

SQL Exercise Solutions (Lecture 3)

January 26, 2025

1. Retrieve the flight details (flight_id, flightno, departure, arrival) for flights departing from 'Heathrow' airport.

```
SELECT f.flight_id, f.flightno, f.departure, f.arrival
FROM flight f
JOIN airport a ON f.from = a.airport_id
WHERE a.name = 'Heathrow';
```

2. Find all passengers who have made bookings for flights that depart from 'Heathrow' airport.

```
SELECT DISTINCT p.*
FROM passenger p
JOIN booking b ON p.passenger_id = b.passenger_id
JOIN flight f ON b.flight_id = f.flight_id
JOIN airport a ON f.from = a.airport_id
WHERE a.name = 'Heathrow';
```

3. List the airlines with more than 1000 flights.

```
SELECT a.name, COUNT(f.flight_id) AS flight_count
FROM airline a
JOIN flight f ON a.airline_id = f.airline_id
GROUP BY a.name
HAVING COUNT(f.flight_id) > 1000;
```

4. Retrieve the passengers who have not made any bookings.

```
SELECT p.*
FROM passenger p
LEFT JOIN booking b ON p.passenger_id = b.passenger_id
WHERE b.booking_id IS NULL;
```

5. Find flights that have at least 3 bookings.

```
SELECT f.*
FROM flight f
JOIN booking b ON f.flight_id = b.flight_id
GROUP BY f.flight_id
HAVING COUNT(b.booking_id) >= 3;
```

6. Categorize flights as 'Cheap', 'Medium', or 'Expensive' based on their average price (i= 300 , 301-400, i 400).

```

SELECT f.*,
       CASE
         WHEN AVG(b.price) <= 300 THEN 'Cheap'
         WHEN AVG(b.price) BETWEEN 301 AND 400 THEN 'Medium'
         ELSE 'Expensive'
       END AS price_category
FROM flight f
JOIN booking b ON f.flight_id = b.flight_id
GROUP BY f.flight_id;

```

7. Identify flights with a duration higher than the average duration for their respective airlines.

```

SELECT f.*
FROM flight f
WHERE TIMESTAMPDIFF(MINUTE, f.departure, f.arrival) >
      (SELECT AVG(TIMESTAMPDIFF(MINUTE, f2.departure, f2.arrival))
       FROM flight f2
       WHERE f2.airline_id = f.airline_id);

```

8. List the airports whose names start with 'Ales' (case-insensitive).

```

SELECT *
FROM airport
WHERE name LIKE 'Ales%';

```

9. Find passengers who have made bookings on flights departing from 'Heathrow' or 'Vienna'.

```

SELECT DISTINCT p.*
FROM passenger p
JOIN booking b ON p.passenger_id = b.passenger_id
JOIN flight f ON b.flight_id = f.flight_id
JOIN airport a ON f.from = a.airport_id
WHERE a.name IN ('Heathrow', 'Vienna');

```

10. Retrieve flights with the number of bookings for each flight.

```

SELECT f.*, COUNT(b.booking_id) AS booking_count
FROM flight f
LEFT JOIN booking b ON f.flight_id = b.flight_id
GROUP BY f.flight_id;

```

11. For each passenger, display their name and the total number of flights they have booked.

```

SELECT p.firstname, p.lastname, COUNT(b.booking_id) AS total_flights
FROM passenger p
LEFT JOIN booking b ON p.passenger_id = b.passenger_id
GROUP BY p.passenger_id;

```

12. Identify flights with the maximum number of bookings.

```

SELECT f.*, COUNT(b.booking_id) AS booking_count
FROM flight f
JOIN booking b ON f.flight_id = b.flight_id
GROUP BY f.flight_id

```

```

HAVING COUNT(b.booking_id) = (SELECT MAX(booking_count)
                               FROM (SELECT COUNT(booking_id) AS booking_count
                                       FROM booking
                                       GROUP BY flight_id) AS counts);

```

13. Retrieve the average price per airline for flights with more than 5 bookings.

```

SELECT a.name, AVG(b.price) AS average_price
FROM airline a
JOIN flight f ON a.airline_id = f.airline_id
JOIN booking b ON f.flight_id = b.flight_id
GROUP BY a.name
HAVING COUNT(b.booking_id) > 5;

```

14. Find passengers who have made bookings for flights departing from airports whose name do not start with a vowel. (use regexp_like)

```

SELECT DISTINCT p.*
FROM passenger p
JOIN booking b ON p.passenger_id = b.passenger_id
JOIN flight f ON b.flight_id = f.flight_id
JOIN airport a ON f.from = a.airport_id
WHERE NOT REGEXP_LIKE(a.name, '^[AEIOU]');

```

15. Display the flight details for flights that do not have any bookings.

```

SELECT f.*
FROM flight f
LEFT JOIN booking b ON f.flight_id = b.flight_id
WHERE b.booking_id IS NULL;

```

16. Categorize flights as 'Popular' if they have more than 10 bookings; otherwise, label them as 'Less Popular'.

```

SELECT f.*,
       CASE
           WHEN COUNT(b.booking_id) > 10 THEN 'Popular'
           ELSE 'Less Popular'
       END AS popularity
FROM flight f
LEFT JOIN booking b ON f.flight_id = b.flight_id
GROUP BY f.flight_id;

```

17. List the airlines and the total number of flights they operate, along with the overall average price for all flights.

```

SELECT a.name, COUNT(f.flight_id) AS total_flights, AVG(b.price) AS average_price
FROM airline a
LEFT JOIN flight f ON a.airline_id = f.airline_id
LEFT JOIN booking b ON f.flight_id = b.flight_id
GROUP BY a.name;

```

18. Using a CTE, retrieve the flights departing from 'Heathrow' airport.

```

WITH HeathrowFlights AS (
    SELECT f.*
    FROM flight f

```

```

        JOIN airport a ON f.from = a.airport_id
        WHERE a.name = 'Heathrow'
    )
    SELECT * FROM HeathrowFlights;

```

19. Find passengers who have booked at least one flight with a price higher than 500.

```

SELECT DISTINCT p.*
FROM passenger p
JOIN booking b ON p.passenger_id = b.passenger_id
WHERE b.price > 500;

```

20. Retrieve the airports with names that do not start with 'Lond'.

```

SELECT *
FROM airport
WHERE name NOT LIKE 'Lond%';

```