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
Assistant Professor-Senior
DSCA, MIT

Archana H
Assistant Professor-Senior
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
Assistant Professor-Senior
DSCA, MIT

Archana H
Assistant Professor-Senior
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

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

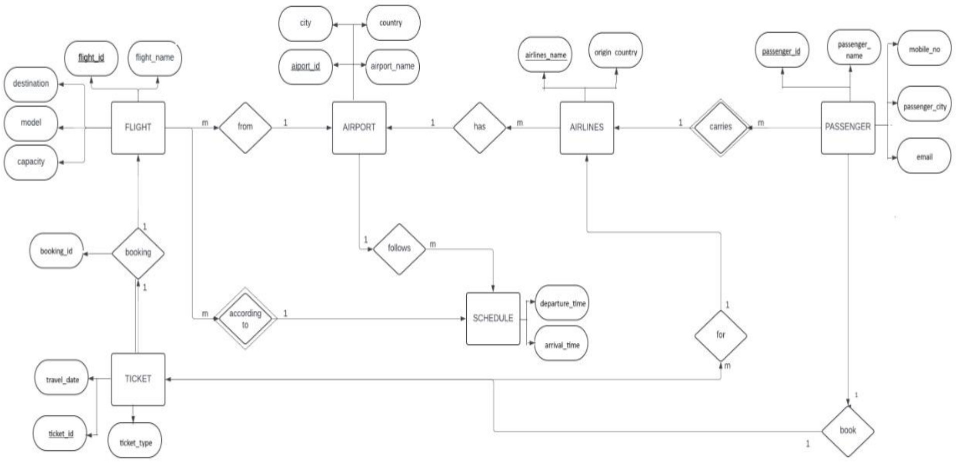


Figure 4.1 Entity Relationship Diagram

4.2 Schema Diagram

Airport(<u>airport-id</u>, airport_name, city,country)
flight(<u>flight_id,</u> flight_name, airlines_name,airport_id,destination,model,capacity)

airport_id References airport

Airlines(<u>airlines_name,origin_country,airport_id</u>)
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

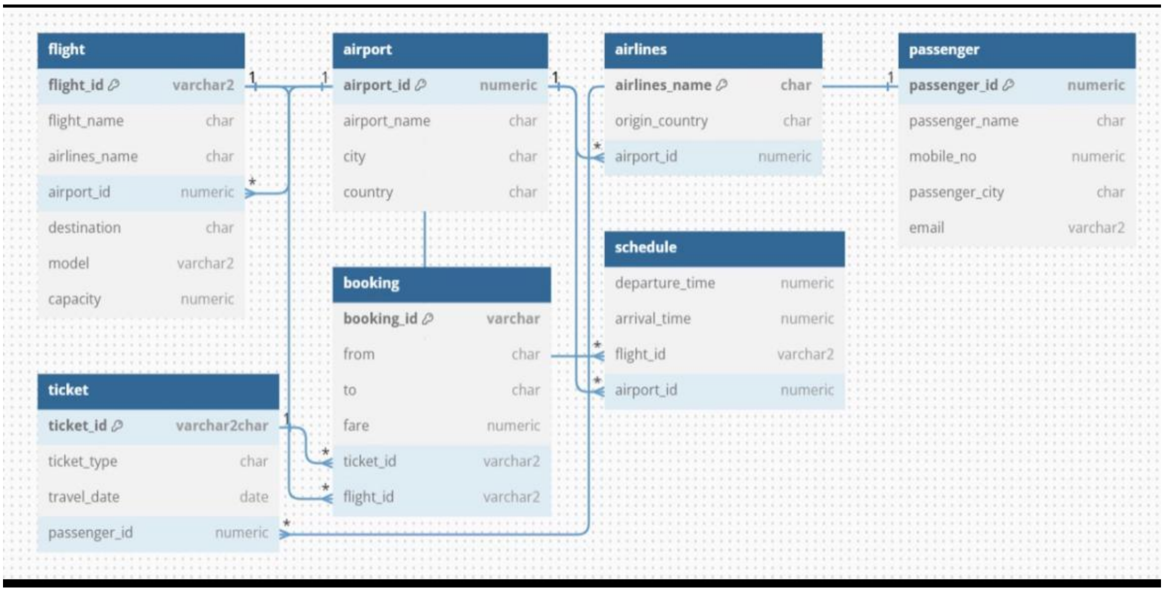


Figure 4.2 Schema Diagram

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 (
    airport_id numeric(4) constraint airport_id_pk primary key,
    airport_name varchar(50),
    city varchar(20),
    country varchar(20)
);
create table flight (
    flight_id numeric(4) constraint flight_id_pk primary key,
    flight_name varchar(50),
    airlines_name varchar(20),
    airport_id numeric(4) references airport,
    destination varchar(20),
    model varchar(10),
    capacity numeric(20) check (capacity > 0)
);
create table airlines(
 airlines_name varchar(50) constraint airlines_name_pk primary key,
 origin_country varchar(20),
 airport_id numeric(4) references airport);
create table passenger (
    passenger_id numeric(7) constraint p_id_pk primary key,
    passenger_name varchar(20),
    mobile_no numeric(10),
    passenger_city varchar(10)
    );
create table ticket (
    ticket_id varchar(10) constraint ticket_id_pk primary key,
    ticket_type varchar(5),
    travel_date date,
    passenger_id numeric(7) references passenger
    );
create table booking (
    booking_id varchar(10) constraint booking_id_pk primary key,
    from_ varchar(15),
    to_ varchar(15),
    fare numeric(10) check (fare >= 0),
    ticket_id varchar(10) references ticket,
    flight_id numeric(4) references flight
    );
create table schedule (
    schedule_id varchar(10) constraint schedule_id_pk primary key,
    departure_time numeric(8) check (departure_time >= 00000000 and departure_time <= 23595959),
    arrival_time numeric(8) check (arrival_time >= 00000000 and arrival_time <= 23595959),
    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
```

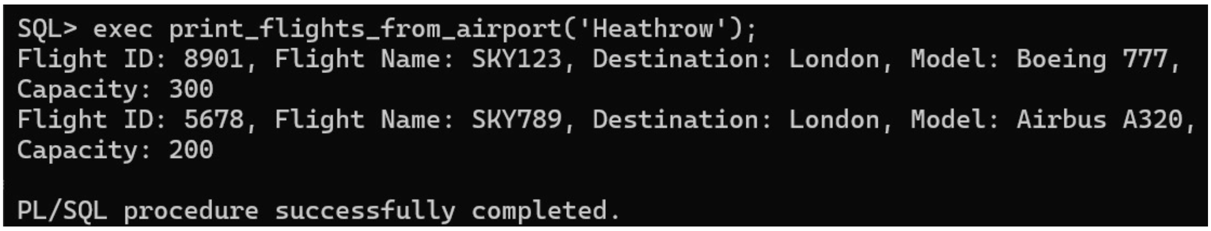
Q Find the total number of bookings for each destination city:

```
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:

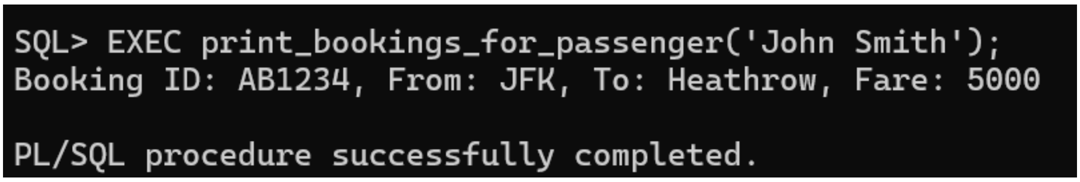
```
create or replace procedure print_flights_from_airport(p_airport_name in varchar2) is
begin
  for rec in (select f.flight_id, f.flight_name, f.destination, f.model, f.capacity
        from flight f
        where f.airport_id = (select airport_id from airport where airport_name =
p_airport_name))
  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;
```



```
SQL> exec print_flights_from_airport('Heathrow');
Flight ID: 8901, Flight Name: SKY123, Destination: London, Model: Boeing 777,
Capacity: 300
Flight ID: 5678, Flight Name: SKY789, Destination: London, Model: Airbus A320,
Capacity: 200

PL/SQL procedure successfully completed.
```

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;
```

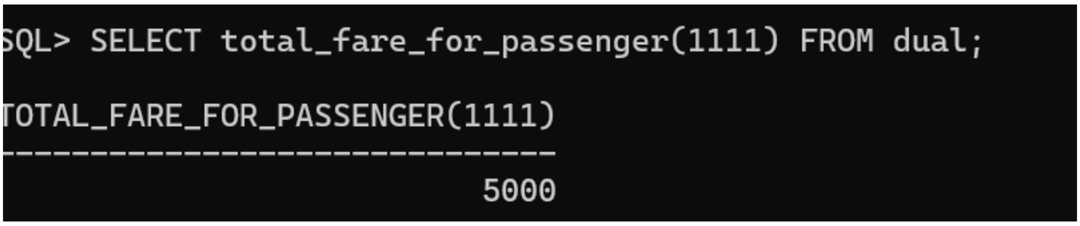


```
SQL> EXEC print_bookings_for_passenger('John Smith');
Booking ID: AB1234, From: JFK, To: Heathrow, Fare: 5000

PL/SQL procedure successfully completed.
```

5.3 Stored Functions

Calculate Total Fare for a Passenger:

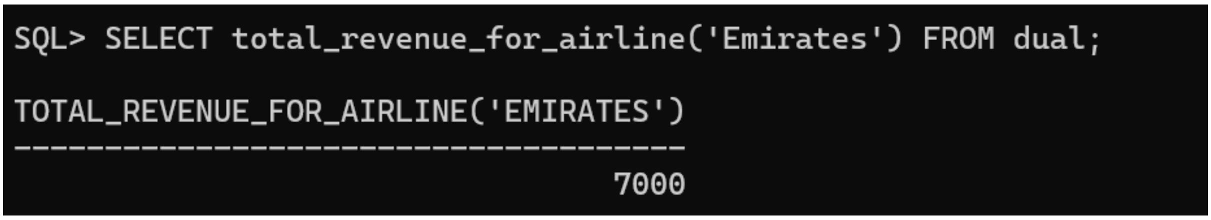
```
create or replace function total_fare_for_passenger(p_passenger_id in numeric)
return numeric is
  v_total_fare numeric := 0;
begin
  select sum(b.fare) into v_total_fare
  from booking b
  join ticket t on b.ticket_id = t.ticket_id
  where t.passenger_id = p_passenger_id;


  return v_total_fare;
exception
  when no_data_found then
    return 0;
  when others then
    raise;
end;
/
```



```
SQL> SELECT total_fare_for_passenger(1111) FROM dual;

TOTAL_FARE_FOR_PASSENGER(1111)
------------------------------
                          5000
```

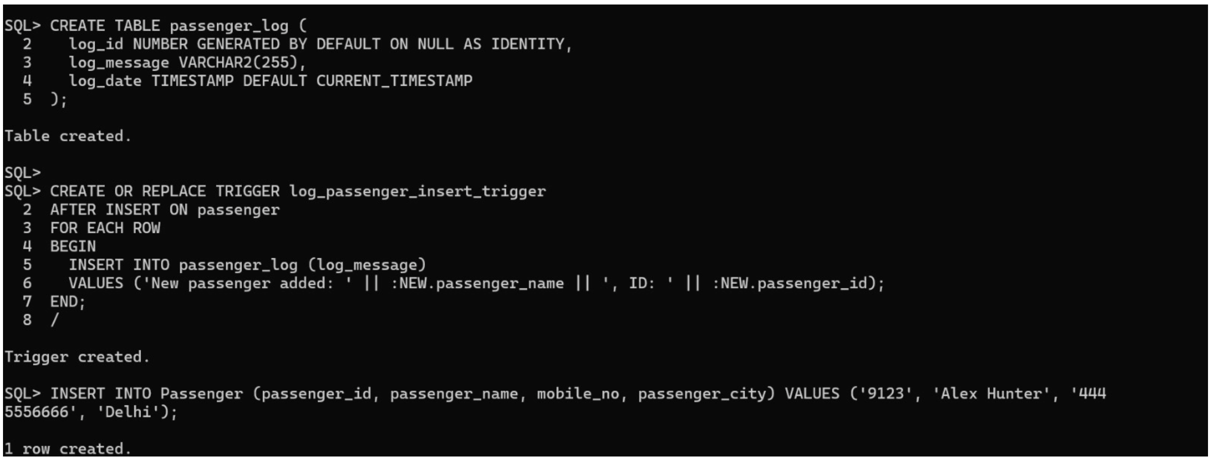
Function to return total revenue of Specific airline:

```
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;
/
```

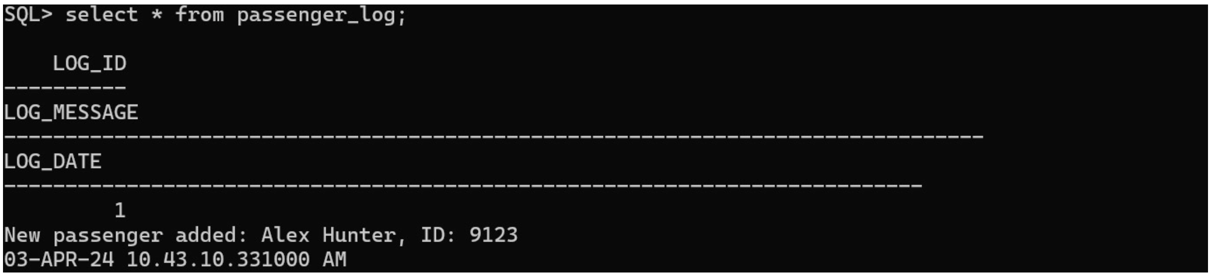


```
SQL> SELECT total_revenue_for_airline('Emirates') FROM dual;

TOTAL_REVENUE_FOR_AIRLINE('EMIRATES')
-------------------------------------
                                 7000
```

5.4 Triggers

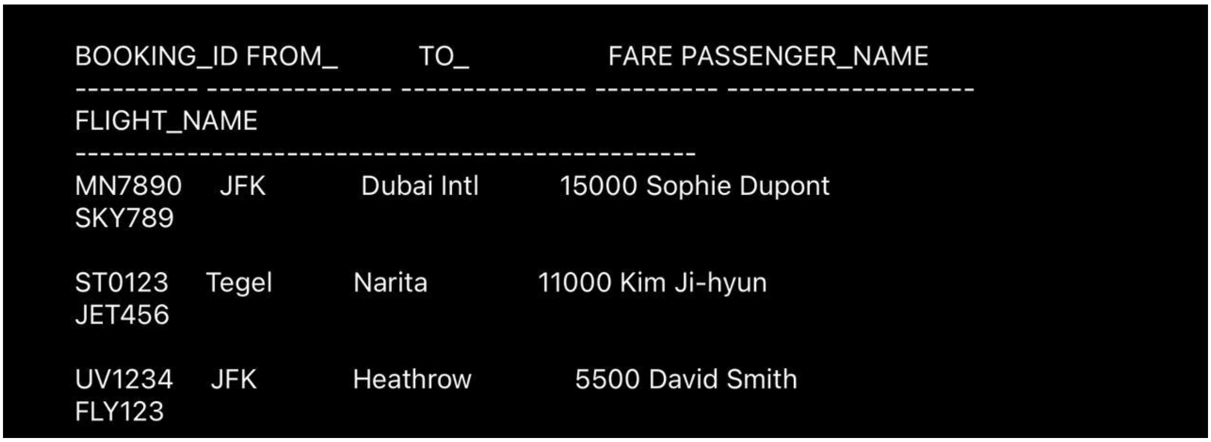


```
SQL> CREATE TABLE passenger_log (
       log_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
       log_message VARCHAR2(255),
       log_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP
     );

Table created.

SQL>
SQL> CREATE OR REPLACE TRIGGER log_passenger_insert_trigger
     AFTER INSERT ON passenger
     FOR EACH ROW
     BEGIN
       INSERT INTO passenger_log (log_message)
       VALUES ('New passenger added: ' || :NEW.passenger_name || ', ID: ' || :NEW.passenger_id);
     END;
     /

Trigger created.

SQL> INSERT INTO Passenger (passenger_id, passenger_name, mobile_no, passenger_city) VALUES ('9123', 'Alex Hunter', '444
5556666', 'Delhi');

1 row created.
```

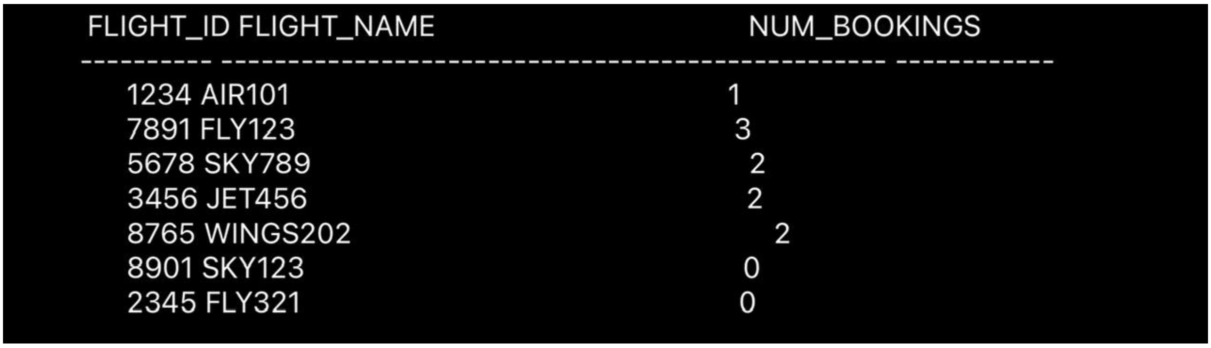


```
SQL> select * from passenger_log;

    LOG_ID
----------
LOG_MESSAGE
--------------------------------------------------------------------------------
LOG_DATE
---------------------------------------------------------------------------
         1
New passenger added: Alex Hunter, ID: 9123
03-APR-24 10.43.10.331000 AM
```

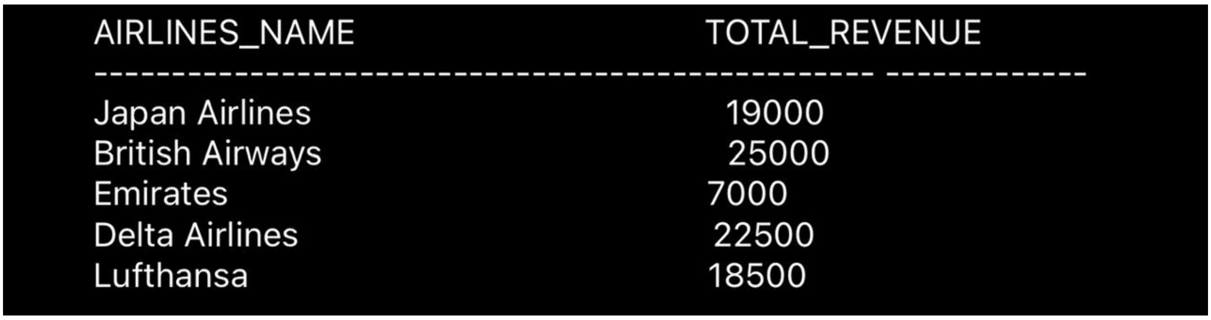
6. Result



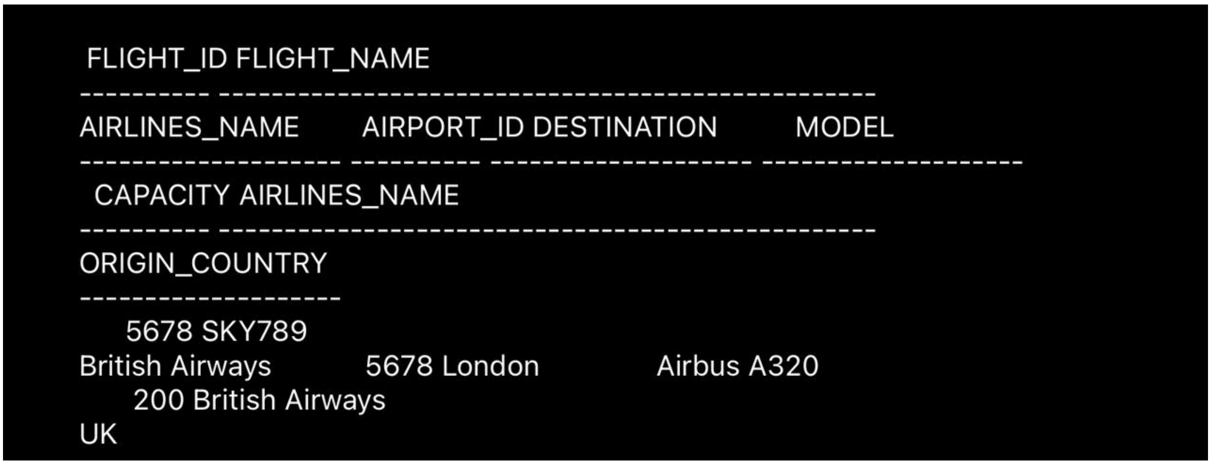
```
BOOKING_ID FROM_           TO_                  FARE PASSENGER_NAME
---------- --------------- --------------- ---------- --------------------
FLIGHT_NAME
--------------------------------------------------
MN7890     JFK             Dubai Intl           15000 Sophie Dupont
SKY789

ST0123     Tegel           Narita               11000 Kim Ji-hyun
JET456

UV1234     JFK             Heathrow              5500 David Smith
FLY123
```



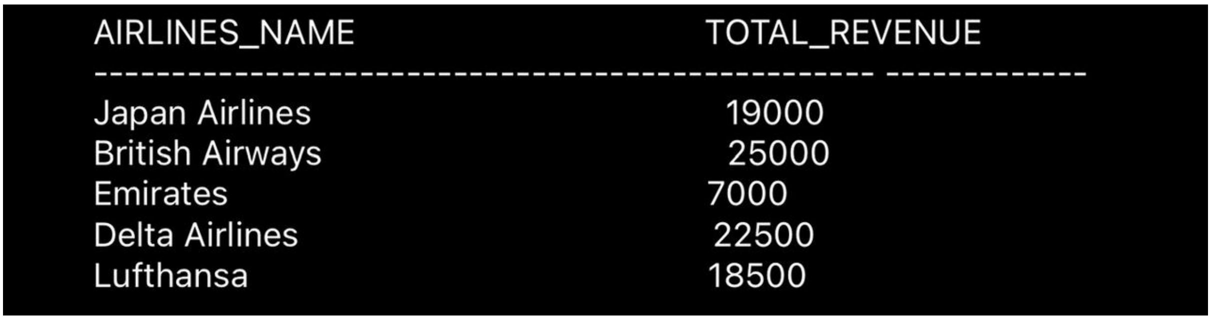
```
 FLIGHT_ID FLIGHT_NAME                                        NUM_BOOKINGS
---------- -------------------------------------------------- ------------
      1234 AIR101                                                        1
      7891 FLY123                                                        3
      5678 SKY789                                                        2
      3456 JET456                                                        2
      8765 WINGS202                                                      2
      8901 SKY123                                                        0
      2345 FLY321                                                        0
```



```
AIRLINES_NAME                                        TOTAL_REVENUE
---------------------------------------------------- -------------
Japan Airlines                                         19000
British Airways                                        25000
Emirates                                             7000
Delta Airlines                                        22500
Lufthansa                                            18500
```



```
 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
```



```
AIRLINES_NAME                                        TOTAL_REVENUE
---------------------------------------------------- -------------
Japan Airlines                                         19000
British Airways                                        25000
Emirates                                             7000
Delta Airlines                                        22500
Lufthansa                                            18500
```

```
 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
```

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