforce critical business rules, such as validating flight types (Domestic/International), ensuring appropriate crew assignments, and standardizing contact information formats. You are required to implement automated procedures to handle tasks like generating tickets, calculating capacities, and querying flight details, with user-friendly error handling and optimized performance for large datasets. Focused on core airline operations without external integrations, your system aims to serve airline administrators, airport staff, and passengers with a reliable, scalable, and intuitive solution that minimizes errors and supports your operational planning.

Tables Before Normalisation

Terminals			
Terminals_ID (PK) INT	Terminal_No INT	Terminal_Name VARCHAR2(20)	Airport_ID INT

Runways			
Runways_ID (PK) INT	Runway_No INT	Runway_Name VARCHAR2(20)	Airport_ID INT

Passengers					
Passengers_ID (PK) INT	Flight_ID INT	Passenger_Name VARCHAR2(50)	Passenger_Gender CHAR(1)	Passenger_Age INT	Passenger_Phone_No INT

Flight_Crew						
Flight_Crew_ID (PK) INT	Flight _ID INT	Pilot _ID INT	Copilot _ID INT	Number_Of _Airhostesses INT	Head_Airhostess _ID INT	Flight_Crew_Hostes s_ID INT

Airline_Crew					
Crew_ID (PK) INT	Crew_First_Name VARCHAR2(50)	Crew_Last_Name VARCHAR2(50)	Crew_Gender CHAR(1)	Crew_Country VARCHAR2(20)	Airline_ID INT

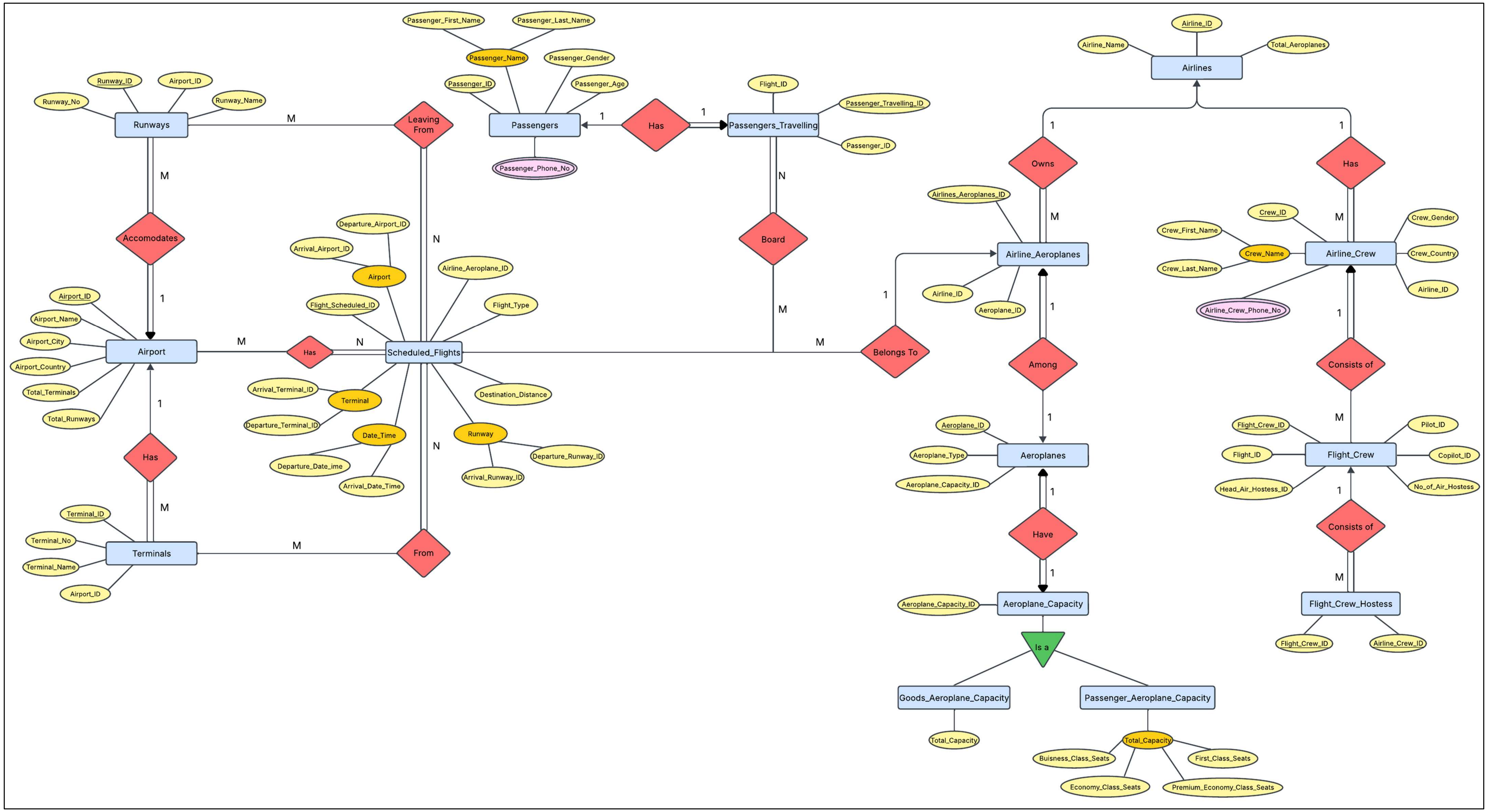
Airlines	
Airline_ID (PK) INT	Airline_Name VARCHAR2(50)

Airports					
Airport_ID (PK) INT	Airport_Name VARCHAR2(60)	Airport_City VARCHAR2(50)	Airport_Country VARCHAR2(50)	Total_Terminals INT	Total_Runways INT

Scheduled_Flights					
Flights _Scheduled_ID (PK) INT	Aeroplanes _ID INT	Departure_ Airport _ID INT	Arrival_Airport _ID INT	Flight_Type VARCHAR2 (50)	Arrival _Time TIMESTAMP
Departure _Time TIMESTAMP	Destination _Distance INT	Arrival_ Terminal _ID INT	Departure _Terminal_ID INT	Arrival _Runway_ID INT	Departure _Runway_ID INT
Airlines_ID INT	Departure _Date Date	Arrival_Date Date			

Aeroplanes			
Aeroplanes_ID (PK) INT	Aeroplane_Type VARCHAR2(30)	Aeroplane_Capacity_ID INT	Total_Capacity INT
First_Class INT	Economy_Class_Capacity INT	Buisiness_Class INT	Premium_Economy_Class INT

ER Diagram



Passenger_First_Name
Passenger_Last_Name
Passenger_Name
Passenger_Gender
Passenger_ID
Passenger_Age
Passengers
Passenger_Phone_No
Has
1
1
Passengers_Travelling
Flight_ID
Passenger_Travelling_ID
Passenger_ID
N
Board
M
Runway_ID
Airport_ID
Runway_No
Runway_Name
Runways
M
Leaving From
M
Accomodates
1
N
Departure_Airport_ID
Arrival_Airport_ID
Airport
Airline_Aeroplane_ID
Flight_Scheduled_ID
Flight_Type
Airport_ID
Airport_Name
Airport_City
Airport_Country
Total_Terminals
Total_Runways
Airport
M
Has
N
Scheduled_Flights
M
Belongs To
1
Arrival_Terminal_ID
Terminal
Departure_Terminal_ID
Destination_Distance
Date_Time
Departure_Date_ime
Arrival_Date_Time
Runway
Departure_Runway_ID
Arrival_Runway_ID
N
1
Has
M
Terminal_ID
Terminal_No
Terminal_Name
Airport_ID
Terminals
M
From
Airline_ID
Airline_Name
Total_Aeroplanes
Airlines
1
Owns
1
Has
Airlines_Aeroplanes_ID
M
Airline_Aeroplanes
Airline_ID
Aeroplane_ID
1
Among
1
Crew_ID
Crew_Gender
Crew_First_Name
Crew_Name
Airline_Crew
Crew_Country
Crew_Last_Name
Airline_ID
Airline_Crew_Phone_No
M
1
Consists of
M
Aeroplane_ID
Aeroplane_Type
Aeroplanes
Aeroplane_Capacity_ID
1
Have
1
Aeroplane_Capacity_ID
Aeroplane_Capacity
Is a
Goods_Aeroplane_Capacity
Total_Capacity
Passenger_Aeroplane_Capacity
Total_Capacity
Buisness_Class_Seats
First_Class_Seats
Economy_Class_Seats
Premium_Economy_Class_Seats
Flight_Crew_ID
Pilot_ID
Flight_ID
Flight_Crew
Copilot_ID
Head_Air_Hostess_ID
No_of_Air_Hostess
1
Consists of
M
Flight_Crew_Hostess
Flight_Crew_ID
Airline_Crew_ID

ER to Table (BCNF-Normalised)

Airlines

Airline_ID (PK)	Airline_Name	Total_Aeroplanes
INT	VARCHAR2(50)	INT

Airline_Crew

Crew_ID (PK)	Crew_First_Name	Crew_Last_Name	Crew_Gender	Crew_Country	Airline_ID
INT	VARCHAR2(50)	VARCHAR2(50)	CHAR(1)	VARCHAR2(20)	INT

Airline_Crew_Phone_No

Crew_ID (PK)	Phone_No (PK)
INT	INT

Flight_Crew

Flight_Crew_ID (PK)	Flight_ID	Pilot_ID	Copilot_ID	Number_Of_Air _Hostesses	Head_Air _Hostess_ID
INT	INT	INT	INT	INT	INT

Flight_Crew_Hostess

Flight_Crew_ID (PK)	Airline_Crew_ID (PK)
INT	INT

Airport

Airport_ID (PK)	Airport_Name	Airport_City	Airport_Country	Total_Terminals	Total_Runways
INT	VARCHAR2(100)	VARCHAR2(100)	VARCHAR2(100)	INT	INT

Terminals

Terminal_ID (PK)	Terminal_No	Terminal_Name	Airport_ID
INT	INT	VARCHAR2(15)	INT

Runways

Runway_ID (PK)	Runway_No	Runway_Name	Airport_ID
INT	INT	VARCHAR2(20)	INT

Passengers_Phone_No

Passenger_ID (PK)	Phone_Number (PK)
INT	INT

Passengers_Traveling

Passengers_Traveling_ID (PK)	Passenger_ID	Flight_ID
INT	INT	INT

Goods_Aeroplane_Capacity

Aeroplane_Capacity_ID (PK)	Total_Capacity
INT	INT

Passenger_Aeroplane_Capacity

Aeroplane_Capacity _ID (PK)	Business_Class _Seats	Economy_Class _Seats	First_Class _Seats	Premium_Economy _Class_Seats
INT	INT	INT	INT	INT

Scheduled_Flights

Flights _Scheduled_ID (PK)	Airlines_ Aeroplanes_ID	Departure_ Airport _ID	Arrival_Airport _ID	Flight_Type	Arrival_Date _Time
INT	INT	INT	INT	VARCHAR2 (50)	TIMESTAMP
Departure_Date _Time	Destination _Distance	Arrival_ Terminal _ID	Departure _Terminal_ID	Arrival _Runway_ID	Departure _Runway_ID
TIMESTAMP	INT	INT	INT	INT	INT

Passengers

Passenger_ID (PK)	Passenger_First_Name	Passenger_Last_Name	Passenger_Gender	Passenger_Age
INT	VARCHAR2(50)	VARCHAR2(50)	CHAR(1)	INT

Aeroplanes

Aeroplane_ID (PK)	Aeroplane_Type	Aeroplane_Capacity_ID
INT	VARCHAR2(10)	INT

Airline_Aeroplanes

Airline_Aeroplanes_ID (PK)	Airline_ID	Aeroplane_ID
INT	INT	INT

Foreign Keys

Child Table	Child Column	Parent Table	Parent Column
Aeroplanes	Aeroplane_Capacity_ID	Goods_Aeroplane_Capacity	Aeroplane_Capacity_ID
Aeroplanes	Aeroplane_Capacity_ID	Passenger_Aeroplane_Capacity	Aeroplane_Capacity_ID
Airline_Crew	Airline_ID	Airlines	Airline_ID
Airline_Aeroplanes	Airline_ID	Airlines	Airline_ID
Airline_Aeroplanes	Aeroplane_ID	Aeroplanes	Aeroplane_ID
Flight_Crew	Pilot_ID	Airline_Crew	Crew_ID
Flight_Crew	Copilot_ID	Airline_Crew	Crew_ID
Flight_Crew	Head_Air_Hostess_ID	Airline_Crew	Crew_ID
Flight_Crew	Flight_ID	Scheduled_Flights	Flight_Scheduled_ID
Flight_Crew_Hostess	Flight_Crew_ID	Flight_Crew	Flight_Crew_ID
Flight_Crew_Hostess	Airline_Crew_ID	Airline_Crew	Crew_ID
Scheduled_Flights	Airline_Aeroplane_ID	Airline_Aeroplanes	Airline_Aeroplanes_ID
Scheduled_Flights	Departure_Airport_ID	Airport	Airport_ID
Scheduled_Flights	Arrival_Airport_ID	Airport	Airport_ID
Scheduled_Flights	Arrival_Terminal_ID	Terminals	Terminal_ID
Scheduled_Flights	Departure_Terminal_ID	Terminals	Terminal_ID
Scheduled_Flights	Arrival_Runway_ID	Runways	Runway_ID
Scheduled_Flights	Departure_Runway_ID	Runways	Runway_ID
Passengers_Traveling	Passenger_ID	Passengers	Passenger_ID
Passengers_Traveling	Flight_ID	Scheduled_Flights	Flight_Scheduled_ID
Passenger_Phone_No	Passenger_ID	Passengers	Passenger_ID
Runways	Airport_ID	Airport	Airport_ID
Terminals	Airport_ID	Airport	Airport_ID

External (User) View Procedures and Triggers

Procedures

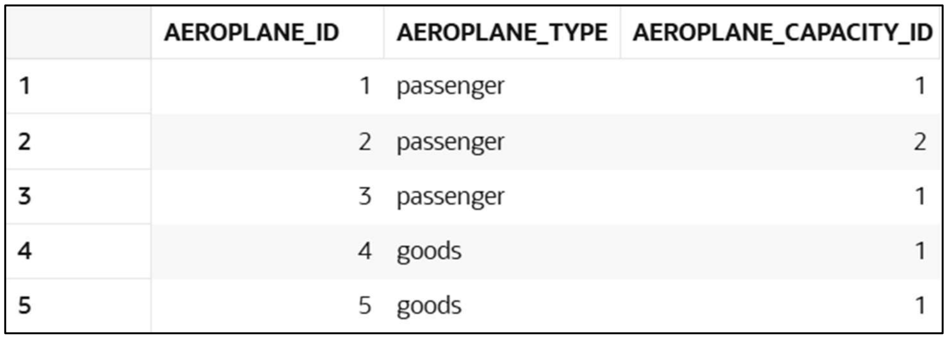
Name	Description
Insert_Passenger_With_Phones	Inserts a passenger and their phone numbers (comma-separated) into the respective tables, handling duplicates and errors.
Insert_Crew_With_Phones	Inserts a crew member and their phone numbers (comma-separated) into the respective tables, handling duplicates and errors.
Print_Passenger_Details	Prints details of all passengers, including their phone numbers, using cursors.
Print_Passenger_Details_pk	Prints details of a specific passenger by ID, including phone numbers, or displays 'N/A' if no phone numbers exist.
Print_All_Passengers_For_Flight	Prints details of all passengers booked on a specific flight by calling Print_Passenger_Details_pk for each passenger.
Print_Crew_Details	Prints details of all crew members, including their phone numbers, or 'N/A' if none exist.
Print_Crew_Details_pk	Prints details of a specific crew member by ID, including phone numbers, or 'N/A' if none exist.
Print_Crew_For_Flight	Prints details of all crew members (pilot, copilot, head hostess, hostesses) for a specific flight, including phone numbers.
Print_Crew_Details_By_Airline	Prints details of all crew members for a specific airline, including phone numbers, or 'N/A' if none exist.
Calculate_Aeroplane_Capacity	Calculates and prints the total capacity of an aeroplane (goods or passenger) based on its type and capacity ID.
Calculate_Flight_Load	Calculates and prints the load percentage of a flight based on capacity and current passengers/goods.
Print_Flights_Same_Departure_Airport	Prints details of all flights departing from a specific airport, including count and flight details.
Print_Flights_Same_Arrival_Airport	Prints details of all flights arriving at a specific airport, including count and flight details.
Print_Flights_Between_Airports	Prints details of flights between two specific airports, including count and flight details.
Print_Flights_On_Date	Prints details of flights departing or arriving at an airport on a specific date, including count and flight details.
Count_Flights_Of_Airline	Counts and prints details of flights for an airline at an airport within a date range.
Print_Flight_Details_By_ID	Prints detailed information about a specific flight by ID, including airport, terminal, runway, and airline details.
Find_Flights_From_Country_To_Country	Prints details of flights between two countries, including flight ID, type, and airport names.
Print_Ticket	Prints a ticket with passenger and flight details for a specific passenger and flight, verifying booking.

Triggers

Name	Description
trg_Check_Flight_Type_Smart	Ensures flight type matches airport countries (Domestic for same country, International for different).
trg_Check_Terminals_Runways	Ensures airports have at least one terminal and runway.
trg_Check_Destination_Distance	Ensures flight destination distance is greater than zero.
trg_Check_Hostess_Insert	Ensures the number of air hostesses does not exceed the allowed limit for a flight.
trg_Check_Passenger_Phone_Digits	Ensures passenger phone numbers are exactly 5 digits.
trg_Check_Crew_Phone_Digits	Ensures crew phone numbers are exactly 5 digits.
trg_Check_PassengerPlane_Capacity	Ensures total passenger plane capacity (sum of seat classes) is greater than zero.
trg_Check_Aeroplane_Capacity	Ensures aeroplane capacity ID exists in the appropriate capacity table (goods or passenger) based on aeroplane type.

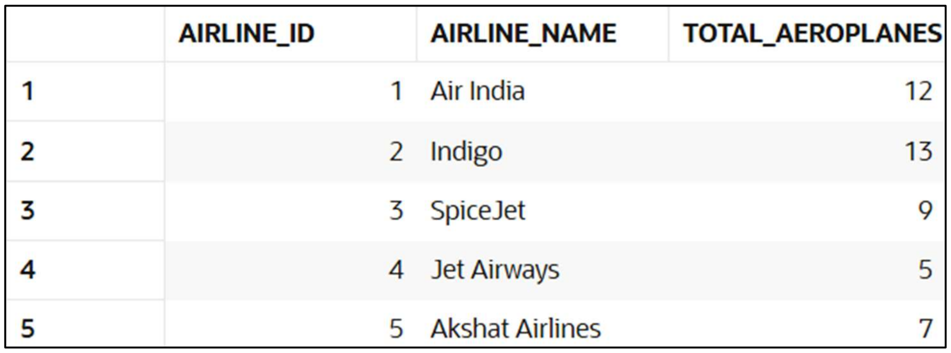
SQL Query Snapshots

1. To print all the records in Aeroplanes table.
   Query: `SELECT * FROM AEROPLANES;`



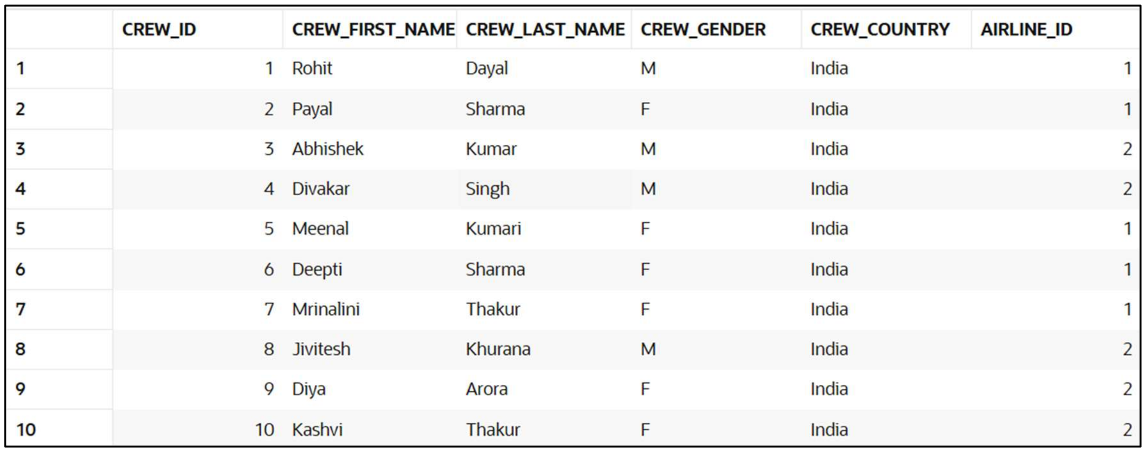
	AEROPLANE_ID	AEROPLANE_TYPE	AEROPLANE_CAPACITY_ID
1	1	passenger	1
2	2	passenger	2
3	3	passenger	1
4	4	goods	1
5	5	goods	1

2. To print all the records in Airlines table.
   Query: `SELECT * FROM AIRLINES;`



	AIRLINE_ID	AIRLINE_NAME	TOTAL_AEROPLANES
1	1	Air India	12
2	2	Indigo	13
3	3	SpiceJet	9
4	4	Jet Airways	5
5	5	Akshat Airlines	7

3. To print all the records of Airline Crew table.
   Query: `SELECT * FROM AIRLINE_CREW;`



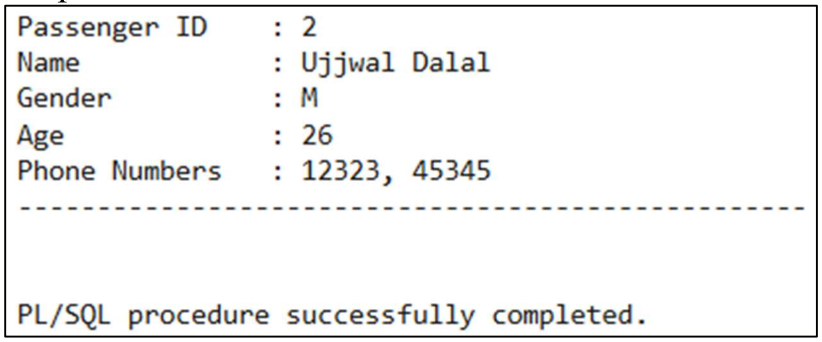
	CREW_ID	CREW_FIRST_NAME	CREW_LAST_NAME	CREW_GENDER	CREW_COUNTRY	AIRLINE_ID
1	1	Rohit	Dayal	M	India	1
2	2	Payal	Sharma	F	India	1
3	3	Abhishek	Kumar	M	India	2
4	4	Divakar	Singh	M	India	2
5	5	Meenal	Kumari	F	India	1
6	6	Deepti	Sharma	F	India	1
7	7	Mrinalini	Thakur	F	India	1
8	8	Jivitesh	Khurana	M	India	2
9	9	Diya	Arora	F	India	2
10	10	Kashvi	Thakur	F	India	2

PL/SQL Snapshots

1. To print Passenger details by ID.

```
CREATE OR REPLACE PROCEDURE Print_Passenger_Details_pk(p_passenger_id
IN INT)
AS
    CURSOR passenger_cur IS
        SELECT Passenger_ID, Passenger_First_Name,
Passenger_Last_Name, Passenger_Gender, Passenger_Age
        FROM Passengers
        WHERE Passenger_ID = p_passenger_id;

    CURSOR phone_cur IS
        SELECT Phone_No
        FROM Passenger_Phone_No
        WHERE Passenger_ID = p_passenger_id;

    v_passenger_rec passenger_cur%ROWTYPE;
    v_phone Passenger_Phone_No.Phone_No%TYPE;
    no_phone_numbers EXCEPTION;
    phone_count INT := 0;
BEGIN
    OPEN passenger_cur;
    FETCH passenger_cur INTO v_passenger_rec;

    IF passenger_cur%NOTFOUND THEN
        DBMS_OUTPUT.PUT_LINE('No passenger found with Passenger_ID = '
|| p_passenger_id);
        CLOSE passenger_cur;
        RETURN;
    END IF;

    DBMS_OUTPUT.PUT_LINE('Passenger ID    : ' ||
v_passenger_rec.Passenger_ID);
    DBMS_OUTPUT.PUT_LINE('Name            : ' ||
v_passenger_rec.Passenger_First_Name || ' ' ||
NVL(v_passenger_rec.Passenger_Last_Name,''));
    DBMS_OUTPUT.PUT_LINE('Gender          : ' ||
v_passenger_rec.Passenger_Gender);
    DBMS_OUTPUT.PUT_LINE('Age             : ' ||
v_passenger_rec.Passenger_Age);
    DBMS_OUTPUT.PUT('Phone Numbers    : ');

    CLOSE passenger_cur;
    OPEN phone_cur;
    LOOP
        FETCH phone_cur INTO v_phone;
        EXIT WHEN phone_cur%NOTFOUND;
        IF phone_count > 0 THEN
            DBMS_OUTPUT.PUT(', ');
        END IF;
        DBMS_OUTPUT.PUT(v_phone);
        phone_count := phone_count + 1;
    END LOOP;

    IF phone_count = 0 THEN
        RAISE no_phone_numbers;
    END IF;

    CLOSE phone_cur;
    DBMS_OUTPUT.NEW_LINE;
    DBMS_OUTPUT.PUT_LINE('--------------------------------------------
------');

EXCEPTION
    WHEN no_phone_numbers THEN
        DBMS_OUTPUT.PUT('N/A');
        CLOSE phone_cur;
        DBMS_OUTPUT.NEW_LINE;
        DBMS_OUTPUT.PUT_LINE('----------------------------------------
----------');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Some unexpected error occurred: ' ||
SQLERRM);
END;
-- LOCAL PROGRAM
DECLARE
l_id PASSENGERS.PASSENGER_ID%TYPE;
BEGIN
l_id := &Enter_Passenger_ID;
PRINT_PASSENGER_DETAILS_PK(l_id);
END;
```

Output:

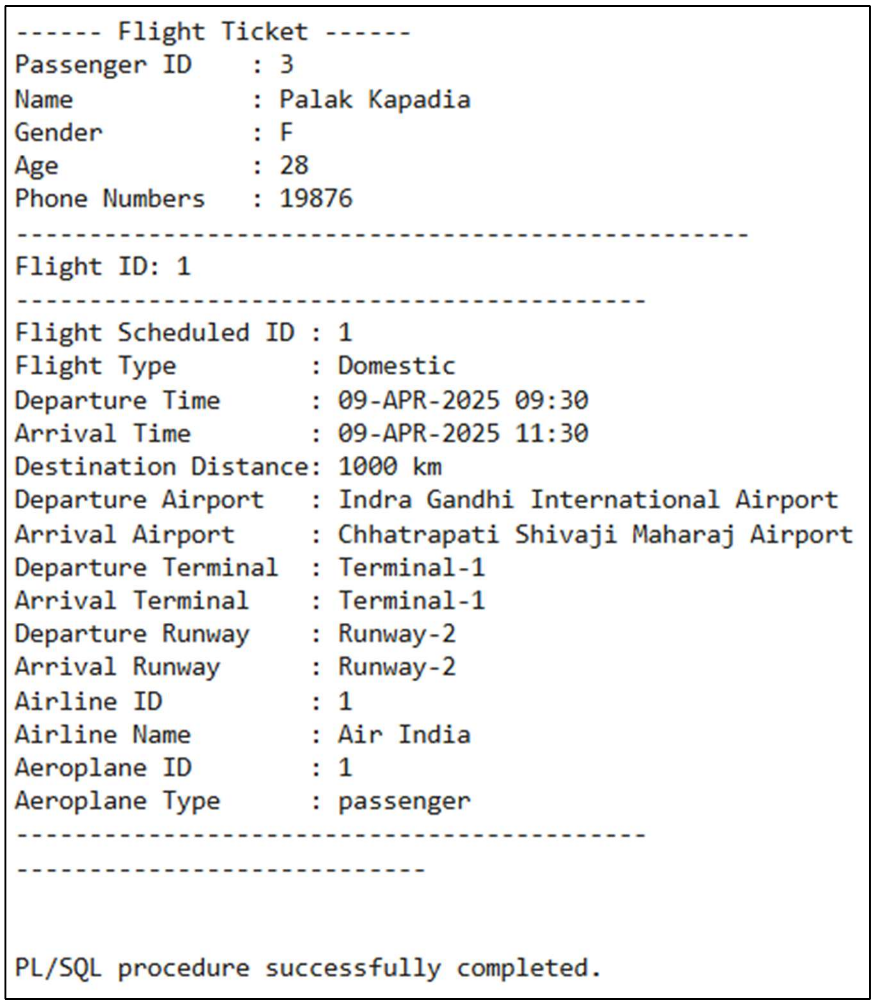


```
Passenger ID    : 2
Name            : Ujjwal Dalal
Gender          : M
Age             : 26
Phone Numbers   : 12323, 45345
---------------------------------------------


PL/SQL procedure successfully completed.
```

2. To print the ticket of Passenger for particular Flight.

```
CREATE OR REPLACE PROCEDURE Print_Ticket(
    p_Passenger_ID IN Passengers.Passenger_ID%TYPE,
    p_Flight_ID    IN Passengers_Traveling.Flight_ID%TYPE
)
IS
    v_Count NUMBER;
BEGIN

    SELECT COUNT(*)
    INTO v_Count
    FROM Passengers_Traveling
    WHERE Passenger_ID = p_Passenger_ID
      AND Flight_ID = p_Flight_ID;

    IF v_Count = 0 THEN
        DBMS_OUTPUT.PUT_LINE('Error: Passenger is not booked on the
given Flight.');
        RETURN;
    END IF;

    DBMS_OUTPUT.PUT_LINE('------ Flight Ticket ------');
    Print_Passenger_Details_pk(p_Passenger_ID);
    DBMS_OUTPUT.PUT_LINE('Flight ID: ' || p_Flight_ID);
    Print_Flight_Details_By_ID(p_Flight_ID);

    DBMS_OUTPUT.PUT_LINE('----------------------------');

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Passenger or Flight not found.');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error occurred: ' || SQLERRM);
END;
-- LOCAL PROGRAM
DECLARE
l_pid Passengers.Passenger_ID%TYPE;
l_fid Passengers_Traveling.Flight_ID%TYPE;
BEGIN
    l_pid := &Enter_Passenger_ID;
    l_fid := &Enter_Flight_ID;
    Print_Ticket(l_pid,l_fid);
END;
```

Output:

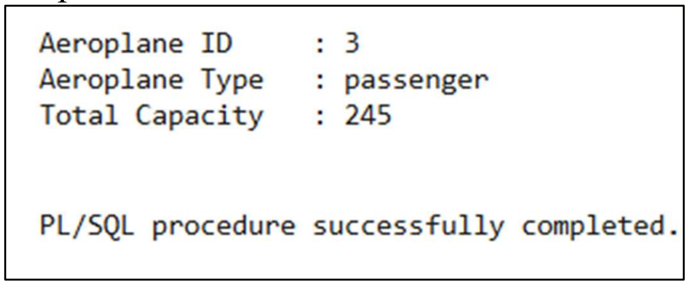


```
------ Flight Ticket ------
Passenger ID    : 3
Name            : Palak Kapadia
Gender          : F
Age             : 28
Phone Numbers   : 19876
--------------------------------------------------
Flight ID: 1
-------------------------------------------
Flight Scheduled ID : 1
Flight Type         : Domestic
Departure Time      : 09-APR-2025 09:30
Arrival Time        : 09-APR-2025 11:30
Destination Distance: 1000 km
Departure Airport   : Indra Gandhi International Airport
Arrival Airport     : Chhatrapati Shivaji Maharaj Airport
Departure Terminal  : Terminal-1
Arrival Terminal    : Terminal-1
Departure Runway    : Runway-2
Arrival Runway      : Runway-2
Airline ID          : 1
Airline Name        : Air India
Aeroplane ID        : 1
Aeroplane Type      : passenger
-------------------------------------------
----------------------------


PL/SQL procedure successfully completed.
```

3. To find/print the capacity of the given aeroplane.

```
CREATE OR REPLACE PROCEDURE
Calculate_Aeroplane_Capacity(p_aeroplane_id IN INT)
AS
    v_aeroplane_type Aeroplanes.Aeroplane_Type%TYPE;
    v_capacity_id Aeroplanes.Aeroplane_Capacity_ID%TYPE;
    v_total_capacity INT;
BEGIN
    SELECT Aeroplane_Type, Aeroplane_Capacity_ID
    INTO v_aeroplane_type, v_capacity_id
    FROM Aeroplanes
    WHERE Aeroplane_ID = p_aeroplane_id;

    IF v_aeroplane_type = 'goods' THEN
        SELECT Total_Capacity
        INTO v_total_capacity
        FROM Goods_Aeroplane_Capacity
        WHERE Aeroplane_Capacity_ID = v_capacity_id;

    ELSIF v_aeroplane_type = 'passenger' THEN
        SELECT NVL(Business_Class_Seats,0) +
NVL(Economy_Class_Seats,0) +
               NVL(Premium_Economy_Class_Seats,0) +
NVL(First_Class_Seats,0)
        INTO v_total_capacity
        FROM Passenger_Aeroplane_Capacity
        WHERE Aeroplane_Capacity_ID = v_capacity_id;
    ELSE
        DBMS_OUTPUT.PUT_LINE('Invalid Aeroplane Type');
        RETURN;
    END IF;

    DBMS_OUTPUT.PUT_LINE('Aeroplane ID      : ' || p_aeroplane_id);
    DBMS_OUTPUT.PUT_LINE('Aeroplane Type    : ' || v_aeroplane_type);
    DBMS_OUTPUT.PUT_LINE('Total Capacity    : ' || v_total_capacity);

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Aeroplane ID not found.');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Some error occurred: ' || SQLERRM);
END;
-- LOCAL PROGRAM
DECLARE
l_id INT;
BEGIN
    l_id := &Enter_Aeroplane_ID;
    Calculate_Aeroplane_Capacity(l_id);
END;
```

Output:



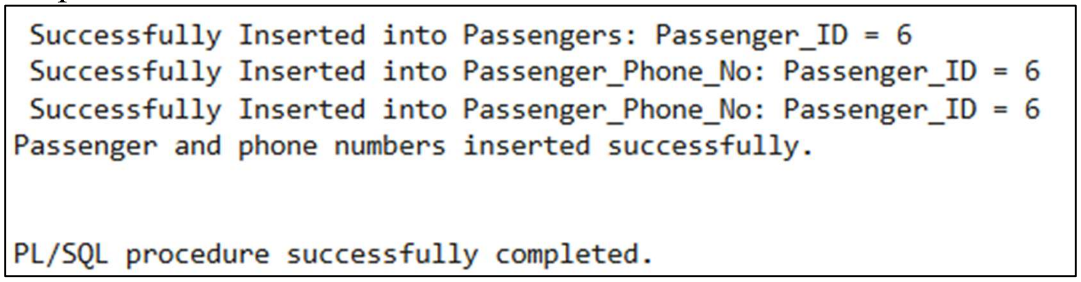
```
Aeroplane ID     : 3
Aeroplane Type   : passenger
Total Capacity   : 245


PL/SQL procedure successfully completed.
```

4. To insert Passenger with Phone Numbers.

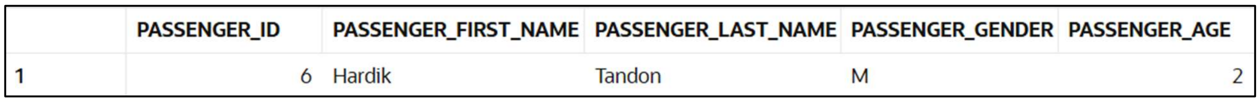
```
CREATE OR REPLACE PROCEDURE Insert_Passenger_With_Phones(
    p_Passenger_ID         IN Passengers.Passenger_ID%TYPE,
    p_First_Name           IN Passengers.Passenger_First_Name%TYPE,
    p_Last_Name            IN Passengers.Passenger_Last_Name%TYPE,
    p_Gender               IN Passengers.Passenger_Gender%TYPE,
    p_Age                  IN Passengers.Passenger_Age%TYPE,
    p_Phone_Nos            IN VARCHAR2
)
IS
    v_phone  VARCHAR2(100);
    v_start  NUMBER := 1;
    v_end    NUMBER;
BEGIN
    INSERT INTO Passengers (
        Passenger_ID,
        Passenger_First_Name,
        Passenger_Last_Name,
        Passenger_Gender,
        Passenger_Age
    ) VALUES (
        p_Passenger_ID,
        p_First_Name,
        p_Last_Name,
        p_Gender,
        p_Age
    );

    LOOP
        v_end := INSTR(p_Phone_Nos, ',', v_start);

        IF v_end = 0 THEN
            v_phone := SUBSTR(p_Phone_Nos, v_start);
            INSERT INTO Passenger_Phone_No (Passenger_ID, Phone_No)
            VALUES (p_Passenger_ID, TO_NUMBER(TRIM(v_phone)));
            EXIT;
        ELSE
            v_phone := SUBSTR(p_Phone_Nos, v_start, v_end - v_start);
            INSERT INTO Passenger_Phone_No (Passenger_ID, Phone_No)
            VALUES (p_Passenger_ID, TO_NUMBER(TRIM(v_phone)));
            v_start := v_end + 1;
        END IF;
    END LOOP;

    COMMIT;

    DBMS_OUTPUT.PUT_LINE('Passenger and phone numbers inserted
successfully.');

EXCEPTION
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE('Passenger ID or Phone number already
exists.');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Some error occurred: ' || SQLERRM);
END;
-- LOCAL PROGRAM
DECLARE
p_Passenger_ID Passengers.Passenger_ID%TYPE;
```

```
p_First_Name Passengers.Passenger_First_Name%TYPE;
p_Last_Name  Passengers.Passenger_Last_Name%TYPE;
p_Gender     Passengers.Passenger_Gender%TYPE;
p_Age        Passengers.Passenger_Age%TYPE;
p_Phone_Nos  VARCHAR2(100);
BEGIN
p_Passenger_ID := &Enter_Passenger_ID;
p_First_Name := '&Enter_First_Name';
p_Last_Name  :='&Enter_Last_Name';
p_Gender := '&Enter_Gender';
p_Age := &Enter_Age;
p_Phone_Nos  := '&Enter_Phone_Nos';
Insert_Passenger_With_Phones(
    p_Passenger_ID,
    p_First_Name,
    p_Last_Name,
    p_Gender,
    p_Age,
    p_Phone_Nos
);
END;
-- OUTPUT VERIFICATION
SELECT * FROM PASSENGERS WHERE Passenger_ID=6;
SELECT * FROM PASSENGER_PHONE_NO WHERE Passenger_ID=6;
```

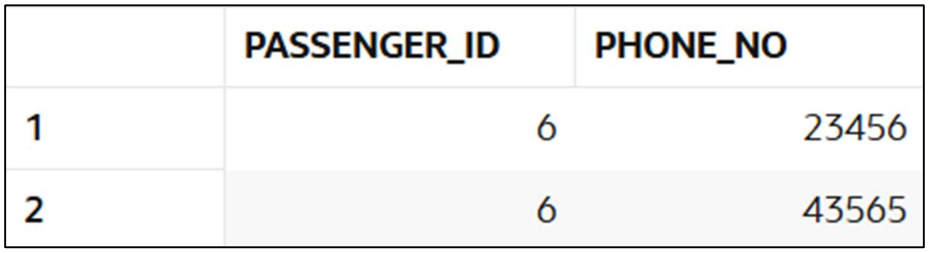
Output:



```
 Successfully Inserted into Passengers: Passenger_ID = 6
 Successfully Inserted into Passenger_Phone_No: Passenger_ID = 6
 Successfully Inserted into Passenger_Phone_No: Passenger_ID = 6
Passenger and phone numbers inserted successfully.


PL/SQL procedure successfully completed.
```



	PASSENGER_ID	PASSENGER_FIRST_NAME	PASSENGER_LAST_NAME	PASSENGER_GENDER	PASSENGER_AGE
1	6	Hardik	Tandon	M	2



	PASSENGER_ID	PHONE_NO
1	6	23456
2	6	43565

Conclusion

The Flight Management System project successfully delivers a robust database solution for managing critical airline operations, including flight scheduling, passenger bookings, crew assignments, aeroplane capacities, and airport infrastructure. Implemented using Oracle SQL/PLSQL, the system features a normalized database schema with 16 tables, ensuring data integrity through primary key, foreign key, and business rule constraints. The 19 user-defined stored procedures enable efficient data insertion, retrieval, and reporting, covering functionalities such as printing passenger and crew details, generating tickets, calculating aeroplane capacities, and querying flight schedules. The 8 validation triggers enforce business rules, such as ensuring correct flight types, valid phone numbers, and appropriate crew assignments, enhancing the system's reliability.

The project meets its objectives by providing a scalable and user-friendly platform for airline staff and airport personnel, with clear error handling and audit logging to support operational decision-making. Query and PLSQL snapshots demonstrate the system's functionality, showcasing practical applications like passenger detail retrieval and ticket generation. While the system is limited to core flight management operations and uses a simplified 5-digit phone number format, it establishes a strong foundation for future enhancements.

Future Enhancements:

- Integration with real-time flight tracking systems for dynamic scheduling.
- Addition of a user interface (e.g., web or mobile app) to improve accessibility.
- Expansion to include payment processing and loyalty program management.
- Support for international phone number formats and additional contact methods.
- Advanced analytics for optimizing flight load and crew scheduling.

The Flight Management System demonstrates the power of database management in streamlining complex airline operations, offering a reliable and extensible solution for stakeholders

References

1. A. Patil, "Airport Management System Database Design," *GitHub*. [Online]. Available: https://github.com/patilankita79/Airport-Management-System-Database-Design
2. GeeksforGeeks, "How to design database for flight reservation system?", *GeeksforGeeks*, Mar. 17, 2021. [Online]. Available: https://www.geeksforgeeks.org/how-to-design-database-for-flight-reservation-system/
3. Slideshare, "Air-line management system DBMS project," *Slideshare*. [Online]. Available: https://www.slideshare.net/slideshow/air-line-management-system-dbms-project/130343920
4. A. Poudel, "Airlines Management System – HTML, PHP Files," *GitHub*. [Online]. Available: https://github.com/ashishpoudel995/Airlines-Management-System/tree/master/HTML%2C%20PHP%20Files
5. R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, 7th ed. Pearson, 2015.
6. A. Silberschatz, H. F. Korth, and S. Sudarshan, *Database System Concepts*, 7th ed. McGraw-Hill Education, 2019.

Source Code Repository

The complete set of SQL scripts for the Flight Management System, including the database schema, stored procedures, and validation triggers, is available on GitHub. These scripts implement the functionalities described in this report, such as flight scheduling, passenger management, and ticket generation.

Link: https://github.com/HardikTandon77/UCS310-DBMS-Flight-Management-System.git

Faculty Signature