

AIRLINE MANAGEMENT SYSTEM PROJECT

INTRODUCTION

The airline management system project aims to streamline airline operations by integrating reservation, scheduling, ticketing, and fleet management processes into a unified platform. In this project, we aim to develop an Airline Management System using SQL and PL/SQL. Key features include passenger booking, seat allocation, flight scheduling, crew management, aircraft maintenance and real-time updates for flight status. The system enhances efficiency, improves customer experience, and ensures optimal resource utilization for airlines.

The airline industry is a complex ecosystem requiring efficient management of various resources and operations to ensure smooth and safe travel for passengers. To streamline these operations and enhance overall efficiency, airline companies often leverage Database Management Systems to organize and manage their data effectively.

An airline management project involves the creation of a comprehensive database that stores and manages critical information related to flights, aircraft, passengers, crew members, reservations, schedules, and more. By centralizing this data, airline companies can optimize their operations, improve customer service, and make data-driven decisions to enhance overall performance.

In this project, we aim to develop an Airline Management System using SQL and PL/SQL. Key features include passenger booking, seat allocation, flight scheduling, crew management, aircraft maintenance and real-time updates for flight status.

By understanding the intricacies of airline management within a DBMS context, we can appreciate how technology plays a pivotal role in transforming the airline industry, making air travel safer, more efficient, and more convenient for passengers worldwide.

REQUIREMENT ANALYSIS

To develop a successful Airline Management System, we need to analyze the requirements of the system. The following are the primary requirements for the system:

- Maintenance of Aircrafts: In the airline management system project, aircraft maintenance functionality is essential for ensuring the safety and reliability of the fleet. It involves inspections, repairs, and component replacements. The system facilitates efficient maintenance planning, monitors aircraft health data, and generates maintenance reports to optimize aircraft availability and minimize downtime.
- Summary of Flights: The flights project aims to optimize the management of airline routes and schedules. It involves creating and managing flight itineraries, including departure and arrival times, aircraft assignments, and seat availability. The system also tracks real-time flight status, handles reservations, and facilitates check-in processes.
- Data Management: Data management in the project includes handling passenger data, flight schedules, crew information, maintenance records, and other relevant data sets. The system ensures data integrity, security, and accessibility, enabling seamless communication between different modules and stakeholders.
- Ticket Management: Ticket management in the project involves handling the entire ticketing process from booking to boarding. It includes functionalities such as ticket reservations, seat assignments, fare calculation, payment processing, and ticket issuance.
- Flight Details: Flight details refer to essential information about a specific flight within the airline management system. This includes the flight number, departure and arrival times, origin and destination airports, seating configuration, and current status. By providing concise details, the system enables efficient flight planning, passenger communication, and operational coordination for the airline.

SOFTWARE REQUIREMENTS

1. SQL
2. PL/SQL
3. Command Prompt

SQL

Creation of Tables:-

Table 1: Country

```
1 create table country(  
2   ct_id int primary key,  
3   country_name varchar2(32) not null  
4* )  
SQL> /
```

Table created.

```
SQL> desc country;
```

Name	Null?	Type
CT_ID	NOT NULL	NUMBER(38)
COUNTRY_NAME	NOT NULL	VARCHAR2(32)

Table 2: State

```
SQL> create table state(  
2   st_id int primary key,  
3   state_name varchar2(32) not null,  
4   country int not null,  
5   foreign key(country) references country(ct_id));
```

Table created.

```
SQL> desc state;
```

Name	Null?	Type
ST_ID	NOT NULL	NUMBER(38)
STATE_NAME	NOT NULL	VARCHAR2(32)
COUNTRY	NOT NULL	NUMBER(38)

Table 3: Contact_Details

```
SQL> create table contact_details(  
2   cn_id int primary key,  
3   email varchar2(50) not null,  
4   mobile varchar2(16) not null,  
5   state int not null,  
6   foreign key(state) references state(st_id));
```

Table created.

```
SQL> desc contact_details;
Name                                         Null?    Type
-----
CN_ID                                         NOT NULL NUMBER(38)
EMAIL                                         NOT NULL VARCHAR2(50)
MOBILE                                         NOT NULL VARCHAR2(16)
STATE                                         NOT NULL NUMBER(38)
```

Table 4: Passenger

```
SQL> create table passenger(
 2  ps_id int primary key,
 3  name varchar2(32) not null,
 4  address varchar2(64) not null,
 5  age int not null,
 6  nationality varchar2(16) not null,
 7  contacts int not null,
 8  foreign key(contacts) references contact_details(cn_id));

Table created.
```

```
SQL> desc passenger;
Name                                         Null?    Type
-----
PS_ID                                         NOT NULL NUMBER(38)
NAME                                         NOT NULL VARCHAR2(32)
ADDRESS                                       NOT NULL VARCHAR2(64)
AGE                                           NOT NULL NUMBER(38)
NATIONALITY                                  NOT NULL VARCHAR2(16)
CONTACTS                                      NOT NULL NUMBER(38)

SQL> |
```

Table 5: Aircraft_Type

```
1  create table aircraft_type(
2  act_id int primary key,
3  type varchar2(32) not null,
4  capacity int not null
5* )
SQL>
SQL> /

Table created.
```

```
SQL> desc aircraft_type;
Name                                         Null?    Type
-----
ACT_ID                                         NOT NULL NUMBER(38)
TYPE                                           NOT NULL VARCHAR2(32)
CAPACITY                                       NOT NULL NUMBER(38)
```

Table 6: Aircraft

```

1  create table aircraft(
2  acid int primary key,
3  ac_type int not null,
4  mfg_date date not null,
5* foreign key(ac_type) references aircraft_type(act_id))
SQL> /
Table created.

```

```

SQL> desc aircraft;
Name                                         Null?    Type
-----
ACID                                         NOT NULL NUMBER(38)
AC_TYPE                                     NOT NULL NUMBER(38)
MFG_DATE                                    NOT NULL DATE

```

Table 7: Airfare

```

SQL> create table airfare(
2  af_id int primary key,
3  route int not null,
4  fare int not null,
5  foreign key(route) references route(rt_id));
Table created.

```

```

SQL> desc airfare;
Name                                         Null?    Type
-----
AF_ID                                         NOT NULL NUMBER(38)
ROUTE                                         NOT NULL NUMBER(38)
FARE                                         NOT NULL NUMBER(38)

```

Table 8: Flight_Schedule

```

SQL> create table flight_schedule(
2  fl_id int primary key,
3  flight_date date not null,
4  departure timestamp not null,
5  arrival timestamp not null,
6  aircraft int not null,
7  netfare int not null,
8  foreign key(aircraft) references aircraft(acid),
9  foreign key(netfare) references airfare(af_id));
Table created.

```

```

SQL> desc flight_schedule;
Name                                         Null?    Type
-----
FL_ID                                         NOT NULL NUMBER(38)
FLIGHT_DATE                                  NOT NULL DATE
DEPARTURE                                    NOT NULL TIMESTAMP(6)
ARRIVAL                                      NOT NULL TIMESTAMP(6)
AIRCRAFT                                     NOT NULL NUMBER(38)
NETFARE                                      NOT NULL NUMBER(38)

```

Table 9: Transaction

```
1 create table transaction(  
2   ts_id int primary key,  
3   booking_date timestamp not null,  
4   passenger int not null,  
5   flight int not null,  
6   foreign key(passenger) references passenger(ps_id),  
7   foreign key(flight) references flight_schedule(fl_id)  
8* )  
SQL> /  
  
Table created.
```

```
SQL> desc transaction;  
Name                               Null?    Type  
-----  
TS_ID                              NOT NULL NUMBER(38)  
BOOKING_DATE                        NOT NULL TIMESTAMP(6)  
PASSENGER                           NOT NULL NUMBER(38)  
FLIGHT                             NOT NULL NUMBER(38)
```

Table 10: Route

```
SQL> create table route(  
2   rt_id int primary key,  
3   airport varchar2(32) not null,  
4   destination varchar2(32) not null,  
5   routecode varchar2(16) not null unique);  
  
Table created.
```

Inserting values into Tables:-

```

SQL> select * from passenger;

  PS_ID NAME
-----
ADDRESS
-----
NATIONALITY CONTACTS
-----
1 John Doe
123 Main Street 30
USA 1
2 Jane Smith
456 Elm Street 25
Canada 2
  PS_ID NAME
-----
ADDRESS
-----
NATIONALITY CONTACTS
-----
3 Ali Khan
789 Oak Avenue 28
Pakistan 3
4 Rachel Lee
321 Pine Road 35
  PS_ID NAME
-----
ADDRESS
-----
NATIONALITY CONTACTS
-----
Singapore 4
5 Neha Sharma
567 Maple Lane 29
India 5
6 Tashi Wangmo
  PS_ID NAME
-----

```

ADDRESS		AGE
NATIONALITY	CONTACTS	
890 Cedar Court Bhutan	6	32
7 Adam Abdul 234 Birch Street Maldives	7	27
PS_ID NAME		
ADDRESS		AGE
NATIONALITY	CONTACTS	
8 May Lin 678 Pine Lane Malaysia	8	31
9 David Nguyen 901 Oak Road Vietnam	9	26
PS_ID NAME		
ADDRESS		AGE
NATIONALITY	CONTACTS	
10 Lisa Santos 345 Cedar Avenue Philippines	10	33
11 Ahmad Khan 567 Pine Avenue		34
PS_ID NAME		

ADDRESS		AGE
NATIONALITY	CONTACTS	
Afghanistan	11	
12 Emily Wong		
890 Oak Lane		27
Malaysia	12	
13 Michael Johnson		
PS_ID	NAME	
ADDRESS		AGE
NATIONALITY	CONTACTS	
123 Elm Road		29
USA	13	
14 Priya Patel		
456 Birch Street		26
India	14	
PS_ID	NAME	
ADDRESS		AGE
NATIONALITY	CONTACTS	
15 Carlos Rodriguez		
789 Maple Avenue		32
Mexico	15	
16 Sophia Adams		
321 Oak Lane		31
Canada	16	
PS_ID	NAME	
ADDRESS		AGE
NATIONALITY	CONTACTS	
17 James Smith		
567 Pine Street		28
USA	17	
18 Amit Patel		
890 Cedar Avenue		30
PS_ID	NAME	
ADDRESS		AGE
NATIONALITY	CONTACTS	
India	18	
19 Maria Garcia		
234 Elm Road		27
Mexico	19	
20 Mohammed Khan		
PS_ID	NAME	
ADDRESS		AGE
NATIONALITY	CONTACTS	
678 Maple Street		29
Pakistan	20	

20 rows selected.

```
SQL> select * from route;
```

RT_ID	AIRPORT	DESTINATION
ROUTE CODE		
PK-IN	1 Pakistan	India
BD-IN	2 Bangladesh	India
LK-IN	3 Sri Lanka	India
ROUTE CODE		
NP-IN	4 Nepal	India
BT-IN	5 Bhutan	India
MV-IN	6 Maldives	India
ROUTE CODE		
AF-IN	7 Afghanistan	India
MM-IN	8 Myanmar	India
MY-IN	9 Malaysia	India
ROUTE CODE		
SG-IN	10 Singapore	India
TH-IN	11 Thailand	India
ID-IN	12 Indonesia	India
ROUTE CODE		
PH-IN	13 Philippines	India
KH-IN	14 Cambodia	India

14 rows selected.

```
SQL> select * from aircraft_type;
```

ACT_ID	TYPE	CAPACITY
1	Boeing 737	150
2	Airbus A320	180
3	Boeing 777	300
4	Airbus A350	280
5	Embraer E190	100
6	Bombardier CRJ900	90
7	ATR 72	70
8	Cessna 172	4
9	Gulfstream G650	14
10	Global 7500	19
11	Boeing 747	400
ACT_ID TYPE CAPACITY		
12	Airbus A380	550
13	Embraer E175	80
14	Bombardier CRJ900	90
15	Cessna Citation X	8

15 rows selected.

```
SQL> select * from aircraft;
```

ACID	AC_TYPE	MFG_DATE
1	1	01-JAN-22
2	2	15-MAY-20
3	3	30-NOV-19
4	1	20-AUG-21
5	4	10-APR-18
6	5	05-DEC-20
7	1	02-SEP-17
8	6	15-MAR-16
9	7	25-JUL-15
10	1	18-FEB-19
11	8	30-JUN-18

ACID	AC_TYPE	MFG_DATE
12	9	12-NOV-17
13	1	04-SEP-16
14	10	22-OCT-20
15	11	01-APR-19

15 rows selected.

```
SQL> select * from airfare;
```

AF_ID	ROUTE	FARE
1	1	200
2	2	150
3	3	180
4	4	120
5	5	140
6	6	250
7	7	300
8	8	180
9	9	280
10	10	350
11	11	400

AF_ID	ROUTE	FARE
12	12	320
13	13	280
14	14	380

14 rows selected.

```
SQL> select * from flight_schedule;
```

FL_ID	FLIGHT_DA
DEPARTURE	
ARRIVAL	
AIRCRAFT	NETFARE

1	15-MAY-24
01-MAY-24	08.00.00.000000 AM
01-MAY-24	10.30.00.000000 AM
1	1

FL_ID	FLIGHT_DA
DEPARTURE	
ARRIVAL	
AIRCRAFT	NETFARE

2	16-MAY-24
01-MAY-24	12.00.00.000000 PM
01-MAY-24	02.30.00.000000 PM
2	2

FL_ID	FLIGHT_DA
DEPARTURE	
ARRIVAL	
AIRCRAFT	NETFARE

3	17-MAY-24
01-MAY-24	04.00.00.000000 PM
01-MAY-24	06.30.00.000000 PM
3	3

```

FL_ID FLIGHT_DA
-----
DEPARTURE
-----
ARRIVAL
-----
AIRCRAFT   NETFARE
-----
4 18-MAY-24
01-MAY-24 08.00.00.000000 PM
01-MAY-24 10.30.00.000000 PM
4 4

```

```

FL_ID FLIGHT_DA
-----
DEPARTURE
-----
ARRIVAL
-----
AIRCRAFT   NETFARE
-----
5 19-MAY-24
01-MAY-24 10.00.00.000000 AM
01-MAY-24 12.30.00.000000 PM
5 5

```

```

FL_ID FLIGHT_DA
-----
DEPARTURE
-----
ARRIVAL
-----
AIRCRAFT   NETFARE
-----
6 20-MAY-24
01-MAY-24 02.00.00.000000 PM
01-MAY-24 04.30.00.000000 PM
6 6

```

```

FL_ID FLIGHT_DA
-----
DEPARTURE
-----
ARRIVAL
-----
AIRCRAFT   NETFARE
-----
7 21-MAY-24
01-MAY-24 06.00.00.000000 PM
01-MAY-24 08.30.00.000000 PM
7 7

```

```

FL_ID FLIGHT_DA
-----
DEPARTURE
-----
ARRIVAL
-----
AIRCRAFT   NETFARE
-----
8 22-MAY-24
01-MAY-24 10.00.00.000000 PM
01-MAY-24 11.30.00.000000 PM
8 8

```

```

FL_ID FLIGHT_DA
-----
DEPARTURE
-----
ARRIVAL
-----
AIRCRAFT   NETFARE
-----
9 23-MAY-24
01-MAY-24 09.00.00.000000 AM
01-MAY-24 11.30.00.000000 AM
9 9

```

```

FL_ID FLIGHT_DA
-----
DEPARTURE
-----
ARRIVAL
-----
AIRCRAFT   NETFARE
-----
10 24-MAY-24
01-MAY-24 01.00.00.000000 PM
01-MAY-24 03.30.00.000000 PM
10 10

```

```
SQL> select * from transaction;
```

```
TS_ID
-----
BOOKING_DATE
-----
PASSENGER  FLIGHT
-----
1
15-MAY-24 10.00.00.000000 AM
7          5
```

```
2
16-MAY-24 12.30.00.000000 PM
8          6
```

```
TS_ID
-----
BOOKING_DATE
-----
PASSENGER  FLIGHT
-----
3
17-MAY-24 03.45.00.000000 PM
9          7
```

```
4
18-MAY-24 06.20.00.000000 PM
```

```
TS_ID
-----
BOOKING_DATE
-----
PASSENGER  FLIGHT
-----
10         8
5
19-MAY-24 08.00.00.000000 PM
11         9
6
```

```
TS_ID
-----
BOOKING_DATE
-----
PASSENGER  FLIGHT
-----
12         10
7
21-MAY-24 01.45.00.000000 PM
13         11
```

```
TS_ID
-----
BOOKING_DATE
-----
PASSENGER  FLIGHT
-----
8
22-MAY-24 04.00.00.000000 PM
14         12
9
23-MAY-24 07.30.00.000000 PM
15         13
```

```
TS_ID
-----
BOOKING_DATE
-----
PASSENGER  FLIGHT
-----
10
24-MAY-24 11.00.00.000000 AM
16         5
11
25-MAY-24 02.15.00.000000 PM
```

```

      TS_ID
-----
BOOKING_DATE
-----
PASSENGER    FLIGHT
-----
20-MAY-24 09.30.00.000000 AM
      12      10
      7
21-MAY-24 01.45.00.000000 PM
      13      11

      TS_ID
-----
BOOKING_DATE
-----
PASSENGER    FLIGHT
-----
      8
22-MAY-24 04.00.00.000000 PM
      14      12
      9
23-MAY-24 07.30.00.000000 PM
      15      13

      TS_ID
-----
BOOKING_DATE
-----
PASSENGER    FLIGHT
-----
      10
24-MAY-24 11.00.00.000000 AM
      16      5
      11
25-MAY-24 02.15.00.000000 PM

```

```

      TS_ID
-----
BOOKING_DATE
-----
PASSENGER    FLIGHT
-----
      17      6
      12
26-MAY-24 05.30.00.000000 PM
      18      7

12 rows selected.

```

PL/SQL

Which aircraft type has highest capacity? Using stored function in PL/SQL:

```
SQL> CREATE OR REPLACE FUNCTION get_highest_capacity_aircraft_type RETURN VARCHAR2
2  IS
3      v_max_capacity NUMBER;
4      v_aircraft_type VARCHAR2(32);
5  BEGIN
6      -- Query to find the aircraft type with the highest capacity
7      SELECT MAX(capacity) INTO v_max_capacity
8      FROM aircraft_type;
9
10     -- Retrieve the name of the aircraft type with the highest capacity
11     SELECT type INTO v_aircraft_type
12     FROM aircraft_type
13     WHERE capacity = v_max_capacity;
14
15     -- Return the name of the aircraft type with the highest capacity
16     RETURN v_aircraft_type;
17 EXCEPTION
18     WHEN NO_DATA_FOUND THEN
19         RETURN NULL; -- Handle no data found scenario
20     WHEN OTHERS THEN
21         RETURN NULL; -- Handle other exceptions
22 END;
23
24 /
Function created.

SQL> -- Execute the function to get the highest capacity aircraft type
SQL> DECLARE
2      v_highest_capacity_aircraft_type VARCHAR2(32);
3  BEGIN
4      v_highest_capacity_aircraft_type := get_highest_capacity_aircraft_type();
5      DBMS_OUTPUT.PUT_LINE('Aircraft type with highest capacity: ' || v_highest_capacity_aircraft_type);
6  END;
7  /
Aircraft type with highest capacity: Airbus A380
PL/SQL procedure successfully completed.
```

Total number of tickets booked from 15th May 2024 to 23rd May 2024:

```
1  DECLARE
2      v_ticket_count INTEGER;
3      v_start_date DATE := TO_DATE('2024-05-15', 'YYYY-MM-DD');
4      v_end_date DATE := TO_DATE('2024-05-23', 'YYYY-MM-DD');
5  BEGIN
6      -- Open a cursor to select tickets booked within the specified date range
7      FOR c_ticket IN (
8          SELECT COUNT(*) AS ticket_count
9          FROM transaction t
10         JOIN passenger p ON t.passenger = p.ps_id
11         JOIN contact_details cd ON p.contacts = cd.cn_id
12         WHERE t.booking_date BETWEEN v_start_date AND v_end_date
13     ) LOOP
14         -- Assign the ticket count from the cursor to a variable
15         v_ticket_count := c_ticket.ticket_count;
16     END LOOP;
17     -- Display or use the ticket count as needed
18     DBMS_OUTPUT.PUT_LINE('Total tickets booked: ' || v_ticket_count);
19* END;
20 /
PL/SQL procedure successfully completed.

SQL> set serveroutput on
SQL> /
Total tickets booked: 8
PL/SQL procedure successfully completed.
```

All flights reaching India 20th May to 25th May:

```
SQL> DECLARE
2     v_person_count INTEGER := 0;
3     v_start_date DATE := TO_DATE('2024-05-20', 'YYYY-MM-DD');
4     v_end_date DATE := TO_DATE('2024-05-25', 'YYYY-MM-DD');
5 BEGIN
6     -- Open a cursor to iterate over transactions within the specified date range
7     FOR c_transaction IN (
8         SELECT t.passenger
9         FROM transaction t
10        JOIN flight_schedule fs ON t.flight = fs.fl_id
11        WHERE fs.flight_date BETWEEN v_start_date AND v_end_date
12    ) LOOP
13        -- Check if the passenger associated with the transaction is from India
14        BEGIN
15            SELECT COUNT(*)
16            INTO v_person_count
17            FROM passenger p
18            WHERE p.ps_id = c_transaction.passenger
19            AND p.nationality = 'India';
20
21            -- Increment the count if the passenger is from India
22            IF v_person_count > 0 THEN
23                v_person_count := v_person_count + 1;
24            END IF;
25        EXCEPTION
26            WHEN NO_DATA_FOUND THEN
27                -- Handle exception (e.g., log message or continue processing)
28                DBMS_OUTPUT.PUT_LINE('No passenger data found for transaction.');
```

Printing all details of Flight No. 5:

```
1  SELECT
2      p.ps_id,
3      p.name AS passenger_name,
4      p.age AS passenger_age,
5      p.nationality AS passenger_nationality,
6      t.ts_id AS transaction_id,
7      t.booking_date AS booking_date
8  FROM
9      passenger p
10 JOIN
11     transaction t ON p.ps_id = t.passenger
12 WHERE
13*    t.flight = 5
SQL> /
```


Passenger's age above 30 years and travelling in Flight No. 5:

```
1  SELECT
2      p.ps_id,
3      p.name AS passenger_name,
4      p.age AS passenger_age,
5      t.ts_id AS transaction_id,
6      t.booking_date AS booking_date,
7      fs.flight_date AS flight_date
8  FROM
9      passenger p
10 JOIN
11     transaction t ON p.ps_id = t.passenger
12 JOIN
13     flight_schedule fs ON t.flight = fs.fl_id
14 WHERE
15     fs.fl_id = 5
16*   AND p.age > 30
SQL> /
```

PS_ID	PASSENGER_NAME	PASSENGER_AGE	TRANSACTION_ID
BOOKING_DATE			
FLIGHT_DA			
16	Sophia Adams	31	10
24-MAY-24	11.00.00.000000 AM		
19-MAY-24			

Printing all details of passenger with a ticket:

```
SQL> SELECT
2      p.ps_id,
3      p.name AS passenger_name,
4      p.address AS passenger_address,
5      p.age AS passenger_age,
6      p.nationality AS passenger_nationality,
7      cd.email AS passenger_email,
8      cd.mobile AS passenger_mobile,
9      t.ts_id AS transaction_id,
10     t.booking_date AS booking_date,
11     fs.flight_date AS flight_date,
12     fs.departure AS departure_time,
13     fs.arrival AS arrival_time
14 FROM
15     passenger p
16 LEFT JOIN
17     contact_details cd ON p.contacts = cd.cn_id
18 LEFT JOIN
19     transaction t ON p.ps_id = t.passenger
20 LEFT JOIN
21     flight_schedule fs ON t.flight = fs.fl_id
22 WHERE
23     p.ps_id = 18;
```

```

1  SELECT
2      p.ps_id,
3      p.name AS passenger_name,
4      p.age AS passenger_age,
5      t.ts_id AS transaction_id,
6      t.booking_date AS booking_date,
7      fs.flight_date AS flight_date
8  FROM
9      passenger p
10 JOIN
11     transaction t ON p.ps_id = t.passenger
12 JOIN
13     flight_schedule fs ON t.flight = fs.fl_id
14 WHERE
15     fs.fl_id = 5
16*    AND p.age > 30
SQL> /

```

PS_ID	PASSENGER_NAME	PASSENGER_AGE	TRANSACTION_ID
16	Sophia Adams	31	10

BOOKING_DATE
FLIGHT_DA

Summary of flight from Maldives to India:

```

SQL> select fl_id,flight_date,fare,departure,arrival from ((flight_schedule inner join airfare on( flight_schedule.netfare=airfare.af_id))) inner join route on route=rt_id where airport='Maldives' and destina
tion='India';

```

FL_ID	FLIGHT_DA	FARE
6	20-MAY-24	250

DEPARTURE
ARRIVAL

01-MAY-24 02.00.00.000000 PM
01-MAY-24 04.30.00.000000 PM

Maintenance of Aircrafts:

```

SQL> select ACID ,mfg_date,type from aircraft inner join aircraft_type on aircraft.ac_type=aircraft_type.act_id where mfg_date< add_months(sysdate,-48);

```

ACID	MFG_DATE	TYPE
7	02-SEP-17	Boeing 737
10	18-FEB-19	Boeing 737
13	04-SEP-16	Boeing 737
3	30-NOV-19	Boeing 777
5	10-APR-18	Airbus A350
8	15-MAR-16	Bombardier CRJ900
9	25-JUL-15	ATR 72
11	30-JUN-18	Cessna 172
12	12-NOV-17	Gulfstream G650
15	01-APR-19	Boeing 747

10 rows selected.

CONCLUSION

The conclusion of the airline management system project represents a significant milestone in the journey towards modernizing and optimizing airline operations. Over the course of the project, a comprehensive system has been developed, encompassing a range of functionalities critical to the smooth functioning of an airline. These functionalities include reservation management, flight scheduling, ticketing, aircraft maintenance, and data management.

One of the primary achievements of the project is the enhanced efficiency achieved in flight planning and scheduling. By leveraging advanced algorithms and real-time data, the system can optimize routes, allocate resources effectively, and minimize delays, leading to improved on-time performance and operational reliability. This not only benefits the airline by reducing costs associated with disruptions but also enhances the overall travel experience for passengers.

Moreover, the implementation of streamlined booking processes has significantly improved the customer experience. With intuitive interfaces for reservation management and ticketing, passengers can easily book their flights, select seats, and complete payments hassle-free. This not only fosters customer satisfaction but also enhances the airline's reputation for providing convenient and reliable services.

Another critical aspect of the project's success is the optimization of resource allocation through effective maintenance management. By automating maintenance scheduling, tracking aircraft health data, and generating maintenance reports, the system ensures that aircraft remain in optimal condition, thereby minimizing downtime and maximizing operational efficiency. This proactive approach to maintenance not only enhances safety but also contributes to cost savings by reducing the need for unscheduled maintenance and repairs.

Looking ahead, the conclusion of the project marks the beginning of a new phase of ongoing monitoring, maintenance, and potential updates to the system. As the aviation industry continues to evolve, it will be essential to ensure that the system remains aligned with emerging trends, regulatory requirements, and customer expectations. By staying agile and adaptable, the airline management system will continue to play a vital role in optimizing airline operations and enhancing the overall travel experience for passengers in the years to come.