

**Project Activity Report**  
**Submitted for Artificial Intelligence**

---

**TOPIC: Sign Language Recognition**  
**System**

---

**GROUP: 2C55**

**Submitted By:**

Mitul Pabri: 102303793

**Submitted to:**

Mrs/Ms. MRINALINI KAKROO



**THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY  
(A DEEMED TO BE UNIVERSITY), PATIALA, PUNJAB, INDIA**  
**(Jan – May 2025)**

## **INDEX**

<b>Sr.No.</b>	<b>Contents</b>	<b>Page no.</b>
1	Introduction	1
2	Objective	2
3	Problem statement	3
4	Scope	4
5	Methodology	5
6	ER Diagram	6
7	ER Diagram Explanation	8
8	Normalization	12
9	Creation of tables	14
10	Inserting values	19
12	Few execution Queries	27
14	Expected Outcome	52
15	Conclusions	53

## **INTRODUCTION**

Managing an airline's complex operations requires precision, coordination, and reliable data management. The Flight Management System project addresses these challenges by creating a centralized platform to handle various facets of airline operations, including flight schedules, crew assignments, and passenger information. At the heart of this system lies the meticulous organization of flight crew data, which ensures every flight is staffed with the right personnel at the right time.

The `flightcrew.sql` script plays a vital role in this ecosystem by automating the process of inserting detailed crew information into the database. It captures essential data such as crew member identities, their specific roles, and their assigned flights, allowing for seamless scheduling and resource allocation. This automation minimizes errors and speeds up the operational workflow, making airline management more efficient and reliable.

By integrating these data management processes, the Flight Management System project aims to transform how airlines coordinate their workforce, ultimately contributing to safer, smoother, and more efficient flights. This foundational script exemplifies the project's commitment to precision and operational excellence through smart data handling.

## **OBJECTIVE**

The primary objective of the Flight Management System project is to develop a robust and efficient platform that simplifies and automates the management of airline operations. Specifically, the project aims to:

- Ensure accurate and systematic storage of flight crew information, including personal details, roles, and flight assignments.
- Automate the insertion of crew data into the database to minimize manual errors and improve data consistency.
- Facilitate effective crew scheduling and allocation to enhance operational efficiency and flight safety.
- Provide a scalable and maintainable database structure that supports the dynamic needs of airline management.
- Support decision-making processes by maintaining up-to-date and reliable crew information.

By achieving these objectives, the system seeks to improve overall airline management, reduce operational delays, and ensure smooth coordination of flight crews.

## **PROBLEM STATEMENT**

In airline operations, effective management of flight crew information is critical for ensuring timely and safe flights. However, traditional crew management systems often face several challenges, including manual data entry, scheduling conflicts, inaccurate crew records, and a lack of real-time updates. These inefficiencies can lead to crew shortages, flight delays, and increased administrative workload.

To address these issues, we propose a Flight Crew Management System that automates and streamlines the process of recording, scheduling, and managing crew assignments. This system ensures:

- Accurate and automated insertion of crew data to minimize manual errors.
- Real-time tracking of crew assignments to avoid scheduling conflicts.
- Centralized storage of crew information for consistency and easy access.
- Efficient management of roles and flight allocations to optimize workforce utilization.

With this system, airline operations can be more reliable, flights can be staffed appropriately without delays, and administrative efforts can be significantly reduced, ensuring that the right crew is available at the right time for safe and smooth flight operations.

## **SCOPE OF THE PROJECT**

The Flight Crew Management System is designed to streamline and enhance the management of airline crew data and scheduling processes. The scope of this project includes:

- **Crew Data Management:** Efficiently storing and maintaining detailed information about crew members, including personal details, roles, certifications, and contact information.
- **Automated Data Insertion:** Automating the process of adding new crew records to the database to reduce manual effort and errors.
- **Crew Assignment and Scheduling:** Managing the assignment of crew members to specific flights, ensuring optimal workforce utilization, and avoiding scheduling conflicts.
- **Real-time Updates:** Providing up-to-date information about crew availability and assignments to support quick decision-making.
- **Role and Flight Linkage:** Clearly defining and maintaining relationships between crew roles (such as pilots, co-pilots, cabin crew) and their respective flights.
- **Data Integrity and Consistency:** Ensuring the database remains accurate and consistent through validation and structured data input.
- **Scalability:** Designing the system to accommodate growth in the number of flights, crew members, and operational complexity.

The project focuses primarily on backend data management and scheduling automation; it does not currently cover passenger management, ticketing, or in-flight services. This system lays a foundation for improved operational efficiency and safety in airline crew management.

## **METHODOLOGY**

The Flight Crew Management System follows a structured Software Development Life Cycle (SDLC) approach to ensure smooth and efficient implementation. The methodology involves the following steps:

### **Requirement Analysis**

- Identify key entities: Flight Crew, Roles, Flights, Assignments
- Define functionalities: Crew data insertion, scheduling, role assignment, real-time updates
- Determine technical feasibility (SQL database with backend automation)

### **System Design**

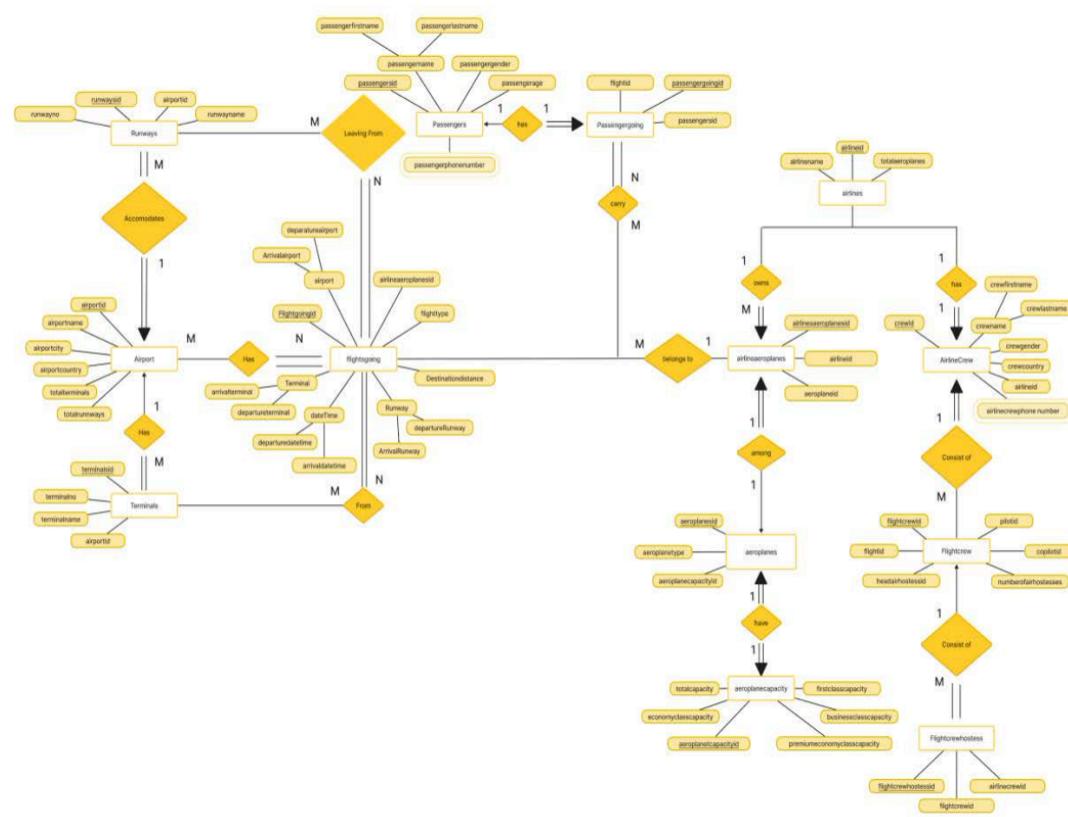
- ER Diagram & Database Schema – Define relationships between crew members, their roles, and flight assignments
- Design database tables with primary and foreign keys to ensure data integrity

### **Implementation**

- Develop SQL scripts (e.g., `flightcrew.sql`) to automate crew data insertion
- Implement backend logic to manage crew assignments and scheduling (can be integrated with a backend language like Python or Java)
- Create and manage the relational database to store crew details and flight associations

## ER DIAGRAM

ER Model stands for Entity Relationship Model is a high-level conceptual data model diagram. ER model helps to systematically analyse data requirements to produce a well-designed database. The ER Model represents real-world entities and the relationships between them. Creating an ER Model in DBMS is considered the best practice before implementing your database. The ER Entity-Relationship diagram is a visual representation of all the entities and their attributes, with their relationship



The ER diagram made for our project consists of several entities with their attributes describing how the various processes and entities are related in our database.

Reference:

- Rectangle – Entity
- (Double Rectangle – Weak Entity, Single rectangle – Strong Entity)
- Oval – Attribute
- (Underlined Attribute – Primary Key)
- Diamonds – Relationship
- (Double Diamond – Weak Relation)
- Double line – Total participation
- Single line – Partial Participation
- Line segment – Many relationships
- Ray – One Relationship

# ER DIAGRAM EXPLANATION

## Entities

<b>Flights_Going</b>	<b>Flights_Crew</b>	<b>Airline_Crew</b>
Airlines_Aeroplanes_ID	Flight_Crew_ID	Crew_ID
Departure_Airport_ID	Flight_ID	Crew_First_Name
Arrival_Airport_ID	Pilot_ID	Crew_Last_Name
Flight_Type	Copilot_ID	Crew_Gender
Arrival_Date_Time	Number_of_Air_Hostesses	Crew_Country
Departure_Date_Time	Head_Air_Hostess_ID	Airline_ID
Destination_Distance		
Arrival_Terminal_ID		
Departure_Terminal_ID		
Arrival_Runway_ID		
Departure_Runway_ID		
<b>Aeroplanes</b>	<b>Airlines_Aeroplanes</b>	
	Aeroplanes_ID	Airlines_Aeroplanes_ID
	Aeroplane_Type	Airline_ID
	Aeroplane_Capacity_ID	Total_Aeroplanes
<b>Flight_Crew_Hostess</b>	<b>Airlines</b>	<b>Airline_Crew_Phone_No</b>
Flight_Crew_Hostess	Airline_ID	Airline_Crew_PhoneNo_ID
Flight_Crew_ID	Airline_Name	Airline_Crew_ID
Airline_Crew_ID	Total_Aeroplanes	Phone_Number
<b>Aeroplane_Capacity</b>		
	Aeroplane_Capacity_ID	
	Total_Capacity	
	Economy_Class_Capacity	
	Premier_Economy_Class_Capacity	
	Business_Class_Capacity	
	First_Class_Capacity	

## Types of Relationships in ER Diagram

Entity 1	Relationship	Entity 2	Cardinality
Airport	Accommodates	Runways	1:m
Airport	Has	Terminals	1:m
Airport	Has	Flights Going	m:n
Terminals	Has	Flights Going	m:n
Flights Going	Leaving From	Runways	n:m
Flights Going	Carry	Passengers Going	m:n
Passengers	Has	Passengers Going	1:1
Flights Going	Belong To	Airlines Aeroplanes	m:1
Airlines	Owns	Airlines Aeroplanes	1:m
Airlines	Has	Airline Crew	1:1
Airlines Aeroplanes	Among	Aeroplanes	1:1
Aeroplanes	Have	Aeroplane Capacity	1:1
Airline Crew	Consist Of	Flight Crew	1:m
Flight Crew	Consist Of	Flight Crew Hostess	1:m

## **EXPLAINING THE RELATIONSHIPS**

1. **Airport Accommodates Runways (1:m)**  
An airport can accommodate multiple runways, but each runway belongs to only one airport.
2. **Airport Has Terminals (1:m)**  
Each airport contains several terminals, but a terminal is located in only one airport.
3. **Airport Has Flights Going (m:n)**  
Multiple flights can operate from multiple airports. One airport can host many flights, and a single flight may be associated with multiple airports (departure and arrival).
4. **Terminals Has Flights Going (m:n)**  
Flights can operate from or to multiple terminals, and each terminal can handle many flights.
5. **Flights Going Leaving From Runways (n:m)**  
Flights can leave from different runways, and a single runway can serve multiple flights.
6. **Flights Going Carry Passengers Going (m:n)**  
Multiple flights carry multiple passengers, and passengers can be on multiple flights (e.g., connecting flights).
7. **Passengers Has Passengers Going (1:1)**  
Each passenger corresponds to a unique "passenger going" instance, likely representing a specific trip or booking.
8. **Flights Going Belong To Airlines Aeroplanes (m:1)**  
Multiple flights can be assigned to one airline's airplane, but a flight is

operated by only one airplane.

**9. Airlines Owns Airlines Aeroplanes (1:m)**

An airline can own many airplanes, but each airplane belongs to only one airline.

**10. Airlines Has Airline Crew (1:1)**

Each airline has exactly one airline crew entity, and each airline crew is associated with only one airline.

**11. Airlines Aeroplanes Among Aeroplanes (1:1)**

Each airline airplane is uniquely associated with an airplane in the general "aeroplanes" category.

**12. Aeroplanes Have Aeroplane Capacity (1:1)**

Each airplane has a specific seating or load capacity, and each capacity value refers to one airplane.

**13. Airline Crew Consist Of Flight Crew (1:m)**

One airline crew is composed of multiple flight crew members.

**14. Flight Crew Consist Of Flight Crew Hostess (1:m)**

Each flight crew includes multiple hostesses, and each hostess is part of only one flight crew.

## **NORMALIZATION**

Normalization is the process of organizing data in a relational database to reduce redundancy (duplicate data) and improve data integrity. The goal is to structure the database in a way that Eliminates duplicate data (e.g., storing the same data in more than one table).

Ensures logical data dependencies, so that data is stored only once and related properly.

### ❖ 1NF (First Normal Form)

Ensure:

- Atomic values (no multi-valued or composite attributes)
- Unique identifiers (primary keys)

Our current schema is already in 1NF.

### ❖ 2NF (Second Normal Form)

Ensure:

- It is in 1NF.
- All non-key attributes are fully functionally dependent on the whole primary key (No partial dependency)  
Our current schema mostly uses single-column primary keys, so 2NF is satisfied.

### ❖ 3NF (Third Normal Form)

Ensure:

- It is in 2NF

- No transitive dependency (non-key attribute depending on another non-key attribute)  
Our current schema is already in 3NF.

❖ BCNF(Boyes Codd Normal Form)

- A relation is in BCNF if:
  1. It is in 3NF, and
  2. For every functional dependency ( $X \rightarrow Y$ ), X must be a super key.This means no non-trivial functional dependency should exist where the determinant is not a super key.  
Our current schema is already in BCNF.

# Creating tables in SQL

Creating tables:

## **1. Aeroplane capacity Table**

```
CREATE TABLE aeroplanecapacity (
    aeroplanecapacityid int NOT NULL,
    totalcapacity int NOT NULL,
    economyclasscapacity int NOT NULL,
    premiumeconomyclasscapacity int NOT NULL,
    businessclasscapacity int NOT NULL,
    firstclasscapacity int NOT NULL
);
```

## **2. aeroplanes table**

```
CREATE TABLE aeroplanes (
    aeroplanesid int NOT NULL,
    aeroplanetype varchar(10) check(aeroplanetype in ('Goods','Passenger')),
    aeroplanecapacityid int NOT NULL
);
```

## **3. Airlinecrew table**

```
CREATE TABLE airlinecrew (
    crewid int NOT NULL,
    crewfirstname varchar(50) NOT NULL,
    crewlastname varchar(50) NOT NULL,
    crewgender char(1) NOT NULL,
    crewcountry varchar(20) NOT NULL,
    airlineid int NOT NULL
);
```

**4. Airlinecrewphonenumbers table**

```
CREATE TABLE airlinecrewphonenumbers (
    airlinecrewphonenumbersid int NOT NULL,
    airlinecrewid int NOT NULL,
    phonenumbers varchar(20) NOT NULL
);
```

**5. Airlines table**

```
CREATE TABLE airlines (
    airlineid int NOT NULL,
    airlinename varchar(50) NOT NULL
);
```

**6. Airlinesaeroplanes table**

```
CREATE TABLE airlinesaeroplanes (
    airlinesaeroplanesid int NOT NULL,
    airlineid int NOT NULL,
    aeroplaneid int NOT NULL
);
```

## **7. Airports table**

```
CREATE TABLE airports (
    airportid int NOT NULL,
    airportname varchar(60) NOT NULL,
    airportcity varchar(50) NOT NULL,
    airportcountry varchar(50) NOT NULL,
    totalterminals int NOT NULL,
    totalrunways int NOT NULL
);
```

## **8. Flightcrew table**

```
CREATE TABLE flightcrew (
    flightcrewid int NOT NULL,
    flightid int NOT NULL,
    pilotid int NOT NULL,
    copilotid int NOT NULL,
    numberoffairhostesses int NOT NULL,
    headairhostessid int NOT NULL
);
```

## **9. Flightcrewhostess table**

```
CREATE TABLE flightcrewhostess (
    flightcrewhostessid int NOT NULL,
    flightcrewid int NOT NULL,
    airlinecrewid int NOT NULL
);
```

## **10.Flightsgoing table**

```
CREATE TABLE flightsgoing (
    flightsgoingid int NOT NULL,
    airlinesaeroplanesid int NOT NULL,
    departureairportid int NOT NULL,
    arrivalairportid int NOT NULL,
    flighttype varchar(20) check(flighttype in ('Domestic','International')) NOT NULL,
    arrivaldatetime timestamp NOT NULL,
    departuredatetime timestamp NOT NULL,
    destinationdistance int NOT NULL,
    arrivalterminalid int NOT NULL,
    departureterminalid int NOT NULL,
    arrivalrunwayid int NOT NULL,
    departurerunwayid int NOT NULL
);
```

## **11.Passengers table**

```
CREATE TABLE passengers (
    passengersid int NOT NULL,
    passengerfirstname varchar(50) NOT NULL,
    passengerlastname varchar(50) NOT NULL,
    passengergender char(1) NOT NULL,
    passengerage int NOT NULL
);
```

## **12.Passengergoing table**

```
CREATE TABLE passengersgoing (
    passengersgoingid int NOT NULL,
    passengerid int NOT NULL,
    flightid int NOT NULL
);
```

### **13. Passengersphonenumbers table**

```
CREATE TABLE passengersphonenumbers (
    passengersphonenumbersid int NOT NULL,
    passengerid int NOT NULL,
    phonenumbers varchar(20) NOT NULL
);
```

### **14. Runways table**

```
CREATE TABLE runways (
    runwaysid int NOT NULL,
    runwayno int NOT NULL,
    runwayname varchar(20) NOT NULL,
    airportid int NOT NULL
);
```

### **15. Terminals table**

```
CREATE TABLE terminals (
    terminalsid int NOT NULL,
    terminalno int NOT NULL,
    terminalname varchar(20) NOT NULL,
    airportid int NOT NULL
);
```

# INSERTING VALUES

## **1. Aeroplane capacity:**

## 2. aeroplanes table

```
1 INSERT INTO aeroplanes (aeroplanesid, aeroplaneType, aeroplaneCapacityId) VALUES (1, 'Passenger', 1);
2 INSERT INTO aeroplanes (aeroplanesid, aeroplaneType, aeroplaneCapacityId) VALUES (2, 'Passenger', 2);
3 INSERT INTO aeroplanes (aeroplanesid, aeroplaneType, aeroplaneCapacityId) VALUES (3, 'Goods', 3);
4 INSERT INTO aeroplanes (aeroplanesid, aeroplaneType, aeroplaneCapacityId) VALUES (4, 'Goods', 4);
5 INSERT INTO aeroplanes (aeroplanesid, aeroplaneType, aeroplaneCapacityId) VALUES (5, 'Passenger', 5);
6 INSERT INTO aeroplanes (aeroplanesid, aeroplaneType, aeroplaneCapacityId) VALUES (6, 'Goods', 6);
7 INSERT INTO aeroplanes (aeroplanesid, aeroplaneType, aeroplaneCapacityId) VALUES (7, 'Passenger', 7);
8 INSERT INTO aeroplanes (aeroplanesid, aeroplaneType, aeroplaneCapacityId) VALUES (8, 'Passenger', 8);
9 INSERT INTO aeroplanes (aeroplanesid, aeroplaneType, aeroplaneCapacityId) VALUES (9, 'Goods', 9);
10 INSERT INTO aeroplanes (aeroplanesid, aeroplaneType, aeroplaneCapacityId) VALUES (10, 'Passenger', 10);
11 INSERT INTO aeroplanes (aeroplanesid, aeroplaneType, aeroplaneCapacityId) VALUES (11, 'Goods', 11);
12 INSERT INTO aeroplanes (aeroplanesid, aeroplaneType, aeroplaneCapacityId) VALUES (12, 'Passenger', 12);
13 INSERT INTO aeroplanes (aeroplanesid, aeroplaneType, aeroplaneCapacityId) VALUES (13, 'Passenger', 13);
14 INSERT INTO aeroplanes (aeroplanesid, aeroplaneType, aeroplaneCapacityId) VALUES (14, 'Goods', 14);
15 INSERT INTO aeroplanes (aeroplanesid, aeroplaneType, aeroplaneCapacityId) VALUES (15, 'Goods', 15);
16 INSERT INTO aeroplanes (aeroplanesid, aeroplaneType, aeroplaneCapacityId) VALUES (16, 'Passenger', 16);
17 INSERT INTO aeroplanes (aeroplanesid, aeroplaneType, aeroplaneCapacityId) VALUES (17, 'Goods', 17);
18 INSERT INTO aeroplanes (aeroplanesid, aeroplaneType, aeroplaneCapacityId) VALUES (18, 'Passenger', 18);
19 INSERT INTO aeroplanes (aeroplanesid, aeroplaneType, aeroplaneCapacityId) VALUES (19, 'Goods', 19);
20 INSERT INTO aeroplanes (aeroplanesid, aeroplaneType, aeroplaneCapacityId) VALUES (20, 'Passenger', 20);
21 INSERT INTO aeroplanes (aeroplanesid, aeroplaneType, aeroplaneCapacityId) VALUES (21, 'Passenger', 21);
```

### **3. Airlinecrew table**

```
1  INSERT INTO airlinecrew (crewid, crewfirstname, crewlastname, crewgender, crewcountry, airlineid)
2  VALUES (1, 'John', 'Smith', 'M', 'United States', 1);
3
4  INSERT INTO airlinecrew (crewid, crewfirstname, crewlastname, crewgender, crewcountry, airlineid)
5  VALUES (2, 'Emily', 'Johnson', 'F', 'United States', 1);
6
7  INSERT INTO airlinecrew (crewid, crewfirstname, crewlastname, crewgender, crewcountry, airlineid)
8  VALUES (3, 'Michael', 'Williams', 'M', 'United Kingdom', 1);
9
10 INSERT INTO airlinecrew (crewid, crewfirstname, crewlastname, crewgender, crewcountry, airlineid)
11 VALUES (4, 'Sophia', 'Brown', 'F', 'United Kingdom', 1);
12
13 INSERT INTO airlinecrew (crewid, crewfirstname, crewlastname, crewgender, crewcountry, airlineid)
14 VALUES (5, 'David', 'Jones', 'M', 'Australia', 1);
15
16 INSERT INTO airlinecrew (crewid, crewfirstname, crewlastname, crewgender, crewcountry, airlineid)
17 VALUES (6, 'Olivia', 'Miller', 'F', 'Australia', 1);
18
19 INSERT INTO airlinecrew (crewid, crewfirstname, crewlastname, crewgender, crewcountry, airlineid)
20 VALUES (7, 'Daniel', 'Wilson', 'M', 'Canada', 1);
```

#### **4. Airlinecrewphonenumbers table**

```
1 -- Records for airlinecrewid 1(US code)
2 INSERT INTO airlinecrewphonenumbers (airlinecrewphonenumbersid, airlinecrewid, phonenumbers) VALUE
3 INSERT INTO airlinecrewphonenumbers (airlinecrewphonenumbersid, airlinecrewid, phonenumbers) VALUE
4
5 -- Records for airlinecrewid (UK code)
6 INSERT INTO airlinecrewphonenumbers (airlinecrewphonenumbersid, airlinecrewid, phonenumbers) VALUE
7 INSERT INTO airlinecrewphonenumbers (airlinecrewphonenumbersid, airlinecrewid, phonenumbers) VALUE
8
9 -- Records for airlinecrewid (Australia code)
10 INSERT INTO airlinecrewphonenumbers (airlinecrewphonenumbersid, airlinecrewid, phonenumbers) VALUE
11 INSERT INTO airlinecrewphonenumbers (airlinecrewphonenumbersid, airlinecrewid, phonenumbers) VALUE
12
13 -- Records for airlinecrewid (Canada code)
14 INSERT INTO airlinecrewphonenumbers (airlinecrewphonenumbersid, airlinecrewid, phonenumbers) VALUE
15 INSERT INTO airlinecrewphonenumbers (airlinecrewphonenumbersid, airlinecrewid, phonenumbers) VALUE
16
17 -- Records for airlinecrewid (France Code)
18 INSERT INTO airlinecrewphonenumbers (airlinecrewphonenumbersid, airlinecrewid, phonenumbers) VALUE
19 INSERT INTO airlinecrewphonenumbers (airlinecrewphonenumbersid, airlinecrewid, phonenumbers) VALUE
```

#### **5. Airlines table**

```
1 INSERT INTO airlines (airlineid, airlinename) VALUES (1, 'American Airlines');
2 INSERT INTO airlines (airlineid, airlinename) VALUES (2, 'Delta Air Lines');
3 INSERT INTO airlines (airlineid, airlinename) VALUES (3, 'United Airlines');
4 INSERT INTO airlines (airlineid, airlinename) VALUES (4, 'Southwest Airlines');
5 INSERT INTO airlines (airlineid, airlinename) VALUES (5, 'Emirates');
6 INSERT INTO airlines (airlineid, airlinename) VALUES (6, 'Lufthansa');
7 INSERT INTO airlines (airlineid, airlinename) VALUES (7, 'British Airways');
8 INSERT INTO airlines (airlineid, airlinename) VALUES (8, 'Air France');
9 INSERT INTO airlines (airlineid, airlinename) VALUES (9, 'Qantas Airways');
10 INSERT INTO airlines (airlineid, airlinename) VALUES (10, 'Singapore Airlines');
11 INSERT INTO airlines (airlineid, airlinename) VALUES (11, 'Cathay Pacific');
12 INSERT INTO airlines (airlineid, airlinename) VALUES (12, 'Qatar Airways');
13 INSERT INTO airlines (airlineid, airlinename) VALUES (13, 'Turkish Airlines');
14 INSERT INTO airlines (airlineid, airlinename) VALUES (14, 'Etihad Airways');
15 INSERT INTO airlines (airlineid, airlinename) VALUES (15, 'ANA (All Nippon Airways)');
16 INSERT INTO airlines (airlineid, airlinename) VALUES (16, 'Virgin Atlantic');
17 INSERT INTO airlines (airlineid, airlinename) VALUES (17, 'KLM Royal Dutch Airlines');
18 INSERT INTO airlines (airlineid, airlinename) VALUES (18, 'Japan Airlines');
19 INSERT INTO airlines (airlineid, airlinename) VALUES (19, 'Air Canada');
20 INSERT INTO airlines (airlineid, airlinename) VALUES (20, 'Air New Zealand');
```

## 6. Airlinesaeroplanes table

---

```
1  -- Insert data into airlinesaeroplanes
2  INSERT INTO airlinesaeroplanes (airlinesaeroplanesid, airlineid, aeroplaneid) VALUES (1, 1, 1);
3  INSERT INTO airlinesaeroplanes (airlinesaeroplanesid, airlineid, aeroplaneid) VALUES (2, 2, 2);
4  INSERT INTO airlinesaeroplanes (airlinesaeroplanesid, airlineid, aeroplaneid) VALUES (3, 3, 5);
5  INSERT INTO airlinesaeroplanes (airlinesaeroplanesid, airlineid, aeroplaneid) VALUES (4, 4, 7);
6  INSERT INTO airlinesaeroplanes (airlinesaeroplanesid, airlineid, aeroplaneid) VALUES (5, 5, 8);
7  INSERT INTO airlinesaeroplanes (airlinesaeroplanesid, airlineid, aeroplaneid) VALUES (6, 6, 10);
8  INSERT INTO airlinesaeroplanes (airlinesaeroplanesid, airlineid, aeroplaneid) VALUES (7, 7, 12);
9  INSERT INTO airlinesaeroplanes (airlinesaeroplanesid, airlineid, aeroplaneid) VALUES (8, 8, 13);
10 INSERT INTO airlinesaeroplanes (airlinesaeroplanesid, airlineid, aeroplaneid) VALUES (9, 9, 16);
11 INSERT INTO airlinesaeroplanes (airlinesaeroplanesid, airlineid, aeroplaneid) VALUES (10, 10, 18);
12 INSERT INTO airlinesaeroplanes (airlinesaeroplanesid, airlineid, aeroplaneid) VALUES (11, 11, 20);
13 INSERT INTO airlinesaeroplanes (airlinesaeroplanesid, airlineid, aeroplaneid) VALUES (12, 12, 22);
14 INSERT INTO airlinesaeroplanes (airlinesaeroplanesid, airlineid, aeroplaneid) VALUES (13, 13, 24);
15 INSERT INTO airlinesaeroplanes (airlinesaeroplanesid, airlineid, aeroplaneid) VALUES (14, 14, 25);
16 INSERT INTO airlinesaeroplanes (airlinesaeroplanesid, airlineid, aeroplaneid) VALUES (15, 15, 27);
17 INSERT INTO airlinesaeroplanes (airlinesaeroplanesid, airlineid, aeroplaneid) VALUES (16, 16, 29);
18 INSERT INTO airlinesaeroplanes (airlinesaeroplanesid, airlineid, aeroplaneid) VALUES (17, 17, 31);
19 INSERT INTO airlinesaeroplanes (airlinesaeroplanesid, airlineid, aeroplaneid) VALUES (18, 18, 33);
20 INSERT INTO airlinesaeroplanes (airlinesaeroplanesid, airlineid, aeroplaneid) VALUES (19, 19, 35);
```

## 7. Airports table

```
1  INSERT INTO airports (airportid, airportname, airportcity, airportcountry, totalterminals, totalrunways)
2  VALUES (1, 'Hartsfield-Jackson Atlanta International Airport', 'Atlanta', 'United States', 7, 5);
3
4  INSERT INTO airports (airportid, airportname, airportcity, airportcountry, totalterminals, totalrunways)
5  VALUES (2, 'Beijing Capital International Airport', 'Beijing', 'China', 5, 3);
6
7  INSERT INTO airports (airportid, airportname, airportcity, airportcountry, totalterminals, totalrunways)
8  VALUES (3, 'Dubai International Airport', 'Dubai', 'United Arab Emirates', 4, 2);
9
10 INSERT INTO airports (airportid, airportname, airportcity, airportcountry, totalterminals, totalrunways)
11 VALUES (4, 'Los Angeles International Airport', 'Los Angeles', 'United States', 8, 4);
12
13 INSERT INTO airports (airportid, airportname, airportcity, airportcountry, totalterminals, totalrunways)
14 VALUES (5, 'Tokyo Haneda Airport', 'Tokyo', 'Japan', 3, 2);
15
16 INSERT INTO airports (airportid, airportname, airportcity, airportcountry, totalterminals, totalrunways)
17 VALUES (6, 'London Heathrow Airport', 'London', 'United Kingdom', 6, 3);
18
19 INSERT INTO airports (airportid, airportname, airportcity, airportcountry, totalterminals, totalrunways)
20 VALUES (7, 'Charles de Gaulle Airport', 'Paris', 'France', 5, 4);
21
```

## **8. Flightcrew table**

```
1  -- Insert record 1
2  INSERT INTO flightcrew (flightcrewid, flightid, pilotid, copilotid, numberoffairhostesses, headairhostessid)
3  VALUES (1, 101, 1, 2, 8, 3);
4
5  -- Insert record 2
6  INSERT INTO flightcrew (flightcrewid, flightid, pilotid, copilotid, numberoffairhostesses, headairhostessid)
7  VALUES (2, 102, 11, 12, 3, 13);
8
9  -- Insert record 3
10 INSERT INTO flightcrew (flightcrewid, flightid, pilotid, copilotid, numberoffairhostesses, headairhostessid)
11 VALUES (3, 103, 16, 17, 3, 18);
12
13 -- Insert record 4
14 INSERT INTO flightcrew (flightcrewid, flightid, pilotid, copilotid, numberoffairhostesses, headairhostessid)
15 VALUES (4, 104, 21, 22, 3, 23);
16
17 -- Insert record 5
18 INSERT INTO flightcrew (flightcrewid, flightid, pilotid, copilotid, numberoffairhostesses, headairhostessid)
19 VALUES (5, 105, 26, 27, 3, 28);
```

## **9. Flightcrewhostess table**

```
1  INSERT INTO flightcrewhostess (flightcrewhostessid, flightcrewid, airelinecrewid)
2  VALUES (1, 1, 1);
3  INSERT INTO flightcrewhostess (flightcrewhostessid, flightcrewid, airelinecrewid)
4  VALUES (2, 1, 2);
5
6  INSERT INTO flightcrewhostess (flightcrewhostessid, flightcrewid, airelinecrewid)
7  VALUES (3, 1, 3);
8
9  INSERT INTO flightcrewhostess (flightcrewhostessid, flightcrewid, airelinecrewid)
10 VALUES (4, 1, 4);
11
12 INSERT INTO flightcrewhostess (flightcrewhostessid, flightcrewid, airelinecrewid)
13 VALUES (5, 1, 5);
14
15 INSERT INTO flightcrewhostess (flightcrewhostessid, flightcrewid, airelinecrewid)
16 VALUES (6, 1, 6);
17
18 INSERT INTO flightcrewhostess (flightcrewhostessid, flightcrewid, airelinecrewid)
19 VALUES (7, 1, 7);
```

## **10. Flightsgoing table**

```
1  INSERT INTO flightsgoing (flightsgoingid, airlinesaeroplanesid, departureairportid, arrivalairportid, arrivaldatetime,
2  VALUES (1, 1, 12, 8, TIMESTAMP '2024-05-05 12:00:00', TIMESTAMP '2024-05-05 10:00:00', 1500, 32, 47, 32, 47);
3
4  INSERT INTO flightsgoing (flightsgoingid, airlinesaeroplanesid, departureairportid, arrivalairportid, arrivaldatetime,
5  VALUES (2, 40, 27, 24, TIMESTAMP '2024-05-06 14:30:00', TIMESTAMP '2024-05-06 12:30:00', 2200, 97, 107, 97, 107);
6
7  INSERT INTO flightsgoing (flightsgoingid, airlinesaeroplanesid, departureairportid, arrivalairportid, arrivaldatetime,
8  VALUES (3, 13, 4, 2, TIMESTAMP '2024-05-07 16:45:00', TIMESTAMP '2024-05-07 14:45:00', 1900, 6, 16, 6, 16);
9
10 INSERT INTO flightsgoing (flightsgoingid, airlinesaeroplanesid, departureairportid, arrivalairportid, arrivaldatetime,
11 VALUES (4, 20, 7, 15, TIMESTAMP '2024-05-08 10:15:00', TIMESTAMP '2024-05-08 08:15:00', 1250, 60, 12, 60, 12);
12
13 INSERT INTO flightsgoing (flightsgoingid, airlinesaeroplanesid, departureairportid, arrivalairportid, arrivaldatetime,
14 VALUES (5, 25, 3, 14, TIMESTAMP '2024-05-09 09:30:00', TIMESTAMP '2024-05-09 07:30:00', 3000, 14, 13, 14, 13);
15
16 INSERT INTO flightsgoing (flightsgoingid, airlinesaeroplanesid, departureairportid, arrivalairportid, arrivaldatetime,
17 VALUES (6, 33, 16, 5, TIMESTAMP '2024-05-10 12:45:00', TIMESTAMP '2024-05-10 10:45:00', 1750, 21, 63, 21, 63);
18
19 INSERT INTO flightsgoing (flightsgoingid, airlinesaeroplanesid, departureairportid, arrivalairportid, arrivaldatetime,
20 VALUES (7, 48, 8, 21, TIMESTAMP '2024-05-11 19:00:00', TIMESTAMP '2024-05-11 17:00:00', 1450, 32, 85, 32, 85);
21
```

## **11. Passengers table**

```
1  INSERT INTO passengers (passengersid, passengerfirstname, passengerlastname, passengergender, passengerage) VALUES (101, 'Rahul', 'Sharma', 'Male', 22);
2  INSERT INTO passengers (passengersid, passengerfirstname, passengerlastname, passengergender, passengerage) VALUES (102, 'Priya', 'Patel', 'Female', 25);
3  INSERT INTO passengers (passengersid, passengerfirstname, passengerlastname, passengergender, passengerage) VALUES (103, 'Amit', 'Singh', 'Male', 28);
4  INSERT INTO passengers (passengersid, passengerfirstname, passengerlastname, passengergender, passengerage) VALUES (104, 'Neha', 'Gupta', 'Female', 23);
5  INSERT INTO passengers (passengersid, passengerfirstname, passengerlastname, passengergender, passengerage) VALUES (105, 'Suresh', 'Kumar', 'Male', 30);
6  INSERT INTO passengers (passengersid, passengerfirstname, passengerlastname, passengergender, passengerage) VALUES (106, 'Anjali', 'Desai', 'Female', 26);
7  INSERT INTO passengers (passengersid, passengerfirstname, passengerlastname, passengergender, passengerage) VALUES (107, 'Vikram', 'Rao', 'Male', 29);
8  INSERT INTO passengers (passengersid, passengerfirstname, passengerlastname, passengergender, passengerage) VALUES (108, 'Kavita', 'Menon', 'Female', 27);
9  INSERT INTO passengers (passengersid, passengerfirstname, passengerlastname, passengergender, passengerage) VALUES (109, 'Manish', 'Sharma', 'Male', 24);
10 INSERT INTO passengers (passengersid, passengerfirstname, passengerlastname, passengergender, passengerage) VALUES (110, 'Pooja', 'Reddy', 'Female', 21);
11 INSERT INTO passengers (passengersid, passengerfirstname, passengerlastname, passengergender, passengerage) VALUES (111, 'Rajesh', 'Krishna', 'Male', 32);
12 INSERT INTO passengers (passengersid, passengerfirstname, passengerlastname, passengergender, passengerage) VALUES (112, 'Divya', 'Sinha', 'Female', 29);
13 INSERT INTO passengers (passengersid, passengerfirstname, passengerlastname, passengergender, passengerage) VALUES (113, 'Sanjay', 'Verma', 'Male', 31);
14 INSERT INTO passengers (passengersid, passengerfirstname, passengerlastname, passengergender, passengerage) VALUES (114, 'Shilpa', 'Chandrasekhar', 'Female', 28);
15 INSERT INTO passengers (passengersid, passengerfirstname, passengerlastname, passengergender, passengerage) VALUES (115, 'Alok', 'Joshua', 'Male', 33);
16 INSERT INTO passengers (passengersid, passengerfirstname, passengerlastname, passengergender, passengerage) VALUES (116, 'Ananya', 'Mullany', 'Female', 22);
17 INSERT INTO passengers (passengersid, passengerfirstname, passengerlastname, passengergender, passengerage) VALUES (117, 'Arjun', 'Nair', 'Male', 26);
18 INSERT INTO passengers (passengersid, passengerfirstname, passengerlastname, passengergender, passengerage) VALUES (118, 'Sneha', 'Rajesh', 'Female', 24);
19 INSERT INTO passengers (passengersid, passengerfirstname, passengerlastname, passengergender, passengerage) VALUES (119, 'Anand', 'Srinivas', 'Male', 35);
20 INSERT INTO passengers (passengersid, passengerfirstname, passengerlastname, passengergender, passengerage) VALUES (120, 'Deepika', 'Gupta', 'Female', 29);
```

## **12. Passengersphonenumbers table**

```
1 -- Passenger 101 with two phone numbers
2 INSERT INTO passengersphonenumbers (passengerid, phonenumbers) VALUES (1, 101, '+91 12345-67890');
3 INSERT INTO passengersphonenumbers (passengerid, phonenumbers) VALUES (2, 101, '+91 98765-43210');
4
5 -- Passenger 102 with one phone number
6 INSERT INTO passengersphonenumbers (passengerid, phonenumbers) VALUES (3, 102, '+91 11111-22222');
7
8 -- Passenger 103 with one phone number
9 INSERT INTO passengersphonenumbers (passengerid, phonenumbers) VALUES (4, 103, '+91 99999-88888');
10
11 -- Passenger 104 with one phone number
12 INSERT INTO passengersphonenumbers (passengerid, phonenumbers) VALUES (5, 104, '+91 77777-66666');
13
14 -- Passenger 105 with one phone number
15 INSERT INTO passengersphonenumbers (passengerid, phonenumbers) VALUES (6, 105, '+91 44444-55555');
16 -- Passenger 106 with two phone numbers (India)
17 INSERT INTO passengersphonenumbers (passengerid, phonenumbers) VALUES (7, 106, '+91 77777-88888');
18 INSERT INTO passengersphonenumbers (passengerid, phonenumbers) VALUES (8, 106, '+91 99999-11111');
19
20
```

## **13. Runways table**

```
1 INSERT INTO runways (runwaysid, runwayno, runwayname, airportid) VALUES (1, 1, 'runway 1', 1);
2 INSERT INTO runways (runwaysid, runwayno, runwayname, airportid) VALUES (2, 2, 'runway 2', 1);
3 INSERT INTO runways (runwaysid, runwayno, runwayname, airportid) VALUES (3, 3, 'runway 3', 1);
4 INSERT INTO runways (runwaysid, runwayno, runwayname, airportid) VALUES (4, 4, 'runway 4', 1);
5 INSERT INTO runways (runwaysid, runwayno, runwayname, airportid) VALUES (5, 5, 'runway 5', 1);
6 INSERT INTO runways (runwaysid, runwayno, runwayname, airportid) VALUES (6, 1, 'runway 1', 2);
7 INSERT INTO runways (runwaysid, runwayno, runwayname, airportid) VALUES (7, 2, 'runway 2', 2);
8 INSERT INTO runways (runwaysid, runwayno, runwayname, airportid) VALUES (8, 3, 'runway 3', 2);
9 INSERT INTO runways (runwaysid, runwayno, runwayname, airportid) VALUES (9, 4, 'runway 4', 2);
10 INSERT INTO runways (runwaysid, runwayno, runwayname, airportid) VALUES (10, 5, 'runway 5', 2);
11 INSERT INTO runways (runwaysid, runwayno, runwayname, airportid) VALUES (11, 1, 'runway 1', 3);
12 INSERT INTO runways (runwaysid, runwayno, runwayname, airportid) VALUES (12, 2, 'runway 2', 3);
13 INSERT INTO runways (runwaysid, runwayno, runwayname, airportid) VALUES (13, 3, 'runway 3', 3);
14 INSERT INTO runways (runwaysid, runwayno, runwayname, airportid) VALUES (14, 4, 'runway 4', 3);
15 INSERT INTO runways (runwaysid, runwayno, runwayname, airportid) VALUES (15, 1, 'runway 1', 4);
16 INSERT INTO runways (runwaysid, runwayno, runwayname, airportid) VALUES (16, 2, 'runway 2', 4);
17 INSERT INTO runways (runwaysid, runwayno, runwayname, airportid) VALUES (17, 3, 'runway 3', 4);
18 INSERT INTO runways (runwaysid, runwayno, runwayname, airportid) VALUES (18, 4, 'runway 4', 4);
19 INSERT INTO runways (runwaysid, runwayno, runwayname, airportid) VALUES (19, 5, 'runway 5', 4);
20 INSERT INTO runways (runwaysid, runwayno, runwayname, airportid) VALUES (20, 1, 'runway 1', 5);
```

## **14.Terminals table**

```
1  INSERT INTO terminals (terminalsid, terminalno, terminalname, airportid) VALUES (1, 1, 'Terminal 1', 1);
2  INSERT INTO terminals (terminalsid, terminalno, terminalname, airportid) VALUES (2, 2, 'Terminal 2', 1);
3  INSERT INTO terminals (terminalsid, terminalno, terminalname, airportid) VALUES (3, 3, 'Terminal 3', 1);
4  INSERT INTO terminals (terminalsid, terminalno, terminalname, airportid) VALUES (4, 4, 'Terminal 4', 1);
5  INSERT INTO terminals (terminalsid, terminalno, terminalname, airportid) VALUES (5, 5, 'Terminal 5', 1);
6  INSERT INTO terminals (terminalsid, terminalno, terminalname, airportid) VALUES (6, 1, 'Terminal 1', 2);
7  INSERT INTO terminals (terminalsid, terminalno, terminalname, airportid) VALUES (7, 2, 'Terminal 2', 2);
8  INSERT INTO terminals (terminalsid, terminalno, terminalname, airportid) VALUES (8, 3, 'Terminal 3', 2);
9  INSERT INTO terminals (terminalsid, terminalno, terminalname, airportid) VALUES (9, 4, 'Terminal 4', 2);
10 INSERT INTO terminals (terminalsid, terminalno, terminalname, airportid) VALUES (10, 5, 'Terminal 5', 2);
11 INSERT INTO terminals (terminalsid, terminalno, terminalname, airportid) VALUES (11, 1, 'Terminal 1', 3);
12 INSERT INTO terminals (terminalsid, terminalno, terminalname, airportid) VALUES (12, 2, 'Terminal 2', 3);
13 INSERT INTO terminals (terminalsid, terminalno, terminalname, airportid) VALUES (13, 3, 'Terminal 3', 3);
14 INSERT INTO terminals (terminalsid, terminalno, terminalname, airportid) VALUES (14, 4, 'Terminal 4', 3);
15 INSERT INTO terminals (terminalsid, terminalno, terminalname, airportid) VALUES (15, 1, 'Terminal 1', 4);
16 INSERT INTO terminals (terminalsid, terminalno, terminalname, airportid) VALUES (16, 2, 'Terminal 2', 4);
17 INSERT INTO terminals (terminalsid, terminalno, terminalname, airportid) VALUES (17, 3, 'Terminal 3', 4);
18 INSERT INTO terminals (terminalsid, terminalno, terminalname, airportid) VALUES (18, 4, 'Terminal 4', 4);
19 INSERT INTO terminals (terminalsid, terminalno, terminalname, airportid) VALUES (19, 5, 'Terminal 5', 4);
20 INSERT INTO terminals (terminalsid, terminalno, terminalname, airportid) VALUES (20, 1, 'Terminal 1', 5);
-----
```

# TRIGGERS

## Trigger to Validate Flight Departure and Arrival Data

```
1  CREATE OR REPLACE TRIGGER flightsgoing_trigger
2  BEFORE INSERT OR UPDATE ON flightsgoing
3  FOR EACH ROW
4  BEGIN
5      -- Terminal check
6      IF :NEW.departureterminalid = :NEW.arrivalterminalid THEN
7          RAISE_APPLICATION_ERROR(-20001, 'Departure and arrival terminals cannot be the same');
8      END IF;
9      -- DateTime check
10     IF :NEW.departuredatetime >= :NEW.arrivaldatetime THEN
11         RAISE_APPLICATION_ERROR(-20002, 'Departure datetime must be before arrival datetime');
12     END IF;
13 END;
14 /
```

## Trigger to Enforce Unique Airline Names on Insert

```
1  CREATE OR REPLACE TRIGGER airline_trigger
2  BEFORE INSERT OR UPDATE ON airlines
3  FOR EACH ROW
4  DECLARE
5      airline_count NUMBER;
6  BEGIN
7      -- Check if the airline name is unique
8      IF INSERTING THEN
9          SELECT COUNT(*)
10         INTO airline_count
11        FROM airlines
12       WHERE airlinename = :NEW.airlinename;
13
14      IF airline_count > 0 THEN
15          RAISE_APPLICATION_ERROR(-20001, 'Airline name must be unique');
16      END IF;
17      END IF;
18  END;
19 /
```

Trigger to Validate Non-Negative Terminals and Runways in Airports

```
1  CREATE OR REPLACE TRIGGER airport_trigger
2  BEFORE INSERT OR UPDATE ON airports
3  FOR EACH ROW
4  BEGIN
5      -- Check if the total number of terminals and runways is positive
6      IF :NEW.totalterminals < 0 OR :NEW.totalrunways < 0 THEN
7          RAISE_APPLICATION_ERROR(-20001, 'Total terminals and runways must be non-negative');
8      END IF;
9  END;
10 /
```

## PROCEDURE

Displaying Airline and Aeroplane Information.

```
1  CREATE OR REPLACE PROCEDURE get_airline_aeroplane_info IS
2  BEGIN
3      FOR airline_info IN (
4          SELECT
5              a.airlineid,
6              a.airlinename,
7              aa.aeroplaneid
8          FROM
9              airlines a
10         JOIN
11             airlinesaeroplanes aa ON a.airlineid = aa.airlineid
12     ) LOOP
13         DBMS_OUTPUT.PUT_LINE('Airline ID: ' || airline_info.airlineid);
14         DBMS_OUTPUT.PUT_LINE('Airline Name: ' || airline_info.airlinename);
15         DBMS_OUTPUT.PUT_LINE('Aeroplane ID: ' || airline_info.aeroplaneid);
16         DBMS_OUTPUT.PUT_LINE('-----');
17     END LOOP;
18 END;
19 /
20
21 begin
22     get_airline_aeroplane_info;
23 end;
```

Output

Airline ID: 1  
Airline Name: American Airlines  
Aeroplane ID: 1

---

Airline ID: 1  
Airline Name: American Airlines  
Aeroplane ID: 75

---

Airline ID: 2  
Airline Name: Delta Air Lines  
Aeroplane ID: 2

---

Airline ID: 2  
Airline Name: Delta Air Lines  
Aeroplane ID: 76

---

Airline ID: 3  
Airline Name: United Airlines  
Aeroplane ID: 5

---

Airline ID: 3  
Airline Name: United Airlines  
Aeroplane ID: 77

---

Airline ID: 4  
Airline Name: Southwest Airlines  
Aeroplane ID: 7

---

### Displaying Aeroplane Capacity Information by Class.

```
1  CREATE OR REPLACE PROCEDURE get_airplane_capacity AS
2  BEGIN
3      FOR rec IN (
4          SELECT a.aeroplanesid,
5              a.aeroplanetype,
6              ac.totalcapacity,
7              ac.economyclasscapacity,
8              ac.premiumeconomyclasscapacity,
9              ac.businessclasscapacity,
10             ac.firstclasscapacity
11     FROM aeroplanes a
12    JOIN aeroplanecapacity ac ON a.aeroplanecapacityid = ac.aeroplanecapacityid
13    ORDER BY ac.totalcapacity, a.aeroplanesid
14 ) LOOP
15     DBMS_OUTPUT.PUT_LINE(
16         'Aeroplane ID: ' || rec.aeroplanesid ||
17         ', Type: ' || rec.aeroplanetype ||
18         ', Total Capacity: ' || rec.totalcapacity ||
19         ', Economy Class Capacity: ' || rec.economyclasscapacity ||
20         ', Premium Economy Class Capacity: ' || rec.premiumeconomyclasscapacity ||
21         ', Business Class Capacity: ' || rec.businessclasscapacity ||
22         ', First Class Capacity: ' || rec.firstclasscapacity
23     );
24     END LOOP;
25 END;
26 /
27
28 begin
29     get_airplane_capacity;
30 end;
```

## Output

## Displaying Flight Crew Information.

```
1  CREATE OR REPLACE PROCEDURE get_crew_info IS
2  BEGIN
3      FOR crew_info IN (
4          SELECT
5              fc.flightcrewid,
6              fc.flightid,
7              ac.crewfirstname AS airline_crew_firstname,
8              ac.crewlastname AS airline_crew_lastname,
9              ac.crewgender AS airline_crew_gender,
10             ac.crewcountry AS airline_crew_country,
11             fh.airlinecrewid AS flight_hostess_crewid,
12             fh.flightcrewhostessid AS flight_crew_hostess_id
13         FROM
14             flightcrew fc
15         JOIN
16             airlinecrew ac ON fc.pilotid = ac.crewid OR fc.copilotid = ac.crewid
17         LEFT JOIN
18             flightcrewhostess fh ON fc.flightcrewid = fh.flightcrewid
19     ) LOOP
20         DBMS_OUTPUT.PUT_LINE('Flight Crew ID: ' || crew_info.flightcrewid);
21         DBMS_OUTPUT.PUT_LINE('Flight ID: ' || crew_info.flightid);
22         DBMS_OUTPUT.PUT_LINE('Airline Crew Firstname: ' || crew_info.airline_crew_firstname);
23         DBMS_OUTPUT.PUT_LINE('Airline Crew Lastname: ' || crew_info.airline_crew_lastname);
24         DBMS_OUTPUT.PUT_LINE('Airline Crew Gender: ' || crew_info.airline_crew_gender);
25         DBMS_OUTPUT.PUT_LINE('Airline Crew Country: ' || crew_info.airline_crew_country);
26         DBMS_OUTPUT.PUT_LINE('Flight Hostess Crew ID: ' || crew_info.flight_hostess_crewid);
27         DBMS_OUTPUT.PUT_LINE('Flight Crew Hostess ID: ' || crew_info.flight_crew_hostess_id);
28         DBMS_OUTPUT.PUT_LINE('-----');
29     END LOOP;
30  END;
31  /
32
33  BEGIN
34      get_crew_info;
35  END;
36  /
```

## Output

```
Flight Crew ID: 1
Flight ID: 101
Airline Crew Firstname: Emily
Airline Crew Lastname: Johnson
Airline Crew Gender: F
Airline Crew Country: United States
Flight Hostess Crew ID: 1
Flight Crew Hostess ID: 1
```

---

```
Flight Crew ID: 1
Flight ID: 101
Airline Crew Firstname: Emily
Airline Crew Lastname: Johnson
Airline Crew Gender: F
Airline Crew Country: United States
Flight Hostess Crew ID: 2
Flight Crew Hostess ID: 2
```

---

```
Flight Crew ID: 1
Flight ID: 101
Airline Crew Firstname: Emily
Airline Crew Lastname: Johnson
Airline Crew Gender: F
Airline Crew Country: United States
Flight Hostess Crew ID: 3
Flight Crew Hostess ID: 3
```

---

```
Flight Crew ID: 1
Flight ID: 101
Airline Crew Firstname: Emily
Airline Crew Lastname: Johnson
Airline Crew Gender: F
Airline Crew Country: United States
Flight Hostess Crew ID: 4
Flight Crew Hostess ID: 4
```

---

```
Flight Crew ID: 1
Flight ID: 101
Airline Crew Firstname: Emily
Airline Crew Lastname: Johnson
Airline Crew Gender: F
Airline Crew Country: United States
Flight Hostess Crew ID: 5
Flight Crew Hostess ID: 5
```

---

```
Flight Crew ID: 1
Flight ID: 101
Airline Crew Firstname: Emily
Airline Crew Lastname: Johnson
Airline Crew Gender: F
Airline Crew Country: United States
Flight Hostess Crew ID: 6
Flight Crew Hostess ID: 6
```

---

```
Flight Crew ID: 1
Flight ID: 101
Airline Crew Firstname: Emily
Airline Crew Lastname: Johnson
Airline Crew Gender: F
Airline Crew Country: United States
Flight Hostess Crew ID: 7
Flight Crew Hostess ID: 7
```

---

## Displaying Flight and Crew Details.

```
1      CREATE OR REPLACE PROCEDURE get_flight_crew_info IS
2      BEGIN
3          FOR crew_info IN (
4              SELECT
5                  fg.flightsgoingid,
6                  fg.departureairportid,
7                  fg.arrivalairportid,
8                  fg.arrivaldatetime,
9                  fg.departuredatetime,
10                 fc.pilotid,
11                 fc.copilotid,
12                 fc.numberoffairhostesses,
13                 fc.headairhostessid
14             FROM
15                 flightsgoing fg
16             JOIN
17                 flightcrew fc ON fg.flightsgoingid = fc.flightid
18         ) LOOP
19             DBMS_OUTPUT.PUT_LINE('Flight ID: ' || crew_info.flightsgoingid);
20             DBMS_OUTPUT.PUT_LINE('Departure Airport ID: ' || crew_info.departureairportid);
21             DBMS_OUTPUT.PUT_LINE('Arrival Airport ID: ' || crew_info.arrivalairportid);
22             DBMS_OUTPUT.PUT_LINE('Arrival Datetime: ' || crew_info.arrivaldatetime);
23             DBMS_OUTPUT.PUT_LINE('Departure Datetime: ' || crew_info.departuredatetime);
24             DBMS_OUTPUT.PUT_LINE('Pilot ID: ' || crew_info.pilotid);
25             DBMS_OUTPUT.PUT_LINE('CoPilot ID: ' || crew_info.copilotid);
26             DBMS_OUTPUT.PUT_LINE('Number of Air Hostesses: ' || crew_info.numberoffairhostesses);
27             DBMS_OUTPUT.PUT_LINE('Head Air Hostess ID: ' || crew_info.headairhostessid);
28             DBMS_OUTPUT.PUT_LINE('-----');
29         END LOOP;
30     END;
31     /
32
33     BEGIN
34         get_flight_crew_info;
35     END;
36     /
```

## Output

```
Flight ID: 1
Departure Airport ID: 12
Arrival Airport ID: 8
Arrival Datetime: 05-MAY-24 12.00.00.000000 PM
Departure Datetime: 05-MAY-24 10.00.00.000000 AM
Pilot ID: 1
CoPilot ID: 2
Number of Air Hostesses: 8
Head Air Hostess ID: 3
-----
Flight ID: 2
Departure Airport ID: 27
Arrival Airport ID: 24
Arrival Datetime: 06-MAY-24 02.30.00.000000 PM
Departure Datetime: 06-MAY-24 12.30.00.000000 PM
Pilot ID: 11
CoPilot ID: 12
Number of Air Hostesses: 3
Head Air Hostess ID: 13
-----
Flight ID: 3
Departure Airport ID: 4
Arrival Airport ID: 2
Arrival Datetime: 07-MAY-24 04.45.00.000000 PM
Departure Datetime: 07-MAY-24 02.45.00.000000 PM
Pilot ID: 16
CoPilot ID: 17
Number of Air Hostesses: 3
Head Air Hostess ID: 18
-----
Flight ID: 4
Departure Airport ID: 7
Arrival Airport ID: 15
Arrival Datetime: 08-MAY-24 10.15.00.000000 AM
Departure Datetime: 08-MAY-24 08.15.00.000000 AM
Pilot ID: 21
CoPilot ID: 22
Number of Air Hostesses: 3
Head Air Hostess ID: 23
-----
Flight ID: 5
Departure Airport ID: 3
Arrival Airport ID: 14
Arrival Datetime: 09-MAY-24 09.30.00.000000 AM
Departure Datetime: 09-MAY-24 07.30.00.000000 AM
Pilot ID: 26
CoPilot ID: 27
Number of Air Hostesses: 3
Head Air Hostess ID: 28
-----
Flight ID: 6
Departure Airport ID: 16
Arrival Airport ID: 5
Arrival Datetime: 10-MAY-24 12.45.00.000000 PM
Departure Datetime: 10-MAY-24 10.45.00.000000 AM
Pilot ID: 31
CoPilot ID: 32
Number of Air Hostesses: 3
Head Air Hostess ID: 33
```

Displaying Airline Crew Member Information.

```
1  CREATE OR REPLACE PROCEDURE get_airline_crew_info IS
2  BEGIN
3      FOR crew_info IN (
4          SELECT
5              ac.crewid,
6              ac.crewfirstname,
7              ac.crewlastname,
8              ac.crewgender,
9              ac.crewcountry,
10             pn.phonenumber
11         FROM
12             airlinecrew ac
13        LEFT JOIN
14            airlinecrewphonenumber pn ON ac.crewid = pn.airlinecrewid
15    ) LOOP
16        DBMS_OUTPUT.PUT_LINE('Crew ID: ' || crew_info.crewid);
17        DBMS_OUTPUT.PUT_LINE('Crew Firstname: ' || crew_info.crewfirstname);
18        DBMS_OUTPUT.PUT_LINE('Crew Lastname: ' || crew_info.crewlastname);
19        DBMS_OUTPUT.PUT_LINE('Crew Gender: ' || crew_info.crewgender);
20        DBMS_OUTPUT.PUT_LINE('Crew Country: ' || crew_info.crewcountry);
21        DBMS_OUTPUT.PUT_LINE('Phone Number: ' || crew_info.phonenumber);
22        DBMS_OUTPUT.PUT_LINE('-----');
23    END LOOP;
24    END;
25  /
26
27  BEGIN
28      get_airline_crew_info;
29  END;
30  /
```

Output

```
Crew ID: 1
Crew Firstname: John
Crew Lastname: Smith
Crew Gender: M
Crew Country: United States
Phone Number: +1 12345-12345
```

```
-----  
Crew ID: 1
Crew Firstname: John
Crew Lastname: Smith
Crew Gender: M
Crew Country: United States
Phone Number: +1 98765-43210
```

```
-----  
Crew ID: 2
Crew Firstname: Emily
Crew Lastname: Johnson
Crew Gender: F
Crew Country: United States
Phone Number: +44 11122-23333
```

```
-----  
Crew ID: 2
Crew Firstname: Emily
Crew Lastname: Johnson
Crew Gender: F
Crew Country: United States
Phone Number: +44 44455-56666
```

```
-----  
Crew ID: 3
Crew Firstname: Michael
Crew Lastname: Williams
Crew Gender: M
Crew Country: United Kingdom
Phone Number: +61 99988-87777
```

```
-----  
Crew ID: 3
Crew Firstname: Michael
Crew Lastname: Williams
Crew Gender: M
Crew Country: United Kingdom
Phone Number: +61 33322-21111
```

```
-----  
Crew ID: 4
Crew Firstname: Sophia
Crew Lastname: Brown
Crew Gender: F
Crew Country: United Kingdom
Phone Number: +1 77777-77777
```

```
-----  
Crew ID: 4
Crew Firstname: Sophia
Crew Lastname: Brown
Crew Gender: F
Crew Country: United Kingdom
Phone Number: +1 88888-88888
```

```
-----  
Crew ID: 5
Crew Firstname: David
Crew Lastname: Jones
Crew Gender: M
Crew Country: Australia
Phone Number: +33 11111-11111
```

# CURSORS

## Retrieving and Displaying Airport Details using a Function and Cursor

```
1  CREATE OR REPLACE FUNCTION get_airport_details
2  RETURN SYS_REFCURSOR
3  IS
4      result_cursor SYS_REFCURSOR;
5  BEGIN
6      OPEN result_cursor FOR
7          SELECT airportname, airportcity, totalterminals, totalrunways
8          FROM airports;
9      RETURN result_cursor;
10 END;
11 /
12
13
14 DECLARE
15     result_cursor SYS_REFCURSOR;
16     airport_name airports.airportname%TYPE;
17     airport_city airports.airportcity%TYPE;
18     num_terminals NUMBER;
19     num_runways NUMBER;
20 BEGIN
21     result_cursor := get_airport_details;
22     LOOP
23         FETCH result_cursor INTO airport_name, airport_city, num_terminals, num_runways;
24         EXIT WHEN result_cursor%NOTFOUND;
25         DBMS_OUTPUT.PUT_LINE('Airport Name: ' || airport_name);
26         DBMS_OUTPUT.PUT_LINE('Airport City: ' || airport_city);
27         DBMS_OUTPUT.PUT_LINE('Number of Terminals: ' || num_terminals);
28         DBMS_OUTPUT.PUT_LINE('Number of Runways: ' || num_runways);
29         DBMS_OUTPUT.PUT_LINE('-----');
30     END LOOP;
31     CLOSE result_cursor;
32 END;
33 /
```

## Output

```
Airport Name: Hartsfield-Jackson Atlanta International Airport
Airport City: Atlanta
Number of Terminals: 7
Number of Runways: 5
-----
Airport Name: Beijing Capital International Airport
Airport City: Beijing
Number of Terminals: 5
Number of Runways: 3
-----
Airport Name: Dubai International Airport
Airport City: Dubai
Number of Terminals: 4
Number of Runways: 2
-----
Airport Name: Los Angeles International Airport
Airport City: Los Angeles
Number of Terminals: 8
Number of Runways: 4
-----
Airport Name: Tokyo Haneda Airport
Airport City: Tokyo
Number of Terminals: 3
Number of Runways: 2
-----
Airport Name: London Heathrow Airport
Airport City: London
Number of Terminals: 6
Number of Runways: 3
-----
Airport Name: Charles de Gaulle Airport
Airport City: Paris
Number of Terminals: 5
Number of Runways: 4
-----
Airport Name: Frankfurt Airport
Airport City: Frankfurt
Number of Terminals: 4
Number of Runways: 3
-----
Airport Name: Singapore Changi Airport
Airport City: Singapore
Number of Terminals: 5
Number of Runways: 4
-----
Airport Name: Incheon International Airport
Airport City: Incheon
Number of Terminals: 6
Number of Runways: 3
-----
Airport Name: Amsterdam Airport Schiphol
Airport City: Amsterdam
Number of Terminals: 4
Number of Runways: 2
-----
Airport Name: Hong Kong International Airport
Airport City: Hong Kong
Number of Terminals: 7
Number of Runways: 5
-----
Airport Name: Denver International Airport
Airport City: Denver
Number of Terminals: 6
Number of Runways: 4
```

Displaying Airline IDs and Names using a Cursor.

```
1  create or replace procedure airline_data as
2      cursor airline_cursor is select * from airlines;
3  begin
4      for rec in airline_cursor loop
5          dbms_output.put_line(rec.airlineid || ' ' || rec.airlinename);
6      end loop;
7  end;
8 /
9
10 BEGIN
11     airline_data;
12 end;
```

Output

```
1 American Airlines
2 Delta Air Lines
3 United Airlines
4 Southwest Airlines
5 Emirates
6 Lufthansa
7 British Airways
8 Air France
9 Qantas Airways
10 Singapore Airlines
11 Cathay Pacific
12 Qatar Airways
13 Turkish Airlines
14 Etihad Airways
15 ANA (All Nippon Airways)
16 Virgin Atlantic
17 KLM Royal Dutch Airlines
18 Japan Airlines
19 Air Canada
20 Air New Zealand
21 Virgin Australia
22 Swiss International Air Lines
23 Austrian Airlines
24 Finnair
25 Korean Air
26 Thai Airways
27 Malaysia Airlines
28 EgyptAir
29 Aeroflot Russian Airlines
30 Scandinavian Airlines (SAS)
31 Norwegian Air Shuttle
32 LATAM Airlines
33 Iberia
34 Garuda Indonesia
35 Alaska Airlines
36 JetBlue Airways
37 Hawaiian Airlines
38 EVA Air
39 Azul Brazilian Airlines
40 AirAsia
41 IndiGo
42 Vueling Airlines
43 Wizz Air
44 EasyJet
45 Ryanair
46 Spirit Airlines
47 Allegiant Air
48 Frontier Airlines
49 Air Arabia
50 Flydubai
```

Displaying Airline Crew Details with Airline Name using an Explicit Cursor.

```
1  DECLARE
2      v_crew_id    airlinecrew.crewid%TYPE;
3      v_first_name airlinecrew.crewfirstname%TYPE;
4      v_last_name  airlinecrew.crewlastname%TYPE;
5      v_gender     airlinecrew.crewgender%TYPE;
6      v_country    airlinecrew.crewcountry%TYPE;
7      v_airline_name airlines.airlinename%TYPE;
8
9      CURSOR crew_cursor IS
10         SELECT ac.crewid, ac.crewfirstname, ac.crewlastname, ac.crewgender, ac.crewcountry, al.airl
11             FROM airlinecrew ac
12             JOIN airlines al ON ac.airlineid = al.airlineid;
13
14 BEGIN
15     OPEN crew_cursor;
16     LOOP
17         FETCH crew_cursor INTO v_crew_id, v_first_name, v_last_name, v_gender, v_country, v_airline
18         EXIT WHEN crew_cursor%NOTFOUND;
19         DBMS_OUTPUT.PUT_LINE('Crew ID: ' || v_crew_id || ', Name: ' || v_first_name || ', Gender: '
20     END LOOP;
21     CLOSE crew_cursor;
22 END;
23 /
```

## Output

```
Crew ID: 1, Name: John, Gender: M, Country: United States, Airline: American Airlines
Crew ID: 2, Name: Emily, Gender: F, Country: United States, Airline: American Airlines
Crew ID: 3, Name: Michael, Gender: M, Country: United Kingdom, Airline: American Airlines
Crew ID: 4, Name: Sophia, Gender: F, Country: United Kingdom, Airline: American Airlines
Crew ID: 5, Name: David, Gender: M, Country: Australia, Airline: American Airlines
Crew ID: 6, Name: Olivia, Gender: F, Country: Australia, Airline: American Airlines
Crew ID: 7, Name: Daniel, Gender: M, Country: Canada, Airline: American Airlines
Crew ID: 8, Name: Emma, Gender: F, Country: Canada, Airline: American Airlines
Crew ID: 9, Name: Matthew, Gender: M, Country: France, Airline: American Airlines
Crew ID: 10, Name: Isabella, Gender: F, Country: France, Airline: American Airlines
Crew ID: 11, Name: Mark, Gender: M, Country: United States, Airline: Delta Air Lines
Crew ID: 12, Name: Ava, Gender: F, Country: United States, Airline: Delta Air Lines
Crew ID: 13, Name: Jacob, Gender: M, Country: United Kingdom, Airline: Delta Air Lines
Crew ID: 14, Name: Mia, Gender: F, Country: United Kingdom, Airline: Delta Air Lines
Crew ID: 15, Name: Ethan, Gender: M, Country: Australia, Airline: Delta Air Lines
Crew ID: 16, Name: Amelia, Gender: F, Country: Germany, Airline: United Airlines
Crew ID: 17, Name: Noah, Gender: M, Country: Germany, Airline: United Airlines
Crew ID: 18, Name: Liam, Gender: M, Country: France, Airline: United Airlines
Crew ID: 19, Name: Charlotte, Gender: F, Country: France, Airline: United Airlines
Crew ID: 20, Name: Ella, Gender: F, Country: Australia, Airline: United Airlines
Crew ID: 21, Name: Lucas, Gender: M, Country: Canada, Airline: Southwest Airlines
Crew ID: 22, Name: Lily, Gender: F, Country: Canada, Airline: Southwest Airlines
Crew ID: 23, Name: Alexander, Gender: M, Country: Italy, Airline: Southwest Airlines
Crew ID: 24, Name: Grace, Gender: F, Country: Italy, Airline: Southwest Airlines
Crew ID: 25, Name: James, Gender: M, Country: Spain, Airline: Southwest Airlines
Crew ID: 26, Name: Oliver, Gender: M, Country: Spain, Airline: Emirates
Crew ID: 27, Name: Harper, Gender: F, Country: Brazil, Airline: Emirates
Crew ID: 28, Name: Benjamin, Gender: M, Country: Brazil, Airline: Emirates
Crew ID: 29, Name: Evelyn, Gender: F, Country: Mexico, Airline: Emirates
Crew ID: 30, Name: Logan, Gender: M, Country: Mexico, Airline: Emirates
Crew ID: 31, Name: Jacob, Gender: M, Country: Malaysia, Airline: Lufthansa
Crew ID: 32, Name: Sophia, Gender: F, Country: Malaysia, Airline: Lufthansa
Crew ID: 33, Name: Daniel, Gender: M, Country: Singapore, Airline: Lufthansa
Crew ID: 34, Name: Emily, Gender: F, Country: Singapore, Airline: Lufthansa
Crew ID: 35, Name: Lucas, Gender: M, Country: South Korea, Airline: Lufthansa
Crew ID: 36, Name: Ava, Gender: F, Country: India, Airline: British Airways
Crew ID: 37, Name: Liam, Gender: M, Country: India, Airline: British Airways
Crew ID: 38, Name: Olivia, Gender: F, Country: India, Airline: British Airways
Crew ID: 39, Name: Ethan, Gender: M, Country: India, Airline: British Airways
Crew ID: 40, Name: Isabella, Gender: F, Country: India, Airline: British Airways
Crew ID: 41, Name: James, Gender: M, Country: UAE, Airline: Air France
Crew ID: 42, Name: Emma, Gender: F, Country: UAE, Airline: Air France
Crew ID: 43, Name: Alexander, Gender: M, Country: UAE, Airline: Air France
Crew ID: 44, Name: Grace, Gender: F, Country: UAE, Airline: Air France
Crew ID: 45, Name: William, Gender: M, Country: UAE, Airline: Air France
Crew ID: 46, Name: Olivia, Gender: F, Country: UAE, Airline: Qantas Airways
Crew ID: 47, Name: Noah, Gender: M, Country: UAE, Airline: Qantas Airways
Crew ID: 48, Name: Mia, Gender: F, Country: UAE, Airline: Qantas Airways
Crew ID: 49, Name: Ethan, Gender: M, Country: UAE, Airline: Qantas Airways
Crew ID: 50, Name: Charlotte, Gender: F, Country: UAE, Airline: Qantas Airways
Crew ID: 51, Name: William, Gender: M, Country: Qatar, Airline: Singapore Airlines
Crew ID: 52, Name: Sophia, Gender: F, Country: Qatar, Airline: Singapore Airlines
Crew ID: 53, Name: Liam, Gender: M, Country: Qatar, Airline: Singapore Airlines
Crew ID: 54, Name: Isabella, Gender: F, Country: Qatar, Airline: Singapore Airlines
Crew ID: 55, Name: Lucas, Gender: M, Country: Qatar, Airline: Singapore Airlines
Crew ID: 56, Name: Oliver, Gender: M, Country: Spain, Airline: Cathay Pacific
Crew ID: 57, Name: Sophia, Gender: F, Country: Spain, Airline: Cathay Pacific
Crew ID: 58, Name: Liam, Gender: M, Country: Spain, Airline: Cathay Pacific
Crew ID: 59, Name: Emma, Gender: F, Country: Spain, Airline: Cathay Pacific
Crew ID: 60, Name: Noah, Gender: M, Country: Spain, Airline: Cathay Pacific
Crew ID: 61, Name: Ava, Gender: F, Country: Mexico, Airline: Qatar Airways
Crew ID: 62, Name: Oliver, Gender: M, Country: Mexico, Airline: Qatar Airways
Crew ID: 63, Name: Emma, Gender: F, Country: Mexico, Airline: Qatar Airways
Crew ID: 64, Name: Liam, Gender: M, Country: Mexico, Airline: Qatar Airways
Crew ID: 65, Name: Sophia, Gender: F, Country: Mexico, Airline: Qatar Airways
Crew ID: 66, Name: Lucas, Gender: M, Country: Brazil, Airline: Turkish Airlines
Crew ID: 67, Name: Ava, Gender: F, Country: Brazil, Airline: Turkish Airlines
Crew ID: 68, Name: Oliver, Gender: M, Country: Brazil, Airline: Turkish Airlines
```

## SQL Queries

Displaying Passenger Details.

```
SELECT
    p.passengerfirstname AS Passenger_FirstName,
    p.passengerlastname AS Passenger_LastName,
    p.passengergender AS S,
    p.passengerage AS Passenger_Age,
    a.airlinename AS Airline_Name,
    fc.flightid AS Flight_ID,
    fc.pilotid AS Pilot_ID,
    fc.headairhostessid AS HeadAirHostess_ID
FROM
    passengers p
JOIN
    passengersgoing pg ON p.passengersid = pg.passengerid
JOIN
    flightsgoing fg ON pg.flightid = fg.flightsgoingid
JOIN
    airlinesaeroplanes aa ON fg.airlinesaeroplanesid = aa.airlinesaeroplanesid
JOIN
    airlines a ON aa.airlineid = a.airlineid
JOIN
    flightcrew fc ON fg.flightsgoingid = fc.flightid;
```

Output

PASSENGER_FIRSTNAME	PASSENGER_LASTNAME	S	PASSENGER_AGE	AIRLINE_NAME	FLIGHT_ID	PILOT_ID	HEADAIRHOSTESS_ID
Rahul	Sharma	M	35	American Airlines	1	1	3
Rahul	Sharma	M	35	AirAsia	2	11	13
Priya	Patel	F	28	Turkish Airlines	3	16	18
Priya	Patel	F	28	Air New Zealand	4	21	23
Amit	Singh	M	42	Korean Air	5	26	28
Amit	Singh	M	42	Iberia	6	31	33
Neha	Gupta	F	30	Frontier Airlines	7	36	38
Neha	Gupta	F	30	Emirates	8	41	43
Suresh	Kumar	M	25	IndiGo	9	46	48
Suresh	Kumar	M	25	Flydubai	10	51	53
Anjali	Desai	F	33	Lufthansa	11	56	58
Anjali	Desai	F	33	ANA (All Nippon Airways)	12	61	63
Vikram	Rao	M	40	Finnair	13	66	68
Vikram	Rao	M	40	LATAM Airlines	14	71	73
Kavita	Menon	F	28	EasyJet	15	76	78
Kavita	Menon	F	28	Delta Air Lines	16	81	83
Manish	Shah	M	45	Singapore Airlines	17	86	88
Manish	Shah	M	45	Alaska Airlines	18	91	93
Pooja	Reddy	F	32	Southwest Airlines	19	96	99
Pooja	Reddy	F	32	Flydubai	20	100	104
Rajesh	Krishnan	M	38	British Airways	21	106	108
Rajesh	Krishnan	M	38	KLM Royal Dutch Airlines	22	110	112
Divya	Sinha	F	27	Thai Airways	23	116	119
Divya	Sinha	F	27	Hawaiian Airlines	24	121	123
Sanjay	Verma	M	41	United Airlines	25	126	128
Sanjay	Verma	M	41	United Airlines	26	131	133
Shilpa	Chopra	F	29	Cathay Pacific	27	136	138
Shilpa	Chopra	F	29	Norwegian Air Shuttle	28	141	143
Alok	Joshi	M	36	Air Arabia	29	146	148
Alok	Joshi	M	36	Air France	30	151	153
Ananya	Mukherjee	F	31	Japan Airlines	31	156	158
Ananya	Mukherjee	F	31	Malaysia Airlines	32	161	163
Arjun	Nair	M	43	EVA Air	33	166	168
Arjun	Nair	M	43	Southwest Airlines	34	171	173
Sneha	Rajput	F	30	Southwest Airlines	35	176	178
Sneha	Rajput	F	30	Emirates	36	181	183
Anand	Srivastava	M	37	Qatar Airways	37	186	188
Anand	Srivastava	M	37	Scandinavian Airlines (SAS)	38	191	193
Deepika	Gandhi	F	26	Allegiant Air	39	196	198
Deepika	Gandhi	F	26	American Airlines	40	201	203
Ajay	Sharma	M	44	Air France	41	206	208
Ajay	Sharma	M	44	Virgin Atlantic	42	211	213
Ritu	Choudhary	F	31	Korean Air	43	216	218
Ritu	Choudhary	F	31	JetBlue Airways	44	221	223
Rajat	Sinha	M	39	Delta Air Lines	45	226	228
Rajat	Sinha	M	39	Delta Air Lines	46	231	233
Vidya	Narayan	F	29	Singapore Airlines	47	236	238
Vidya	Narayan	F	29	Aeroflot Russian Airlines	48	241	243
Anil	Pillai	M	42	Spirit Airlines	49	246	248
Anil	Pillai	M	42	Emirates	50	251	253
Kirti	Patil	F	28	Air Canada	51	1	3
Kirti	Patil	F	28	EgyptAir	52	11	13
Raj	Malhotra	M	33	Azul Brazilian Airlines	53	16	18
Raj	Malhotra	M	33	Flydubai	54	21	23
Pallavi	Iyer	F	29	British Airways	55	26	28
Pallavi	Iyer	F	29	ANA (All Nippon Airways)	56	31	33

Displaying Top 5 Busiest Airports.

```
SELECT * FROM (
    SELECT
        ap.AIRPORTNAME,
        ap.AIRPORTCITY,
        ap.AIRPORTCOUNTRY,
        COUNT(*) AS TOTAL_FLIGHTS
    FROM
        AIRPORTS ap
    JOIN FLIGHTSGOING fg
        ON ap.AIRPORTID = fg.DEPARTUREAIRPORTID
        OR ap.AIRPORTID = fg.ARRIVALAIRPORTID
    GROUP BY
        ap.AIRPORTNAME, ap.AIRPORTCITY, ap.AIRPORTCOUNTRY
    ORDER BY
        TOTAL_FLIGHTS DESC
)
WHERE ROWNUM <= 5;
```

Output

AIRPORTNAME	AIRPORTCITY	AIRPORTCOUNTRY	TOTAL_FLIGHTS
Incheon International Airport	Incheon	South Korea	15
Beijing Capital International Airport	Beijing	China	10
Zurich Airport	Zurich	Switzerland	10
Frankfurt Airport	Frankfurt	Germany	10
Amsterdam Airport Schiphol	Amsterdam	Netherlands	9

Displaying Average Distance Airline.

```
SELECT * FROM (
    SELECT
        al.AIRLINENAME,
        COUNT(fg.FLIGHTSGOINGID) AS TOTAL_FLIGHTS,
        ROUND(AVG(fg.DESTINATIONDISTANCE), 2) AS AVG_DISTANCE
    FROM
        AIRLINES al
    JOIN AIRLINESAEROPLANES aa ON al.AIRLINEID = aa.AIRLINEID
    JOIN FLIGHTSGOING fg ON aa.AIRLINESAEROPLANESID = fg.AIRLINESAEROPLANESID
    GROUP BY
        al.AIRLINENAME
    HAVING
        AVG(fg.DESTINATIONDISTANCE) > (
            SELECT AVG(DESTINATIONDISTANCE)
            FROM FLIGHTSGOING
        )
    ORDER BY
        AVG_DISTANCE DESC
)
WHERE ROWNUM <= 5;
```

Output

AIRLINENAME	TOTAL_FLIGHTS	AVG_DISTANCE
Virgin Australia	2	13400
Azul Brazilian Airlines	2	12200
Qantas Airways	2	11200
Aeroflot Russian Airlines	3	9933.33
Virgin Atlantic	3	9066.67

Displaying Domestic And International Flights.

```
SELECT
    fg.FLIGHTSGOINGID AS flight_id,
    fg.DEPARTUREAIRPORTID AS source_airport_code,
    fg.ARRIVALAIRPORTID AS destination_airport_code,
    a1.AIRPORTCOUNTRY AS source_country,
    a2.AIRPORTCOUNTRY AS destination_country,
    CASE
        WHEN a1.AIRPORTCOUNTRY = a2.AIRPORTCOUNTRY THEN 'Domestic'
        ELSE 'International'
    END AS flight_type
FROM FLIGHTSGOING fg
JOIN Airports a1 ON fg.DEPARTUREAIRPORTID = a1.AIRPORTID
JOIN Airports a2 ON fg.ARRIVALAIRPORTID = a2.AIRPORTID;
```

## Output

FLIGHT_ID	SOURCE_AIRPORT_CODE	DESTINATION_AIRPORT_CODE	SOURCE_COUNTRY	DESTINATION_COUNTRY	FLIGHT_TYPE
161	2	1	China	United States	International
36	9	1	Singapore	United States	International
136	9	1	Singapore	United States	International
79	40	1	Austria	United States	International
160	1	2	United States	China	International
3	4	2	United States	China	International
103	4	2	United States	China	International
25	13	2	United States	China	International
125	13	2	United States	China	International
68	22	2	United States	China	International
85	23	3	Australia	United Arab Emirates	International
52	37	3	Spain	United Arab Emirates	International
98	30	4	Ireland	United States	International
6	16	5	Australia	Japan	International
106	16	5	Australia	Japan	International
81	37	5	Spain	Japan	International
46	49	5	New Zealand	Japan	International
146	49	5	New Zealand	Japan	International
18	42	6	United States	United Kingdom	International
118	42	6	United States	United Kingdom	International
73	44	7	Morocco	France	International
30	48	7	Hungary	France	International
130	48	7	Hungary	France	International
55	6	8	United Kingdom	Germany	International
1	12	8	China	Germany	International
101	12	8	China	Germany	International
93	14	8	United States	Germany	International
66	18	9	Italy	Singapore	International
91	36	9	South Korea	Singapore	International
15	17	10	India	South Korea	International
115	17	10	India	South Korea	International
49	19	10	Israel	South Korea	International
149	19	10	Israel	South Korea	International
40	25	10	Switzerland	South Korea	International
140	25	10	Switzerland	South Korea	International
76	27	10	Malaysia	South Korea	International
32	41	11	United States	Netherlands	International
132	41	11	United States	Netherlands	International
42	42	11	United States	Netherlands	International
142	42	11	United States	Netherlands	International
9	21	12	Canada	China	International
109	21	12	Canada	China	International
22	38	12	United Kingdom	China	International
122	38	12	United Kingdom	China	International
61	45	12	Sweden	China	International
59	4	13	United States	United States	Domestic
97	17	13	India	United States	International
5	3	14	United Arab Emirates	United States	International
105	3	14	United Arab Emirates	United States	International
74	19	14	Israel	United States	International
4	7	15	France	Spain	International
104	7	15	France	Spain	International
24	32	15	United Arab Emirates	Spain	International
124	32	15	United Arab Emirates	Spain	International
116	1	16	United States	Australia	International
16	1	16	United States	Australia	International
84	8	16	Germany	Australia	International
34	29	17	Germany	India	International
134	29	17	Germany	India	International
44	35	17	United States	India	International
144	35	17	United States	India	International
13	5	18	Japan	Italy	International
88	27	18	Malaysia	Italy	International
54	33	18	Canada	Italy	International
64	8	19	Germany	Israel	International
11	9	19	Singapore	Israel	International
111	9	19	Singapore	Israel	International

Displaying Passenger Details Of Passengers Taking A Round Trip.

```
SELECT
    p.PASSENGERSID,
    p.PASSENGERFIRSTNAME || ' ' || p.PASSENGERLASTNAME AS PASSENGERNAME,
    a1.AIRPORTNAME AS OUTBOUND_FROM,
    a2.AIRPORTNAME AS OUTBOUND_TO,
    a3.AIRPORTNAME AS RETURN_FROM,
    a4.AIRPORTNAME AS RETURN_TO
FROM Passengers p
JOIN PassengersGoing pg1 ON p.PASSENGERSID = pg1.PASSENGERID
JOIN FlightsGoing fg1 ON pg1.FLIGHTID = fg1.FLIGHTSGOINGID
JOIN PassengersGoing pg2 ON p.PASSENGERSID = pg2.PASSENGERID
JOIN FlightsGoing fg2 ON pg2.FLIGHTID = fg2.FLIGHTSGOINGID
JOIN Airports a1 ON fg1.DEPARTUREAIRPORTID = a1.AIRPORTID
JOIN Airports a2 ON fg1.ARRIVALAIRPORTID = a2.AIRPORTID
JOIN Airports a3 ON fg2.DEPARTUREAIRPORTID = a3.AIRPORTID
JOIN Airports a4 ON fg2.ARRIVALAIRPORTID = a4.AIRPORTID
WHERE fg1.DEPARTUREAIRPORTID = fg2.ARRIVALAIRPORTID
    AND fg1.ARRIVALAIRPORTID = fg2.DEPARTUREAIRPORTID
    AND fg1.DEPARTUREDATETIME < fg2.DEPARTUREDATETIME;
```

Output

PASSENGERSID	PASSENGERNAME	OUTBOUND_FROM	OUTBOUND_TO	RETURN_FROM	RETURN_TO
251	John Doe	Hartsfield-Jackson Atlanta International Airport	Beijing Capital International Airport	Beijing Capital International Airport	Hartsfield-Jackson Atlanta International Airport
252	Jane Smith	Hartsfield-Jackson Atlanta International Airport	Beijing Capital International Airport	Beijing Capital International Airport	Hartsfield-Jackson Atlanta International Airport
253	Emily Johnson	Hartsfield-Jackson Atlanta International Airport	Beijing Capital International Airport	Beijing Capital International Airport	Hartsfield-Jackson Atlanta International Airport

Elapsed: 00:00:00.020  
3 rows selected.

# **EXPECTED OUTCOMES**

By implementing the Flight Management System, we anticipate the following outcomes:

## **Comprehensive Flight Data Management**

- Centralized Schema: A well-defined `schema.sql` ensures structured storage of flight-related data, including flights, crew, and schedules.
- Data Integrity: Consistent schema design minimizes redundancy and maintains data accuracy.

## **Efficient Data Operations**

- Automated Data Insertion: The `Insert Data Script` facilitates bulk data entry, reducing manual effort and errors.
- Dynamic Data Handling: SQL scripts enable dynamic operations like updates and deletions, ensuring up-to-date information.

## **Enhanced Crew Management**

- Cursor Implementation: The `curu3_crew_dets.sql` script utilizes cursors to process crew details sequentially, allowing for complex data manipulations.
- Detailed Reporting: Ability to generate comprehensive reports on crew assignments and schedules.

## **Improved Data Accessibility**

- Structured SQL Codebase: Organized SQL scripts enhance readability and maintainability, facilitating easier updates and scalability.
- Modular Design: Separation of concerns in scripts allows for targeted modifications without affecting the entire system.

## **Scalability and Future Integration**

- Foundation for Expansion: The current structure supports potential integration with front-end applications or APIs for real-time data access.
- Adaptability: Modular SQL scripts can be adapted for other transportation management systems with minimal changes.

## **CONCLUSION**

The Flight Management System is a robust, SQL-driven solution aimed at improving the efficiency and accuracy of airline operations. By centralizing flight schedules, crew information, and other operational data, the system enhances overall data management and supports seamless coordination between different functional units.

Through well-structured SQL scripts—including schema creation, data insertion, and cursor-based operations—the project demonstrates strong backend capabilities that can be easily integrated into larger aviation management platforms. It reduces the reliance on manual processes, minimizes errors, and ensures that flight and crew details are processed in a structured and logical manner.

This system lays the groundwork for future enhancements such as web-based user interfaces, real-time flight tracking, and integration with booking and logistics modules—paving the way for a comprehensive digital transformation in flight operations.