**Airport Database Management System**

Mini Project Report -Database Lab (DSE 2241)

Department of Data Science & Computer Applications



B. Tech Data Science

4th Semester – Batch: B1- Group: 3

Submitted By

|  |  |
| --- | --- |
| Hemang Malik | 220968092 |
| Akhil Ajai Kumar | 220968062 |
| Kshitij Sohni | 220968054 |
| Dhanush Varma | 220968082 |
| Sreenivasa Sai Krishna Kartik | 220968090 |

**Mentored By**

|  |  |
| --- | --- |
| Vinayak M | Archana H |
| Assistant Professor-Senior | Assistant Professor-Senior |
| DSCA, MIT | DSCA, MIT |

Date: 03/04/24

**CERTIFICATE**

This is to certify that Kshitij Sohoni(220968054), Akhil Ajai Kumar(220968062), Hemang Malik(220968092), M Dhanush Varma(220968082), Sreenivasa Sai Krishna Kartik(220968090), have successfully executed a mini project titled “Hotel Management System” rightly bringing fore the competencies and skill sets they have gained during the course- Database Lab (DSE 2241), thereby resulting in the culmination of this project.

|  |  |
| --- | --- |
| **Vinayak M** | **Archana H** |
| **Assistant Professor-Senior** | **Assistant Professor-Senior** |
| **DSCA, MIT** | **DSCA, MIT** |

**ABSTRACT**

**Methodology:** The project employs SQL Plus as the primary tool for the database management system. It aims to integrate a user- friendly interface along with comprehensive, relational data to safeguard sensitive passenger information and implement data validation checks. This would enable users to access information regarding their flight and booking details such as flight number, ticket id, booking fare, flight schedule, arrival time, departure time etc

**Results and Significance**: Through implementation, the Airport-DBMS prioritizes simplicity for users and ensures secure, scalable interface for large-scale usage. It would enable improved efficiency for the airport and robust security for the user as well. It aims to foster better decision-making, resource allocation, and collaboration, leading to smoother airport operations.

**Contents**

[**1. Introduction 1**](#_heading=h.gjdgxs)

[**2. Synopsis 2**](#_heading=h.30j0zll)

[**2.1 Proposed System 2**](#_heading=h.1fob9te)

[**2.2 Objectives 2**](#_heading=h.3znysh7)

[**3. Functional Requirements 3**](#_heading=h.2et92p0)

### 4. Detailed Design 4

**4.1 ER Diagram 4 4.2 Schema Diagram 4 4.3 Data Dictionary 5**

**4.4 Relational Model Implementation 7**

### 5. Implementation 9

#### 5.1 Queries 9

**5.2 Triggers 10**

**5.3 Stored Functions 12**

### 6. Result 15

### 7. Conclusion and Future Work 16

**Chapter 1**

**Introduction**

This project aims to develop an efficient airport database management system (ADBMS). It is designed to store and manage critical information related to airport operations such as data concerning flights, passengers, airport resources, and more all in real-time. This centralized database plays a crucial role in optimizing airport operations, streamlining processes, and enhancing the overall passenger experience.

**Chapter 2**

# **Synopsis**

## 2.1 Proposed System

To create a proposed Airport Management System database, we will consider the main entities involved in airport operations such as flights, airlines, passengers, bookings, and airports.

## 2.2 Objectives

The Main Objective of the work are

* **Efficient Data Management:** ADMS aims to efficiently store, organize, and manage vast amounts of data related to passengers, flights, cargo, employees, facilities, and security measures.
* **Streamlined Operations**: The system seeks to streamline airport operations by automating processes, reducing manual tasks, and improving the overall efficiency of tasks such as flight scheduling, passenger check-in, fare of flights.
* **Enhanced Passenger Experience**: ADMS aims to enhance the passenger experience by providing real-time flight information. The system mitigates the risk of inconvenience and data loss.
* **Optimized Resource Allocation**: The system helps in optimizing resource allocation by providing insights flight schedules and facility utilization, thereby maximizing operational efficiency and minimizing costs.

**Chapter 3**

# **Functional Requirements**

**Data management**: This includes functionalities for creating, storing, retrieving, updating, and deleting data within the database for airline scheduling, flight details, passenger details, ticket information etc

**Data querying:** Users should be able to query the database to retrieve specific information based on defined criteria especially regarding ticket fare, details and flight schedules.

This includes:

· Checking of ticket fare by passengers

· Retrieval of flight details like timings, gate etc

· Retrieval of schedules like destination, arrival time, departure time etc

**Data manipulation:** This involves functionalities for sorting, filtering, aggregating, and performing calculations on the stored data for creating reports.

This includes:

· Filtering data by airline

· Sorting data by flight routes

etc

**User interaction:** A user-friendly interface for users to interact with the database, such as a web interface or a command-line interface.

### Chapter 4 Detailed Design

#### 4.1 ER Diagram

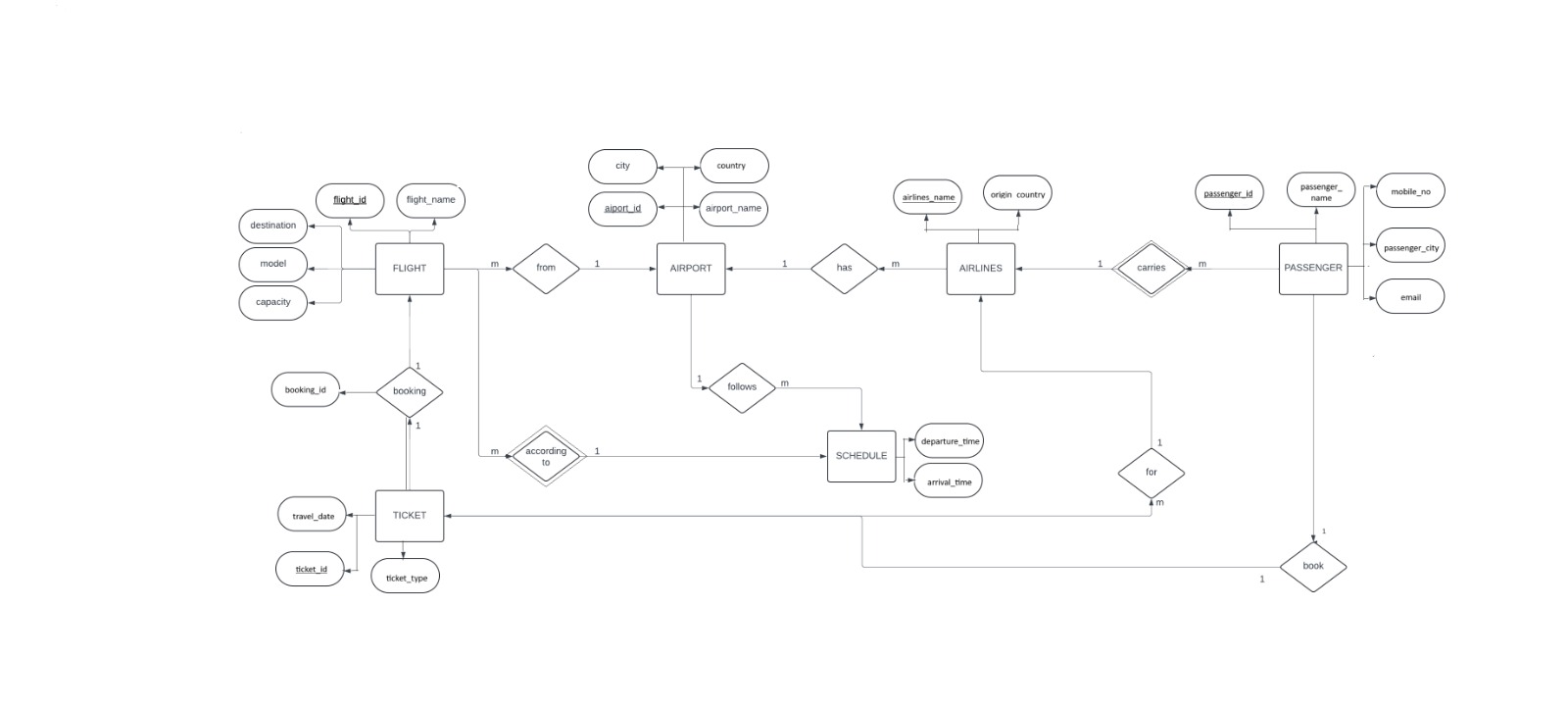


Figure 4.1 Entity Relationship Diagram

#### 4.2 Schema Diagram

**Airport(**airport-id, airport\_name, city,country)

**flight(**flight\_id, flight\_name, airlines\_name,airport\_id,destination,model,capacity)

airport\_id References airport

**Airlines(**airlines\_name,origin\_country,airport\_id )

airport-id References Airport

**Passenger(** passenger\_id, passenger\_name,mobile\_no,passenger\_city,email)

**Ticket(**ticket\_id, ticket\_type,travel\_date,passenger\_id)

passenger\_id References Passenger

**Booking(**Booking\_id,from,to,fare,ticket\_id,flight\_id)

ticket-id References ticket  
 flight\_id References flight

**Schedule(**flight\_id,airport\_id,departure\_time,arrival\_time)

flight\_id References flight, airport\_id References airport

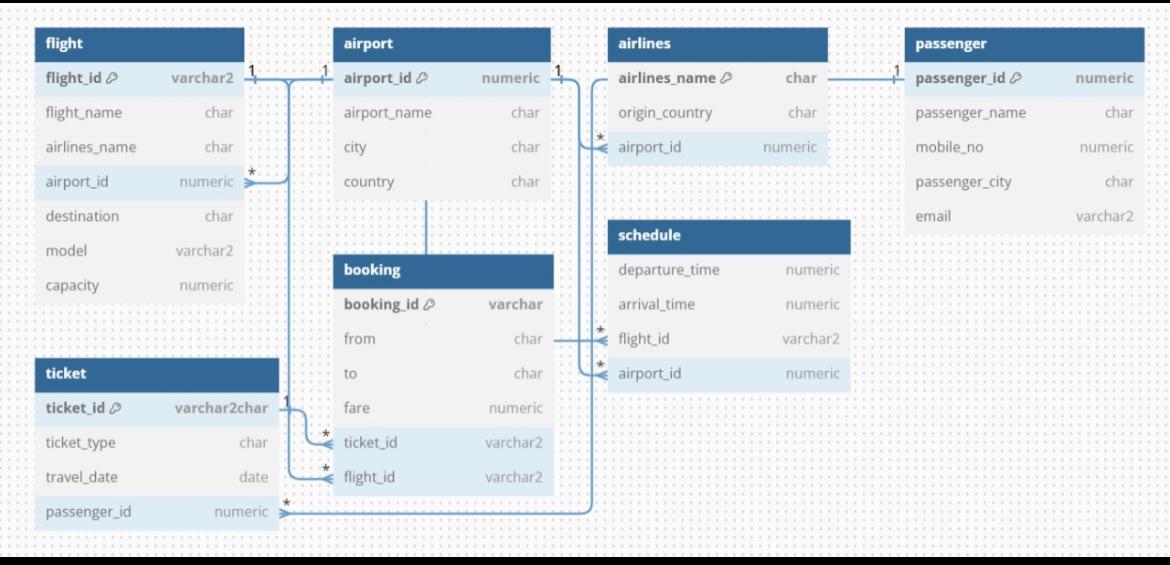


Figure 4.2 Schema Diagram

**4.3 Data Dictionary**

#### AIRPORT

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Data type (size) | Constraint | Constraint Name |
| Airport\_Id | numeric(4) | Primary Key | airport\_id\_pk |
| Airport\_Name | VARCHAR2(20) |  |  |
| city | VARCHAR2(20) |  |  |
| country | CHAR(1) |  |  |

**FLIGHT**

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Data type (size) | Constraint | Constraint Name |
| Flight\_Id | Numeric(4) | Primary Key | flight\_id\_pk |
| Flight\_Name | Varchar2(20) |  |  |
| Airlines\_Name | Varchar2(20) |  |  |
| Airport\_id | Number | Foreign Key |  |
| destination | varchar(20) |  |  |
| model | varchar(10) |  |  |
| capacity | numeric(20) |  |  |

**AIRLINES**

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Data type (size) | Constraint | Constraint Name |
| Airlines\_Name | varchar(50) | Primary Key | airlines\_name\_pk |
| origin\_country | varchar(20) |  |  |
| Airport\_Id | Numeric(4) | Foreign Key |  |

**PASSENGER**

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Data type (size) | Constraint | Constraint Name |
| Passenger\_Id | numeric(7) | Primary Key | p\_id\_pk |
| Passenger\_name | Varchar2(20) |  |  |
| Mobile\_no | Numeric(10) |  |  |
| Passenger\_City | Varchar(10) |  |  |
| email | varchar(20) |  |  |

TICKET

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Data type (size) | Constraint | Constraint Name |
| Ticket\_id | Varchar(10) | Primary Key | Ticket\_Id\_PK |
| Ticket\_Type | Varchar(5) |  |  |
| Passenger\_ID | Numeric(7) | Foreign Key |  |
| Travel Date | Date |  |  |

BOOKING

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Data type (size) | Constraint | Constraint Name |
| Booking\_id | Varchar(10) | Primary Key | booking\_ID\_PK |
| From | Varchar2(15) |  |  |
| To | Varchar(15) |  |  |
| Fare | Numeric(10) |  |  |
| Ticket\_id | Varchar(10) | Foreign Key |  |
| Flight\_id | numeric(4) | Foreign Key |  |

**SCHEDULE**

|  |  |  |  |
| --- | --- | --- | --- |
| column | Data type(size) | Constraint | Constraint Name |
| Schedule\_id | varchar(10) | Primary Key | schedule\_id\_pk |
| Departure\_time | numeric**(**8**)** |  |  |
| arrival\_ttime | numeric(8) |  |  |
| flight\_id | numeric(4) | Foreign Key |  |
| airport\_id | numeric(4) | Foreign Key |  |

##### 4.4 Relational Model Implementation

**create table airport** (

2 airport\_id numeric(4) constraint airport\_id\_pk primary key,

3 airport\_name varchar(50),

4 city varchar(20),

5 country varchar(20)

6 );

**create table flight** (

2 flight\_id numeric(4) constraint flight\_id\_pk primary key,

3 flight\_name varchar(50),

4 airlines\_name varchar(20),

5 airport\_id numeric(4) references airport,

6 destination varchar(20),

7 model varchar(10),

8 capacity numeric(20) check (capacity > 0)

9 );

**create table airlines**(

2 airlines\_name varchar(50) constraint airlines\_name\_pk primary key,

3 origin\_country varchar(20),

4 airport\_id numeric(4) references airport);

create table passenger (

2 passenger\_id numeric(7) constraint p\_id\_pk primary key,

3 passenger\_name varchar(20),

4 mobile\_no numeric(10),

5 passenger\_city varchar(10)

6 );

**create table ticket** (

2 ticket\_id varchar(10) constraint ticket\_id\_pk primary key,

3 ticket\_type varchar(5),

4 travel\_date date,

5 passenger\_id numeric(7) references passenger

6 );

**create table booking** (

2 booking\_id varchar(10) constraint booking\_id\_pk primary key,

3 from\_ varchar(15),

4 to\_ varchar(15),

5 fare numeric(10) check (fare >= 0),

6 ticket\_id varchar(10) references ticket,

7 flight\_id numeric(4) references flight

8 );

**create table schedule** (

2 schedule\_id varchar(10) constraint schedule\_id\_pk primary key,

3 departure\_time numeric(8) check (departure\_time >= 00000000 and departure\_time <= 23595959),

4 arrival\_time numeric(8) check (arrival\_time >= 00000000 and arrival\_time <= 23595959),

5 flight\_id numeric(4) references flight,

6 airport\_id numeric(4) references airport

7 );

##### 4.4.1 Additional Constraints

SQL> alter table flight add constraint flight\_id\_length check (length(flight\_id) = 4)

SQL> alter table flight add constraint airport\_id\_length check (length(airport\_id) = 4);

SQL> alter table flight add constraint model\_prefix check (model like 'airbus%' or model like 'boeing%');

SQL> alter table passenger add constraint passenger\_id\_length check (length(passenger\_id) = 4);

SQL> alter table passenger add constraint mobile\_no\_length check (length(mobile\_no) = 10);

SQL> alter table passenger add constraint email\_suffix check (email like '%@gmail.com');

SQL> alter table ticket add constraint ticket\_type\_values check (ticket\_type in ('economy', 'business', 'first class'));

SQL> alter table booking add constraint booking\_id\_length check (length(booking\_id) = 6);

**5. Implementation**

#### 5.1 Queries

**Q Retrieve all flights departing from a specific airport:**

SQL> SELECT f.flight\_id, f.flight\_name, f.destination, f.model, f.capacity

2 FROM flight f

3 WHERE f.airport\_id = (SELECT airport\_id FROM airport WHERE airport\_name = 'Heathrow');

FLIGHT\_ID FLIGHT\_NAME

---------- --------------------------------------------------

DESTINATION MODEL CAPACITY

-------------------- -------------------- ----------

8901 SKY123

London Boeing 777 300

5678 SKY789

London Airbus A320 200

**Q List all bookings made by a specific passenger:**

SQL> SELECT b.booking\_id, b.from\_, b.to\_, b.fare

2 FROM booking b

3 JOIN ticket t ON b.ticket\_id = t.ticket\_id

4 JOIN passenger p ON t.passenger\_id = p.passenger\_id

5 WHERE p.passenger\_name = 'John Smith';

BOOKING\_ID FROM\_ TO\_ FARE

---------- --------------- --------------- ----------

AB1234 JFK Heathrow 5000

**Q Find all flights operated by a specific airline:**

SQL> SELECT f.flight\_id, f.flight\_name, f.destination, f.model, f.capacity

2 FROM flight f

3 JOIN airlines a ON f.airlines\_name = a.airlines\_name

4 WHERE a.airlines\_name = 'Emirates';

FLIGHT\_ID FLIGHT\_NAME

---------- --------------------------------------------------

DESTINATION MODEL CAPACITY

-------------------- -------------------- ----------

1234 AIR101

Dubai Airbus A380 300

**Q Retrieve the number of bookings for each flight:**

SQL> SELECT f.flight\_id, f.flight\_name, COUNT(b.booking\_id) AS num\_bookings

2 FROM flight f

3 LEFT JOIN booking b ON f.flight\_id = b.flight\_id

4 GROUP BY f.flight\_id, f.flight\_name;

FLIGHT\_ID FLIGHT\_NAME NUM\_BOOKINGS

---------- -------------------------------------------------- ------------

1234 AIR101 1

7891 FLY123 3

5678 SKY789 2

3456 JET456 2

8765 WINGS202 2

8901 SKY123 0

2345 FLY321 0

**Q Count the number of flights departing from each airport:**

SQL> SELECT airport\_id, COUNT(flight\_id) AS num\_flights FROM flight GROUP BY airport\_id;

AIRPORT\_ID NUM\_FLIGHTS

---------- -----------

1234 2

7890 1

5678 2

9876 1

3456 1

**Q List all bookings with their corresponding passengers and flights:**

SQL> SELECT b.booking\_id, b.from\_, b.to\_, b.fare, p.passenger\_name, f.flight\_name

2 FROM booking b

3 JOIN ticket t ON b.ticket\_id = t.ticket\_id

4 JOIN passenger p ON t.passenger\_id = p.passenger\_id

5 JOIN flight f ON b.flight\_id = f.flight\_id;

BOOKING\_ID FROM\_ TO\_ FARE PASSENGER\_NAME

---------- --------------- --------------- ---------- --------------------

FLIGHT\_NAME

--------------------------------------------------

GH4567 Dubai Intl Heathrow 7000 Maria Garcia

AIR101

AB1234 JFK Heathrow 5000 John Smith

FLY123

CD2345 Heathrow Narita 10000 Emily Johnson

SKY789

BOOKING\_ID FROM\_ TO\_ FARE PASSENGER\_NAME

---------- --------------- --------------- ---------- --------------------

FLIGHT\_NAME

--------------------------------------------------

EF3456 Narita JFK 8000 Satoshi Tanaka

JET456

IJ5678 Tegel Narita 9000 Wei Chen

WINGS202

KL6789 Narita JFK 12000 Ahmed Khan

FLY123

BOOKING\_ID FROM\_ TO\_ FARE PASSENGER\_NAME

---------- --------------- --------------- ---------- --------------------

FLIGHT\_NAME

--------------------------------------------------

MN7890 JFK Dubai Intl 15000 Sophie Dupont

SKY789

ST0123 Tegel Narita 11000 Kim Ji-hyun

JET456

UV1234 JFK Heathrow 5500 David Smith

FLY123

**Q Find the total revenue generated by each airline from ticket sales:**

SQL> SELECT a.airlines\_name, SUM(b.fare) AS total\_revenue

2 FROM booking b

3 JOIN flight f ON b.flight\_id = f.flight\_id

4 JOIN airlines a ON f.airlines\_name = a.airlines\_name

5 GROUP BY a.airlines\_name;

AIRLINES\_NAME TOTAL\_REVENUE

-------------------------------------------------- -------------

Japan Airlines 19000

British Airways 25000

Emirates 7000

Delta Airlines 22500

Lufthansa 18500

**Q List all flights with their corresponding airlines and destinations:**

SQL> SELECT f.\*, a.airlines\_name, a.origin\_country

2 FROM flight f

3 JOIN airlines a ON f.airlines\_name = a.airlines\_name;

FLIGHT\_ID FLIGHT\_NAME

---------- --------------------------------------------------

AIRLINES\_NAME AIRPORT\_ID DESTINATION MODEL

-------------------- ---------- -------------------- --------------------

CAPACITY AIRLINES\_NAME

---------- --------------------------------------------------

ORIGIN\_COUNTRY

--------------------

7891 FLY123

Delta Airlines 1234 New York Boeing 737

150 Delta Airlines

USA

FLIGHT\_ID FLIGHT\_NAME

---------- --------------------------------------------------

AIRLINES\_NAME AIRPORT\_ID DESTINATION MODEL

-------------------- ---------- -------------------- --------------------

CAPACITY AIRLINES\_NAME

---------- --------------------------------------------------

ORIGIN\_COUNTRY

--------------------

3456 JET456

Japan Airlines 7890 Tokyo Boeing 787

180 Japan Airlines

Japan

FLIGHT\_ID FLIGHT\_NAME

---------- --------------------------------------------------

AIRLINES\_NAME AIRPORT\_ID DESTINATION MODEL

-------------------- ---------- -------------------- --------------------

CAPACITY AIRLINES\_NAME

---------- --------------------------------------------------

ORIGIN\_COUNTRY

--------------------

8765 WINGS202

Lufthansa 3456 Berlin Boeing 747

250 Lufthansa

Germany

FLIGHT\_ID FLIGHT\_NAME

---------- --------------------------------------------------

AIRLINES\_NAME AIRPORT\_ID DESTINATION MODEL

-------------------- ---------- -------------------- --------------------

CAPACITY AIRLINES\_NAME

---------- --------------------------------------------------

ORIGIN\_COUNTRY

--------------------

8901 SKY123

British Airways 5678 London Boeing 777

300 British Airways

UK

FLIGHT\_ID FLIGHT\_NAME

---------- --------------------------------------------------

AIRLINES\_NAME AIRPORT\_ID DESTINATION MODEL

-------------------- ---------- -------------------- --------------------

CAPACITY AIRLINES\_NAME

---------- --------------------------------------------------

ORIGIN\_COUNTRY

--------------------

5678 SKY789

British Airways 5678 London Airbus A320

200 British Airways

UK

FLIGHT\_ID FLIGHT\_NAME

---------- --------------------------------------------------

AIRLINES\_NAME AIRPORT\_ID DESTINATION MODEL

-------------------- ---------- -------------------- --------------------

CAPACITY AIRLINES\_NAME

---------- --------------------------------------------------

ORIGIN\_COUNTRY

--------------------

1234 AIR101

Emirates 9876 Dubai Airbus A380

300 Emirates

UAE

FLIGHT\_ID FLIGHT\_NAME

---------- --------------------------------------------------

AIRLINES\_NAME AIRPORT\_ID DESTINATION MODEL

-------------------- ---------- -------------------- --------------------

CAPACITY AIRLINES\_NAME

---------- --------------------------------------------------

ORIGIN\_COUNTRY

--------------------

2345 FLY321

Delta Airlines 1234 New York Airbus A330

180 Delta Airlines

USA

**Q Find the total number of bookings for each destination city:**

SQL> SELECT f.destination, COUNT(b.booking\_id) AS num\_bookings

2 FROM booking b

3 JOIN flight f ON b.flight\_id = f.flight\_id

4 GROUP BY f.destination;

DESTINATION NUM\_BOOKINGS

-------------------- ------------

London 2

Tokyo 2

New York 3

Dubai 1

Berlin 2

#### 5.2 Stored Procedures

**Procedure to print all flights for a given airport:**

create or replace procedure print\_flights\_from\_airport(p\_airport\_name in varchar2) is

begin

for rec in (select f.flight\_id, f.flight\_name, f.destination, f.model, f.capacity

from flight f

where f.airport\_id = (select airport\_id from airport where airport\_name = p\_airport\_name))

loop

dbms\_output.put\_line('flight id: ' || rec.flight\_id || ', flight name: ' || rec.flight\_name ||

', destination: ' || rec.destination || ', model: ' || rec.model ||

', capacity: ' || rec.capacity);

end loop;

exception

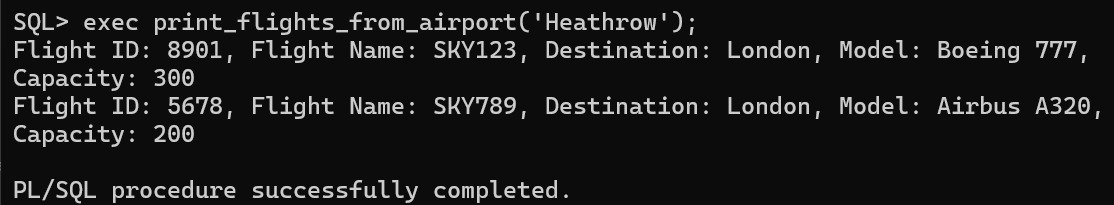
when no\_data\_found then

dbms\_output.put\_line('no flights found for the airport ' || p\_airport\_name);

when others then

dbms\_output.put\_line('error: ' || sqlerrm);

end;



**Procedure to print all bookings for a given passenger:**

create or replace procedure print\_bookings\_for\_passenger(p\_passenger\_name in varchar2) is

begin

for rec in (select b.booking\_id, b.from\_, b.to\_, b.fare

from booking b

join ticket t on b.ticket\_id = t.ticket\_id

join passenger p on t.passenger\_id = p.passenger\_id

where p.passenger\_name = p\_passenger\_name)

loop

dbms\_output.put\_line('booking id: ' || rec.booking\_id || ', from: ' || rec.from\_ ||

', to: ' || rec.to\_ || ', fare: ' || rec.fare);

end loop;

exception

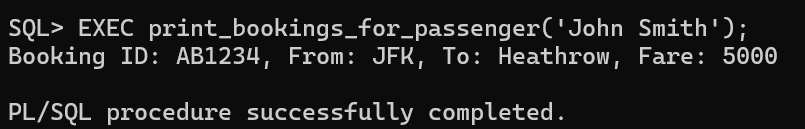
when no\_data\_found then

dbms\_output.put\_line('no bookings found for passenger ' || p\_passenger\_name);

when others then

dbms\_output.put\_line('error: ' || sqlerrm);

end;



**5.3 Stored Functions**

**Calculate Total Fare for a Passenger:**

create or replace function total\_fare\_for\_passenger(p\_passenger\_id in numeric)

return numeric is

v\_total\_fare numeric := 0;

begin

select sum(b.fare) into v\_total\_fare

from booking b

join ticket t on b.ticket\_id = t.ticket\_id

where t.passenger\_id = p\_passenger\_id;

return v\_total\_fare;

exception

when no\_data\_found then

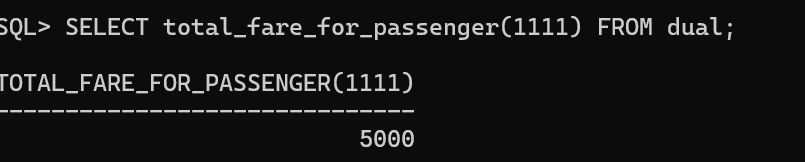
return 0;

when others then

raise;

end;

/



**Function to return total revenue of Specific airline:**

create or replace function total\_revenue\_for\_airline(p\_airline\_name in varchar2)

return number is

v\_total\_revenue number := 0;

begin

select sum(b.fare) into v\_total\_revenue

from booking b

join flight f on b.flight\_id = f.flight\_id

join airlines a on f.airlines\_name = a.airlines\_name

where a.airlines\_name = p\_airline\_name;

return v\_total\_revenue;

exception

when no\_data\_found then

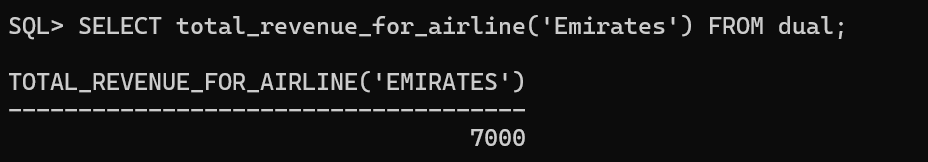
return 0;

when others then

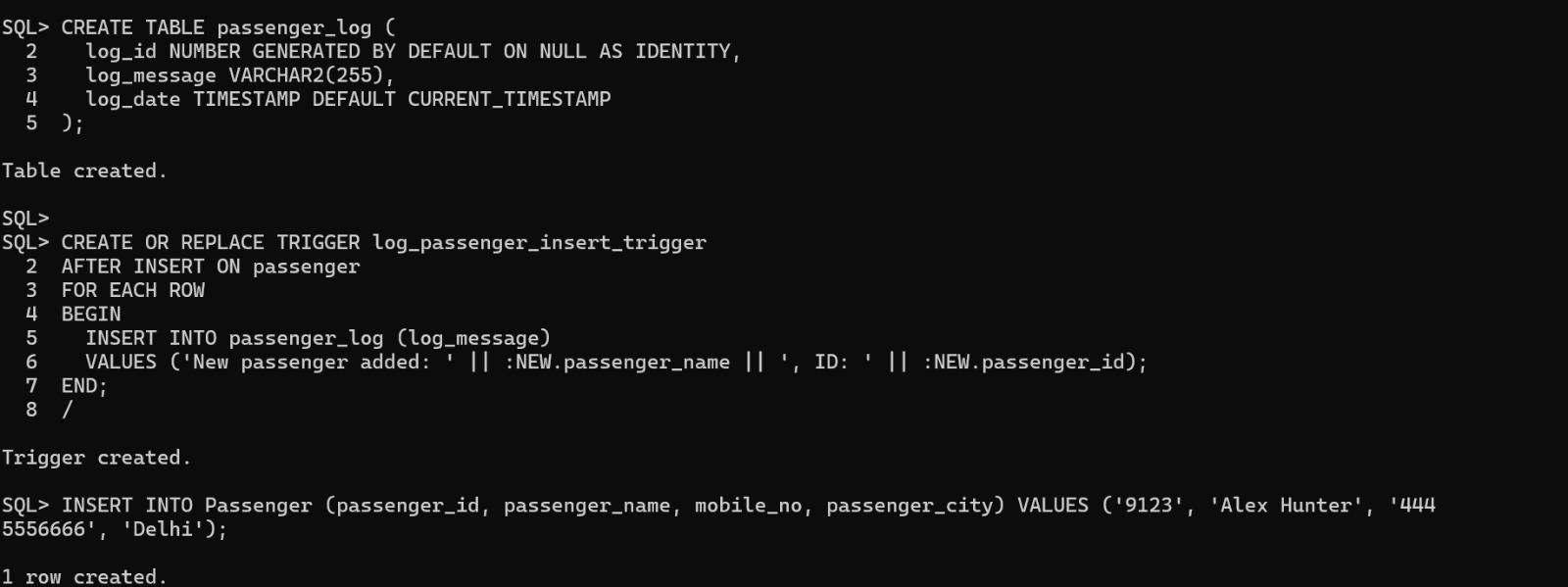
raise;

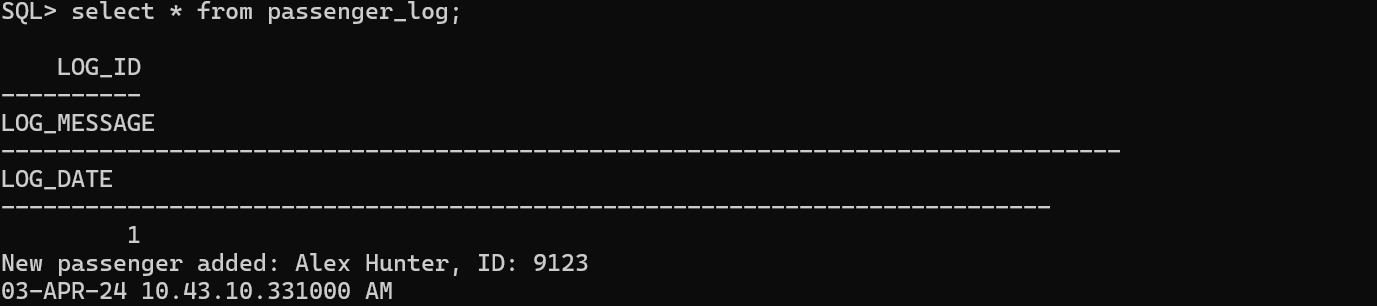
end;

/

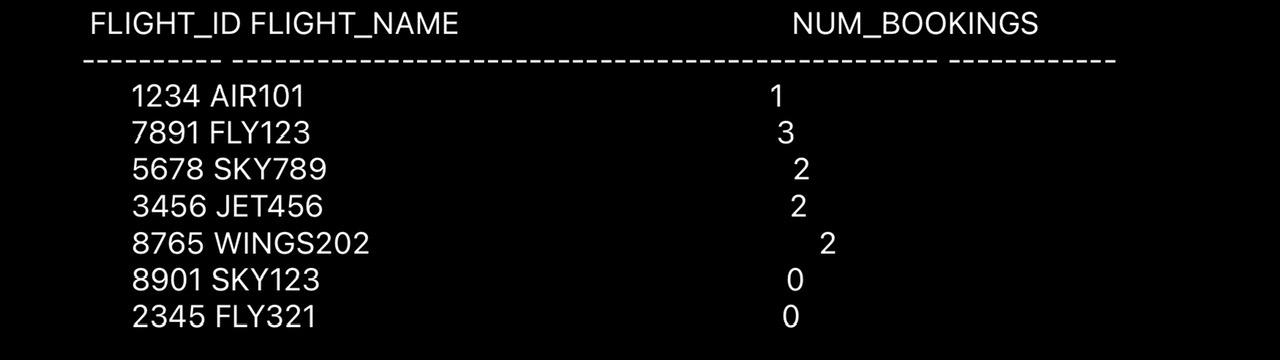
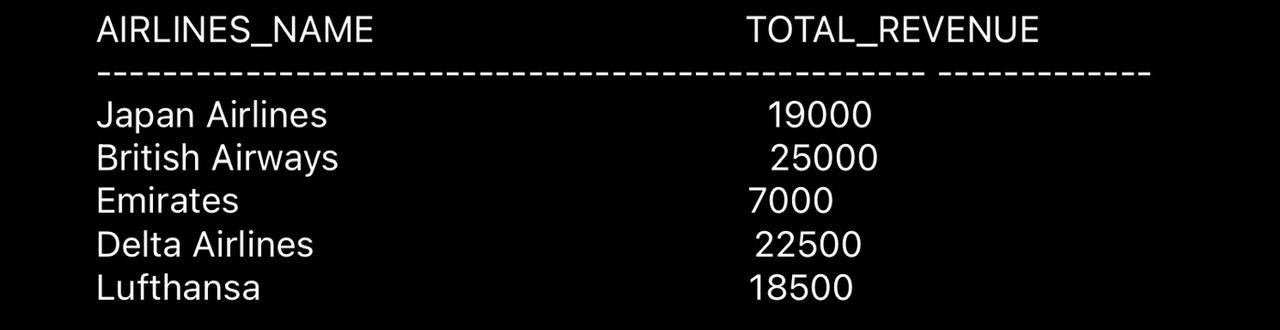


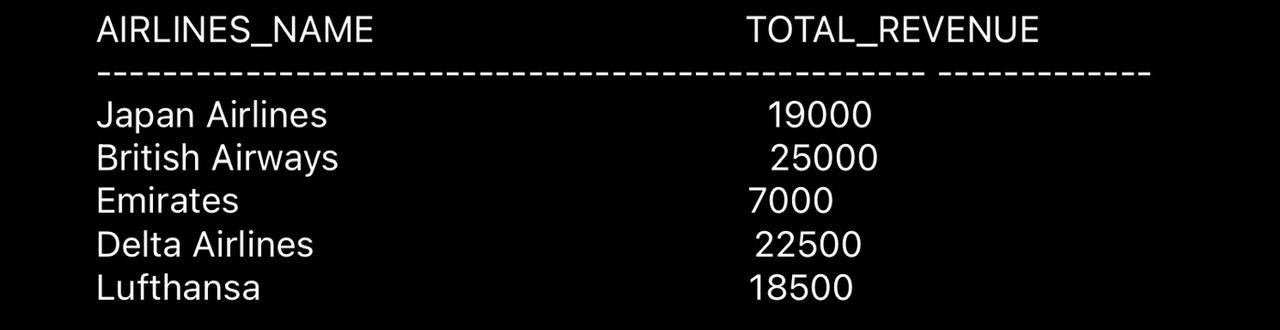
**5.4 Triggers**





**6. Result**

** **

****

### 7.Conclusion and Future Work

**7.1 Conclusion**

In conclusion, the implementation of customer details integrated within the airport database management system, complemented by the ease of accessing information through multiple primary keys, establishes a robust system for efficient airport operations and passenger service. By ensuring data consistency and providing identifiers such as Passenger ID, the system mitigates the risk of inconvenience and data loss. The system can identifying common user queries such as flight schedule, details, fares etc which not only enhances passenger convenience but also maximizes revenue potential and operational efficiency for airports.

#### 7.2 Scope for future work

The system has scope for expansion by leveraging insights from passenger behaviour and trends can provide additional enhancements in our airport database management system. By identifying trends such as peak travel seasons and popular routes, we can optimize resources and services to meet passenger demand effectively. For example, we can allocate additional staff and security measures during peak periods to ensure smooth operations and enhance passenger satisfaction. Furthermore, implementing dynamic pricing strategies for airline tickets and airport services during high-demand periods can maximize revenue generation.

Each Team Member Contribution:

|  |  |
| --- | --- |
| Team Member | Contribution |
| Hemang Malik | Inserted data in tables, formatted the report and wrote PL/SQL functions. |
| Kshitij Sohoni | Created tables and Data Dictionary |
| M Dhanush Varma | Intro, Report and Conclusion |
| Akhil Ajai Kumar | Wrote Functional Requirements |
| Sreenivasa Sai Krishna Kartik | Established Relationships and ER Diagram |