USE CASE STUDY REPORT

Group No.: Group 13

Names: Amruta Sudhakar Shinde & Pranjal Umesh Kalekar

Topic: Airport Management System

Executive Summary: Everyday, enormous volumes of information are created at airports. Managing this constantly changing information is critical, as any out-of-date information can cause problems when planning future operations. As a result, we propose an 'Airport Management and Monitoring System' model that would contain all data associated to the entities stated below for simplicity of executing any database operations. To build this model, we need a collection of data for each entity stated in the conceptual model, as well as a platform to run queries for each task.

We will use MySQL Workbench to develop and run queries for this project, and Python to render the analytics. The information gathered from each entity may subsequently be utilized to analyse various criteria. These may include tasks such as determining which day of the week the most flights took off in order to calculate the maximum revenue generated, determining when the least number of taxis were available at the airport in order to increase the number of taxis on those specific days, determining which airline is preferred by the majority of customers in order to determine the airline with the highest demand, determining which offers should be made available to customers based on their travel history, and determining which airlines are most preferred by customers.

This concept will make airline administration and passenger coordination easier as well as help the stakeholders to determine what strategies can be used to increase their customers, revenue and resources.

I. Introduction

Our motivation for selecting an airport management system was to create a system that could handle many jobs while being simple to use. Another motivator was the massive volume of data created at airports every day, as well as its usefulness. This data may be utilized in a variety of ways to generate important insights that will help the aviation business thrive if handled correctly.

Background

Airport operations encompass a wide range of activities, including check-in, luggage checks, and security, as well as aircraft handling, parking facilities, retailing, maintenance, and client service. Each stratum has its own set of requirements, demands, and obstacles, and addressing all aspects of an airport may aid in the creation of a profitable business across all tiers. Because airports have so many moving parts, competent management is critical. Conventional, low-tech airport operations are time-consuming, expensive, and require a significant number of personnel to function well. Running an airport incurs several expenditures that, if not properly planned for and managed, can stifle economic growth. Airport management that has been updated provides teams with a complete view at all procedures to determine where they can effectively contribute funds or remove wasteful spending.

Requirements:

Following are the different entities and their attributes required in this model, as well as information on the data that these entities will hold.

1) Airport - This entity will hold all of the airport's information, such as its name, location, and ID (as one city can have two or more airports).

2) Flight – This entity will house all information pertaining to planes arriving and departing from the specific airport.

3) Route - This entity will keep track of how many paths are available to go to a certain place and how much each route costs.

4) Terminal – This is a weak entity that will store information on each terminal at the airport, including its location and which flights are available at which terminal.

5) Ticket – This entity will maintain track of each ticket and the information linked with it, which will be used to estimate the suggestions based on prior decisions.

6) Passenger This entity will include all of the passenger's information. This information will also include past flight history. Further specialised into 3 more entities namely, premium customer, first time customer, and general customer. These entities are overlapped.

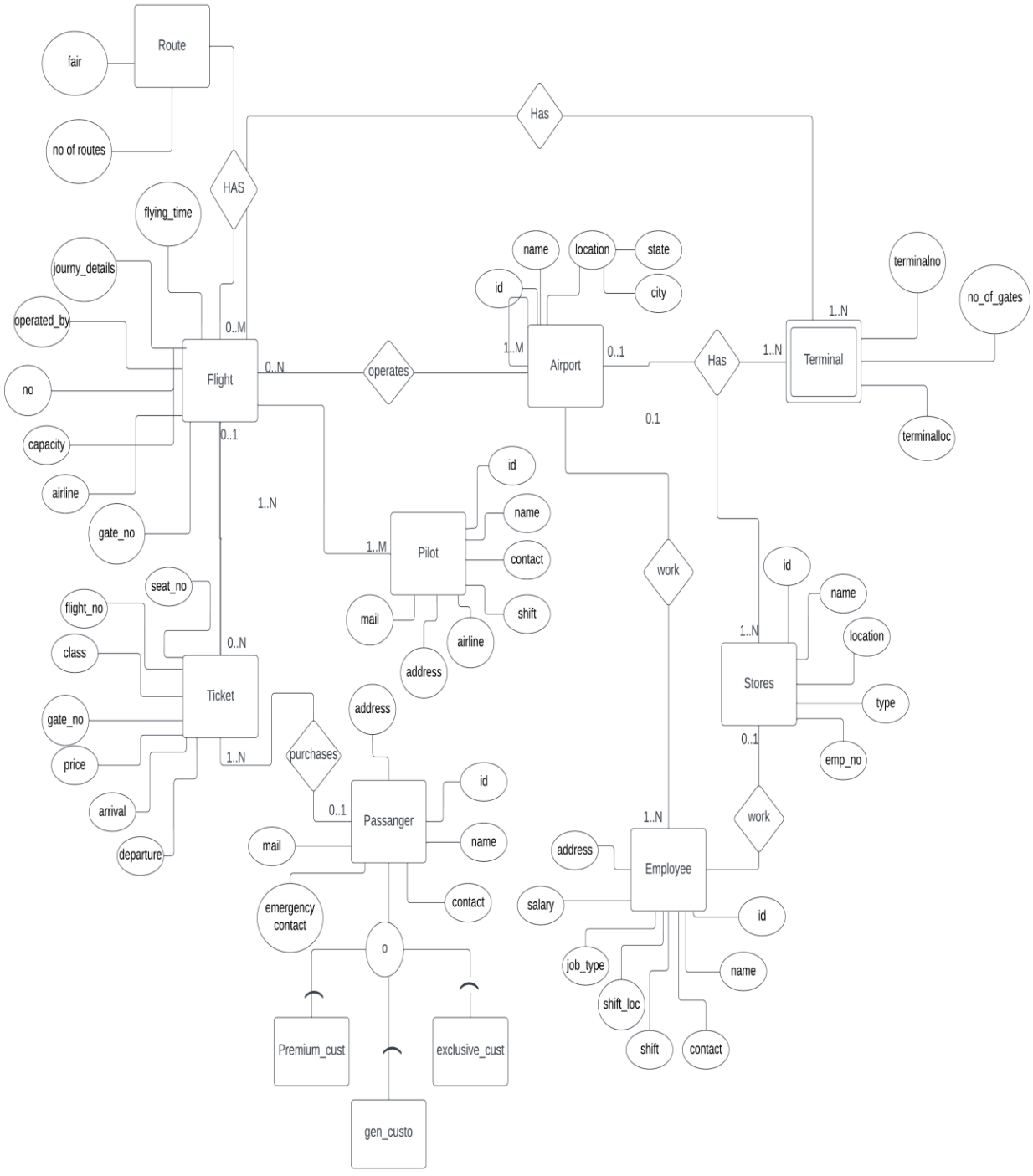
7) Employee – This entity will aid in the tracking of all airport employees.

8) Stores – This provides information on all of the stores and the items offered at each store at a specific airport.

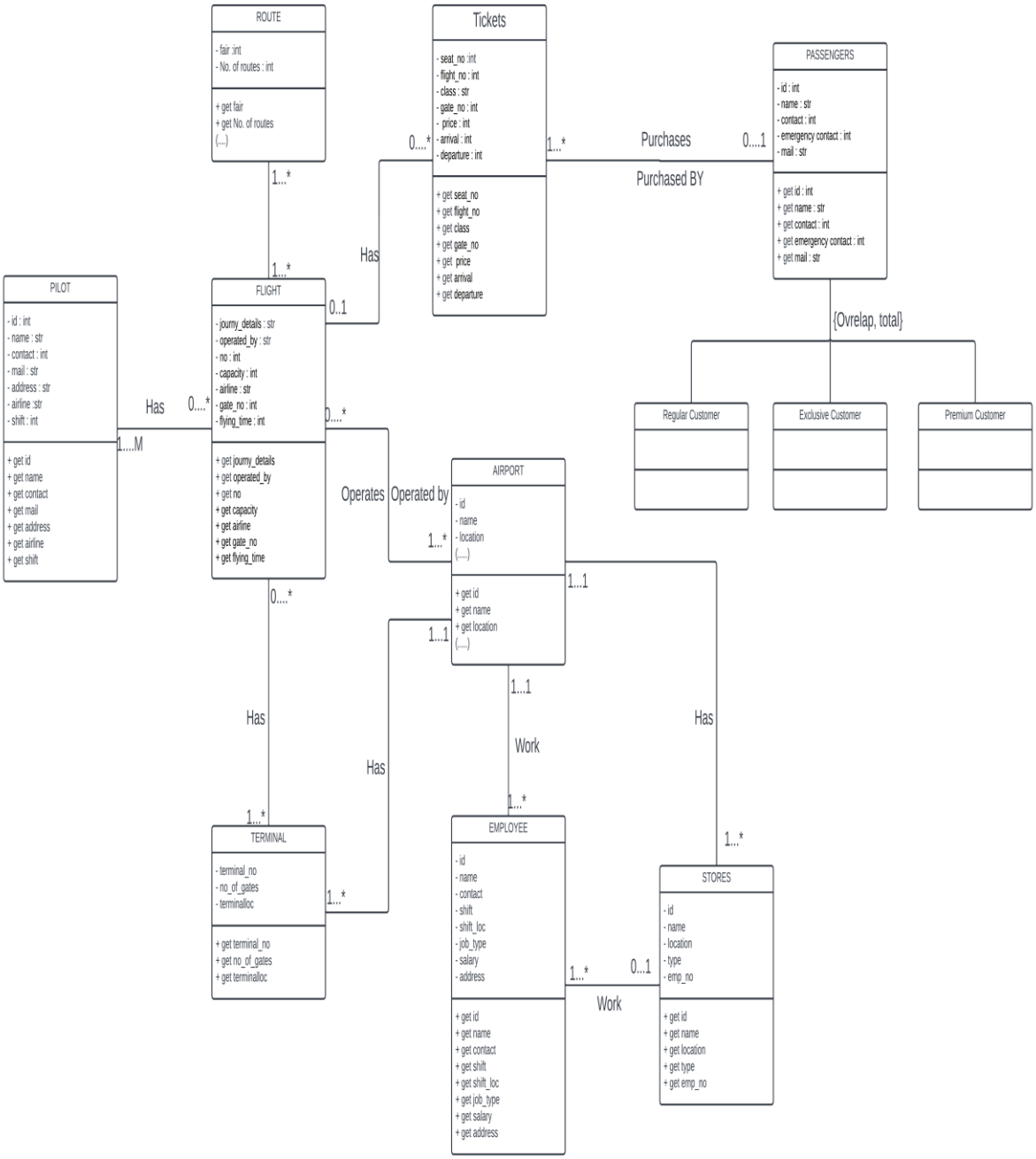
9) Pilot – This will contain information on each pilot linked with flights that arrive or depart from the airport.

II. Conceptual Data Modelling

a) EER Diagram



b) UML Diagram:



Normalizing the schema

- Airport (ID, name, location)
- Location (state, city)
- Terminal (terminal_no, no_of_gates, terminal_location, AirportID)
- Flight (flight_no, operated_by, capacity, airline, gate_no, flying_time)
- Flight_terminal (flightno*, terminalno*)
- Route (fair, no_of_gates)
- Flight_route (flightno*, routefair*)
- Flight_airport (AirportID*, flightID*)
- Flight_analysis(year, month , day, day_of_week, airline, flight_no*, tail_number, origin_airport, destination_airport, scheduled_departure, departure_time, departure_delay, taxi_out, wheels_off, scheduled_time, elapsed_time, air_time, distance, wheels_on, taxi_in, scheduled_arrival, arrival_time, arrival_delay, diverted, cancelled)
- Ticket (ID, seatno, class, flightno*, gate_no, price, arrival, departure, passID *)
- Passenger (passID, passname, contact, emergencycontact, mail, address, ticketID*)
- Regular_Customer (passID*)
- Premium_Customer (passID*)
- Exclusive_Customer (passID*)
- Pilot (ID, name, contact, shift, airline, address, mail)
- Pilot_flight (pilotID*, flightID*)
- Store (ID, name, location, type, emp_no, AirportID*)
- Employee (ID, name, contact, shift, shiftloc, job_type, salary, address, AirportID*, storeID*)

IV. Implementation of Relation Model via MySQL and NoSQL

MySQL:

--Inserting table Airport—

```
DROP TABLE IF EXISTS Airport
CREATE TABLE Airport ( ID int,
Name varchar(50),
Loaction varchar(50));
LOAD DATA INFILE “C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Airport.csv”
INTO TABLE Airport
FIELDS TERMINATED BY ‘,’
ENCLOSED BY “”
LINES TERMINATED BY ‘\n’
IGNORE ROWS 1;
```

--Inserting table Location—

```
DROP TABLE IF EXISTS Location
CREATE TABLE Location ( state varchar(20),
 City varchar(20));
```

```
LOAD DATA INFILE “C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/
Location.csv”
INTO TABLE Location
FIELDS TERMINATED BY ‘,’
ENCLOSED BY “”
LINES TERMINATED BY ‘\n’
IGNORE ROWS 1;
```

--Inserting table Terminal—

```
DROP TABLE IF EXISTS Terminal
CREATE TABLE Terminal ( terminal_no int,
no_of_gates int,
terminal_location varchar(20));
LOAD DATA INFILE “C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/
Terminal.csv”
INTO TABLE Terminal
FIELDS TERMINATED BY ‘,’
ENCLOSED BY “”
LINES TERMINATED BY ‘\n’
IGNORE ROWS 1;
```

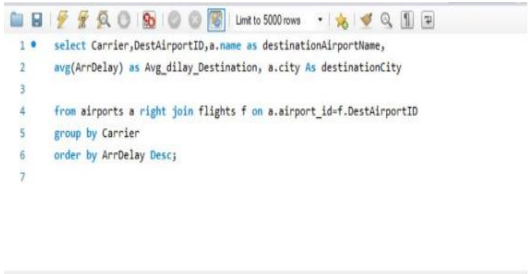
--Inserting table Flight—

```
DROP TABLE IF EXISTS Flight
CREATE TABLE Flight ( flight_no int,
operated_by varchar(20),
capacity int,
airline varchar(20),
gate_no int,
flying_time TIME);
LOAD DATA INFILE “C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/ Flight.csv”
INTO TABLE Flight
FIELDS TERMINATED BY ‘,’
ENCLOSED BY “”
LINES TERMINATED BY ‘\n’
IGNORE ROWS 1;
```

SQL Queries implementation:

1.Get the average arrival delay for the carrier at the destination Airport In descending order of the delay.

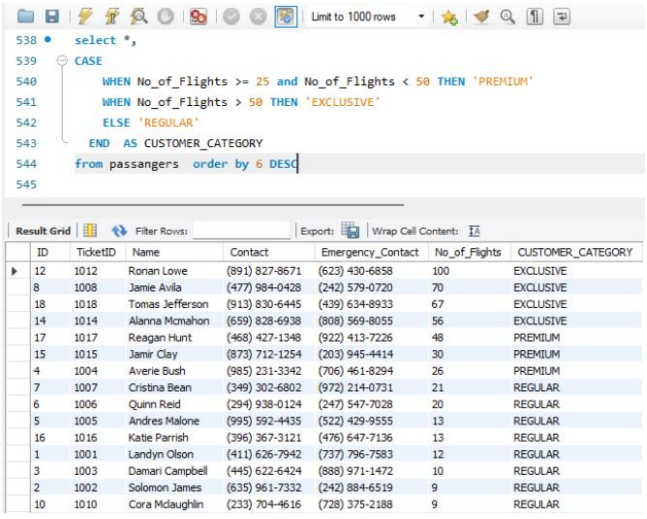
2.Categorise the customers into Premium, Exclusive and Regular Customers



```
select Carrier,DestAirportID,a.name as destinationAirportName,
avg(ArrDelay) as Avg_dilay_Destination, a.city As destinationCity

from airports a right join flights f on a.airport_id=f.DestAirportID
group by Carrier
order by ArrDelay Desc;
```

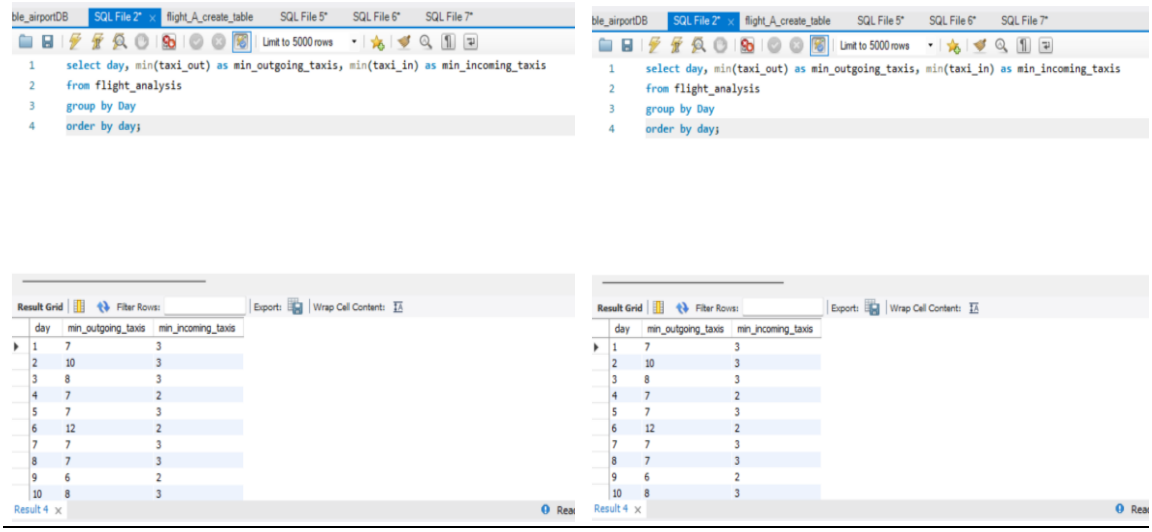
Carrier	DestAirportID	destinationAirportName	Avg_dilay_Destination	destinationCity
US	14771	San Francisco International	25.6000	San Francisco
AS	14747	Seattle/Tacoma International	5.0000	Seattle
UA	13930	Chicago O'Hare International	-6.1818	Chicago
9E	14492	Raleigh-Durham International	2.0000	Raleigh/Durham
MQ	13930	Chicago O'Hare International	45.5000	Chicago
WN	10140	Albuquerque International Sunport	5.2143	Albuquerque
OO	14771	San Francisco International	7.6250	San Francisco
FL	10821	Baltimore/Washington International Thurgood M...	0.0000	Baltimore
F9	11292	Denver International	-9.5000	Denver
AA	11298	Dallas/Fort Worth International	-5.0833	Dallas/Fort Worth



```
select *,
CASE
    WHEN No_of_Flights >= 25 and No_of_Flights < 50 THEN 'PREMIUM'
    WHEN No_of_Flights > 50 THEN 'EXCLUSIVE'
    ELSE 'REGULAR'
  END  AS CUSTOMER_CATEGORY
from passangers  order by 6 DESC
```

ID	TicketID	Name	Contact	Emergency_Contact	No_of_Flights	CUSTOMER_CATEGORY
12	1012	Ronan Lowe	(891) 827-8671	(623) 430-6858	100	EXCLUSIVE
8	1008	Jamie Avila	(477) 984-0428	(242) 579-0720	70	EXCLUSIVE
18	1018	Tomas Jefferson	(913) 830-6445	(439) 634-8933	67	EXCLUSIVE
14	1014	Alanna Mcmahon	(659) 828-6938	(808) 569-8055	56	EXCLUSIVE
17	1017	Reagan Hunt	(468) 427-1348	(922) 413-7226	48	PREMIUM
15	1015	Jamir Clay	(873) 712-1254	(203) 945-4414	30	PREMIUM
4	1004	Averie Bush	(985) 231-3342	(706) 461-8294	26	PREMIUM
7	1007	Cristina Bean	(349) 302-6802	(972) 214-0731	21	REGULAR
6	1006	Quinn Reid	(294) 938-0124	(247) 547-7028	20	REGULAR
5	1005	Andres Malone	(995) 592-4435	(522) 429-9555	13	REGULAR
16	1016	Katie Parrish	(396) 367-3121	(476) 647-7136	13	REGULAR
1	1001	Landyn Olson	(411) 626-7942	(737) 796-7583	12	REGULAR
3	1003	Damari Campbell	(445) 622-6424	(888) 971-1472	10	REGULAR
2	1002	Solomon James	(635) 961-7332	(242) 884-6519	9	REGULAR
10	1010	Cora Mclaughlin	(233) 704-4616	(728) 375-2188	9	REGULAR

3.Least number of taxis arrived and went from the airport on a particular day of the month.
4.Maximum number of flights on a particular day in the month of February 2015



```
select day, min(taxi_out) as min_outgoing_taxis, min(taxi_in) as min_incoming_taxis
from flight_analysis
group by Day
order by day;
```

day	min_outgoing_taxis	min_incoming_taxis
1	7	3
2	10	3
3	8	3
4	7	2
5	7	3
6	12	2
7	7	3
8	7	3
9	6	2
10	8	3

Populated all other tables specified in the similar way using csv file.

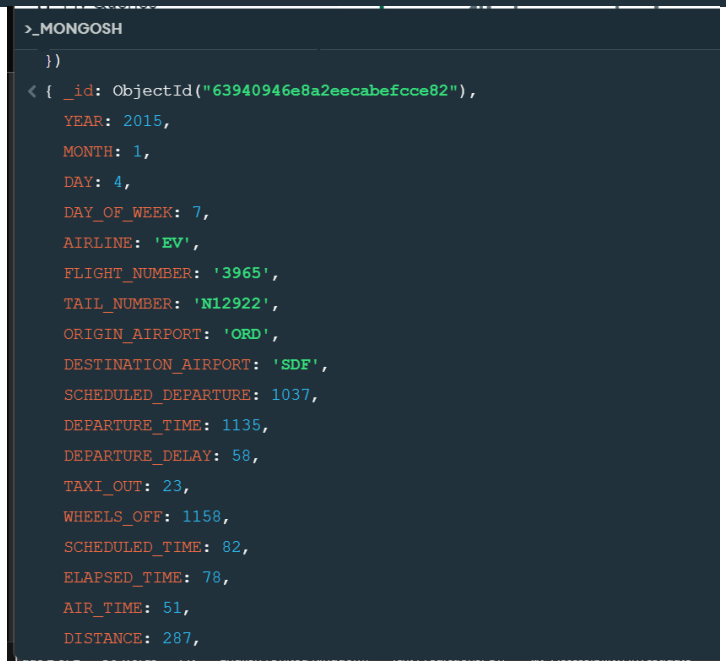
Implementation of the relational model in NoSQL:

1) To find all the details of the city from airport table

```
2. db.airports.find({
3.   "city": "Anchorage"
4. })
```

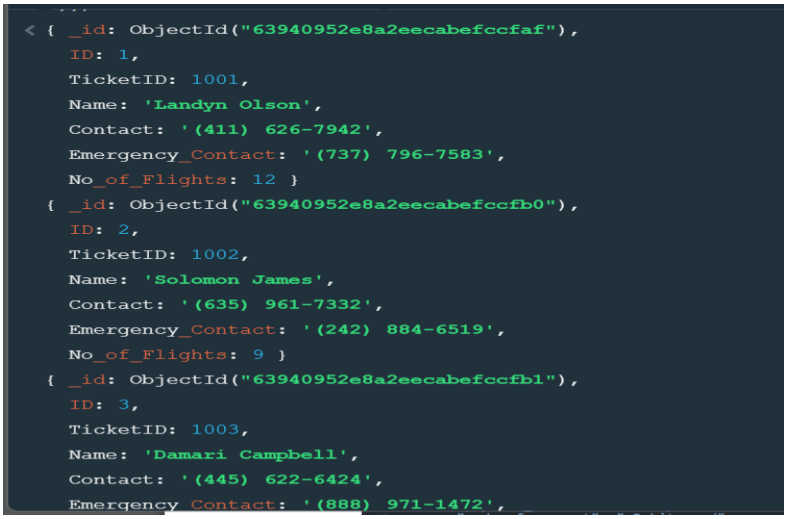
2) To find the list of “FLIGHT_NUMBER” FLIGHTS with TAIL_NUMBER “N12922” from orders table

```
 "FLIGHT_NUMBER": "3965",
 "TAIL_NUMBER": "N12922"
})
```



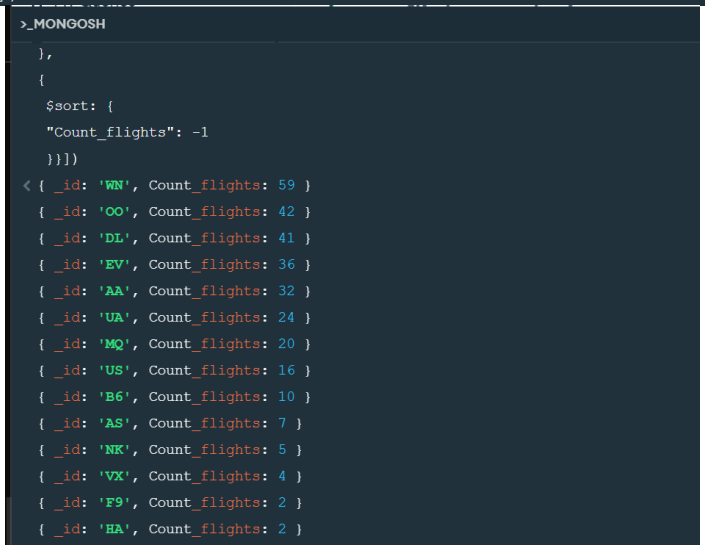
3) To find the list of all passengers that are less than 25 no of flights from passenger table

```
db.passengers.find({
 "No_of_Flights": {
 $lt: 25
 }})
```



4) To find the number of flights respectively in descending order from orders table

```
db.flight_analysis.aggregate([
 {
 $group: {
 _id: "$AIRLINE",
 Count_flights: {
 $count: {}
 }
 }
},
{
 $sort: {
 "Count_flights": -1
 }}])
```

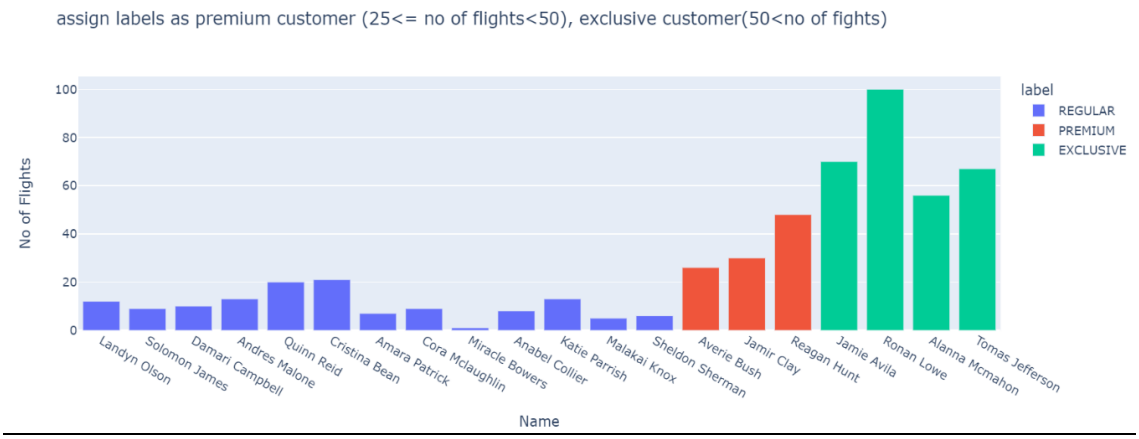


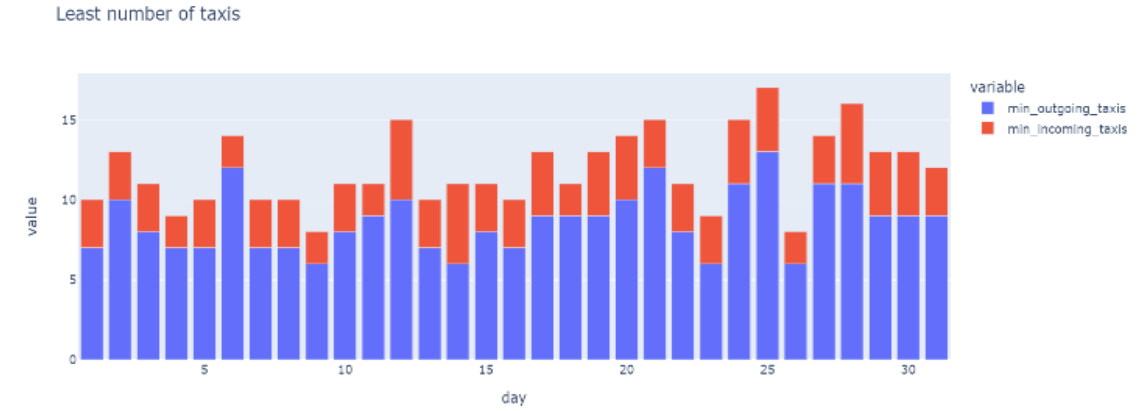
V. Database Access via R or Python

#example of python connecting to MySQL server and databases
#import mysql.connector
#from mysql.connector import Error
try:
    connection = mysql.connector.connect(host='localhost',
                                         database='airport',
                                         user='root',

```
                                password=******)
    if connection.is_connected():
        db_Info = connection.get_server_info()
        print("Connected to MySQL Server version ", db_Info)
        cursor = connection.cursor()
        cursor.execute("select database();")
        record = cursor.fetchone()
        print("Your connected to database: ", record)
# sql_select_Query = "SELECT city FROM airports ORDER BY city ASC;"
        cursor = connection.cursor()
        cursor.execute(sql_select_Query)
        records = cursor.fetchall()
        print("passangers")
        for row in records:
            print('row  =',row,"\n")
#except Error as e:
    print("Error while connecting to MySQL", e)
finally:
    if (connection.is_connected()):
        cursor.close()
        connection.close()
        print("MySQL connection is closed")
```

Analytics based on the queries:





VII. Summary and recommendation

Creating a database is never an easy task. During the process of creating the database, we learned many things such as taking real-world objects into evaluation and designing entities and attributes, normalizing the entire schema and assessing logical connections. In our scenario, "Airport Management System" will stand in for real-time anomalies.

Advantage - The benefits of our system include its ability to carry out a variety of tasks that can help lower the airport's operational costs by reducing the buffer time between flights, preventing flight cancellations, or suggesting the best routes to avoid climatic challenges. It can also draw conclusions that can help stakeholders increase their business.

Disadvantage - Because the airport contains a ton of data, our solution has the disadvantage that it could lag or take longer to deliver results. Additionally, we don't currently have any functionality to identify and remove old data.

Future Scope/ Recommendation - At higher levels, this model may be improved to forecast occurrences and generate alerts before an issue occurs. Airport data can be combined with open linked data such as weather, population, and environment to predict early critical failures and maintenance needs, optimize flight paths, reschedule routes in real-time, improve operational efficiency, provide a seamless ground/air passenger experience, protect the environment, and monitor safety and risk threats.