**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-MM-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 tickets t*

*join boarding\_passes b*

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

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

**Answer:**

*Select*

*seat\_no, count(seat\_no) as allocated\_seats*

*from boarding\_passes*

*group by 1*

*order by 2 asc*

*limit 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 t1 as (select*

*to\_char(book\_date, 'mon-yy') as Month\_name,*

*passenger\_id, passenger\_name, sum(total\_amount) as total\_amount,*

*rank() over(partition by to\_char(book\_date, 'mon-yy') order by sum(total\_amount)desc) as paying\_rank*

*from tickets t*

*join bookings b*

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

*group by 1,2,3)*

*select \* from t1*

*where paying\_rank = 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 t1 as (select*

*to\_char(book\_date, 'mon-yy') as Month\_name,*

*passenger\_id, passenger\_name, sum(total\_amount) as total\_amount,*

*rank() over(partition by to\_char(book\_date, 'mon-yy') order by sum(total\_amount)asc) as paying\_rank*

*from tickets t*

*join bookings b*

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

*group by 1,2,3)*

*select \* from t1*

*where paying\_rank = 1*

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*

*passenger\_id, passenger\_name,t.ticket\_no,*

*count(tf.flight\_id) as flight\_count*

*from tickets t*

*join ticket\_flights tf*

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

*group by 1,2,3*

*having count(tf.flight\_id) > 1*

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

Expected Output: just one number is required.

**Answer:**

*select*

*count(t.ticket\_no) as tickets\_without\_boarding\_passes*

*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,*

*departure\_airport,*

*arrival\_airport,*

*aircraft\_code,*

*scheduled\_Arrival-scheduled\_departure as duration*

*from flights*

*order by duration 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,*

*to\_char(scheduled\_departure, 'HH24:MI:SS AM') as scheduled\_departure,*

*to\_char(scheduled\_arrival, 'HH24:MI:SS AM') as scheduled\_arrival,*

*concat(to\_char(scheduled\_departure, 'HH24:MI:SS AM'), ' - ', to\_char(scheduled\_arrival, 'HH24:MI:SS AM')) as timings*

*from*

*flights*

*where*

*extract(hour from scheduled\_departure) between 6 and 11*

*order by scheduled\_departure*

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*

*flight\_id,*

*flight\_no,*

*to\_char(scheduled\_departure, 'HH24:MI:SS AM') as scheduled\_departure,*

*to\_char(scheduled\_arrival, 'HH24:MI:SS AM') as scheduled\_arrival,*

*departure\_airport,*

*concat(TO\_CHAR(scheduled\_departure, 'HH24:MI:SS AM'), ' - ',TO\_CHAR(scheduled\_arrival, 'HH24:MI:SS AM')) AS timings*

*from flights*

*where*

*TO\_CHAR(scheduled\_departure, 'HH24'):: integer BETWEEN 2 AND 6*

*and to\_char(scheduled\_departure, 'AM') = 'AM'*

*order by*

*scheduled\_departure*

*limit 1*

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*

*aircraft\_code,*

*fare\_conditions,*

*count(seat\_no) as seat\_count,*

*rank() over(partition by aircraft\_code order by count(seat\_no)desc)*

*from seats*

*group by 1,2*

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

Expected Output : Count of aircraft codes

*Answer:*

*select*

*count(aircraft\_code) as aircraft\_codes*

*from seats*

*where fare\_conditions = 'Business'*

*having count(seat\_no) >=1*

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

Expected Output : Airport\_name

**Answer:**

*with t1 as (select*

*airport\_name,*

*COUNT(flight\_id) as departure\_flight*

*from airports a*

*join flights f*

*on a.airport\_code = f.departure\_airport*

*GROUP BY airport\_name*

*ORDER BY departure\_flight DESC)*

*select*

*airport\_name*

*from t1*

*limit 1*

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

Expected Output : Airport\_name

**Answer:**

*with t1 as (select*

*airport\_name,*

*COUNT(flight\_id) as departure\_flight*

*from airports a*

*join flights f*

*on a.airport\_code = f.departure\_airport*

*GROUP BY airport\_name*

*ORDER BY departure\_flight ASC)*

*select*

*airport\_name*

*from t1*

*limit 1*

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

Expected Output : Flight Count

**Answer:**

*Select*

*Count(Flight\_id) AS FLIGHT\_COUNT*

*From Flights*

*Where status = 'Scheduled' and departure\_airport = 'DME'*

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

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

**Answer:**

*Select*

*flight\_no as flight\_number,*

*a.aircraft\_code,*

*range*

*from aircrafts a*

*join flights f*

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

*where range between 3000 and 6000*

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

Expected Output : Flight\_count

**Answer:**

*select*

*count(flight\_id) as flight\_count*

*from flights*

*where  (departure\_airport = 'URS' AND arrival\_airport = 'KUF') OR (departure\_airport = 'KUF' AND arrival\_airport = 'URS')*

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

Expected Output : Flight count

**Answer:**

*select*

*count(flight\_id) 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(flight\_id) as count\_of\_flights\_flying*

*from flights*

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

*group by 1*

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*

*flight\_no,*

*a.aircraft\_code,*

*range,*

*departure\_airport*

*from flights f*

*join aircrafts a*

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

*where departure\_airport = 'DME' and range between 3000 and 6000*

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,*

*model,*

*status*

*from flights f*

*join aircrafts a*

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

*where model like '%Airbus%' and status in ('Delayed', 'Cancelled')*

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,*

*model,*

*status*

*from flights f*

*join aircrafts a*

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

*where model like '%Boeing%' and status in ('Cancelled', 'Delayed')*

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

Expected Output : Airport\_name

**Answer:**

*Select*

*airport\_name,*

*Count(\*) as cancelled\_flights*

*From flights f*

*Join*

*Airports a*

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

*Where actual\_arrival IS NULL*

*Group by airport\_name*

*Order by cancelled\_flights DESC*

*Limit 1*

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

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

**Answer:**

*select*

*flight\_id,*

*model*

*from flights f*

*join aircrafts a*

*on a.aircraft\_code = f.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:**

*with t1 as (select*

*flight\_id,*

*flight\_no,*

*Date(scheduled\_departure) as schedule\_departure,*

*departure\_airport,*

*dense\_rank()over(partition by departure\_airport order by date(scheduled\_departure)desc) as departurerank*

*from flights)*

*select*

*flight\_id, flight\_no, schedule\_departure, departure\_airport from t1*

*where departurerank =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*

*passenger\_name,*

*sum(tf.amount) as total\_refund*

*from*

*Flights f*

*join*

*Ticket\_flights tf*

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

*join*

*Tickets t*

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

*where status = 'Cancelled'*

*group by 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:**

*with t1 as (select*

*flight\_id,*

*flight\_no,*

*date(scheduled\_departure) as schedule\_departure,*

*departure\_airport,*

*row\_number() over(partition by departure\_airport order by date(scheduled\_departure)asc)*

*from flights f*

*where status = 'Cancelled')*

*select*

*flight\_id,flight\_no,schedule\_departure, departure\_airport from t1*

*where row\_number = 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 a.aircraft\_code = f.aircraft\_code*

*where status = 'Cancelled' and model like '%Airbus%'*

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

*Expected Output : Flight\_no, range*

**Answer:**

*with t1 as (select*

*flight\_id, range,*

*rank()over(order by range desc)*

*from flights f*

*join aircrafts a*

*on a.aircraft\_code = f.aircraft\_code)*

*select flight\_id, range from t1 where rank = 1*