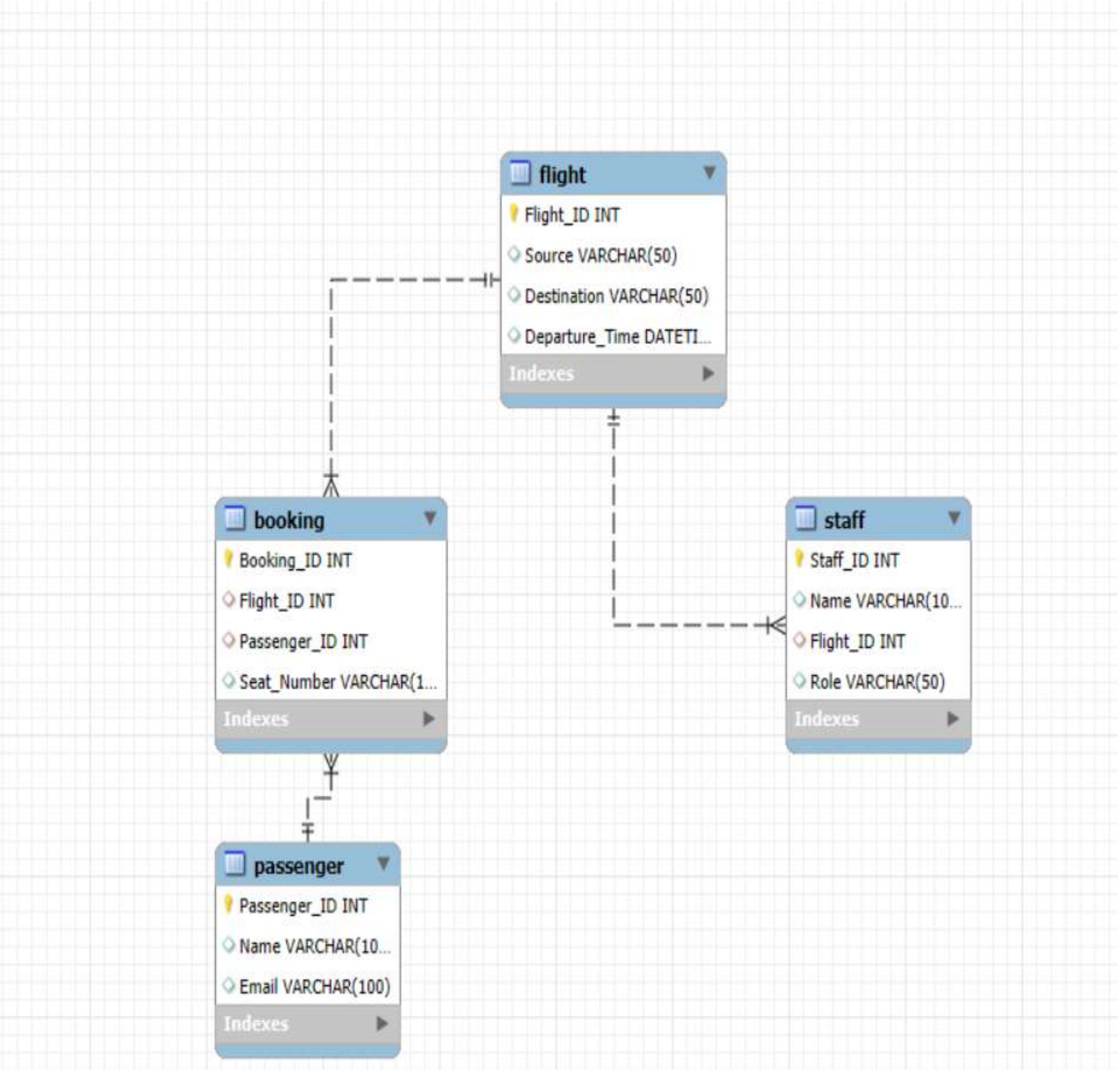
SQL TOPIC PROJECT: AIR INDIA MANAGEMENT

Name : Maithili Ahire

ABSTRACT

The **Air India Management System** project is developed using **MySQL** with a focus on implementing **SQL JOIN operations**. It simulates real-world airline management by creating and linking four main tables: **Flights**, **Passengers**, **Bookings**, and **Staff**. the project demonstrates how to fetch combined data across tables—for example, viewing passenger details along with their booked flights. An **ER diagram** is also created to define table relationships. This project helps in understanding how relational databases are used in aviation for better data management and decision-making

ER DIAGRAM



flight
- Flight_ID INT
- Source VARCHAR(50)
- Destination VARCHAR(50)
- Departure_Time DATETI...

Indexes

booking
- Booking_ID INT
- Flight_ID INT
- Passenger_ID INT
- Seat_Number VARCHAR(1...

Indexes

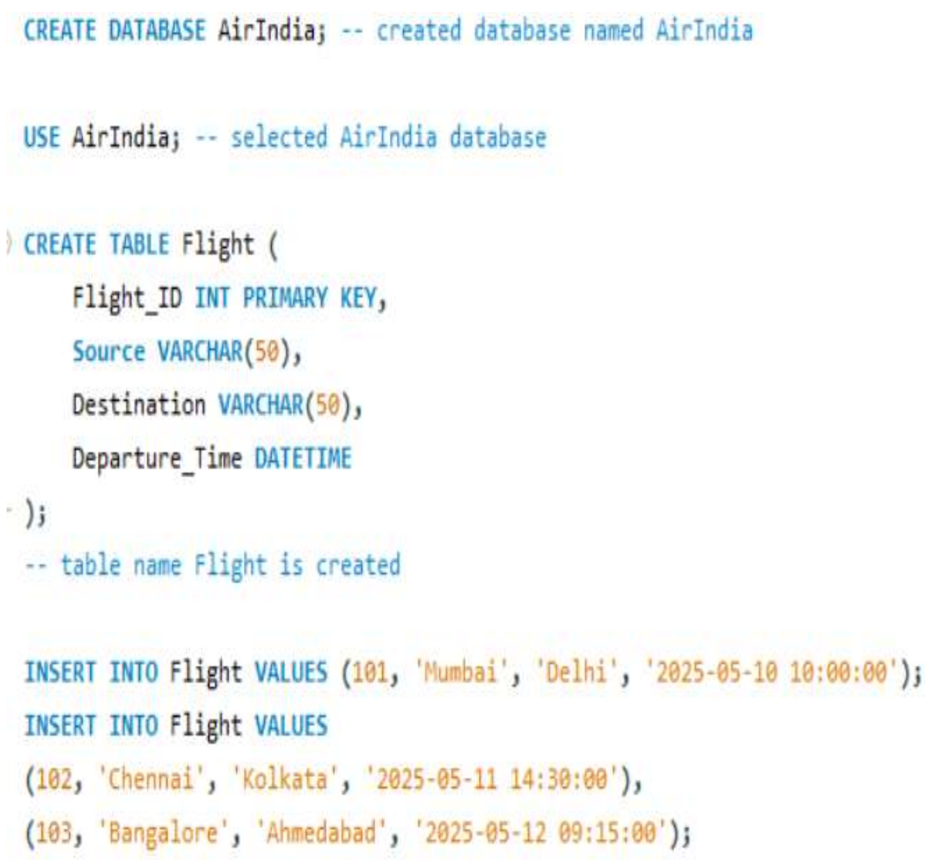
staff
- Staff_ID INT
- Name VARCHAR(10...
- Flight_ID INT
- Role VARCHAR(50)

Indexes

passenger
- Passenger_ID INT
- Name VARCHAR(10...
- Email VARCHAR(100)

Indexes

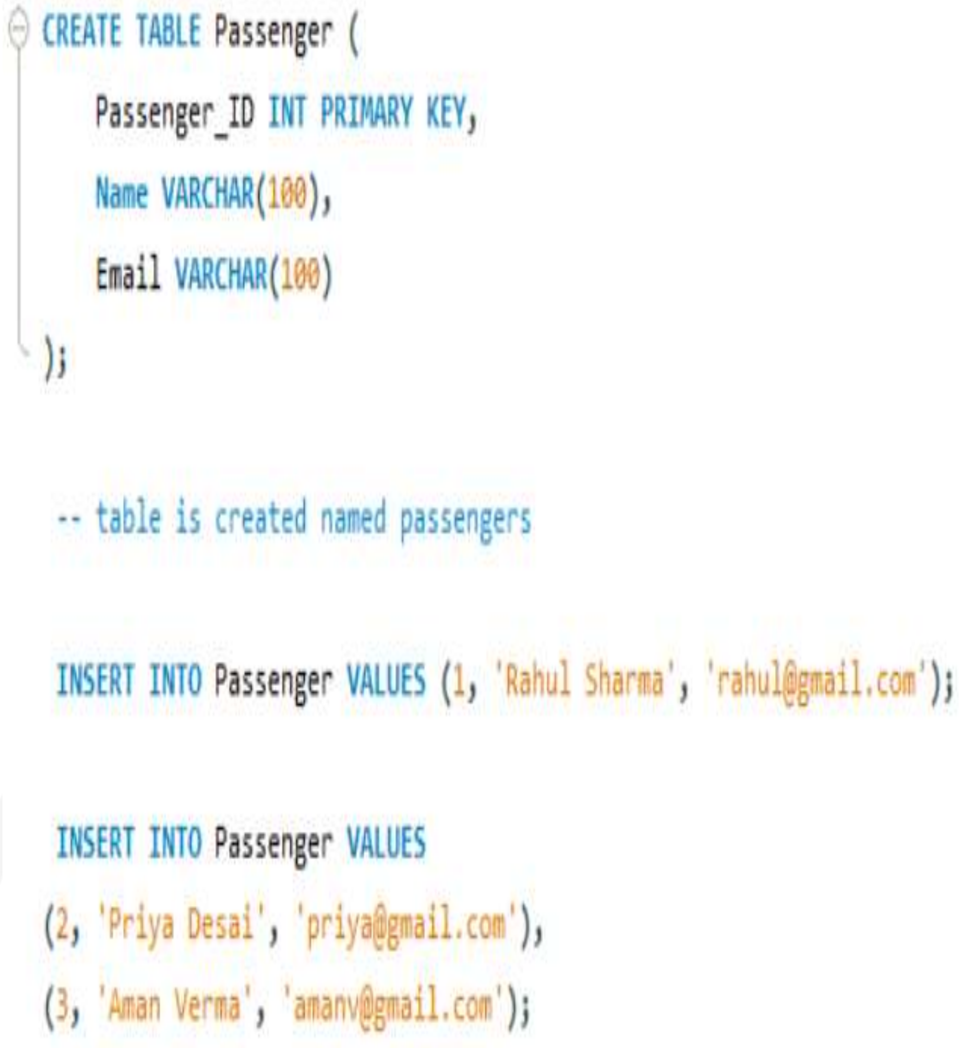
Structure of table 1



```
CREATE DATABASE AirIndia; -- created database named AirIndia

USE AirIndia; -- selected AirIndia database

CREATE TABLE Flight (
    Flight_ID INT PRIMARY KEY,
    Source VARCHAR(50),
    Destination VARCHAR(50),
    Departure_Time DATETIME
);
-- table name Flight is created

INSERT INTO Flight VALUES (101, 'Mumbai', 'Delhi', '2025-05-10 10:00:00');
INSERT INTO Flight VALUES
(102, 'Chennai', 'Kolkata', '2025-05-11 14:30:00'),
(103, 'Bangalore', 'Ahmedabad', '2025-05-12 09:15:00');
```

This table is called **Flight** and it keeps details of Air India flights. It shows where the flight is going from (**Source**), where it is going to (**Destination**), and what time it leaves **departure time**. Each flight has a special number **flight id** Three flights are added in this example.

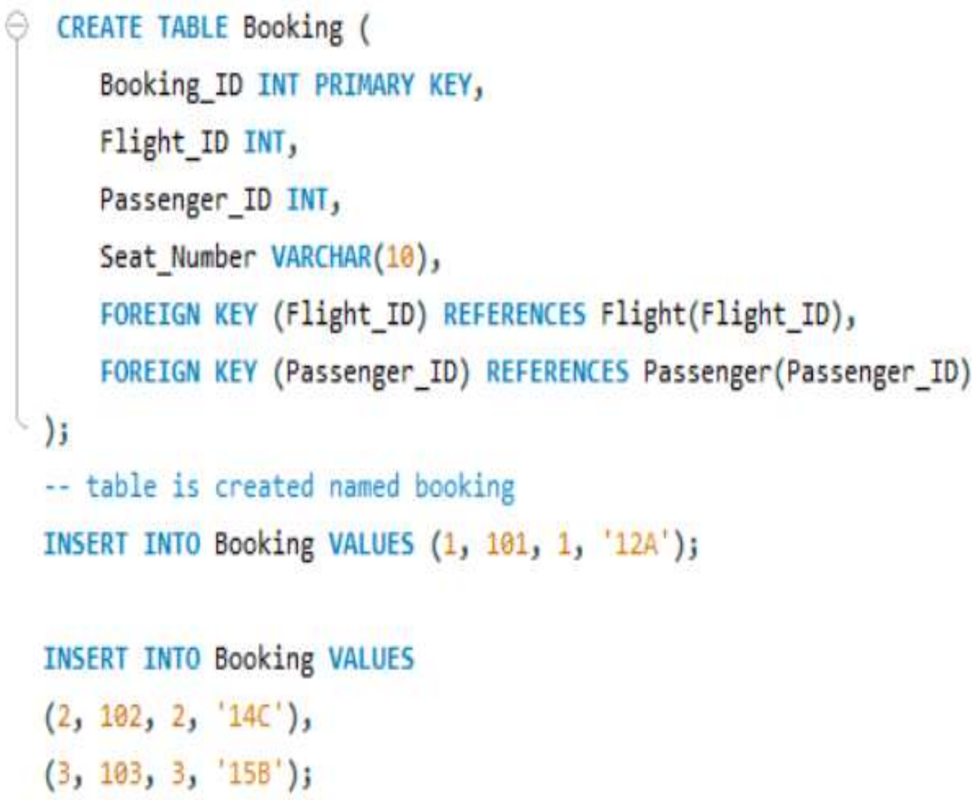
STRUCTURE OF TABLE 2



```sql
CREATE TABLE Passenger (
    Passenger_ID INT PRIMARY KEY,
    Name VARCHAR(100),
    Email VARCHAR(100)
);

 -- table is created named passengers

 INSERT INTO Passenger VALUES (1, 'Rahul Sharma', 'rahul@gmail.com');

 INSERT INTO Passenger VALUES
(2, 'Priya Desai', 'priya@gmail.com'),
(3, 'Aman Verma', 'amanv@gmail.com');
```

This table is named **Passenger** and it stores information about people who are flying. Each passenger has a unique **Passenger ID**, along with their **Name** and **Email**. Three passengers are added here: Rahul Sharma, Priya Desai, and Aman Verma.

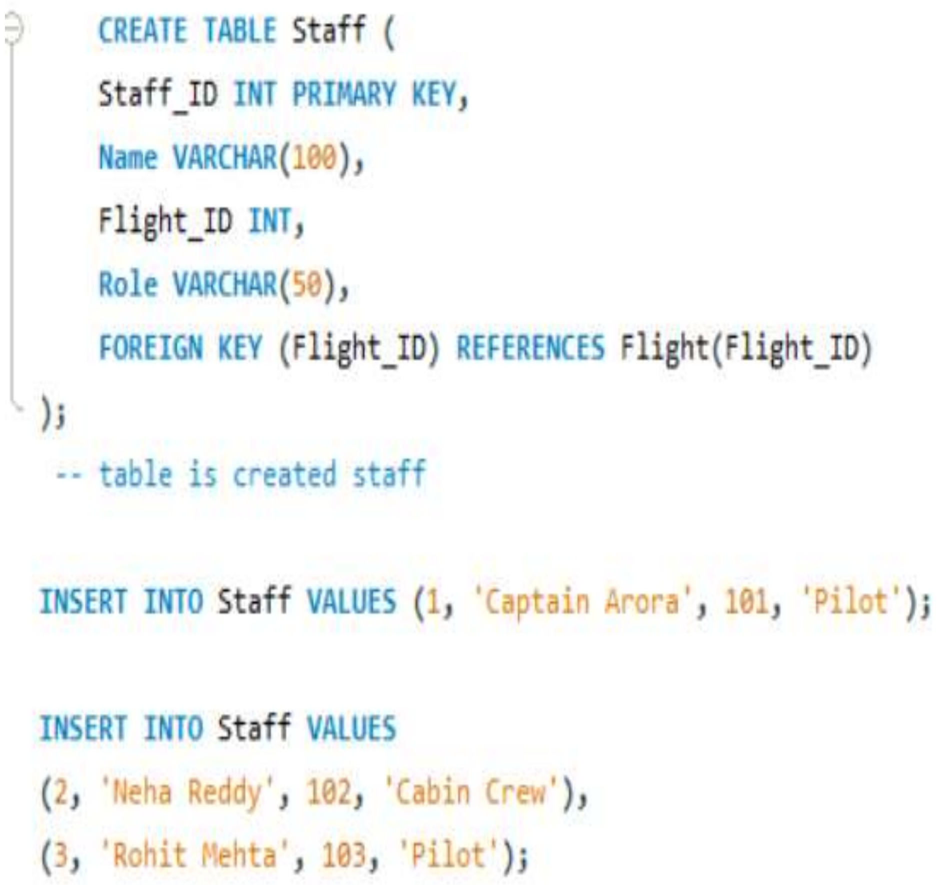
Structure of table 3



```
CREATE TABLE Booking (
    Booking_ID INT PRIMARY KEY,
    Flight_ID INT,
    Passenger_ID INT,
    Seat_Number VARCHAR(10),
    FOREIGN KEY (Flight_ID) REFERENCES Flight(Flight_ID),
    FOREIGN KEY (Passenger_ID) REFERENCES Passenger(Passenger_ID)
);
-- table is created named booking
INSERT INTO Booking VALUES (1, 101, 1, '12A');

INSERT INTO Booking VALUES
(2, 102, 2, '14C'),
(3, 103, 3, '15B');
```

- This table is called **Booking** and it stores details of flight bookings. It links each booking to a **FlightID** and a **PassengerID**, and also records the **Seat Number**. It connects the **Flight** and **Passenger** tables using foreign keys. Three bookings are added here, showing which passenger booked which seat on which flight.

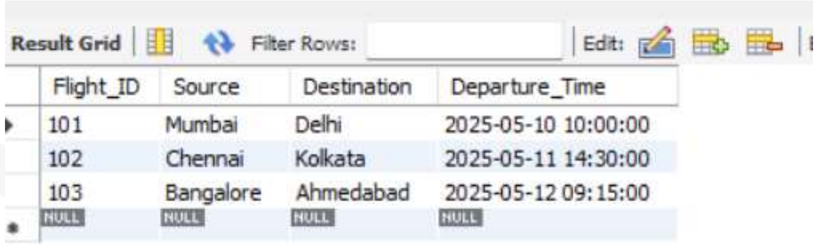
Structure of table 4



```
    CREATE TABLE Staff (
    Staff_ID INT PRIMARY KEY,
    Name VARCHAR(100),
    Flight_ID INT,
    Role VARCHAR(50),
    FOREIGN KEY (Flight_ID) REFERENCES Flight(Flight_ID)
);
 -- table is created staff

INSERT INTO Staff VALUES (1, 'Captain Arora', 101, 'Pilot');

INSERT INTO Staff VALUES
(2, 'Neha Reddy', 102, 'Cabin Crew'),
(3, 'Rohit Mehta', 103, 'Pilot');
```

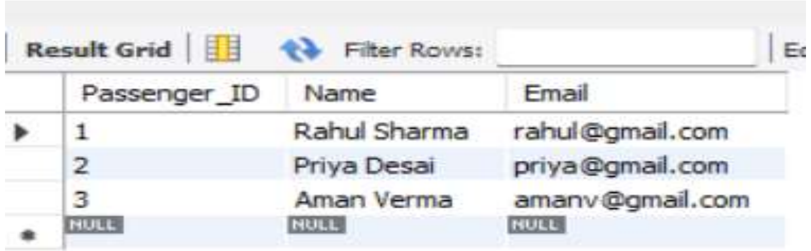
This table is named **Staff** and it stores details about airline staff members. It includes the **Staff ID**, their **Name**, the **Flight ID** they are assigned to, and their **Role** (like Pilot or Cabin Crew). The table is linked to the **Flight** table using a foreign key. Three staff members are added in this example.

Contents of table



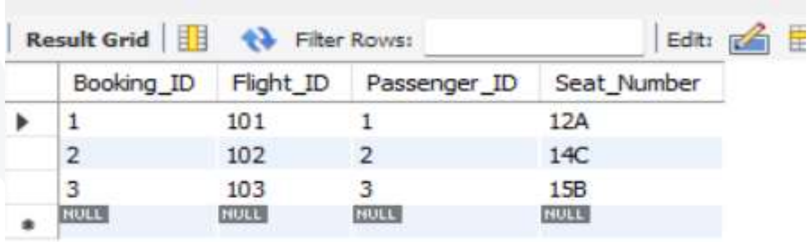
Flight_ID	Source	Destination	Departure_Time
101	Mumbai	Delhi	2025-05-10 10:00:00
102	Chennai	Kolkata	2025-05-11 14:30:00
103	Bangalore	Ahmedabad	2025-05-12 09:15:00
NULL	NULL	NULL	NULL

Flight – Lists flight ID, source, destination, and departure time.



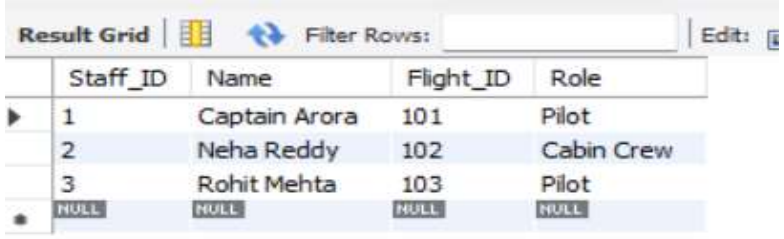
Passenger_ID	Name	Email
1	Rahul Sharma	rahul@gmail.com
2	Priya Desai	priya@gmail.com
3	Aman Verma	amanv@gmail.com
NULL	NULL	NULL

Passenger – Contains passenger ID, name, and email details.



Booking_ID	Flight_ID	Passenger_ID	Seat_Number
1	101	1	12A
2	102	2	14C
3	103	3	15B
NULL	NULL	NULL	NULL

Booking – Records bookings with booking ID, flight ID, passenger ID, and seat number

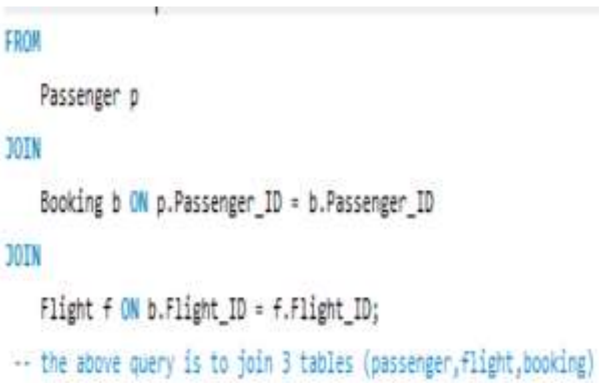


Staff_ID	Name	Flight_ID	Role
1	Captain Arora	101	Pilot
2	Neha Reddy	102	Cabin Crew
3	Rohit Mehta	103	Pilot
NULL	NULL	NULL	NULL

Staff – Shows staff ID, name, assigned flight ID, and their role (e.g., Pilot, Cabin Crew)

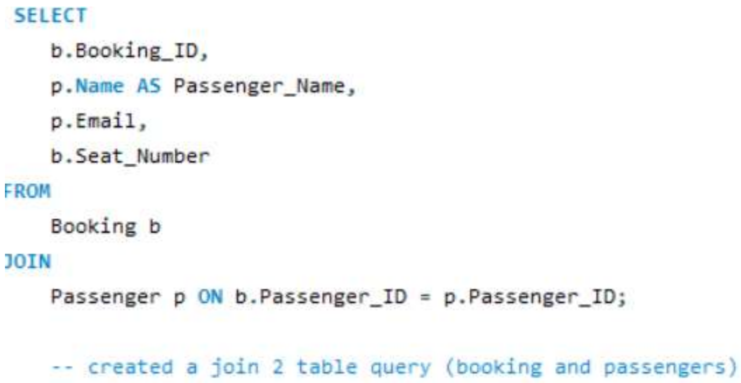
Join

1



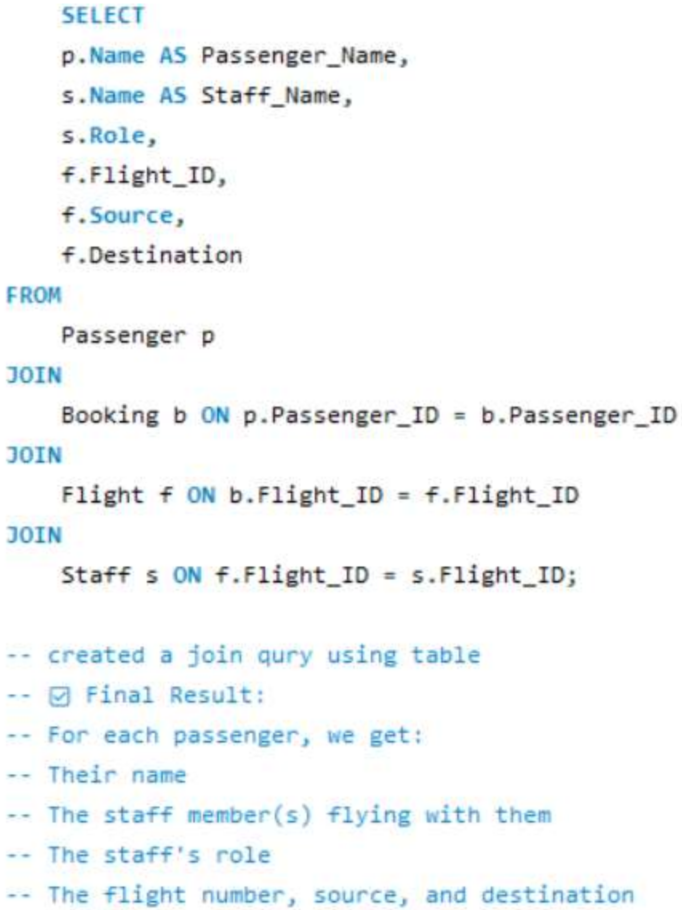
```
FROM
    Passenger p
JOIN
    Booking b ON p.Passenger_ID = b.Passenger_ID
JOIN
    Flight f ON b.Flight_ID = f.Flight_ID;
-- the above query is to join 3 tables (passenger,flight,booking)
```

2



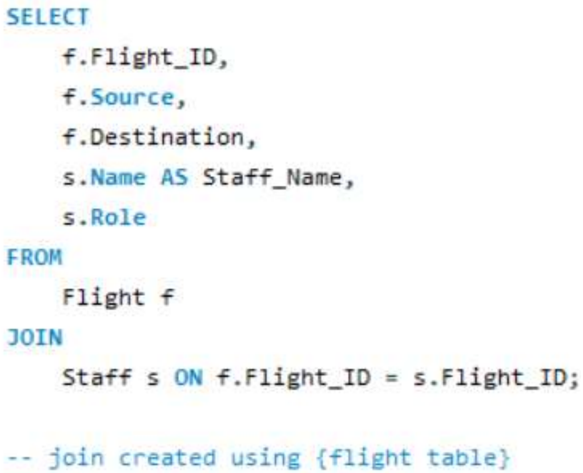
```
 SELECT
    b.Booking_ID,
    p.Name AS Passenger_Name,
    p.Email,
    b.Seat_Number
FROM
    Booking b
JOIN
    Passenger p ON b.Passenger_ID = p.Passenger_ID;

    -- created a join 2 table query (booking and passengers)
```

3



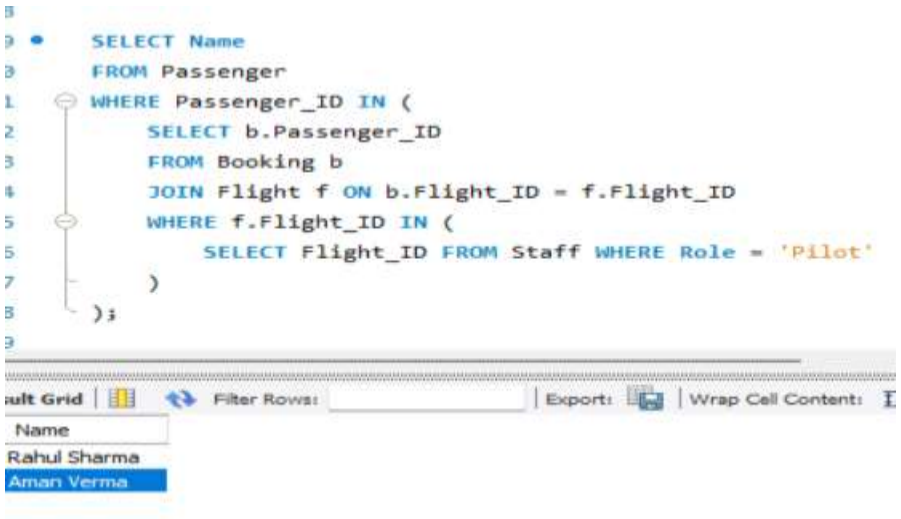
```
    SELECT
    p.Name AS Passenger_Name,
    s.Name AS Staff_Name,
    s.Role,
    f.Flight_ID,
    f.Source,
    f.Destination
FROM
    Passenger p
JOIN
    Booking b ON p.Passenger_ID = b.Passenger_ID
JOIN
    Flight f ON b.Flight_ID = f.Flight_ID
JOIN
    Staff s ON f.Flight_ID = s.Flight_ID;

-- created a join qury using table
-- ☑ Final Result:
-- For each passenger, we get:
-- Their name
-- The staff member(s) flying with them
-- The staff's role
-- The flight number, source, and destination
```

4



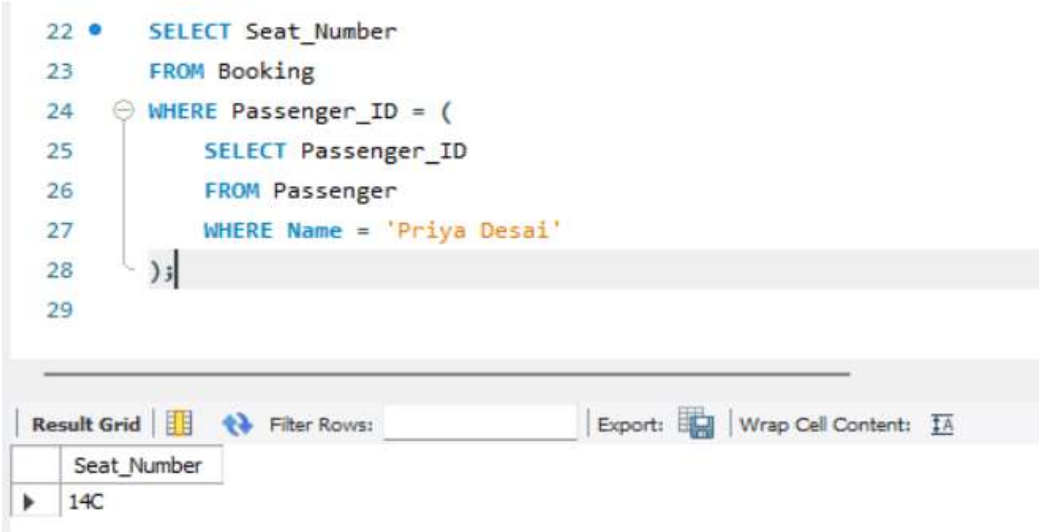
```
SELECT
    f.Flight_ID,
    f.Source,
    f.Destination,
    s.Name AS Staff_Name,
    s.Role
FROM
    Flight f
JOIN
    Staff s ON f.Flight_ID = s.Flight_ID;

-- join created using {flight table}
```

subquery



```
SELECT Name
FROM Passenger
WHERE Passenger_ID IN (
    SELECT b.Passenger_ID
    FROM Booking b
    JOIN Flight f ON b.Flight_ID = f.Flight_ID
    WHERE f.Flight_ID IN (
        SELECT Flight_ID FROM Staff WHERE Role = 'Pilot'
    )
);
```

```
Name
Rahul Sharma
Aman Verma
```



```
SELECT Seat_Number
FROM Booking
WHERE Passenger_ID = (
    SELECT Passenger_ID
    FROM Passenger
    WHERE Name = 'Priya Desai'
);
```

```
Seat_Number
14C
```

- The innermost subquery gets all Flight ID where the staff's role is **'Pilot'**.
- Then, we find all PassengerID in the Booking table who are on those flights.
- Finally, we retrieve the **names of those passengers** from the Passenger table.

- The inner query finds the PassengerID of **'Priya Desai'**
- The outer query then finds the Seat Number from the Booking table using that ID.

Conclusion

This project shows how SQL can be used to manage airline data efficiently. By linking flights, passengers, bookings, and staff using JOINs and subqueries, we can get useful information easily. It helps understand how databases support real-world systems like airline management.

THANK YOU