**Please answer the following questions using Airline DB database.**

**Instruction to attempt questions:**

* Students need to write queries for the questions mentioned in the using Airline DB database
* Read the questions carefully before writing the query in **Airline Playground** (in the Playground chapter of SQL)
* Airline DB: [https://www.skillovilla.com/playground/sql?exerciseId=0181e251-6ea8-4595-ae2b-0c690119f8db](•%09https:/www.skillovilla.com/playground/sql?exerciseId=0181e251-6ea8-4595-ae2b-0c690119f8db)

**How to submit the capstone:**

* Copy the SQL query code and paste it in the answer section in this file.
* Once the assignment is done, submit the file over LMS.

**Invalid Submissions:**

* Pasting pictures of the code as answer is **NOT** acceptable.
* Uploading output data (CSVs) of the SQL queries is **NOT** acceptable.

**Write your answers(query) in the answer and submit it. To write the answer in the assignment, please follow the below example in yellow**

Example:

Questions*: Extract all the columns of the flights table*

Answer: *SELECT \* FROM flights*

**Attempt the following 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 t.ticket\_no,**

**bp.boarding\_no,**

**bp.seat\_no,**

**t.passenger\_id,**

**t.passenger\_name FROM tickets t**

**JOIN boarding\_passes bp on bp.ticket\_no = t.ticket\_no**

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

**Answer: SELECT seat\_no FROM (SELECT seat\_no, COUNT(\*) as seat\_count FROM boarding\_passes**

**GROUP BY seat\_no**

**ORDER BY seat\_count**

**LIMIT 1) as seat\_no**

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 MaxMonthAmount as ( SELECT TO\_CHAR(book\_date,'MON-YY') as Month\_name,**

**t.passenger\_id,**

**t.passenger\_name,**

**b.total\_amount,**

**ROW\_NUMBER() OVER(PARTITION BY TO\_CHAR(book\_date,'MON-YY') ORDER BY b.total\_amount DESC) as row\_no**

**from bookings b join tickets t on**

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

**SELECT Month\_name,**

**passenger\_id,**

**passenger\_name,**

**total\_amount**

**FROM MaxMonthAmount**

**WHERE row\_no = 1**

**ORDER BY Month\_name ASC**

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 MinMonthAmount as ( SELECT TO\_CHAR(book\_date,'MON-YY') as Month\_name,**

**t.passenger\_id,**

**t.passenger\_name,**

**b.total\_amount,**

**ROW\_NUMBER() OVER(PARTITION BY TO\_CHAR(book\_date,'MON-YY') ORDER BY b.total\_amount ASC) as row\_no**

**from bookings b join tickets t on**

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

**SELECT Month\_name,**

**passenger\_id,**

**passenger\_name,**

**total\_amount**

**FROM MinMonthAmount**

**WHERE row\_no = 1**

**ORDER BY Month\_name ASC**

1. **Identify the travel details of non stop 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 as ticket\_number,**

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

**FROM tickets t join ticket\_flights f**

**on t.ticket\_no = f.ticket\_no**

**GROUP BY 1,2,3**

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

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

Expected Output: just one number is required.

**Answer:  SELECT COUNT(\*) as ticket\_count FROM tickets t LEFT JOIN boarding\_passes b**

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

**WHERE b.ticket\_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:  SELECT**

**flight\_no as flight\_number,**

**departure\_airport,**

**arrival\_airport,**

**aircraft\_code,**

**(scheduled\_arrival-scheduled\_departure) as durations FROM flights**

**ORDER BY 5 DESC**

**LIMIT 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 as flight\_number,**

**scheduled\_departure,**

**scheduled\_arrival,**

**CAST(scheduled\_departure AS time) AS timings**

**FROM flights**

**WHERE CAST(scheduled\_departure AS time) BETWEEN '06:00:00' AND '11:00:00'**

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:  WITH MorningFlight AS ( SELECT**

**flight\_id,**

**flight\_no ,**

**scheduled\_departure,**

**scheduled\_arrival,**

**departure\_airport,**

**CAST(scheduled\_departure AS time) AS timings,**

**ROW\_NUMBER() OVER (PARTITION BY departure\_airport ORDER BY scheduled\_departure) AS row\_no**

**FROM flights**

**WHERE CAST(scheduled\_departure AS time) BETWEEN '06:00:00' AND '11:00:00')**

**SELECT**

**flight\_id,**

**flight\_no as flight\_number,**

**scheduled\_departure,**

**scheduled\_arrival,**

**departure\_airport,**

**timings**

**FROM MorningFlight**

**WHERE row\_no = 1**

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

Expected Output: Airport\_code.

**Answer: SELECT DISTINCT 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 aircraft\_code, fare\_conditions, COUNT (\*) as seat\_count FROM seats**

**GROUP BY 1,2**

**ORDER BY 1,2**

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\_aircraft from seats**

**WHERE fare\_conditions = 'Business'**

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

Expected Output : Airport\_name

**Answer:  SELECT airport\_name FROM airports WHERE**

**airport\_code =( SELECT departure\_airport FROM Flights**

**GROUP BY 1**

**ORDER BY COUNT(\*) DESC**

**LIMIT 1)**

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

Expected Output : Airport\_name

**Answer:  SELECT airport\_name FROM airports WHERE**

**airport\_code =( SELECT departure\_airport FROM Flights**

**GROUP BY 1**

**ORDER BY COUNT(\*) DESC**

**LIMIT 1)**

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

Expected Output : Flight Count

**Answer:  SELECT COUNT(\*) 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 f.flight\_no AS flight\_number , a.aircraft\_code, a.range as ranges FROM**

**aircrafts a**

**join flights f on a.aircraft\_code = f.aircraft\_code**

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

**GROUP BY 1,2,3**

**ORDER BY 3**

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 LIKE 'URS'**

**AND arrival\_airport LIKE '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 LIKE 'NOZ'**

**OR departure\_airport LIKE '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 1**

**ORDER BY 2**

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 f.flight\_no, a.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 F.departure\_airport like 'DME'**

**GROUP BY 1,2,3,4**

**ORDER BY a.range**

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 f.flight\_id , a.model AS aircraft\_model FROM flights f**

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

**WHERE a.model like '%Airbus%'**

**AND f.status = 'Cancelled'**

**OR f.status = 'Delayed'**

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 f.flight\_id , a.model AS aircraft\_model FROM flights f**

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

**WHERE a.model like '%Boeing%'**

**AND f.status = 'Cancelled'**

**OR f.status = 'Delayed'**

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

Expected Output : Airport\_name

**Answer:**

SELECT a.airport\_name FROM airports a

 JOIN flights f ON a.airport\_code = f.arrival\_airport

 WHERE f.status = 'Cancelled'

 GROUP BY a.airport\_name

 ORDER BY COUNT(\*)DESC

 LIMIT 1

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

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

**Answer:  SELECT f.flight\_id, a.model AS aircraft\_model FROM flights f**

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

**WHERE a.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 as flight\_number,**

**scheduled\_departure,**

**departure\_airport FROM**

**(SELECT \*,Rank()OVER(PARTITION BY departure\_airport ORDER BY scheduled\_departure DESC )**

**FROM flights ) as flight**

**WHERE rank = 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, MAX(tf.amount) AS total\_refund FROM**

**tickets t**

**JOIN ticket\_flights tf**

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

**JOIN flights f**

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

**GROUP BY 1**

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 as flight\_number, scheduled\_departure, scheduled\_arrival,departure\_airport**

**FROM (SELECT \*,RANK()OVER(PARTITION BY departure\_airport ORDER BY scheduled\_departure )**

**FROM flights where status='Cancelled') AS time WHERE rank = 1**

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

*Expected Output : Flight\_id*

**Answer:  SELECT f.flight\_id FROM flights f JOIN**

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

**WHERE a.model LIKE '%Airbus%'**

**AND f.status = 'Cancelled'**

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

*Expected Output : Flight\_no, range*

**Answer: SELECT DISTINCT f.flight\_no,max(a.range) AS range FROM flights f JOIN**

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

**GROUP BY 1**