

▮ Data Analysis With SQL

Navneet

Data cleaning and Exploration

- Before the keys is defined, we need to apply the “NOT NULL” constraints.

The datatype has been SET to “VARCHAR(100)” as the columns which are used as foreign keys have to have the same datatype.

```
ALTER TABLE airports
```

```
ALTER COLUMN airport_id TYPE VARCHAR(100),
```

```
ALTER COLUMN airport_id SET NOT NULL;
```

```
ALTER TABLE planes
```

```
ALTER COLUMN plane_id TYPE VARCHAR(100),
```

```
ALTER COLUMN plane_id SET NOT NULL;
```

```
ALTER TABLE flights_data
```

```
ALTER COLUMN flight_id TYPE VARCHAR(100),
```

```
ALTER COLUMN flight_id SET NOT NULL;
```

```
ALTER TABLE flights_data
```

```
ALTER COLUMN arrival_airport_id TYPE VARCHAR(100),
```

```
ALTER COLUMN arrival_airport_id SET NOT NULL;
```

```
ALTER TABLE flights_data
```

```
ALTER COLUMN departure_airport_id TYPE VARCHAR(100),
```

```
ALTER COLUMN departure_airport_id SET NOT NULL;
```

```
ALTER TABLE flights_data
```

```
ALTER COLUMN plane_id TYPE VARCHAR(100),
```

```
ALTER COLUMN plane_id SET NOT NULL;
```

Fights – Defining the keys

- We can now define the primary key and foreign key for the table based on the design of our table schema.

we added the airport's key to the arrival airport key and departure airport key because we need data in both the arrival airport and departure airport.

```
ALTER TABLE airports  
ADD PRIMARY KEY (airport_id);
```

```
ALTER TABLE planes  
ADD PRIMARY KEY (plane_id);
```

```
ALTER TABLE flights_data  
ADD PRIMARY KEY (flight_id);
```

```
ALTER TABLE flights_data  
ADD FOREIGN KEY (departure_airport_id)  
REFERENCES airports (airport_id);
```

```
ALTER TABLE flights_data  
ADD FOREIGN KEY (arrival_airport_id)  
REFERENCES airports (airport_id);
```

```
ALTER TABLE flights_data  
ADD FOREIGN KEY (plane_id)  
REFERENCES planes (plane_id);
```

• Joining multiple tables together

joining multiple tables together to see all data First, we have to select all(*) from flight_data and then join the departure_airport_id column from the flight_data table to the airport_id column from the airport table and then join the plane_id column from the flight_data table to plane_id from planes table.

```
SELECT *  
FROM flights_data as f  
INNER JOIN airports as a  
ON f.departure_airport_id = a.airport_id  
INNER JOIN airports as c  
ON f.arrival_airport_id = c.airport_id  
INNER JOIN planes as p  
ON f.plane_id = p.plane_id
```

Flights – Master Table

Creating a master table for our analysis which is consist of all three table

```
SELECT fd.Flight_id,  
fd.airline,  
fd.departure_airport_id,  
fd.arrival_airport_id,  
fd.plane_id,  
fd.flight_delay_flag,  
fd.flight_delay_time_mins,  
a.airport_country AS departure_airport_country,  
a.opening_year AS departure_airport_year,  
a.customer_satisfaction_rating AS departure_airport_customer_satisfaction_rating,  
ap.airport_country AS arrival_airport_country,  
ap.opening_year AS arrival_airport_year,
```

```

ap.customer_satisfaction_rating AS arrival_airport_customer_satisfaction_rating,
p.plane_name,
p.suppliers_name,
p.passenger_capacity,
p.commission_year,
p.life_time
INTO flights_master
FROM flights_data AS fd
INNER JOIN airports AS a
ON a.airport_id = fd.departure_airport_id
INNER JOIN airports AS ap
ON ap.airport_id = fd.arrival_airport_id
INNER JOIN planes AS p
ON fd.plane_id = p.plane_id

```

• Flights – Analysis

As there are three tables in this case study, there is more data which we can use for our analysis.

We will look at the following aspects in the data :

- 1) The metrics surrounding the departure and arrival airports
- 2) The customer satisfaction scores and relating it to airport activity
- 3) The planes which are used for flights

1) The metrics surrounding the departure and arrival airports

- The departure and arrival airports can be included in an aggression
- This tells us how often a flight route occurred in the lists of flights present in the table

```
SELECT arrival_airport_id, arrival_airport_country,  
departure_airport_country, departure_airport_id,  
COUNT(*) AS airport_frequency  
FROM flights_master  
GROUP BY arrival_airport_id, arrival_airport_country,  
departure_airport_country, departure_airport_id
```

2) The customer satisfaction scores and relating it to airport activity

- The airport customer satisfaction rating analyzed to see how it relates to the number of flights in their respective airports.

```
SELECT departure_airport_id,  
departure_airport_country,  
departure_airport_customer_satisfaction_rating,  
COUNT(*) AS departure_airport_frequency  
FROM flights_master  
GROUP BY departure_airport_id,  
departure_airport_country,  
departure_airport_customer_satisfaction_rating  
ORDER BY COUNT(*) DESC
```

For arrival airport data analysis

```
SELECT arrival_airport_id,  
arrival_airport_country,  
arrival_airport_customer_satisfaction_rating,  
COUNT(*) AS departure_airport_frequency  
FROM flights_master  
GROUP BY arrival_airport_id,
```

```
arrival_airport_country,  
arrival_airport_customer_satisfaction_rating  
ORDER BY COUNT(*) DESC
```

3) The planes which are used for flights

- The different types of planes used for the flights can be analyzed. If we had the number of passengers present in each flight, we would be able to take our analysis further !

```
SELECT plane_id,  
plane_name,  
suppliers_name,  
passenger_capacity,  
commission_year,  
COUNT(*) AS flight_count  
FROM flights_master  
GROUP BY plane_id,  
plane_name,  
suppliers_name,  
passenger_capacity,  
commission_year  
ORDER BY COUNT(*) DESC
```