*In this project, I worked on designing, developing, and analyzing a structured SQL database schema to support a real-world business scenario. The database was designed to efficiently manage data related to [insert the type of data or industry here], ensuring data integrity, optimization, and scalability.*

**Different Scienario questions**

1. ***Represent the “book\_date” column in “yyyy-mmm-dd” format using Bookings table***

*Expected output: book\_ref, book\_date (in “yyyy-mmm-dd” format) , total amount*

**Answer:**

**SELECT**

**book\_ref,**

**TO\_CHAR (book\_date,'yyyy-mon-dd') AS book\_date ,**

**total\_amount**

**FROM bookings**

1. **Get the following columns in the exact same sequence.**

Expected columns in the output: ticket\_no, boarding\_no, seat\_number, passenger\_id, passenger\_name.

**Answer:**

**SELECT**

**b.ticket\_no,**

**b.boarding\_no,**

**b.seat\_no,**

**t.passenger\_id,**

**t.passenger\_name**

**FROM boarding\_passes b**

**JOIN tickets t**

**ON b.ticket\_no = t.ticket\_no**

1. **Write a query to find the seat number which is least allocated among all the seats?**

**Answer:**

**WITH seat\_count AS**

**(SELECT**

**seat\_no,**

**COUNT(seat\_no) AS seat\_count**

**FROM boarding\_passes**

**GROUP BY 1**

**ORDER BY 2 ASC)**

**SELECT**

**seat\_no**

**FROM seat\_count**

**WHERE seat\_count BETWEEN 0 AND 1**

1. ***In the database, identify the month wise highest paying passenger name and passenger id.***

Expected output: Month\_name(“mmm-yy” format), passenger\_id, passenger\_name and total amount

**ANSWER:**

**WITH passenger\_spending AS**

**(SELECT**

**TO\_CHAR (book\_date,'mon-YY') AS month\_name,**

**t.passenger\_id,**

**t.passenger\_name,**

**SUM(total\_amount) AS total\_spending**

**FROM bookings b**

**JOIN tickets t**

**ON b.book\_ref = t.book\_ref**

**GROUP BY 1,2,3),**

**RANK\_TABLE AS**

**(SELECT**

**\*,**

**ROW\_NUMBER () OVER (PARTITION BY month\_name ORDER BY total\_spending DESC) AS row\_num**

**FROM passenger\_spending)**

**SELECT**

**month\_name,**

**passenger\_id,**

**passenger\_name,**

**total\_spending**

**FROM RANK\_TABLE**

**WHERE row\_num = 1**

1. ***In the database, identify the month wise least paying passenger name and passenger id?***

Expected output: Month\_name(“mmm-yy” format), passenger\_id, passenger\_name and total amount

**ANSWER:**

**WITH passenger\_spending AS**

**(SELECT**

**TO\_CHAR (book\_date,'mon-YY') AS month\_name,**

**t.passenger\_id,**

**t.passenger\_name,**

**SUM(total\_amount) AS total\_spending**

**FROM bookings b**

**JOIN tickets t**

**ON b.book\_ref = t.book\_ref**

**GROUP BY 1,2,3),**

**RANK\_TABLE AS**

**(SELECT**

**\*,**

**ROW\_NUMBER () OVER (PARTITION BY month\_name ORDER BY total\_spending ASC) AS row\_num**

**FROM passenger\_spending)**

**SELECT**

**month\_name,**

**passenger\_id,**

**passenger\_name,**

**total\_spending**

**FROM RANK\_TABLE**

**WHERE row\_num = 1**

1. **Identify the travel details of nonstop journeys or return journeys (having more than 1 flight).**

Expected Output: Passenger\_id, passenger\_name, ticket\_number and flight count.

**Answer:**

**SELECT**

**t.passenger\_id,**

**t.passenger\_name,**

**t.ticket\_no,**

**COUNT(f.flight\_id) AS flight\_count**

**FROM ticket\_flights tf**

**JOIN flights f**

**ON tf.flight\_id = f.flight\_id**

**JOIN tickets t**

**ON tf.ticket\_no = t.ticket\_no**

**GROUP BY 1,2,3**

**HAVING**

**COUNT(f.flight\_id)= 1 OR COUNT(f.flight\_id) > 1**

1. **How many tickets are there without boarding passes?**

Expected Output: just one number is required.

**Answer:**

**SELECT**

**COUNT (ticket\_no)**

**FROM boarding\_passes**

**WHERE boarding\_no is null;**

1. **Identify details of the longest flight (using flights table)?**

Expected Output: Flight number, departure airport, arrival airport, aircraft code and durations.

**Answer:**

**WITH duration\_table AS (SELECT**

**DISTINCT flight\_no,**

**departure\_airport,**

**arrival\_airport,**

**aircraft\_code,**

**(**

**(EXTRACT (HOUR FROM scheduled\_arrival) \* 3600 + EXTRACT (MINUTE FROM scheduled\_arrival) \* 60 + EXTRACT (SECOND FROM scheduled\_arrival)) -**

**(EXTRACT (HOUR FROM scheduled\_departure) \* 3600 + EXTRACT (MINUTE FROM scheduled\_departure) \* 60 + EXTRACT (SECOND FROM scheduled\_departure))**

**) / 60 AS duration\_minutes**

**FROM**

**flights**

**ORDER BY**

**duration\_minutes DESC),**

**rank\_table AS**

**(SELECT \*,**

**RANK () OVER (ORDER BY duration\_minutes DESC) AS rank**

**FROM duration\_table)**

**SELECT**

**DISTINCT flight\_no,**

**departure\_airport,**

**arrival\_airport,**

**aircraft\_code,**

**duration\_minutes**

**FROM rank\_table**

**WHERE rank = 1**

1. **Identify details of all the morning flights (morning means between 6AM to 11 AM, using flights table)?**

Expected output: flight\_id, flight\_number, scheduled\_departure, scheduled\_arrival and timings.

**Answer:**

**SELECT flight\_id, flight\_no, scheduled\_departure, scheduled\_arrival,**

**scheduled\_arrival - scheduled\_departure AS timings**

**FROM flights**

**WHERE EXTRACT (HOUR FROM scheduled\_departure) BETWEEN 6 AND 10**

**ORDER BY scheduled\_departure ASC;**

1. **Identify the earliest morning flight available from every airport.**

Expected output: flight\_id, flight\_number, scheduled\_departure, scheduled\_arrival, departure airport and timings.

**Answer:**

**SELECT f.flight\_id, f.flight\_no, f.scheduled\_departure, f.scheduled\_arrival,**

**f.departure\_airport,**

**f.scheduled\_arrival-f.scheduled\_departure AS timings**

**FROM flights f**

**JOIN (**

**SELECT departure\_airport, MIN(scheduled\_departure) AS earliest\_departure**

**FROM flights**

**WHERE EXTRACT(HOUR FROM scheduled\_departure) BETWEEN 6 AND 10**

**GROUP BY departure\_airport**

**) AS earliest\_flights**

**ON f.departure\_airport = earliest\_flights.departure\_airport**

**AND f.scheduled\_departure = earliest\_flights.earliest\_departure;**

1. **Questions:** **Find list of airport codes in Europe/Moscow timezone**

Expected Output: Airport\_code.

**Answer:**

**SELECT**

**airport\_code**

**FROM airports**

**WHERE timezone = 'Europe/Moscow'**

1. **Write a query to get the count of seats in various fare condition for every aircraft code?**

Expected Outputs: Aircraft\_code, fare\_conditions ,seat count

**Answer:**

**SELECT**

**fare\_conditions,**

**COUNT (seat\_no) seat\_count**

**FROM seats**

**GROUP BY fare\_conditions**

1. **How many aircrafts codes have at least one Business class seats?**

Expected Output : Count of aircraft codes

**Answer:**

**SELECT**

**COUNT (DISTINCT aircraft\_code) AS count\_of\_aircrafts\_code**

**FROM seats**

**WHERE class = 'Business';**

1. **Find out the name of the airport having maximum number of departure flight**

Expected Output : Airport\_name

**Answer: WITH DepartureCounts AS (**

**SELECT departure\_airport, COUNT(\*) AS departure\_count,**

**RANK () OVER (ORDER BY COUNT (\*) DESC) AS departure\_rank**

**FROM flights**

**GROUP BY departure\_airport)**

**SELECT airport\_name**

**FROM DepartureCounts**

**JOIN airports ON DepartureCounts.departure\_airport = airports.airport\_code**

**WHERE departure\_rank = 1**

1. **Find out the name of the airport having least number of scheduled departure flights**

Expected Output : Airport\_name

**Answer:**

**WITH DepartureCounts AS (**

**SELECT departure\_airport, COUNT (scheduled\_departure) AS departure\_count,**

**RANK () OVER (ORDER BY COUNT (scheduled\_departure) ASC) AS departure\_rank**

**FROM flights**

**GROUP BY departure\_airport**

**)**

**SELECT airport\_name**

**FROM DepartureCounts**

**JOIN airports ON DepartureCounts.departure\_airport = airports.airport\_code**

**WHERE departure\_rank = 1;**

1. **How many flights from ‘DME’ airport don’t have actual departure?**

Expected Output : Flight Count

**Answer:**

**SELECT**

**COUNT (\*) AS Flight\_Count**

**FROM flights**

**WHERE departure\_airport = 'DME' AND actual\_departure IS NULL;**

1. **Identify flight ids having range between 3000 to 6000**

Expected Output : Flight\_Number , aircraft\_code, ranges

**Answer:**

**SELECT**

**DISTINCT flight\_no,**

**f.aircraft\_code,**

**a.range**

**FROM flights f**

**JOIN aircrafts a**

**ON f.aircraft\_code = a.aircraft\_code**

**WHERE a.range BETWEEN 3000 AND 6000**

**ORDER BY range ASC**

1. **Write a query to get the count of flights flying between URS and KUF?**

Expected Output : Flight\_count

**Answer:**

**SELECT**

**COUNT (\*) AS Flight\_count**

**FROM flights**

**WHERE departure\_airport = 'URS' AND arrival\_airport = 'KUF';**

1. **Write a query to get the count of flights flying from either from NOZ or KRR?**

Expected Output : Flight count

**Answer:**

**SELECT COUNT (\*) AS Flight\_count**

**FROM flights**

**WHERE departure\_airport IN ('NOZ', 'KRR');**

1. **Write a query to get the count of flights flying from KZN,DME,NBC,NJC,GDX,SGC,VKO,ROV**

Expected Output : Departure airport ,count of flights flying from these airports.

**Answer:**

**SELECT departure\_airport,**

**COUNT (\*) AS Flight\_count**

**FROM flights**

**WHERE departure\_airport IN ('KZN', 'DME', 'NBC', 'NJC', 'GDX', 'SGC', 'VKO', 'ROV')**

**GROUP BY departure\_airport**

**ORDER BY Flight\_count DESC;**

1. **Write a query to extract flight details having range between 3000 and 6000 and flying from DME**

Expected Output :Flight\_no,aircraft\_code,range,departure\_airport

**Answer:**

**SELECT**

**DISTINCT flight\_no,**

**f.aircraft\_code,**

**a.range,**

**f.departure\_airport**

**FROM flights f**

**JOIN aircrafts a**

**ON f.aircraft\_code = a.aircraft\_code**

**WHERE a.range BETWEEN 3000 AND 6000**

**AND departure\_airport = 'DME'**

**ORDER BY range ASC**

1. **Find the list of flight ids which are using aircrafts from “Airbus” company and got cancelled or delayed**

Expected Output : Flight\_id,aircraft\_model

**Answer:**

**SELECT**

**flight\_id,**

**(a.model::jsonb)->>'en' AS aircraft\_model**

**FROM flights f**

**JOIN aircrafts a**

**ON f.aircraft\_code = a.aircraft\_code**

**WHERE status IN ('Delayed','Cancelled') AND model like '%Airbus%'**

1. **Find the list of flight ids which are using aircrafts from “Boeing” company and got cancelled or delayed**

Expected Output : Flight\_id,aircraft\_model

**Answer:**

**SELECT**

**flight\_id,**

**(a.model::jsonb)->>'en' AS aircraft\_model**

**FROM flights f**

**JOIN aircrafts a**

**ON f.aircraft\_code = a.aircraft\_code**

**WHERE status IN ('Delayed','Cancelled') AND model like '%Boeing%'**

1. **Which airport(name) has most cancelled flights (arriving)?**

Expected Output : Airport\_name

Answer

**WITH cancelled\_flight AS (SELECT**

**(airport\_name::jsonb)->>'en' AS airport\_name,**

**count(f.\*) AS cancelled\_flight**

**FROM airports a**

**JOIN flights f**

**ON a.airport\_code = f.arrival\_airport**

**WHERE status = 'Cancelled'**

**GROUP BY (airport\_name::jsonb)->>'en'**

**ORDER BY 2 DESC),**

**Rank AS**

**(**

**SELECT**

**\*,**

**ROW\_NUMBER () OVER(ORDER BY cancelled\_flight DESC)**

**FROM cancelled\_flight**

**)**

**SELECT**

**airport\_name**

**FROM rank**

**WHERE ROW\_NUMBER = 1**

1. ***Identify flight ids which are using “Airbus aircrafts”***

*Expected Output : Flight\_id,aircraft\_model*

**Answer:**

**SELECT**

**flight\_id,**

**(a.model::jsonb)->>'en' AS aircraft\_model**

**FROM flights f**

**JOIN aircrafts a**

**ON f.aircraft\_code = a.aircraft\_code**

**WHERE model like '%Airbus%'**

1. ***Identify date-wise last flight id flying from every airport?***

*Expected Output: Flight\_id,flight\_number,schedule\_departure,departure\_airport*

**Answer**

**SELECT flight\_id, flight\_no, scheduled\_departure, departure\_airport**

**FROM (**

**SELECT flight\_id, flight\_no, scheduled\_departure, departure\_airport,**

**ROW\_NUMBER() OVER(PARTITION BY departure\_airport, DATE(scheduled\_departure) ORDER BY scheduled\_departure DESC) AS row\_num**

**FROM flights**

**) AS ranked\_flights**

**WHERE row\_num = 1;**

1. ***Identify list of customers who will get the refund due to cancellation of the flights and how much amount they will get?***

*Expected Output : Passenger\_name,total\_refund.*

**Answer:**

**SELECT**

**t.passenger\_name,**

**SUM(b.total\_amount) AS total\_refund**

**FROM**

**tickets t**

**JOIN bookings b ON t.book\_ref = b.book\_ref**

**JOIN ticket\_flights tf ON t.ticket\_no = tf.ticket\_no**

**JOIN flights f ON tf.flight\_id = f.flight\_id**

**WHERE status = 'Cancelled'**

**GROUP BY t.passenger\_name**

1. ***Identify date wise first cancelled flight id flying for every airport?***

*Expected Output : Flight\_id,flight\_number,schedule\_departure,departure\_airport*

**Answer:**

**SELECT flight\_id, flight\_no, scheduled\_departure, departure\_airport**

**FROM (**

**SELECT flight\_id, flight\_no, scheduled\_departure, departure\_airport,**

**ROW\_NUMBER () OVER (PARTITION BY departure\_airport, DATE (scheduled\_departure) ORDER BY scheduled\_departure ASC) AS row\_num**

**FROM flights**

**WHERE status = 'Cancelled'**

**) AS ranked\_flights**

**WHERE row\_num = 1**

1. ***Identify list of Airbus flight ids which got cancelled.***

*Expected Output : Flight\_id*

**Answer:**

**SELECT**

**flight\_id**

**FROM flights f**

**JOIN aircrafts a**

**ON f.aircraft\_code = a.acraft\_code**

**WHERE model like '%Airbus%' AND status = 'Cancelled'**

1. ***Identify list of flight ids having highest range.***

*Expected Output : Flight\_no, range*

**Answer:**

**WITH range\_table AS (SELECT**

**flight\_no,**

**a.range**

**FROM flights f**

**JOIN aircrafts a**

**ON f.aircraft\_code = a.aircraft\_code**

**ORDER BY 1 ASC),**

**rank\_table AS (**

**SELECT**

**\*,**

**RANK () OVER (ORDER BY range DESC) AS range\_rank**

**FROM range\_table**

**)**

**SELECT**

**flight\_no,**

**range**

**FROM rank\_table**

**WHERE range\_rank = 1**