

# Flight Pricing Case Study - SQL Analysis

## Introduction

### Problem Statement:

To analyze flight ticket pricing behavior using SQL queries on a dataset of 300,000+ records. The objective is to understand how prices vary based on airline, class, city, duration, and other parameters to support business decision-making in dynamic pricing models.

### Solution Overview:

This case study includes 15 real-time SQL queries utilizing CTEs, window functions, joins, regex, and aggregations.

## Q1. Find the top 5 most expensive flights (Economy class)

```
SELECT airline, source_city, destination_city, price
FROM flight_pricing
WHERE class = 'Economy'
ORDER BY price DESC
LIMIT 5;
```

## Q2. Get average flight price per airline

```
SELECT airline, ROUND(AVG(price), 2) AS avg_price
FROM flight_pricing
GROUP BY airline;
```

## Q3. Which airline has the highest average ticket price for Business class?

```
SELECT airline, AVG(price) AS avg_business_price
FROM flight_pricing
WHERE class = 'Business'
GROUP BY airline
ORDER BY avg_business_price DESC
LIMIT 1;
```

## Q4. Use a CTE to find the difference between average price of Business and Economy class per airline

```
WITH class_avg AS (
```

## Flight Pricing Case Study - SQL Analysis

```
SELECT airline, class, AVG(price) AS avg_price
FROM flight_pricing
GROUP BY airline, class
)
SELECT a.airline,
       MAX(CASE WHEN a.class = 'Business' THEN a.avg_price END) -
       MAX(CASE WHEN a.class = 'Economy' THEN a.avg_price END) AS price_diff
FROM class_avg a
GROUP BY a.airline;
```

### Q5. List flights with duration above the average duration using a window function

```
SELECT *,
       AVG(duration) OVER () AS avg_duration
FROM flight_pricing
WHERE duration > (SELECT AVG(duration) FROM flight_pricing);
```

### Q6. Use REGEXP to find flights that contain numeric characters in their airline name

```
SELECT DISTINCT airline
FROM flight_pricing
WHERE airline REGEXP '[0-9]';
```

### Q7. Create a query to rank flights by price within each source city

```
SELECT flight, source_city, destination_city, price,
       RANK() OVER (PARTITION BY source_city ORDER BY price DESC) AS price_rank
FROM flight_pricing;
```

### Q8. Find the number of flights available for each combination of source and destination cities

```
SELECT source_city, destination_city, COUNT(*) AS total_flights
FROM flight_pricing
GROUP BY source_city, destination_city;
```

### Q9. Find the cheapest flight (per airline) with less than 2 stops

```
SELECT airline, MIN(price) AS cheapest_flight
FROM flight_pricing
```

## Flight Pricing Case Study - SQL Analysis

```
WHERE stops != 'two_or_more'

GROUP BY airline;
```

### Q10. Categorize flights into short (<2 hrs), medium (2-3 hrs), and long (>3 hrs)

```
SELECT flight, duration,
       CASE
         WHEN duration < 2 THEN 'Short'
         WHEN duration BETWEEN 2 AND 3 THEN 'Medium'
         ELSE 'Long'
       END AS duration_category
FROM flight_pricing;
```

### Q11. Use a window function to get the average price for each source-destination pair

```
SELECT *,
       AVG(price) OVER (PARTITION BY source_city, destination_city) AS avg_route_price
FROM flight_pricing;
```

### Q12. Identify flights that have the same price and duration as any other flight (self-join)

```
SELECT a.flight AS flight1, b.flight AS flight2, a.price, a.duration
FROM flight_pricing a
JOIN flight_pricing b
  ON a.price = b.price AND a.duration = b.duration AND a.flight != b.flight;
```

### Q13. Create a summary showing average price grouped by class and number of days left

```
SELECT class, days_left, ROUND(AVG(price), 2) AS avg_price
FROM flight_pricing
GROUP BY class, days_left
ORDER BY days_left;
```

### Q14. Find flights that depart in the morning and arrive at night

```
SELECT flight, departure_time, arrival_time
FROM flight_pricing
WHERE departure_time = 'Morning' AND arrival_time = 'Night';
```

## Flight Pricing Case Study - SQL Analysis

### Q15. Use CTE to get top 3 shortest flights per airline

```
WITH RankedFlights AS (  
    SELECT airline, flight, duration,  
           RANK() OVER (PARTITION BY airline ORDER BY duration) AS rnk  
    FROM flight_pricing  
)  
  
SELECT *  
FROM RankedFlights  
WHERE rnk <= 3;
```