AIRPORT MANAGEMENT SYSTEM

TABLE OF CONTENTS

INTRODUCTION

Our DBMS project focuses on the development of an advanced Airport Management System (AMS) tailored to meet the diverse needs of modern airports. By leveraging the power of database management systems, our solution aims to streamline airport operations, enhance resource utilisation, and improve overall efficiency. Our Airport Management System offers a comprehensive suite of features designed to address the challenges faced in airport operation and optimise airport performance.

It will be useful in:

- Keeping track of flight status and schedules for travellers to know about flight cancellations and delays.
- Regulating retrieval of information regarding flights and bookings.

Through this project, by harnessing the capabilities of modern database technologies, we aspire to contribute to the advancement of airport management practices and enhance the overall travel experience for passengers.

REQUIREMENT ANALYSIS

Functional Requirements

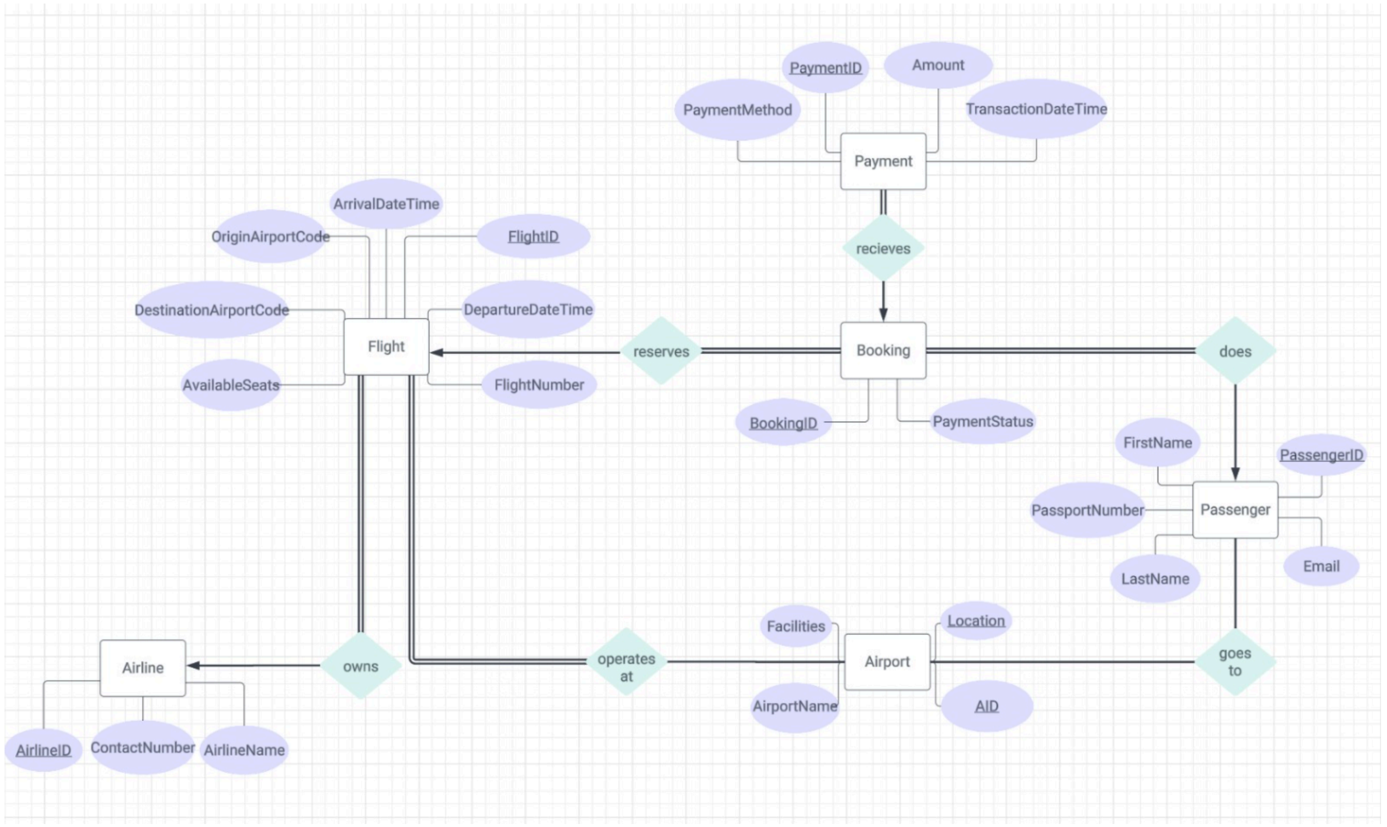
- **Flight Management**: The system should allow inserting, updating, and deleting flight information such as flight number, departure, and arrival times.
- **Passenger Management**: The system should store passenger information including adding, updating, and deleting passenger details such as name, email, and passport number.
- **Airport Management**: Airport information like name, location, and facilities should be managed within the system. This includes adding new airports, updating existing ones, and removing outdated entries.
- **Airline Management:** The system should enable the management of airline details such as name, contact number, and operating region. It should support adding, updating, and removing airlines from the database.
- **Booking Management:** The system should allow passengers to view bookings for flights.It should also allow them to cancel bookings.
- **Data Integrity and Consistency:** The system should maintain data integrity and consistency across all tables. This includes enforcing referential integrity through foreign key constraints and ensuring that updates and deletions follow predefined rules to prevent data inconsistencies.
- **User Authentication and Authorization:** The system should provide user authentication mechanisms to ensure that only authorised users can access and modify sensitive information. Different levels of access rights may be necessary for administrators, and passengers.

By fulfilling these functional requirements, the system will effectively manage flight bookings, passenger information, and payment processing while ensuring data integrity and security.

Non-Functional Requirements

- **Performance**: The system should be responsive and able to handle multiple requests without significant delays, ensuring quick access to flight, passenger, airport, and booking information.
- **Scalability**: The system should be able to accommodate an increasing number of users, flights, and data without compromising performance. It should scale up seamlessly as the airport operations expand.
- **Reliability**: The system should be dependable and available 24/7 to handle critical operations such as flight scheduling, passenger check-ins, and booking management without unexpected downtime.
- **Usability:** The system should be intuitive and easy to use for both airport staff and passengers. It should have a user-friendly interface with clear navigation and minimal training requirements.
- **Availability**: The system should have a high level of availability, ensuring that essential functions like flight status updates, booking modifications, and passenger check-ins are always accessible, even during peak times or system maintenance.

ER DIAGRAM


PaymentID
Amount
PaymentMethod
TransactionDateTime
Payment
recieves
ArrivalDateTime
OriginAirportCode
FlightID
DestinationAirportCode
DepartureDateTime
Flight
reserves
Booking
does
AvailableSeats
FlightNumber
BookingID
PaymentStatus
FirstName
PassengerID
PassportNumber
Passenger
LastName
Email
Facilities
Location
Airline
owns
operates at
Airport
goes to
AirportName
AID
AirlineID
ContactNumber
AirlineName

NORMALISATION

BEFORE-

-- Combined Flight and Airport Information

```
CREATE TABLE FlightInfo (
   FlightID INT PRIMARY KEY,
   FlightNumber VARCHAR(20) UNIQUE,
   DepartureDateTime TIMESTAMP,
   ArrivalDateTime TIMESTAMP,,
   OriginAirportCode VARCHAR(3),
   DestinationAirportCode VARCHAR(3),
   AvailableSeats INT,
   AirlineID INT,
   AirlineName VARCHAR(100),
   ContactNumber VARCHAR(20),
   OperatingRegion VARCHAR(100),
   AirportName VARCHAR(100),
   Location VARCHAR(255),
   Facilities VARCHAR(255),
   FOREIGN KEY (OriginAirportCode) REFERENCES Airport(AirportCode),
   FOREIGN KEY (DestinationAirportCode) REFERENCES Airport(AirportCode),
   FOREIGN KEY (AirlineID) REFERENCES Airline(AirlineID)
);
```

-- Combined Passenger and Booking Information

```
CREATE TABLE PassengerBooking (
   BookingID INT PRIMARY KEY,
   FlightID INT,
   PassengerID INT,
   PaymentStatus VARCHAR(20),
   FirstName VARCHAR(50),
   LastName VARCHAR(50),
   Email VARCHAR(100),
   PassportNumber VARCHAR(20),
   FOREIGN KEY (FlightID) REFERENCES Flight(FlightID),
   FOREIGN KEY (PassengerID) REFERENCES Passenger(PassengerID)
);
```

-- Payment Table (Unchanged)

```
CREATE TABLE Payment (
   PaymentID INT PRIMARY KEY,
   BookingID INT UNIQUE,
   PaymentMethod VARCHAR(50),
   Amount DECIMAL(10, 2),
   TransactionDateTime DATETIME,
   FOREIGN KEY (BookingID) REFERENCES Booking(BookingID)
);
```

AFTER-

```
-- Flight Table
CREATE TABLE Flight (
    FlightID INT PRIMARY KEY,
    FlightNumber VARCHAR(20) UNIQUE,
    DepartureDateTime DATETIME,
    ArrivalDateTime DATETIME,
    OriginAirportCode VARCHAR(3),
    DestinationAirportCode VARCHAR(3),
    AvailableSeats INT,
    AirlineID INT,
    FOREIGN KEY (OriginAirportCode) REFERENCES Airport(AirportCode),
    FOREIGN KEY (DestinationAirportCode) REFERENCES Airport(AirportCode),
    FOREIGN KEY (AirlineID) REFERENCES Airline(AirlineID)
);

-- Passenger Table
CREATE TABLE Passenger (
    PassengerID INT PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Email VARCHAR(100),
    PassportNumber VARCHAR(20)
);

-- Booking Table
CREATE TABLE Booking (
    BookingID INT PRIMARY KEY,
    FlightID INT,
    PassengerID INT,
    PaymentStatus VARCHAR(20),
    FOREIGN KEY (FlightID) REFERENCES Flight(FlightID),
    FOREIGN KEY (PassengerID) REFERENCES Passenger(PassengerID)
);

-- Payment Table (Unchanged)
CREATE TABLE Payment (
    PaymentID INT PRIMARY KEY,
    BookingID INT UNIQUE,
    PaymentMethod VARCHAR(50),
    Amount DECIMAL(10, 2),
    TransactionDateTime DATETIME,
    FOREIGN KEY (BookingID) REFERENCES Booking(BookingID)
);

-- Airport Table (Unchanged)
CREATE TABLE Airport (
    AirportCode VARCHAR(3) PRIMARY KEY,
    AirportName VARCHAR(100),
    Location VARCHAR(255),
    Facilities VARCHAR(255)
);

-- Airline Table (Unchanged)
CREATE TABLE Airline (
    AirlineID INT PRIMARY KEY,
    AirlineName VARCHAR(100),
    ContactNumber VARCHAR(20),
    OperatingRegion VARCHAR(100)
);
```

REASONS FOR CHANGE-

Flight Table:

Previous Normal Form Issue: The denormalized FlightInfo table was not adhering to at least Second Normal Form (2NF) because it combined flight-specific information with airline-specific information, leading to partial dependencies.

Passenger Table:

No Normal Form Issue: The Passenger table was already in at least Third Normal Form (3NF) before any changes were made, with each attribute representing an indivisible piece of information about a passenger.

Booking Table:

Previous Normal Form Issue: The denormalized PassengerBooking table was not adhering to at least Second Normal Form (2NF) because it contained redundant information related to passengers (e.g., FirstName, LastName, Email, PassportNumber), leading to partial dependencies.

Payment Table:

No Normal Form Issue: The Payment table was already in at least Third Normal Form (3NF) before any changes were made, with each attribute representing an indivisible piece of information about a payment transaction.

Airport Table:

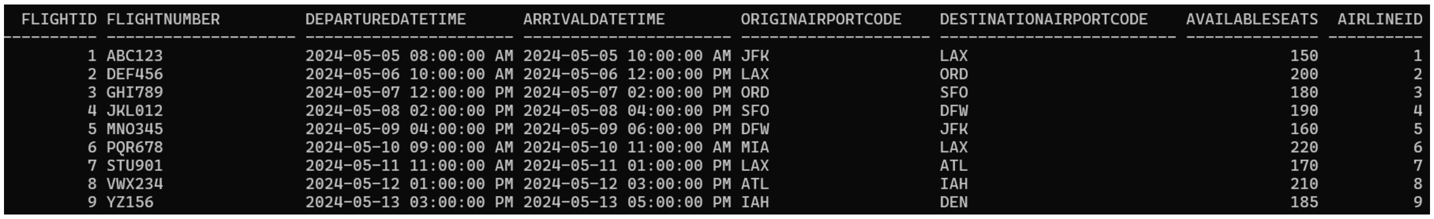
No Normal Form Issue: The Airport table was already in at least Third Normal Form (3NF) before any changes were made, with each attribute representing an indivisible piece of information about an airport.

Airline Table:

No Normal Form Issue: The Airline table was already in at least Third Normal Form (3NF) before any changes were made, with each attribute representing an indivisible piece of information about an airline.

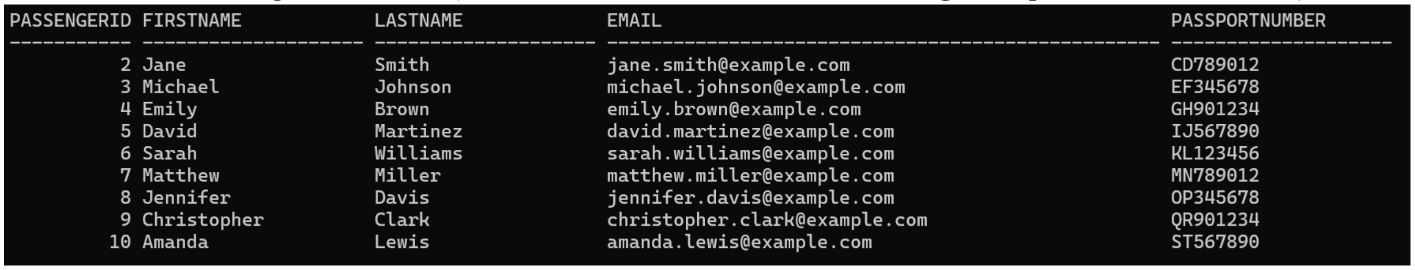
```
-- Flight Table
```

```
CREATE TABLE Flight (
   FlightID NUMBER PRIMARY KEY,
   FlightNumber VARCHAR(20) UNIQUE,
   DepartureDateTime TIMESTAMP,
   ArrivalDateTime  TIMESTAMP,
   OriginAirportCode VARCHAR(20),
   DestinationAirportCode VARCHAR(25),
   AvailableSeats number,
   AirlineID number,
   FOREIGN KEY (OriginAirportCode) REFERENCES Airport(AID),
   FOREIGN KEY (DestinationAirportCode) REFERENCES Airport(AID),
   FOREIGN KEY(AirlineID) REFERENCES Airline(AirlineID)
);

INSERT INTO Flight VALUES (1, 'ABC123', TO_TIMESTAMP('2024-05-05 08:00:00', 'YYYY-MM-DD
         HH24:MM:SS') TO_TIMESTAMP('2024-05-05 10:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'JFK',
         'LAX', 150,1);
INSERT INTO Flight VALUES(2, 'DEF456', TO_TIMESTAMP('2024-05-06 10:00:00', 'YYYY-MM-DD
         HH24:MM:SS'), TO_TIMESTAMP('2024-05-06 12:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'LAX',
         'ORD', 200,2);
INSERT INTO Flight VALUES(3, 'GHI789', TO_TIMESTAMP('2024-05-07 12:00:00', 'YYYY-MM-DD
         HH24:MM:SS'), TO_TIMESTAMP('2024-05-07 14:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'ORD',
         'SFO', 180,3);
INSERT INTO Flight VALUES(4, 'JKL012', TO_TIMESTAMP('2024-05-08 14:00:00', 'YYYY-MM-DD
         HH24:MM:SS'), TO_TIMESTAMP('2024-05-08 16:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'SFO',
         'DFW', 190,4);
INSERT INTO Flight VALUES(5, 'MNO345', TO_TIMESTAMP('2024-05-09 16:00:00', 'YYYY-MM-DD
         HH24:MI:SS'), TO_TIMESTAMP('2024-05-09 18:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'DFW', 'JFK',
         160,5);
INSERT INTO Flight VALUES (6, 'PQR678', TO_TIMESTAMP('2024-05-10 09:00:00', 'YYYY-MM-DD
         HH24:MI:SS'), TO_TIMESTAMP('2024-05-10 11:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'MIA',
         'LAX', 220,6);
INSERT INTO Flight VALUES (7, 'STU901', TO_TIMESTAMP('2024-05-11 11:00:00', 'YYYY-MM-DD
         HH24:MI:SS'), TO_TIMESTAMP('2024-05-11 13:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'LAX', 'ATL',
         170,7);
INSERT INTO Flight VALUES (8, 'VWX234', TO_TIMESTAMP('2024-05-12 13:00:00', 'YYYY-MM-DD
         HH24:MI:SS'), TO_TIMESTAMP('2024-05-12 15:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'ATL', 'IAH',
         210,8);
INSERT INTO Flight VALUES (9, 'YZ156', TO_TIMESTAMP('2024-05-13 15:00:00', 'YYYY-MM-DD
         HH24:MI:SS'), TO_TIMESTAMP('2024-05-13 17:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'IAH', 'DEN',
         185,9);
INSERT INTO Flight VALUES (10, 'ABC123', TO_TIMESTAMP('2024-05-14 17:00:00', 'YYYY-MM-DD
         HH24:MI:SS'), TO_TIMESTAMP('2024-05-14 19:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'DEN',
         'SEA', 230,10);
```



```
  FLIGHTID FLIGHTNUMBER         DEPARTUREDATETIME      ARRIVALDATETIME        ORIGINAIRPORTCODE    DESTINATIONAIRPORTCODE    AVAILABLESEATS  AIRLINEID
---------- -------------------- ---------------------- ---------------------- -------------------- ------------------------- -------------- ----------
         1 ABC123               2024-05-05 08:00:00 AM 2024-05-05 10:00:00 AM JFK                  LAX                                  150          1
         2 DEF456               2024-05-06 10:00:00 AM 2024-05-06 12:00:00 PM LAX                  ORD                                  200          2
         3 GHI789               2024-05-07 12:00:00 PM 2024-05-07 02:00:00 PM ORD                  SFO                                  180          3
         4 JKL012               2024-05-08 02:00:00 PM 2024-05-08 04:00:00 PM SFO                  DFW                                  190          4
         5 MNO345               2024-05-09 04:00:00 PM 2024-05-09 06:00:00 PM DFW                  JFK                                  160          5
         6 PQR678               2024-05-10 09:00:00 AM 2024-05-10 11:00:00 AM MIA                  LAX                                  220          6
         7 STU901               2024-05-11 11:00:00 AM 2024-05-11 01:00:00 PM LAX                  ATL                                  170          7
         8 VWX234               2024-05-12 01:00:00 PM 2024-05-12 03:00:00 PM ATL                  IAH                                  210          8
         9 YZ156                2024-05-13 03:00:00 PM 2024-05-13 05:00:00 PM IAH                  DEN                                  185          9
```

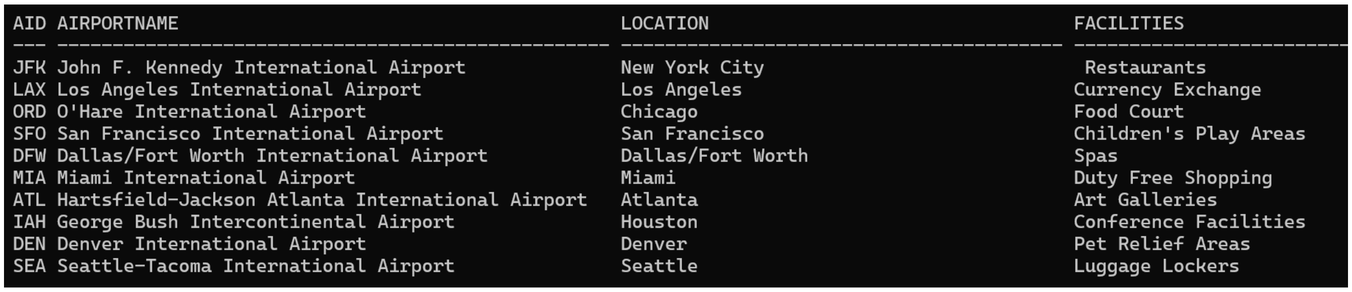
```
-- Passenger Table
CREATE TABLE Passenger (
   PassengerID INT PRIMARY KEY,
   FirstName VARCHAR(20),
   LastName VARCHAR(20),
   Email VARCHAR(50),
   PassportNumber VARCHAR(20)
);

INSERT INTO Passenger (PassengerID, FirstName, LastName, Email, PassportNumber)
VALUES
INSERT INTO Passenger VALUES (1, 'John', 'Doe', 'john.doe@example.com', 'AB123456');
INSERT INTO Passenger VALUES (2, 'Jane', 'Smith', 'jane.smith@example.com', 'CD789012');
INSERT INTO Passenger VALUES (3, 'Michael', 'Johnson', 'michael.johnson@example.com', 'EF345678');
INSERT INTO Passenger VALUES (4, 'Emily', 'Brown', 'emily.brown@example.com', 'GH901234');
INSERT INTO Passenger VALUES (5, 'David', 'Martinez', 'david.martinez@example.com', 'IJ567890');
INSERT INTO Passenger VALUES (6, 'Sarah', 'Williams', 'sarah.williams@example.com', 'KL123456');
INSERT INTO Passenger VALUES (7, 'Matthew', 'Miller', 'matthew.miller@example.com', 'MN789012');
INSERT INTO Passenger VALUES (8, 'Jennifer', 'Davis', 'jennifer.davis@example.com', 'OP345678');
INSERT INTO Passenger VALUES (9, 'Christopher', 'Clark', 'christopher.clark@example.com', 'QR901234');
INSERT INTO Passenger VALUES (10, 'Amanda', 'Lewis', 'amanda.lewis@example.com', 'ST567890');
```



```
PASSENGERID FIRSTNAME            LASTNAME             EMAIL                                              PASSPORTNUMBER
----------- -------------------- -------------------- -------------------------------------------------- --------------------
          2 Jane                 Smith                jane.smith@example.com                             CD789012
          3 Michael              Johnson              michael.johnson@example.com                        EF345678
          4 Emily                Brown                emily.brown@example.com                            GH901234
          5 David                Martinez             david.martinez@example.com                         IJ567890
          6 Sarah                Williams             sarah.williams@example.com                         KL123456
          7 Matthew              Miller               matthew.miller@example.com                         MN789012
          8 Jennifer             Davis                jennifer.davis@example.com                         OP345678
          9 Christopher          Clark                christopher.clark@example.com                      QR901234
         10 Amanda               Lewis                amanda.lewis@example.com                           ST567890
```

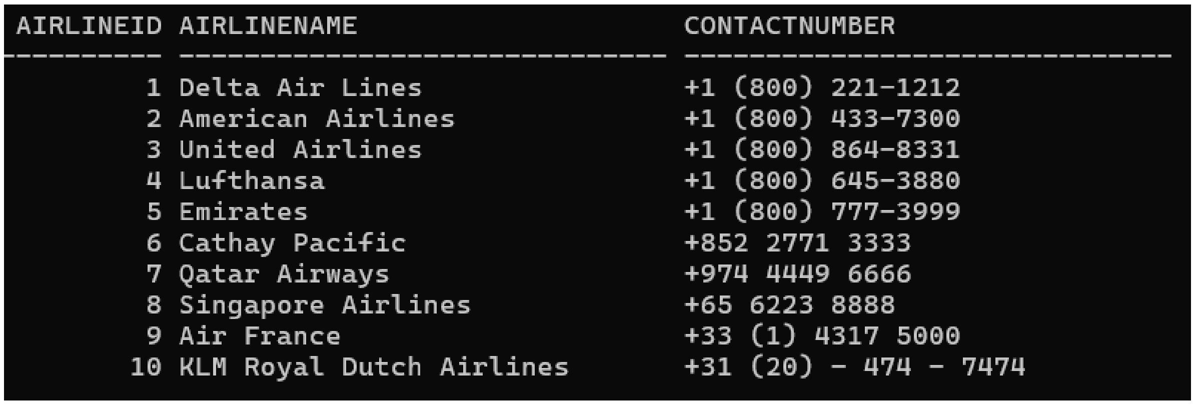
```
-- Airport Table
CREATE TABLE Airport (
   AID VARCHAR(3) PRIMARY KEY,
   AirportName VARCHAR(50),
   Location VARCHAR(40),
   Facilities VARCHAR(25)
);
INSERT INTO Airport VALUES ('JFK', 'John F. Kennedy International Airport', 'New York City', '
         Restaurants');
```

```
INSERT INTO Airport VALUES('LAX', 'Los Angeles International Airport', 'Los Angeles', 'Currency
        Exchange');
INSERT INTO Airport VALUES('ORD', 'O''Hare International Airport', 'Chicago', 'Food Court');
INSERT INTO Airport VALUES('SFO', 'San Francisco International Airport', 'San Francisco', 'Children''s Play
        Areas');
INSERT INTO Airport VALUES('DFW', 'Dallas/Fort Worth International Airport', 'Dallas/Fort Worth', 'Spas');
INSERT INTO Airport VALUES ('MIA', 'Miami International Airport', 'Miami', 'Duty Free Shopping');
INSERT INTO Airport VALUES ('ATL', 'Hartsfield-Jackson Atlanta International Airport', 'Atlanta', 'Art
        Galleries');
INSERT INTO Airport VALUES('IAH', 'George Bush Intercontinental Airport', 'Houston', 'Conference
        Facilities');
INSERT INTO Airport VALUES ('DEN', 'Denver International Airport', 'Denver', 'Pet Relief Areas');
INSERT INTO Airport VALUES ('SEA', 'Seattle-Tacoma International Airport', 'Seattle', 'Luggage Lockers');
```



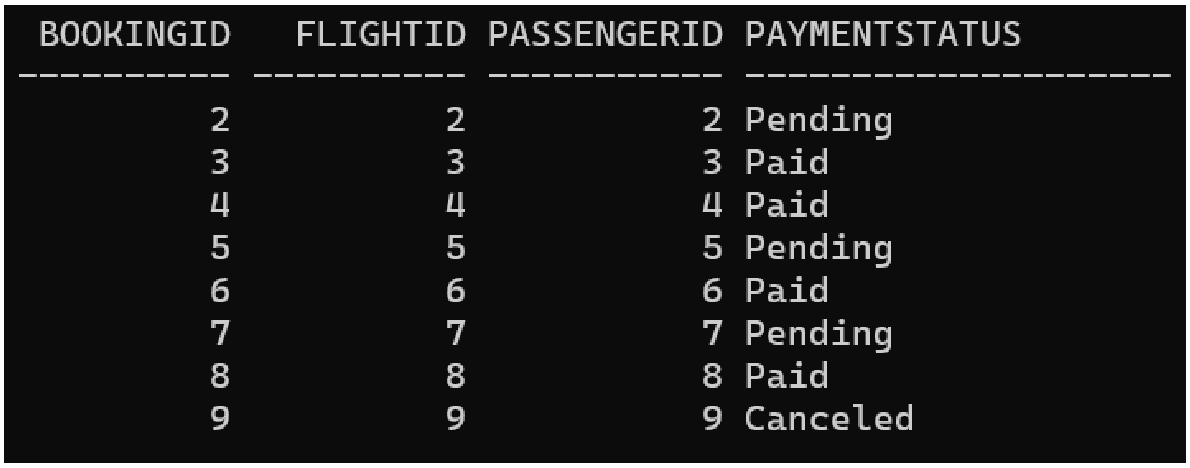
```
AID AIRPORTNAME                                        LOCATION                                 FACILITIES
--- -------------------------------------------------- ---------------------------------------- -------------------------
JFK John F. Kennedy International Airport              New York City                             Restaurants
LAX Los Angeles International Airport                  Los Angeles                              Currency Exchange
ORD O'Hare International Airport                       Chicago                                  Food Court
SFO San Francisco International Airport                San Francisco                            Children's Play Areas
DFW Dallas/Fort Worth International Airport            Dallas/Fort Worth                        Spas
MIA Miami International Airport                        Miami                                    Duty Free Shopping
ATL Hartsfield-Jackson Atlanta International Airport   Atlanta                                  Art Galleries
IAH George Bush Intercontinental Airport               Houston                                  Conference Facilities
DEN Denver International Airport                       Denver                                   Pet Relief Areas
SEA Seattle-Tacoma International Airport               Seattle                                  Luggage Lockers
```

```
-- Airline Table
CREATE TABLE Airline (
   AirlineID NUMBER PRIMARY KEY,
   AirlineName VARCHAR(30),
   ContactNumber VARCHAR(50)
);
INSERT INTO Airline VALUES (1, 'Delta Air Lines', '+1 (800) 221-1212');
INSERT INTO Airline VALUES (2, 'American Airlines', '+1 (800) 433-7300');
INSERT INTO Airline VALUES (3, 'United Airlines', '+1 (800) 864-8331');
INSERT INTO Airline VALUES (4, 'Lufthansa', '+1 (800) 645-3880');
INSERT INTO Airline VALUES (5, 'Emirates', '+1 (800) 777-3999');
INSERT INTO Airline VALUES (6, 'Cathay Pacific', '+852 2771 3333');
INSERT INTO Airline VALUES (7, 'Qatar Airways', '+974 4449 6666');
INSERT INTO Airline VALUES (8, 'Singapore Airlines', '+65 6223 8888');
INSERT INTO Airline VALUES (9, 'Air France', '+33 (1) 4317 5000');
INSERT INTO Airline VALUES (10, 'KLM Royal Dutch Airlines', '+31 (20) - 474 - 7474');
```



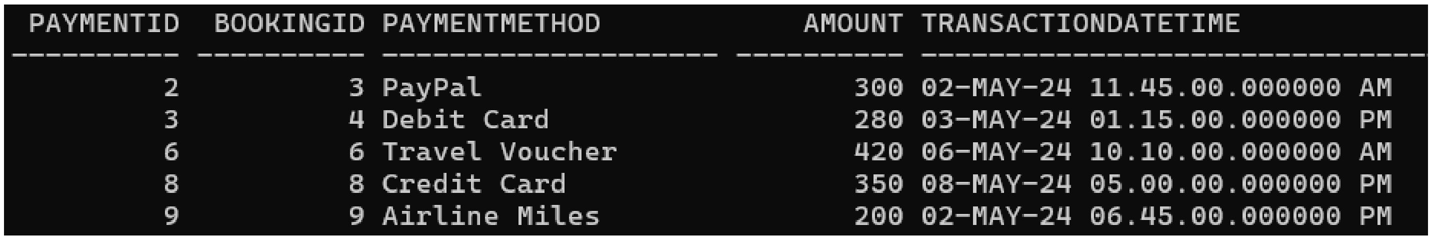
```
AIRLINEID AIRLINENAME                    CONTACTNUMBER
--------- ------------------------------ ------------------------------
        1 Delta Air Lines                +1 (800) 221-1212
        2 American Airlines              +1 (800) 433-7300
        3 United Airlines                +1 (800) 864-8331
        4 Lufthansa                      +1 (800) 645-3880
        5 Emirates                       +1 (800) 777-3999
        6 Cathay Pacific                 +852 2771 3333
        7 Qatar Airways                  +974 4449 6666
        8 Singapore Airlines             +65 6223 8888
        9 Air France                     +33 (1) 4317 5000
       10 KLM Royal Dutch Airlines       +31 (20) - 474 - 7474
```

```
-- Booking Table
CREATE TABLE Booking (
   BookingID INT PRIMARY KEY,
   FlightID INT,
   PassengerID INT,
   PaymentStatus VARCHAR(20),
   FOREIGN KEY (FlightID) REFERENCES Flight(FlightID),
   FOREIGN KEY (PassengerID) REFERENCES Passenger(PassengerID)
);

INSERT INTO Booking VALUES (1, 1, 1, 'Paid');
INSERT INTO Booking VALUES (2, 2, 2, 'Pending');
INSERT INTO Booking VALUES (3, 3, 3, 'Paid');
INSERT INTO Booking VALUES (4, 4, 4, 'Paid');
INSERT INTO Booking VALUES (5, 5, 5, 'Pending');
INSERT INTO Booking VALUES (6, 6, 6, 'Paid');
INSERT INTO Booking VALUES (7, 7, 7, 'Pending');
INSERT INTO Booking VALUES (8, 8, 8, 'Paid');
INSERT INTO Booking VALUES (9, 9, 9, 'Canceled');
INSERT INTO Booking VALUES (10, 10, 10, 'Pending');
```



```
 BOOKINGID   FLIGHTID PASSENGERID PAYMENTSTATUS
---------- ---------- ----------- --------------------
         2          2           2 Pending
         3          3           3 Paid
         4          4           4 Paid
         5          5           5 Pending
         6          6           6 Paid
         7          7           7 Pending
         8          8           8 Paid
         9          9           9 Canceled
```

```
-- Payment Table
CREATE TABLE Payment (
    PaymentID NUMBER PRIMARY KEY,
    BookingID NUMBER UNIQUE,
    PaymentMethod VARCHAR(20),
    Amount DECIMAL(10, 2),
    TransactionDateTime TIMESTAMP,
    FOREIGN KEY (BookingID) REFERENCES Booking(BookingID)
)
/

INSERT INTO Payment VALUES (2, 3, 'PayPal', 300.00, TO_TIMESTAMP('2024-05-02 11:45:00',
        'YYYY-MM-DD HH24:MI:SS'));
INSERT INTO Payment VALUES (3, 4, 'Debit Card', 280.00, TO_TIMESTAMP('2024-05-03 13:15:00',
        'YYYY-MM-DD HH24:MI:SS'));
INSERT INTO Payment VALUES (6, 6, 'Travel Voucher', 420.00, TO_TIMESTAMP('2024-05-06 10:10:00',
        'YYYY-MM-DD HH24:MI:SS'));
INSERT INTO Payment VALUES (8, 8, 'Credit Card', 350.00, TO_TIMESTAMP('2024-05-08 17:00:00',
        'YYYY-MM-DD HH24:MI:SS'));
INSERT INTO Payment VALUES (9, 9, 'Airline Miles', 200.00, TO_TIMESTAMP('2024-05-02 18:45:00',
        'YYYY-MM-DD HH24:MI:SS'));
```

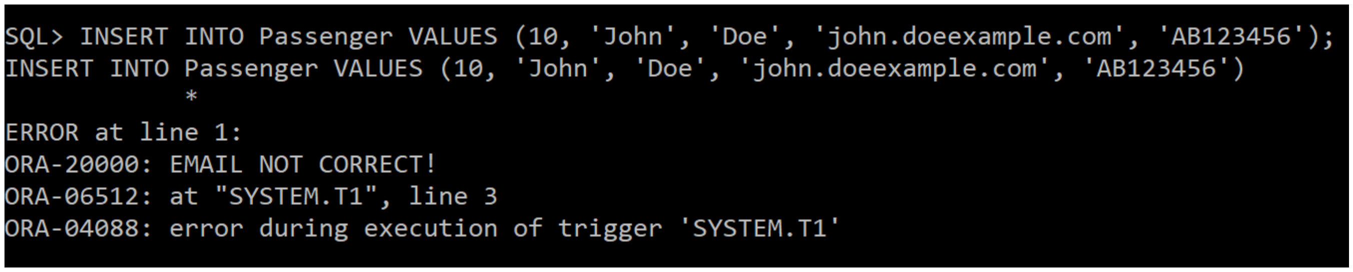


```
 PAYMENTID  BOOKINGID PAYMENTMETHOD            AMOUNT TRANSACTIONDATETIME
---------- ---------- -------------------- ---------- ------------------------------
         2          3 PayPal                      300 02-MAY-24 11.45.00.000000 AM
         3          4 Debit Card                  280 03-MAY-24 01.15.00.000000 PM
         6          6 Travel Voucher              420 06-MAY-24 10.10.00.000000 AM
         8          8 Credit Card                 350 08-MAY-24 05.00.00.000000 PM
         9          9 Airline Miles               200 02-MAY-24 06.45.00.000000 PM
```

TRIGGERS

Trigger to enforce valid email format

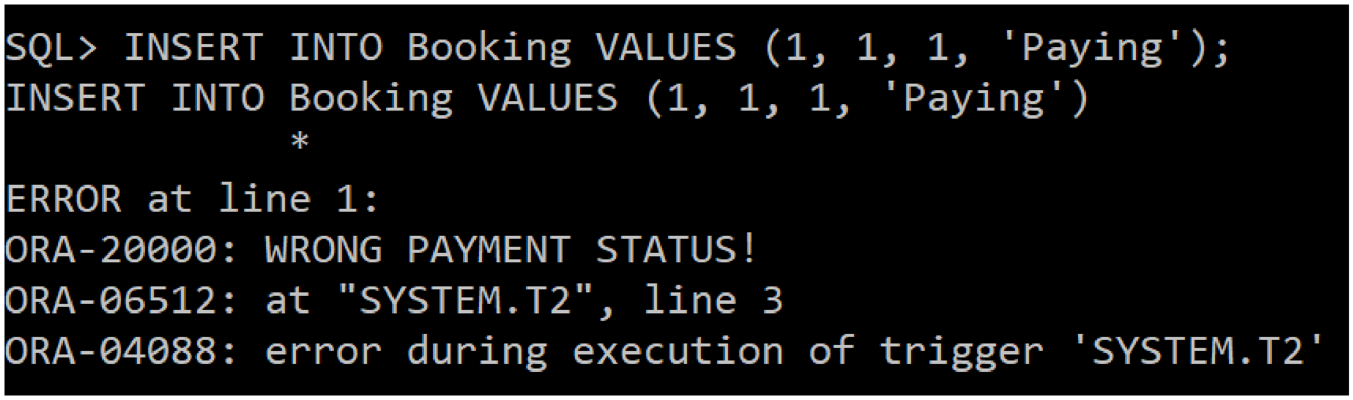
```
CREATE OR REPLACE TRIGGER T1
BEFORE INSERT OR UPDATE ON PASSENGER
FOR EACH ROW
BEGIN
IF :NEW.EMAIL NOT LIKE '%@%' THEN
RAISE_APPLICATION_ERROR(-20000,'EMAIL NOT CORRECT!');
END IF;
END;
/
```



```
SQL> INSERT INTO Passenger VALUES (10, 'John', 'Doe', 'john.doeexample.com', 'AB123456');
INSERT INTO Passenger VALUES (10, 'John', 'Doe', 'john.doeexample.com', 'AB123456')
            *
ERROR at line 1:
ORA-20000: EMAIL NOT CORRECT!
ORA-06512: at "SYSTEM.T1", line 3
ORA-04088: error during execution of trigger 'SYSTEM.T1'
```

Trigger to enforce valid payment status

```
CREATE OR REPLACE TRIGGER T2
BEFORE INSERT OR UPDATE ON BOOKING
FOR EACH ROW
BEGIN
IF :NEW.PAYMENTSTATUS NOT IN ('Paid','Pending','Canceled') THEN
RAISE_APPLICATION_ERROR(-20000,'WRONG PAYMENT STATUS!');
END IF;
END;
/
```

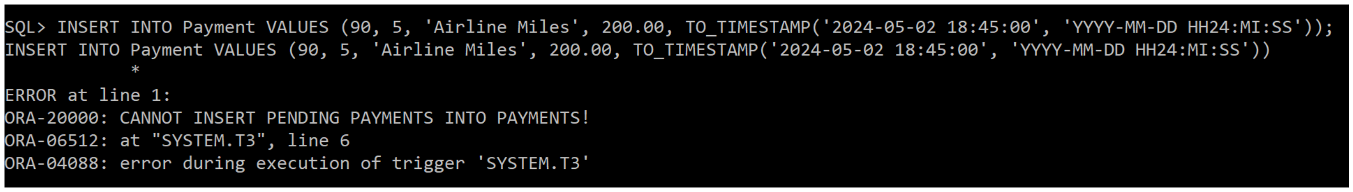


```
SQL> INSERT INTO Booking VALUES (1, 1, 1, 'Paying');
INSERT INTO Booking VALUES (1, 1, 1, 'Paying')
            *
ERROR at line 1:
ORA-20000: WRONG PAYMENT STATUS!
ORA-06512: at "SYSTEM.T2", line 3
ORA-04088: error during execution of trigger 'SYSTEM.T2'
```

Trigger to Prevent Pending Payment Insertion

```
CREATE OR REPLACE TRIGGER T3
```

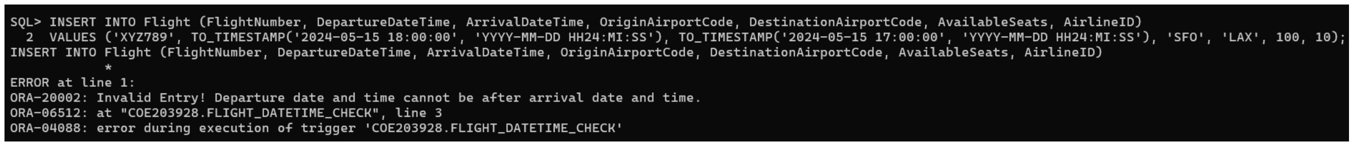
```
BEFORE INSERT OR UPDATE ON PAYMENT
FOR EACH ROW
DECLARE
N VARCHAR2(20);
BEGIN
SELECT PAYMENTSTATUS INTO N FROM BOOKING WHERE BOOKINGID=:NEW.BOOKINGID;
IF N='Pending' THEN
RAISE_APPLICATION_ERROR(-20000,'CANNOT INSERT PENDING PAYMENTS INTO PAYMENTS!');
END IF;
END;
/
```



```
SQL> INSERT INTO Payment VALUES (90, 5, 'Airline Miles', 200.00, TO_TIMESTAMP('2024-05-02 18:45:00', 'YYYY-MM-DD HH24:MI:SS'));
INSERT INTO Payment VALUES (90, 5, 'Airline Miles', 200.00, TO_TIMESTAMP('2024-05-02 18:45:00', 'YYYY-MM-DD HH24:MI:SS'))
            *
ERROR at line 1:
ORA-20000: CANNOT INSERT PENDING PAYMENTS INTO PAYMENTS!
ORA-06512: at "SYSTEM.T3", line 6
ORA-04088: error during execution of trigger 'SYSTEM.T3'
```

Flight Datetime Validation Trigger (ensures departure before arrival)

```
 CREATE OR REPLACE TRIGGER FLIGHT_DATETIME_CHECK
   BEFORE INSERT OR UPDATE ON FLIGHT
   FOR EACH ROW
   BEGIN
   IF :NEW.DEPARTUREDATETIME>=:NEW.ARRIVALDATETIME THEN
   RAISE_APPLICATION_ERROR(-20002,'Invalid Entry! Departure date and time cannot be after arrival date
         and time.');
   END IF;
   END;
   /
```

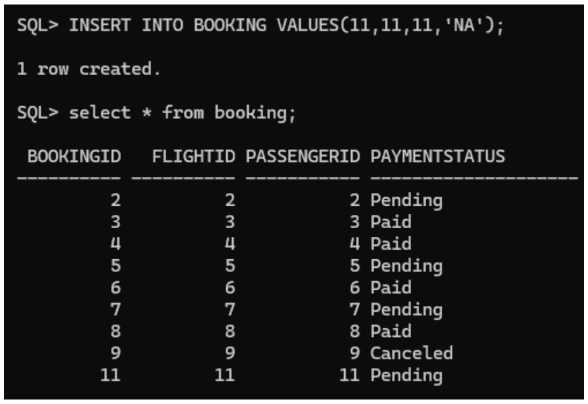


```
SQL> INSERT INTO Flight (FlightNumber, DepartureDateTime, ArrivalDateTime, OriginAirportCode, DestinationAirportCode, AvailableSeats, AirlineID)
  2  VALUES ('XYZ789', TO_TIMESTAMP('2024-05-15 18:00:00', 'YYYY-MM-DD HH24:MI:SS'), TO_TIMESTAMP('2024-05-15 17:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'SFO', 'LAX', 100, 10);
INSERT INTO Flight (FlightNumber, DepartureDateTime, ArrivalDateTime, OriginAirportCode, DestinationAirportCode, AvailableSeats, AirlineID)
            *
ERROR at line 1:
ORA-20002: Invalid Entry! Departure date and time cannot be after arrival date and time.
ORA-06512: at "COE203928.FLIGHT_DATETIME_CHECK", line 3
ORA-04088: error during execution of trigger 'COE203928.FLIGHT_DATETIME_CHECK'
```

Trigger to set payment status

```
CREATE OR REPLACE TRIGGER SET_PAYMENT_STATUS
   BEFORE INSERT OR UPDATE ON BOOKING
   FOR EACH ROW
DECLARE
dummy NUMBER;
BEGIN
   BEGIN
      SELECT 1 INTO dummy FROM PAYMENT WHERE PAYMENT.BOOKINGID = :NEW.BOOKINGID;
   EXCEPTION
      WHEN NO_DATA_FOUND THEN
         :NEW.PAYMENTSTATUS := 'Pending';
```

```
    END;
END;
/
```



```
SQL> INSERT INTO BOOKING VALUES(11,11,11,'NA');

1 row created.

SQL> select * from booking;

 BOOKINGID   FLIGHTID PASSENGERID PAYMENTSTATUS
---------- ---------- ----------- --------------------
         2          2           2 Pending
         3          3           3 Paid
         4          4           4 Paid
         5          5           5 Pending
         6          6           6 Paid
         7          7           7 Pending
         8          8           8 Paid
         9          9           9 Canceled
        11         11          11 Pending
```

FUNCTIONS

--Query to generate boarding pass of passenger:

```
CREATE OR REPLACE FUNCTION generate_boarding_pass( p_BookingID NUMBER)
RETURN VARCHAR2 AS
   v_BoardingPass VARCHAR2(4000);
BEGIN
   SELECT 'Passenger Name: ' || p.FirstName || ' ' || p.LastName || CHR(10) ||
          'Flight Name: ' || f.FlightNumber || CHR(10) ||
          'Source: ' || a1.AirportName || ' (' || f.OriginAirportCode || ')' || CHR(10) ||
          'Destination: ' || a2.AirportName || ' (' || f.DestinationAirportCode || ')' || CHR(10) ||
          'Boarding Time: ' || TO_CHAR(f.DepartureDateTime, 'DD-MON-YYYY HH24:MI')
   INTO v_BoardingPass
   FROM Passenger p
   JOIN Booking b ON p.PassengerID = b.PassengerID
   JOIN Flight f ON b.FlightID = f.FlightID
   JOIN Airport a1 ON f.OriginAirportCode = a1.AirportCode
   JOIN Airport a2 ON f.DestinationAirportCode = a2.AirportCode
   WHERE b.BookingID = p_BookingID;

   RETURN v_BoardingPass;
EXCEPTION
   WHEN NO_DATA_FOUND THEN
      RETURN 'Booking not found.';
   WHEN OTHERS THEN
      RETURN 'An error occurred.';
END;



DECLARE
display_ticket varchar2(4000);
enter_b_id number;
BEGIN
enter_b_id=&enter_b_id;
display_ticket=generate_boarding_pass(enter_b_id);
dbms_output.put_line(display_ticket);
end;
```

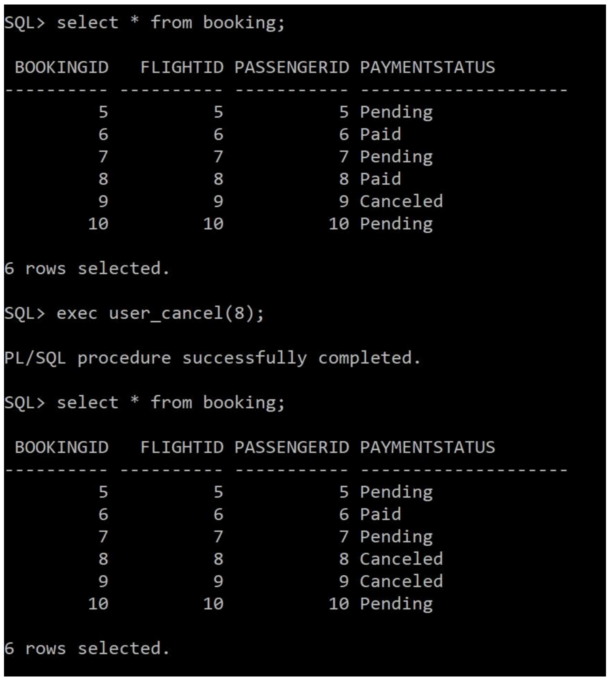
PROCEDURE

--Query to fetch all flights scheduled for the present day:

```
CREATE OR REPLACE PROCEDURE display_all_flights_today AS
   CURSOR c IS
      SELECT
         f.FlightNumber,
         f.DepartureDateTime AS DepartureTime,
         f.ArrivalDateTime AS ArrivalTime,
         a1.AirportName AS Origin,
         a2.AirportName AS Destination
      FROM
         Flight f
      JOIN
         Airport a1 ON f.OriginAirportCode = a1.AirportCode
      JOIN
         Airport a2 ON f.DestinationAirportCode = a2.AirportCode
      WHERE
         TRUNC(f.DepartureDateTime) = TRUNC(SYSDATE);
BEGIN
   FOR flight_rec IN c LOOP
      DBMS_OUTPUT.PUT_LINE('Flight Number: ' || flight_rec.FlightNumber);
      DBMS_OUTPUT.PUT_LINE('Departure Time: ' || TO_CHAR(flight_rec.DepartureTime,
         'DD-MON-YYYY HH24:MI'));
      DBMS_OUTPUT.PUT_LINE('Arrival Time: ' || TO_CHAR(flight_rec.ArrivalTime, 'DD-MON-YYYY
         HH24:MI'));
      DBMS_OUTPUT.PUT_LINE('Origin: ' || flight_rec.Origin);
      DBMS_OUTPUT.PUT_LINE('Destination: ' || flight_rec.Destination);
      DBMS_OUTPUT.PUT_LINE('----------------------------------------');
   END LOOP;
EXCEPTION
   WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
END display_all_flights_today;
```

--Query to set payment status to ‘Canceled’ when user cancels a ticket

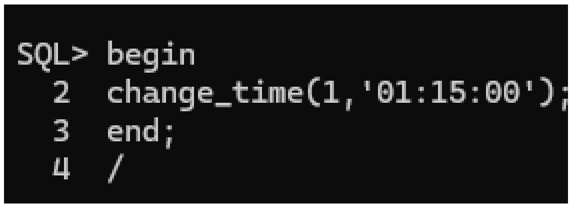
```
CREATE OR REPLACE PROCEDURE USER_CANCEL(N IN NUMBER) IS
BEGIN
UPDATE BOOKING SET PAYMENTSTATUS = ‘Canceled’ WHERE BOOKINGID = N;
END;
```



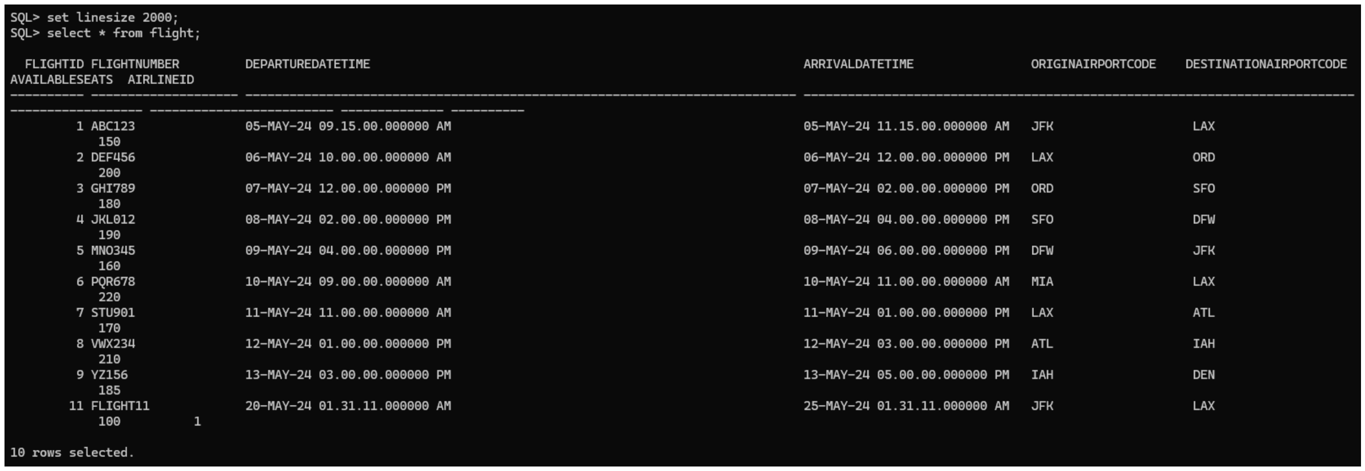
```
SQL> select * from booking;

 BOOKINGID   FLIGHTID PASSENGERID PAYMENTSTATUS
---------- ---------- ----------- --------------------
         5          5           5 Pending
         6          6           6 Paid
         7          7           7 Pending
         8          8           8 Paid
         9          9           9 Canceled
        10         10          10 Pending

6 rows selected.

SQL> exec user_cancel(8);

PL/SQL procedure successfully completed.

SQL> select * from booking;

 BOOKINGID   FLIGHTID PASSENGERID PAYMENTSTATUS
---------- ---------- ----------- --------------------
         5          5           5 Pending
         6          6           6 Paid
         7          7           7 Pending
         8          8           8 Canceled
         9          9           9 Canceled
        10         10          10 Pending

6 rows selected.
```

--Query to update arrival and departure time of flights:

```
CREATE OR REPLACE PROCEDURE change_time(
   flight_id_input IN NUMBER,
   time_difference_input IN VARCHAR2
) IS
   CURSOR c1 IS SELECT * FROM flight WHERE FlightID = flight_id_input;
   time_interval INTERVAL DAY TO SECOND;
BEGIN
      time_interval := TO_DSINTERVAL('0 ' || time_difference_input);

   FOR rec IN c1 LOOP
      DBMS_OUTPUT.PUT_LINE('Updating flight ' || rec.FlightID);
      UPDATE flight
      SET ArrivalDateTime = rec.ArrivalDateTime + time_interval,
         DepartureDateTime = rec.DepartureDateTime + time_interval
      WHERE FlightID = rec.FlightID;
      DBMS_OUTPUT.PUT_LINE('Flight ' || rec.FlightID || ' updated.');
   END LOOP;
   COMMIT;
END;
```



```
SQL> begin
  2  change_time(1,'01:15:00');
  3  end;
  4  /
```



```
SQL> set linesize 2000;
SQL> select * from flight;

  FLIGHTID FLIGHTNUMBER         DEPARTUREDATETIME                                                           ARRIVALDATETIME                ORIGINAIRPORTCODE    DESTINATIONAIRPORTCODE
AVAILABLESEATS  AIRLINEID
---------- -------------------- --------------------------------------------------------------------------- ------------------------------------------------------------------------------
----------------- ------------------------- -------------- -----------
         1 ABC123               05-MAY-24 09.15.00.000000 AM                                                05-MAY-24 11.15.00.000000 AM   JFK                   LAX
          150
         2 DEF456               06-MAY-24 10.00.00.000000 AM                                                06-MAY-24 12.00.00.000000 PM   LAX                   ORD
          200
         3 GHI789               07-MAY-24 12.00.00.000000 PM                                                07-MAY-24 02.00.00.000000 PM   ORD                   SFO
          180
         4 JKL012               08-MAY-24 02.00.00.000000 PM                                                08-MAY-24 04.00.00.000000 PM   SFO                   DFW
          190
         5 MNO345               09-MAY-24 04.00.00.000000 PM                                                09-MAY-24 06.00.00.000000 PM   DFW                   JFK
          160
         6 PQR678               10-MAY-24 09.00.00.000000 AM                                                10-MAY-24 11.00.00.000000 AM   MIA                   LAX
          220
         7 STU901               11-MAY-24 11.00.00.000000 AM                                                11-MAY-24 01.00.00.000000 PM   LAX                   ATL
          170
         8 VWX234               12-MAY-24 01.00.00.000000 PM                                                12-MAY-24 03.00.00.000000 PM   ATL                   IAH
          210
         9 YZ156                13-MAY-24 03.00.00.000000 PM                                                13-MAY-24 05.00.00.000000 PM   IAH                   DEN
          185
        11 FLIGHT11             20-MAY-24 01.31.11.000000 AM                                                25-MAY-24 01.31.11.000000 AM   JFK                   LAX
          100           1

10 rows selected.
```

Main Block

```
declare
   n number;
          c number;
          choice number;
          function_choice number;
          function_choose number;
          booking_id number;
          d VARCHAR2(4000);
          password varchar(30);
          flight_id_input NUMBER;
   time_difference_input VARCHAR2(20);
begin
   n:=1;
          while n=1 loop
      dbms_output.put_line('enter 1 for PASSENGER LOGIN, 2 for MANAGEMENT LOGIN, 3 to NOT CARRY
          FURTHER FUNCTIONS:');
                    choice:=&choice;
                    case choice
         when 1 then
                                        booking_id:=&booking_id;
                                        dbms_output.put_line('enter 1 to DISPLAY PASSENGER DETAILS, 2 to CANCEL
          FLIGHT:');
                                        function_choice:=&function_choice;
                                        case function_choice
               when 1 then
                    d:=generate_boarding_pass(booking_id);
                    dbms_output.put_line('boarding pass: '||d);
               when 2 then
                  USER_CANCEL(booking_id);
                              dbms_output.put_line(booking id ||'ticket cancelled');
               else
                  dbms_output.put_line('WRONG INPUT!');
                                        end case;
                                        when 2 then
   password:=&password_for_management_login;
                                        if password='dbmsproject' then
                                        dbms_output.put_line('enter 1 to DISPLAY TODAYS FLIGHT DETAILS, 2 to UPDATE
        TIME OF FLIGHTS:');
                                                function_choose:=&function_choose;
                                                case function_choose
                when 1 then
                                                                 dbms_output.put_line('today flights: ');
                                                                display_all_flights_today;
                when 2 then
                                         flight_id_input:= &flight_id_input;
                                                        time_difference_input:= '&time_difference_input';
                                                        change_time(flight_id_input, time_difference_input);
                        else
                        dbms_output.put_line('WRONG INPUT!');
                                                        end case;
                                        else
                                        dbms_output.put_line('WRONG PASSWORD! you cannot access management functions,
        try again.');
                                        end if;
                        when 3 then
   n:=0;
                        else
                                        dbms_output.put_line('WRONG INPUT!');
                end case;
        end loop;
end;
```

Conclusion

We have completed the Airport Management System (AMS) project, which has been an excellent opportunity to learn about managing databases in the aviation sector. Overall, we identified the requirements for a solid system for streamlining airport operations in this project and then worked on building and implementing our database structures and functions. The lessons we have learned have been invaluable in learning about database operations using Oracle SQL.
We started by understanding the specific requirements of our AMS. Requirement analysis is an essential part of the planning process to ensure that the AMS meets the operational needs of airports, enhancing efficiency and service quality. We highlighted our project's functional and non-functional requirements to guarantee that all our needs were fulfilled and to avoid planning-related errors.
Creating our Entity-Relationship (ER) diagram was essential to the project. It helped us visualise the connections between flights, passengers, bookings, and airlines, providing a clear roadmap as we constructed our database.
The next step in this process was to convert our ER diagram to a table. This step involved adding the correct foreign keys to each table, ensuring tables had primary keys and the correct constraints attached to them.
Ensuring consistency throughout the table was a problematic roadblock to cross. However, by adhering to normalisation principles, we ensured that our data was organised efficiently and accurately, setting the stage for a reliable system that could grow with our needs.
We added PL/SQL queries to build a robust system, prevent errors and make our system more user-friendly. We can now handle tasks like updating boarding passes and displaying flight information using functions and procedures. Triggers prevent any inconsistent data from being added to the project.

Ultimately, our AMS project has highlighted the importance of a well-designed database system in keeping airports operating efficiently. The skills we have acquired and the lessons we have learned will undoubtedly be invaluable as we tackle real-world challenges in airport management and beyond.
In summary, our AMS project has been a significant milestone in our journey towards mastering database management, and we are eager to see where this journey takes us next.

REFERENCES

1. https://www.javatpoint.com/dbms-er-model-concept
2. https://lucid.co/product/lucidchart
3. https://www.geeksforgeeks.org/normal-forms-in-dbms/
4. https://www.simplilearn.com/what-is-requirement-analysis-article#:~:text=Requirements%20analysis%20or%20requirements%20engineering,document%20all%20the%20key%20requirements
5. https://docs.oracle.com/database/121/LNPLS/toc.htm