SQL Capstone

Airline Database & Bike Sharing Database

Problem Statement 1:
How many tickets are there without boarding passes?
Expected Output: Just one number is required

Query:

```
select
        Count(T.ticket_no) as tickets_without_boarding_passes
from tickets as T
left join boarding_passes as B
on T.ticket_no = B.ticket_no
where B.ticket_no is null
```

Problem Statement 2:
Represent the "book_date" column in "yyyy-mm-dd" format using Bookings table.

Expected Output: book_ref, book_date(in "yyyy-mm-dd" format) & total amount. Output Columns should be in exact same sequence as given in Expected Output.

Query:

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

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

Expected Output: store_name, product_name, quantity_sold

Query:

```
with ranked_products as (
        select s.store_name, p.product_name, sum(oi.quantity) as total_quantity_sold,
        rank()over(partition by s.store_name order by sum(oi.quantity) desc) as ranking
        from stores as s
        join orders as o on s.store_id = o.store_id
        join order_items as oi on o.order_id = oi.order_id
        join products as p on oi.product_id = p.product_id
```

```
    group by s.store_name, p.product_name)
select
    store_name, product_name, total_quantity_sold
from ranked_products
where ranking =1
```

Problem Statement 4:
The Management team is interested in a detailed comparison of sales performance across all stores on a quarterly basis. Your objective is to calculate the total sales for each store for each quarter and then rank the stores based on these sales figures. This analysis will help identify top-performing stores and those that may require strategic adjustments.

Expected Output:

- year_quarter: The year Quarter format should be “YYYY-Q”
- store_name
- total_sales: Quantity*list_price*(1-discount)
- performance_rank

Query:

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

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

Expected Output: departure_airport, total_flights and airport_rank in the exact same sequence

Query:

```
select
    departure_airport, count(flight_id) as total_flights,
    rank()over(order by count(flight_id) desc) as airport_rank
```

```
from flights
group by departure_airport
```