SQL Capstone Project - AirlineDB (PostgreSQL)

Candidate: Shantu Sharma

Database: AirlineDB

SQL Engine: PostgreSQL

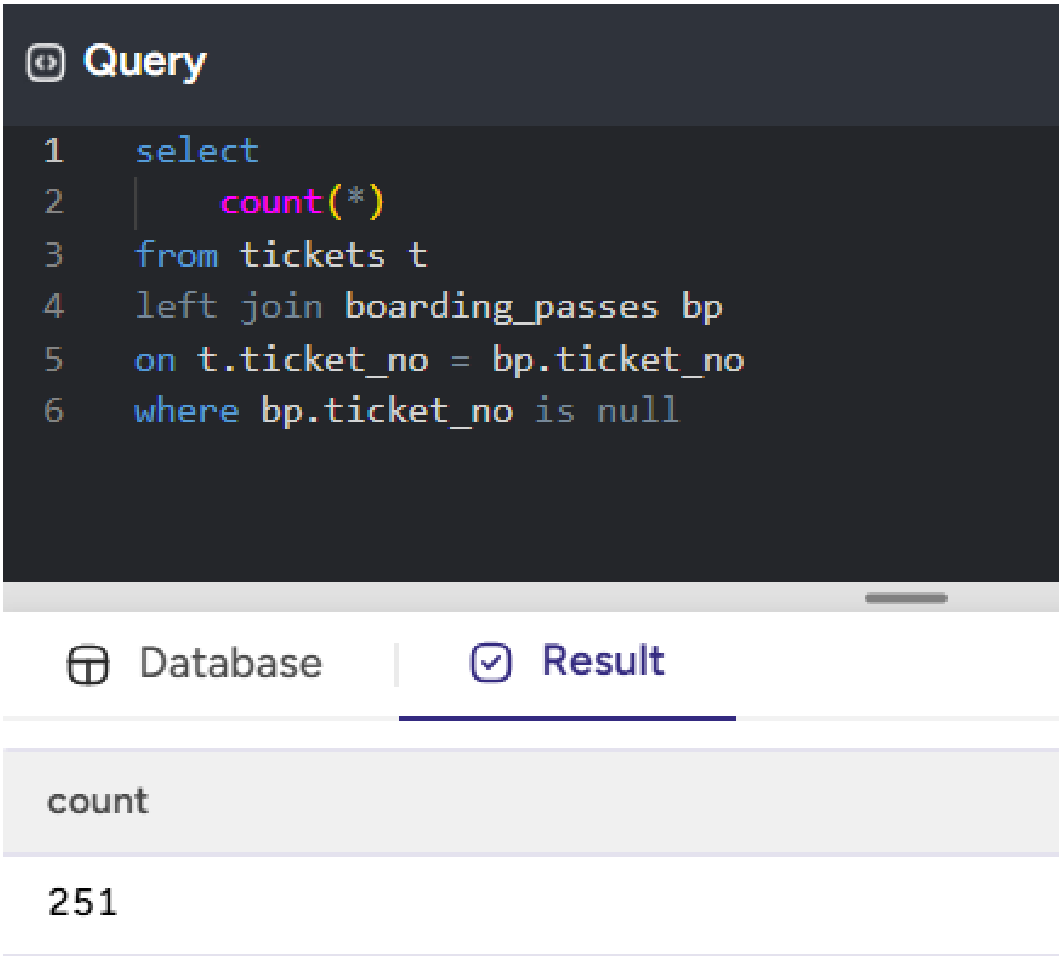
Completed on: October 2015

Question 1: How many tickets are there without a boarding pass?

SQL Solution:

```
SELECT COUNT(*)
FROM tickets t
LEFT JOIN boarding_passes bp
  ON t.ticket_no = bp.ticket_no
WHERE bp.ticket_no IS NULL;
```

Query Output:



```
select
    count(*)
from tickets t
left join boarding_passes bp
on t.ticket_no = bp.ticket_no
where bp.ticket_no is null
```

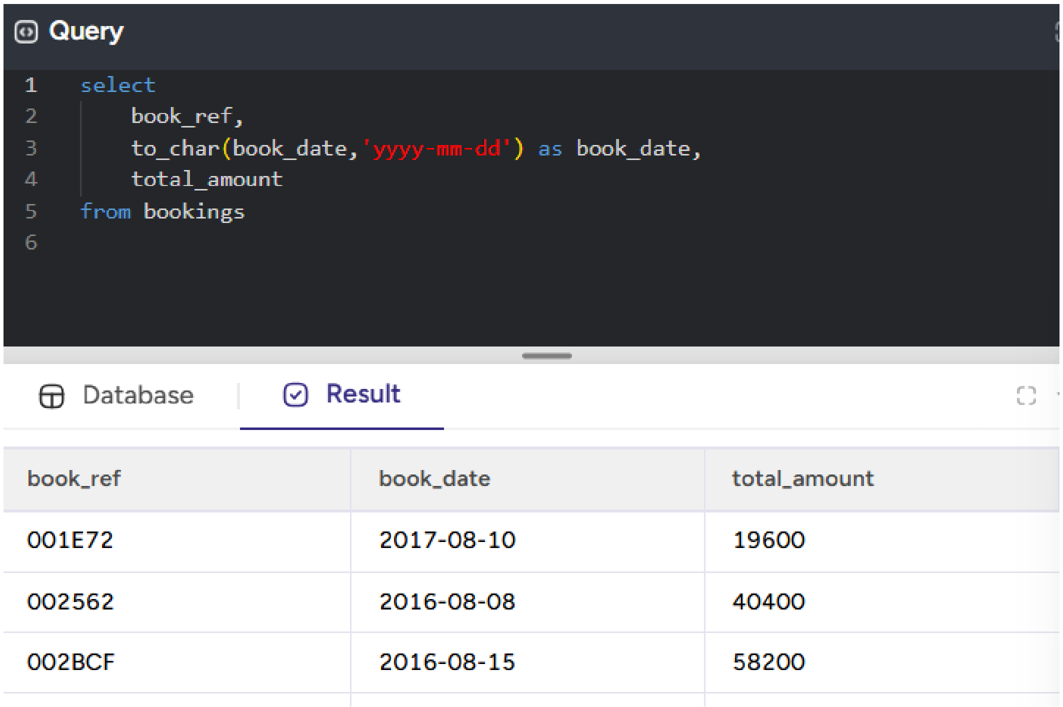
count
251

Question 2: Represent the book_date column in YYYY-MM-DD format using bookings table.

SQL Solution:

```
SELECT
   book_ref,
   TO_CHAR(book_date, 'YYYY-MM-DD') AS book_date,
   total_amount
FROM bookings;
```

Query Output:



```
select
    book_ref,
    to_char(book_date,'yyyy-mm-dd') as book_date,
    total_amount
from bookings
```

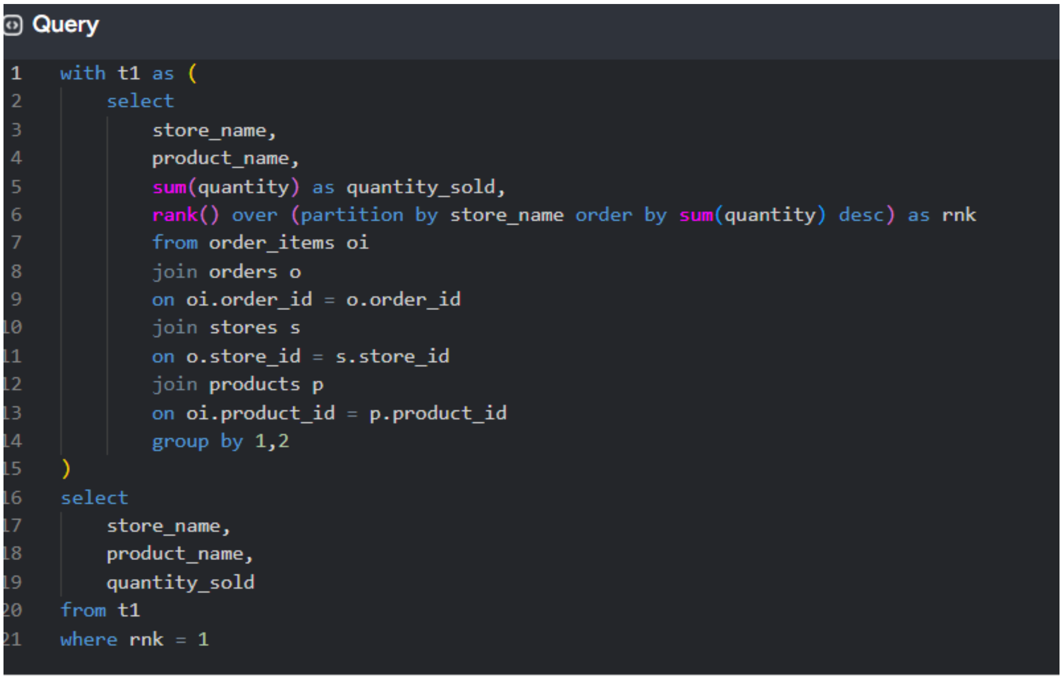
book_ref	book_date	total_amount
001E72	2017-08-10	19600
002562	2016-08-08	40400
002BCF	2016-08-15	58200

Question 3: Identify the most popular product in each store based on quantity sold.

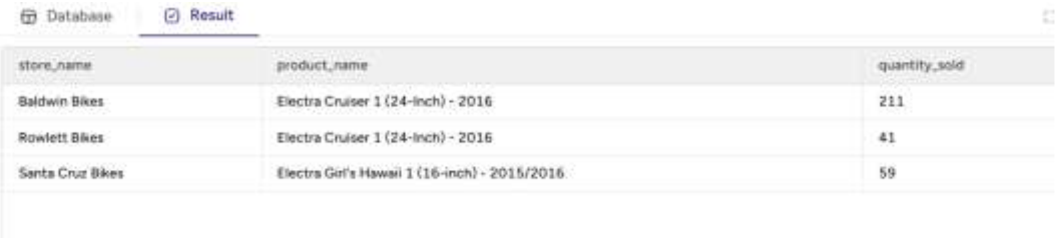
SQL Solution:

```
WITH t1 AS (
  SELECT
    s.store_name,
    p.product_name,
    SUM(oi.quantity) AS quantity_sold,
    RANK() OVER (
      PARTITION BY s.store_name
      ORDER BY SUM(oi.quantity) DESC
    ) AS rnk
  FROM order_items oi
  JOIN orders o ON oi.order_id = o.order_id
  JOIN stores s ON o.store_id = s.store_id
  JOIN products p ON oi.product_id = p.product_id
  GROUP BY 1,2
)
SELECT store_name, product_name, quantity_sold
FROM t1
WHERE rnk = 1;
```

Query Output:



```
with t1 as (
    select
        store_name,
        product_name,
        sum(quantity) as quantity_sold,
        rank() over (partition by store_name order by sum(quantity) desc) as rnk
        from order_items oi
        join orders o
        on oi.order_id = o.order_id
        join stores s
        on o.store_id = s.store_id
        join products p
        on oi.product_id = p.product_id
        group by 1,2
)
select
    store_name,
    product_name,
    quantity_sold
from t1
where rnk = 1
```



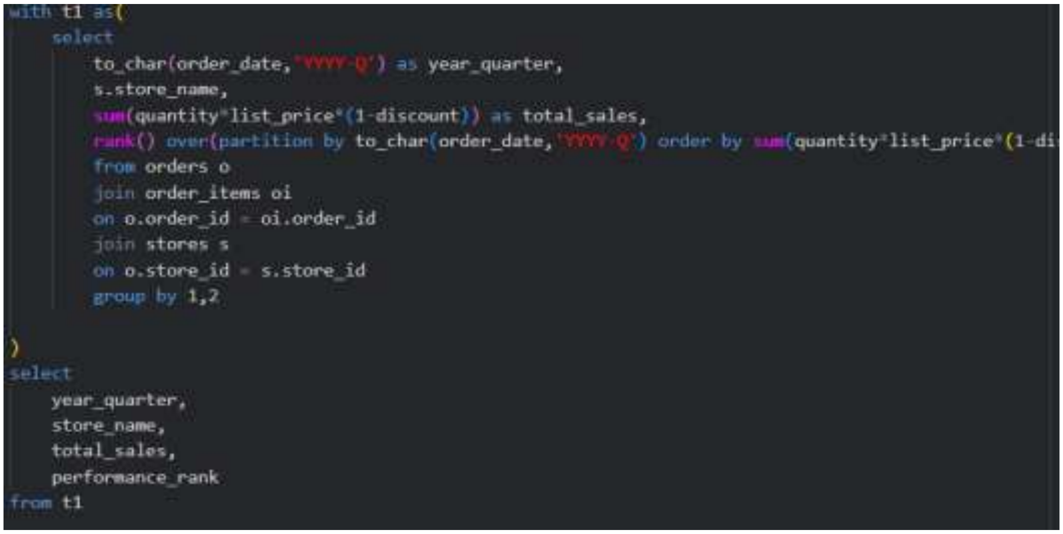
store_name	product_name	quantity_sold
Baldwin Bikes	Electra Cruiser 1 (24-Inch) - 2016	211
Rowlett Bikes	Electra Cruiser 1 (24-Inch) - 2016	41
Santa Cruz Bikes	Electra Girl's Hawaii 1 (16-inch) - 2015/2016	59

Question 4: Compare quarterly sales performance across stores and rank them.

SQL Solution:

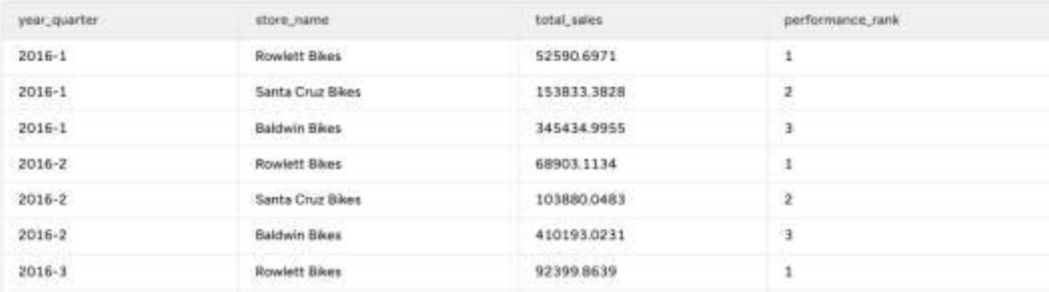
```
WITH t1 AS (
   SELECT
      TO_CHAR(order_date, 'YYYY-Q') AS year_quarter,
        s.store_name,
        SUM(quantity * list_price * (1 - discount)) AS total_sales,
        RANK() OVER (
            PARTITION BY TO_CHAR(order_date, 'YYYY-Q')
            ORDER BY SUM(quantity * list_price * (1 - discount)) DESC
        ) AS performance_rank
    FROM orders o
    JOIN order_items oi ON o.order_id = oi.order_id
    JOIN stores s ON o.store_id = s.store_id
    GROUP BY 1,2
)
SELECT year_quarter, store_name, total_sales, performance_rank
FROM t1;
```

Query Output:



```
with t1 as(
    select
        to_char(order_date,'YYYY-Q') as year_quarter,
        s.store_name,
        sum(quantity*list_price*(1-discount)) as total_sales,
        rank() over(partition by to_char(order_date,'YYYY-Q') order by sum(quantity*list_price*(1-di
        from orders o
        join order_items oi
        on o.order_id = oi.order_id
        join stores s
        on o.store_id = s.store_id
        group by 1,2

)
select
    year_quarter,
    store_name,
    total_sales,
    performance_rank
from t1
```



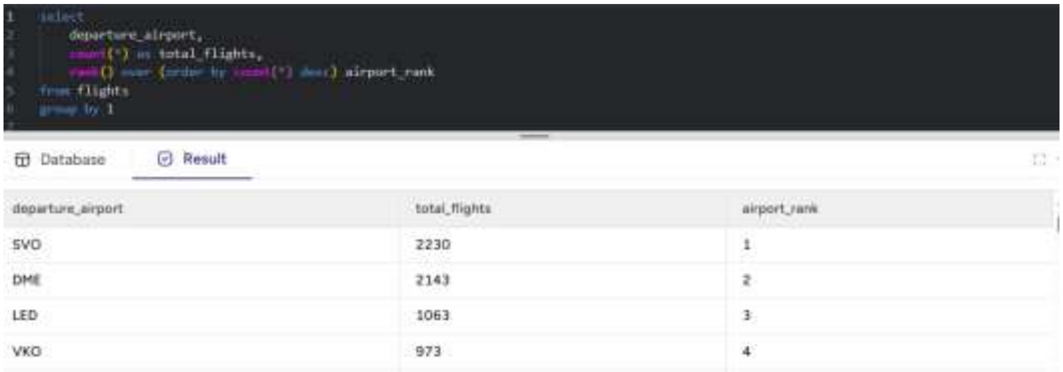
year_quarter	store_name	total_sales	performance_rank
2016-1	Rowlett Bikes	52590.6971	1
2016-1	Santa Cruz Bikes	153833.3828	2
2016-1	Baldwin Bikes	345434.9955	3
2016-2	Rowlett Bikes	68903.1134	1
2016-2	Santa Cruz Bikes	103880.0483	2
2016-2	Baldwin Bikes	410193.0231	3
2016-3	Rowlett Bikes	92399.8639	1

Question 5: Rank airports based on the number of flights departing from them.

SQL Solution:

```
SELECT
   departure_airport,
   COUNT(*) AS total_flights,
   RANK() OVER (ORDER BY COUNT(*) DESC) AS airport_rank
FROM flights
GROUP BY departure_airport;
```

Query Output:



```
select
    departure_airport,
    count(*) as total_flights,
    rank() over (order by count(*) desc) airport_rank
from flights
group by 1
```

departure_airport	total_flights	airport_rank
SVO	2230	1
DME	2143	2
LED	1063	3
VKO	973	4

Final Conclusion

This SQL capstone demonstrates the ability to extract operational insights from AirlineDB using PostgreSQL.

The queries validate data integrity (tickets vs boarding passes), standardize reporting formats, identify store-level product demand, compare quarterly sales performance, and analyze airport traffic concentration.

Together, these analyses reflect practical SQL application in aggregation, ranking, and relational data analysis to support operational and strategic decision-making.