# Airline Revenue & Settlement Analytics System

# Aim of the Project

To design and develop an Airline Revenue & Settlement Analytics System using SQL to analyze airline booking transactions, revenue generation, cancellations, and refund impact, and to generate meaningful business insights through structured data analysis.

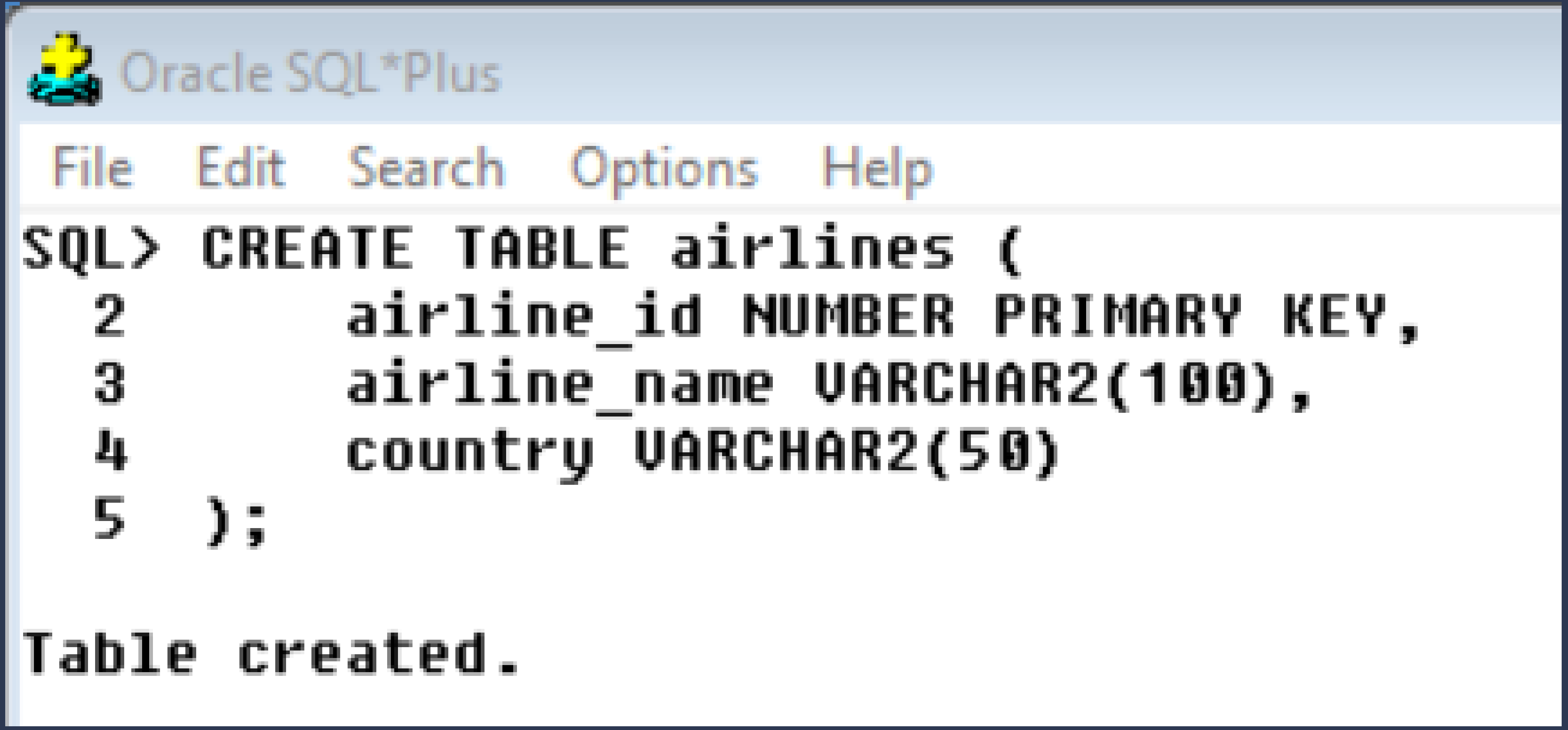
# OBJECTIVES

- Design a normalized relational database for airline operations
- Store and manage booking, payment, and refund transactions
- Perform revenue and settlement analysis using SQL
- Analyze route performance and cancellation rates
- Implement window functions and advanced SQL queries
- Generate business insights for financial decision-making
- Visualize analytics results using dashboards

# TABLES INCLUDED

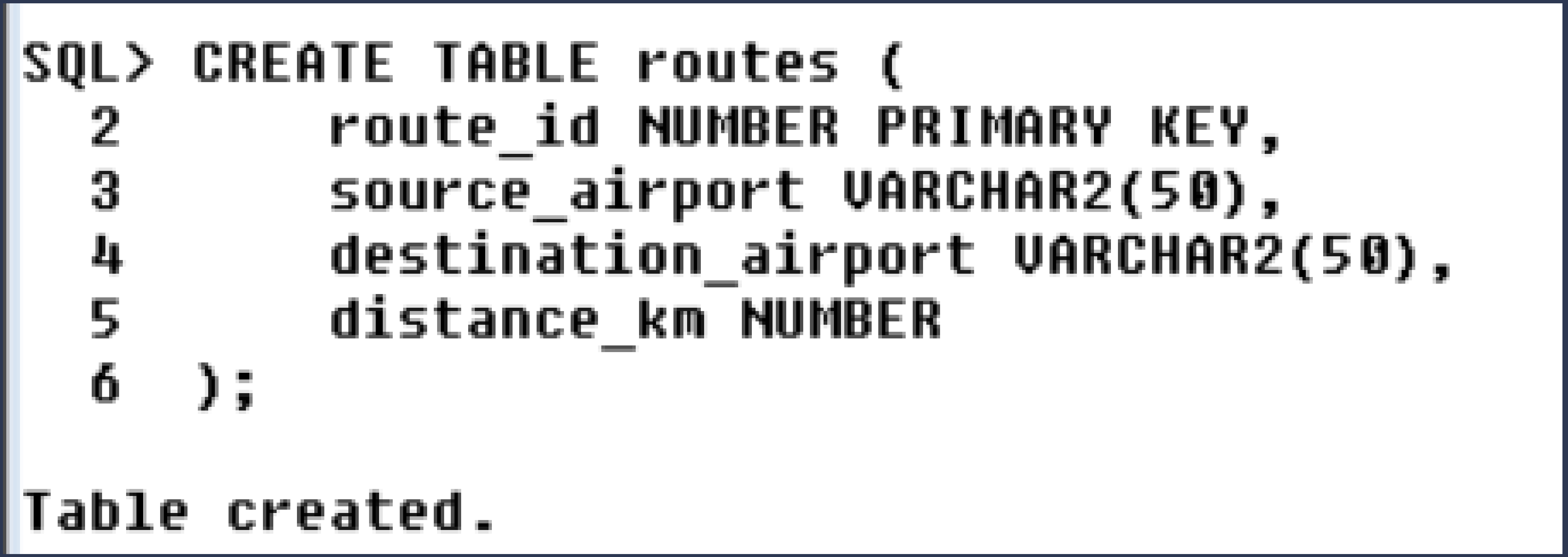
1. Airlines
2. Routes
3. Flights
4. Passengers
5. Bookings
6. Payments
7. Refunds

# Airlines



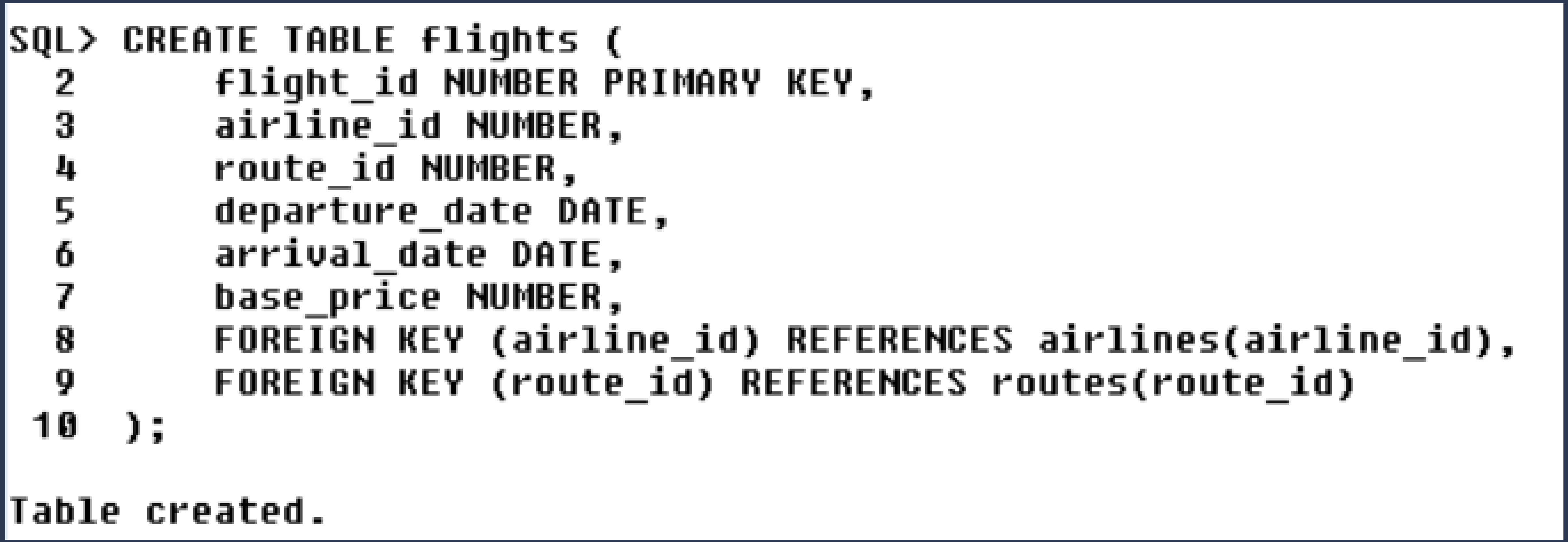
```
SQL> CREATE TABLE airlines (
  2      airline_id NUMBER PRIMARY KEY,
  3      airline_name VARCHAR2(100),
  4      country VARCHAR2(50)
  5  );

Table created.
```

# Routes



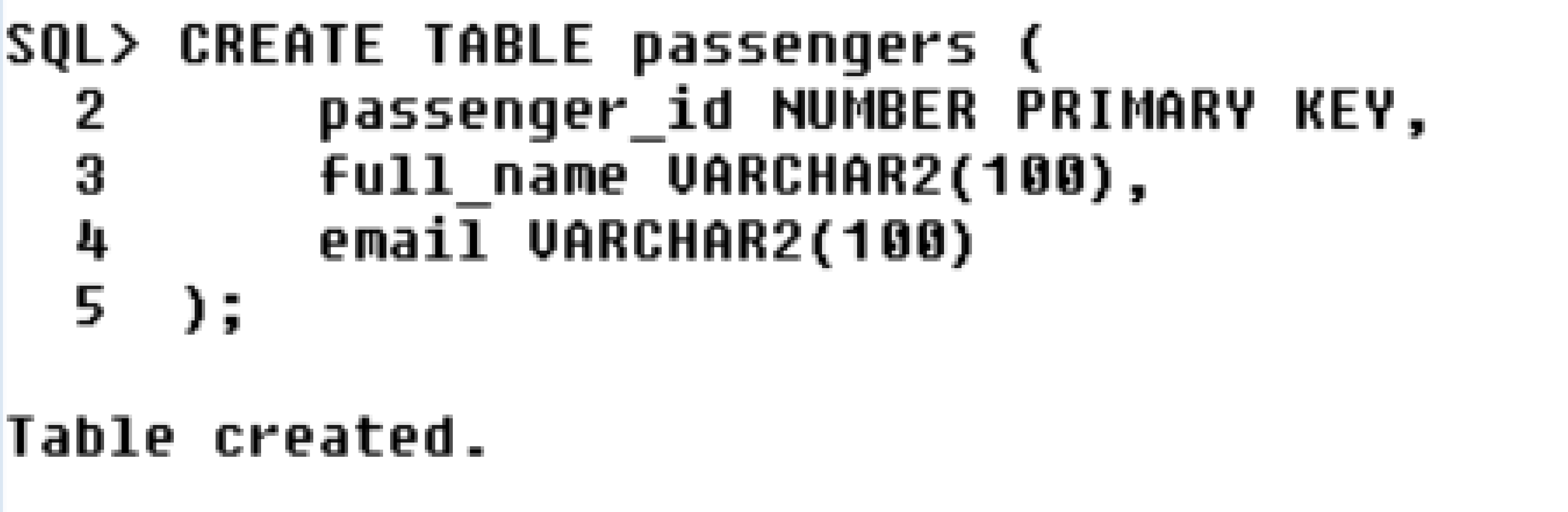
```
SQL> CREATE TABLE routes (
  2      route_id NUMBER PRIMARY KEY,
  3      source_airport VARCHAR2(50),
  4      destination_airport VARCHAR2(50),
  5      distance_km NUMBER
  6  );

Table created.
```

# Flights



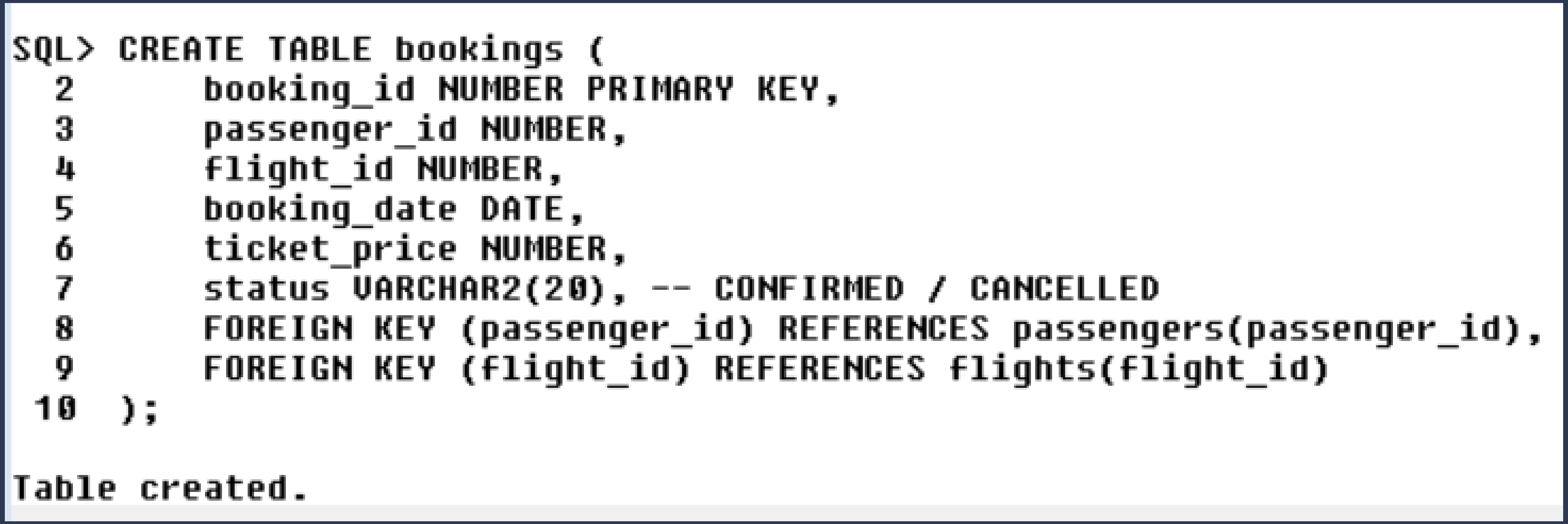
```
SQL> CREATE TABLE flights (
  2       flight_id NUMBER PRIMARY KEY,
  3       airline_id NUMBER,
  4       route_id NUMBER,
  5       departure_date DATE,
  6       arrival_date DATE,
  7       base_price NUMBER,
  8       FOREIGN KEY (airline_id) REFERENCES airlines(airline_id),
  9       FOREIGN KEY (route_id) REFERENCES routes(route_id)
 10  );

Table created.
```

# Passengers



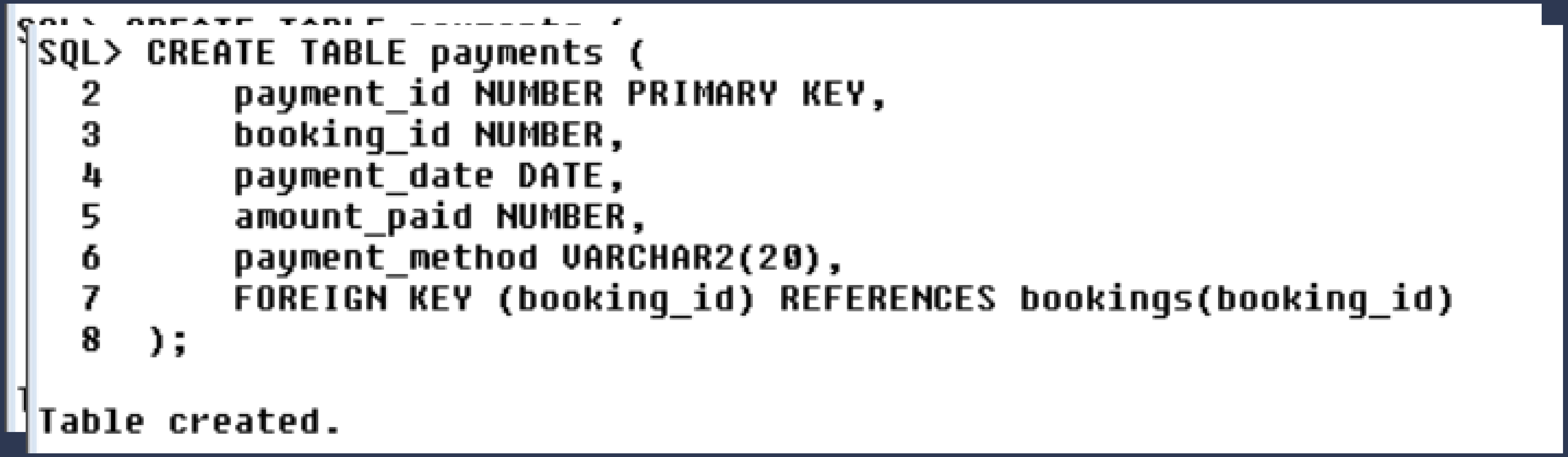
```
SQL> CREATE TABLE passengers (
  2      passenger_id NUMBER PRIMARY KEY,
  3      full_name VARCHAR2(100),
  4      email VARCHAR2(100)
  5  );

Table created.
```

# Bookings



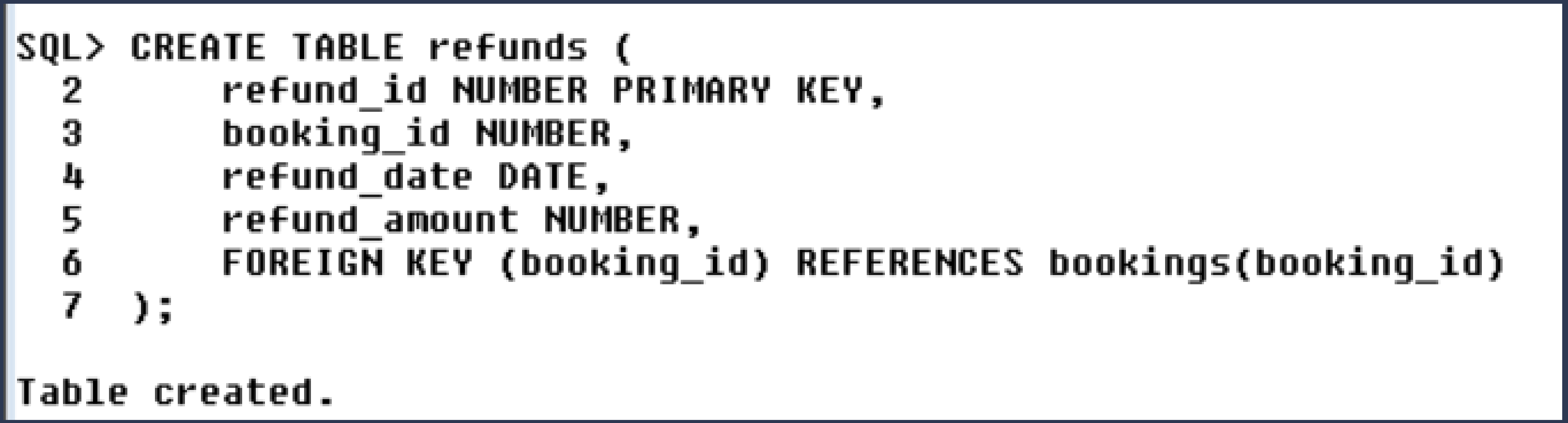
```
SQL> CREATE TABLE bookings (
  2      booking_id NUMBER PRIMARY KEY,
  3      passenger_id NUMBER,
  4      flight_id NUMBER,
  5      booking_date DATE,
  6      ticket_price NUMBER,
  7      status VARCHAR2(20), -- CONFIRMED / CANCELLED
  8      FOREIGN KEY (passenger_id) REFERENCES passengers(passenger_id),
  9      FOREIGN KEY (flight_id) REFERENCES flights(flight_id)
 10  );

Table created.
```

# Payments



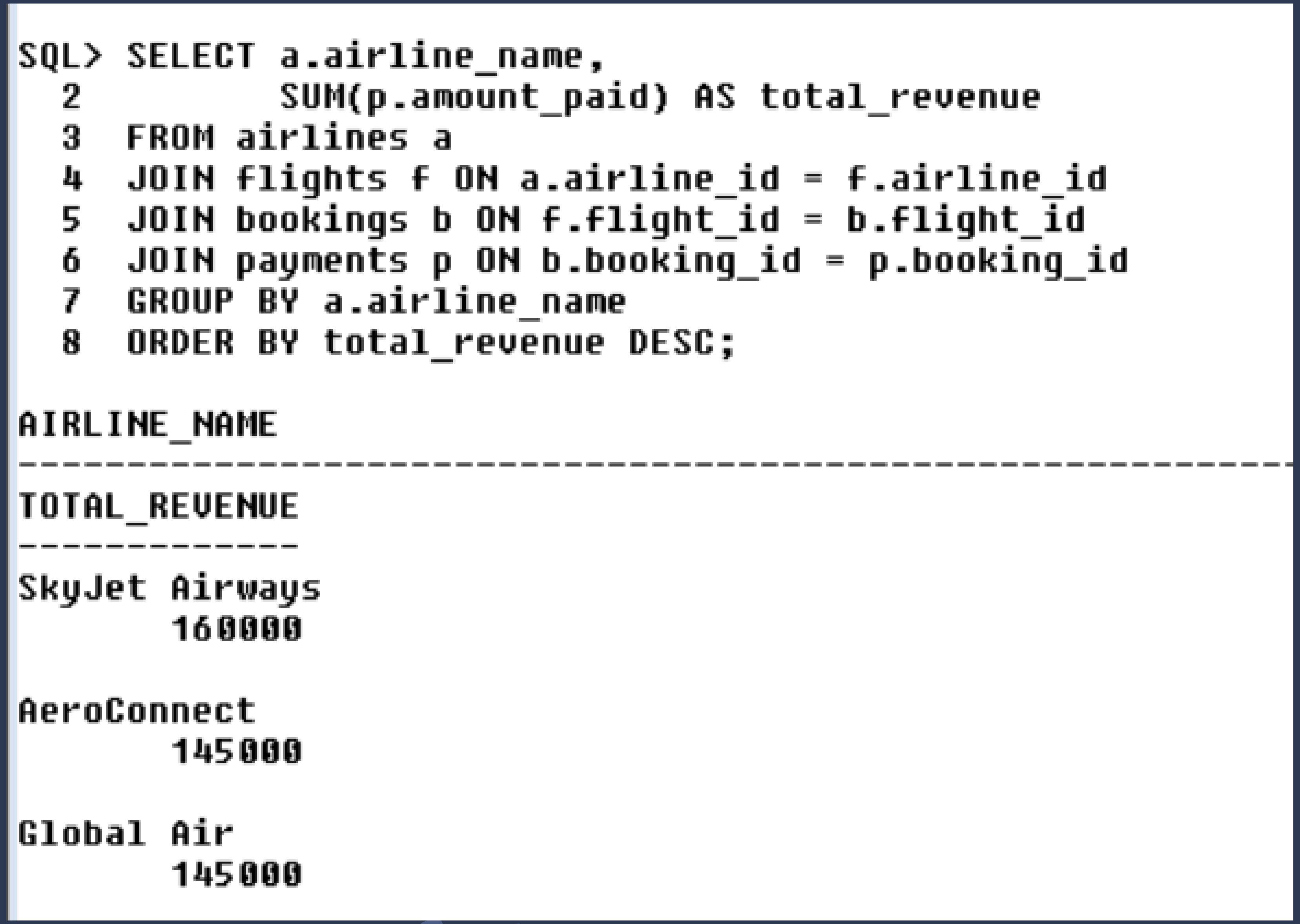
```
SQL> CREATE TABLE payments (
  2      payment_id NUMBER PRIMARY KEY,
  3      booking_id NUMBER,
  4      payment_date DATE,
  5      amount_paid NUMBER,
  6      payment_method VARCHAR2(20),
  7      FOREIGN KEY (booking_id) REFERENCES bookings(booking_id)
  8  );

Table created.
```

# Refunds



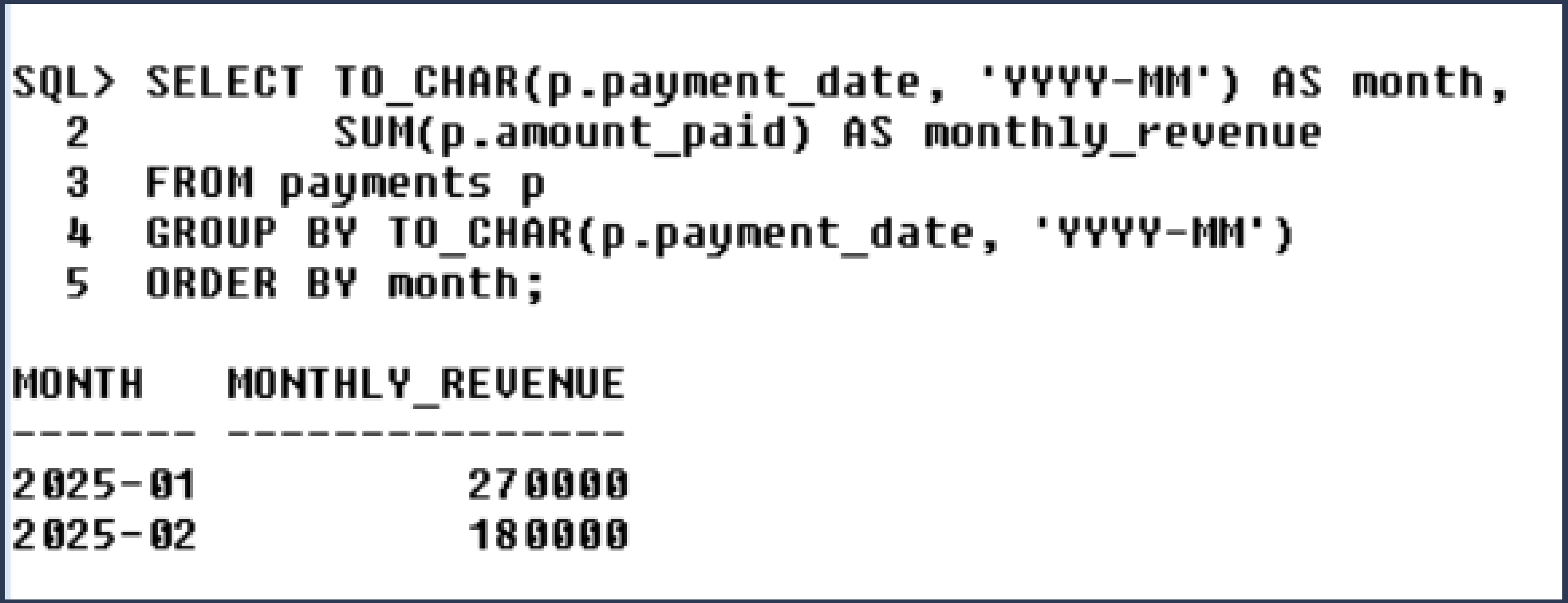
```
SQL> CREATE TABLE refunds (
         refund_id NUMBER PRIMARY KEY,
         booking_id NUMBER,
         refund_date DATE,
         refund_amount NUMBER,
         FOREIGN KEY (booking_id) REFERENCES bookings(booking_id)
     );

Table created.
```

# 1. Total Revenue Per Airline



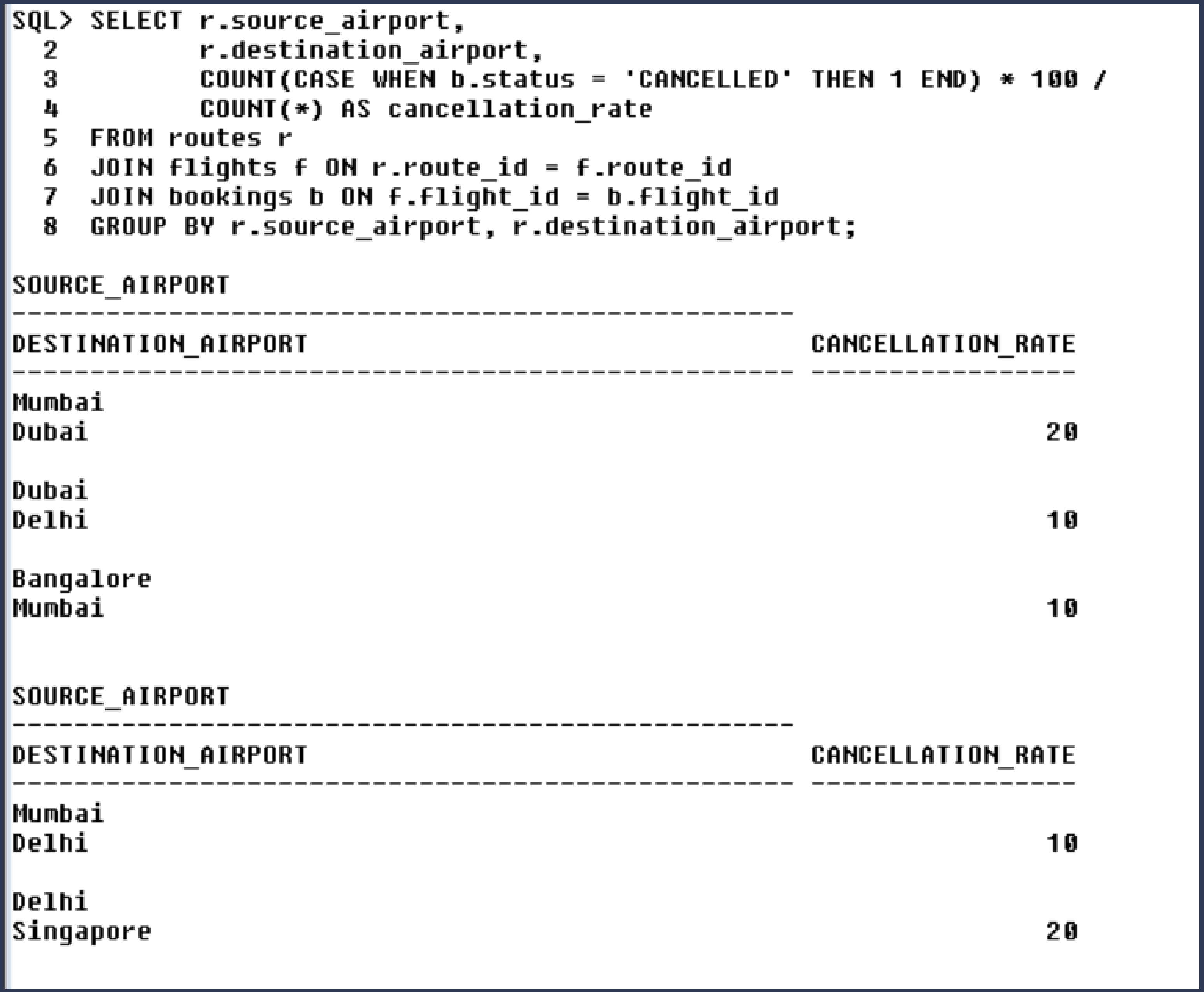
```
SQL> SELECT a.airline_name,
            SUM(p.amount_paid) AS total_revenue
     FROM airlines a
     JOIN flights f ON a.airline_id = f.airline_id
     JOIN bookings b ON f.flight_id = b.flight_id
     JOIN payments p ON b.booking_id = p.booking_id
     GROUP BY a.airline_name
     ORDER BY total_revenue DESC;

AIRLINE_NAME
------------------------------------------------------------
TOTAL_REVENUE
-------------
SkyJet Airways
       160000

AeroConnect
       145000

Global Air
       145000
```

# 2. Monthly Revenue Trend



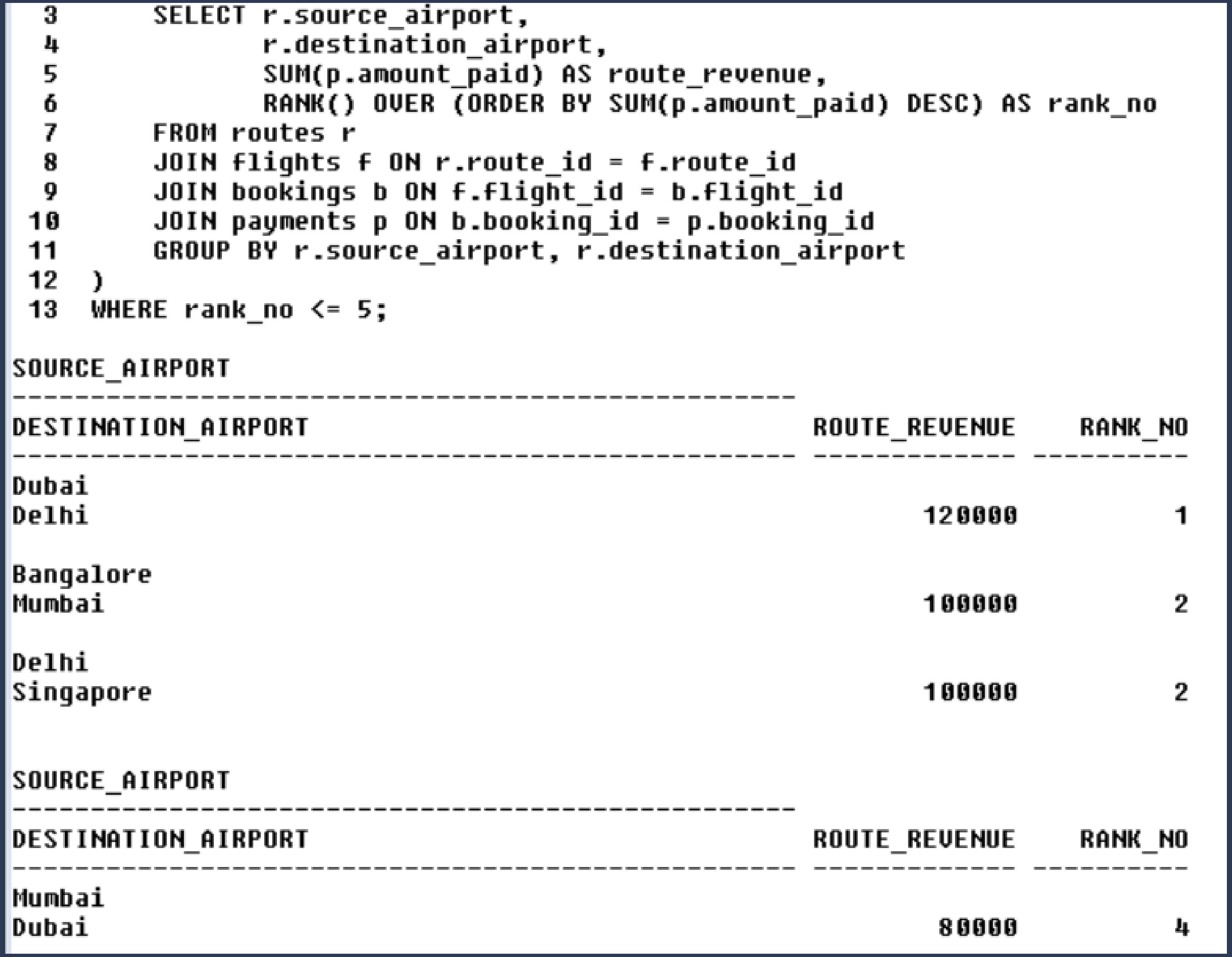
```
SQL> SELECT TO_CHAR(p.payment_date, 'YYYY-MM') AS month,
            SUM(p.amount_paid) AS monthly_revenue
     FROM payments p
     GROUP BY TO_CHAR(p.payment_date, 'YYYY-MM')
     ORDER BY month;

MONTH   MONTHLY_REVENUE
------- ---------------
2025-01          270000
2025-02          180000
```

# 3. Cancellation Rate Per Route



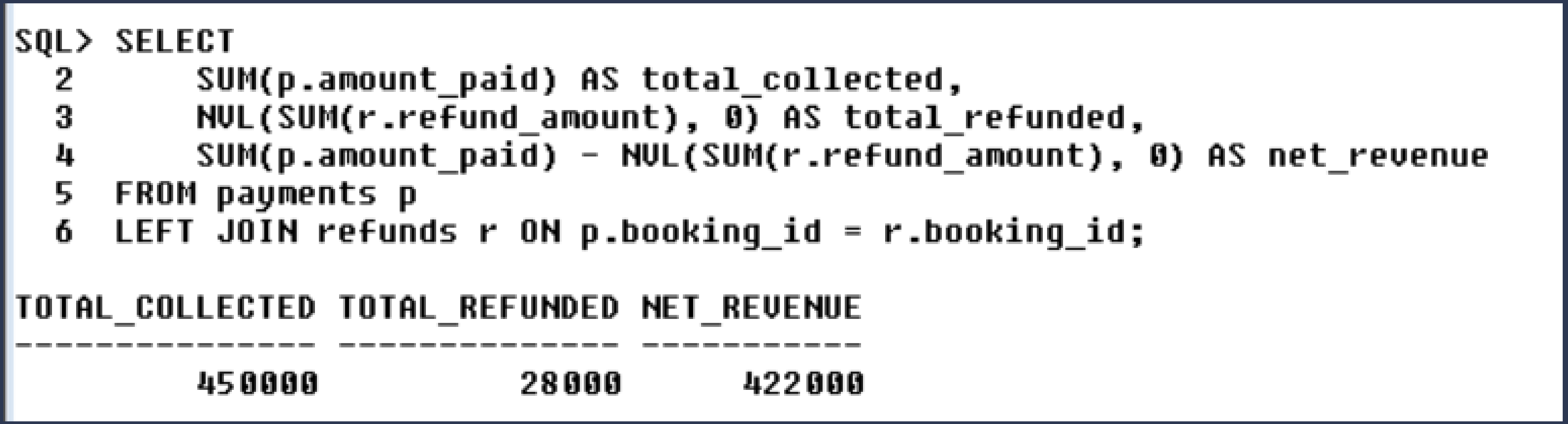
```
SQL> SELECT r.source_airport,
  2         r.destination_airport,
  3         COUNT(CASE WHEN b.status = 'CANCELLED' THEN 1 END) * 100 /
  4         COUNT(*) AS cancellation_rate
  5  FROM routes r
  6  JOIN flights f ON r.route_id = f.route_id
  7  JOIN bookings b ON f.flight_id = b.flight_id
  8  GROUP BY r.source_airport, r.destination_airport;

SOURCE_AIRPORT
--------------------------------------------------
DESTINATION_AIRPORT                                CANCELLATION_RATE
-------------------------------------------------- -----------------
Mumbai
Dubai                                                             20

Dubai
Delhi                                                             10

Bangalore
Mumbai                                                            10


SOURCE_AIRPORT
--------------------------------------------------
DESTINATION_AIRPORT                                CANCELLATION_RATE
-------------------------------------------------- -----------------
Mumbai
Delhi                                                             10

Delhi
Singapore                                                         20
```

# 4. Top 5 Revenue Routes



```
    SELECT r.source_airport,
           r.destination_airport,
           SUM(p.amount_paid) AS route_revenue,
           RANK() OVER (ORDER BY SUM(p.amount_paid) DESC) AS rank_no
    FROM routes r
    JOIN flights f ON r.route_id = f.route_id
    JOIN bookings b ON f.flight_id = b.flight_id
    JOIN payments p ON b.booking_id = p.booking_id
    GROUP BY r.source_airport, r.destination_airport
)
WHERE rank_no <= 5;

SOURCE_AIRPORT
--------------------------------------------------
DESTINATION_AIRPORT                                ROUTE_REVENUE    RANK_NO
-------------------------------------------------- ------------- ----------
Dubai
Delhi                                                     120000          1

Bangalore
Mumbai                                                    100000          2

Delhi
Singapore                                                 100000          2


SOURCE_AIRPORT
--------------------------------------------------
DESTINATION_AIRPORT                                ROUTE_REVENUE    RANK_NO
-------------------------------------------------- ------------- ----------
Mumbai
Dubai                                                      80000          4
```

# 5. Refund Impact Analysis



```
SQL> SELECT
         SUM(p.amount_paid) AS total_collected,
         NVL(SUM(r.refund_amount), 0) AS total_refunded,
         SUM(p.amount_paid) - NVL(SUM(r.refund_amount), 0) AS net_revenue
     FROM payments p
     LEFT JOIN refunds r ON p.booking_id = r.booking_id;

TOTAL_COLLECTED TOTAL_REFUNDED NET_REVENUE
--------------- -------------- -----------
         450000          28000      422000
```

# CONCLUSION

The Airline Revenue & Settlement Analytics System successfully demonstrates the use of SQL for analyzing large transactional datasets in the aviation domain. The system enables revenue tracking, route performance analysis, cancellation monitoring, and refund impact evaluation.

Through the implementation of advanced SQL queries including joins, aggregations, CTEs, and window functions, the project simulates real-world airline financial reporting and settlement processes.

This project reflects practical industry alignment with airline technology companies and showcases strong data analysis, database design, and business intelligence skills.

# Thank You