



Transform Yourself

AIRLINE TRAFFIC MANAGEMENT SYSTEM



PROJECT MEMBER

KOVARTHINI R - 23ADR092

PROBLEM STATEMENT

- The **Airline Traffic Management System** aims to streamline the operations of managing flights, bookings, and passengers in an airline.
- The system should handle these operations through a user-friendly interface, with data stored in a relational database (MySQL) to ensure data integrity and consistency
- The system should also manage common issues such as seat availability and ensure proper updates in the database when bookings are made or canceled.

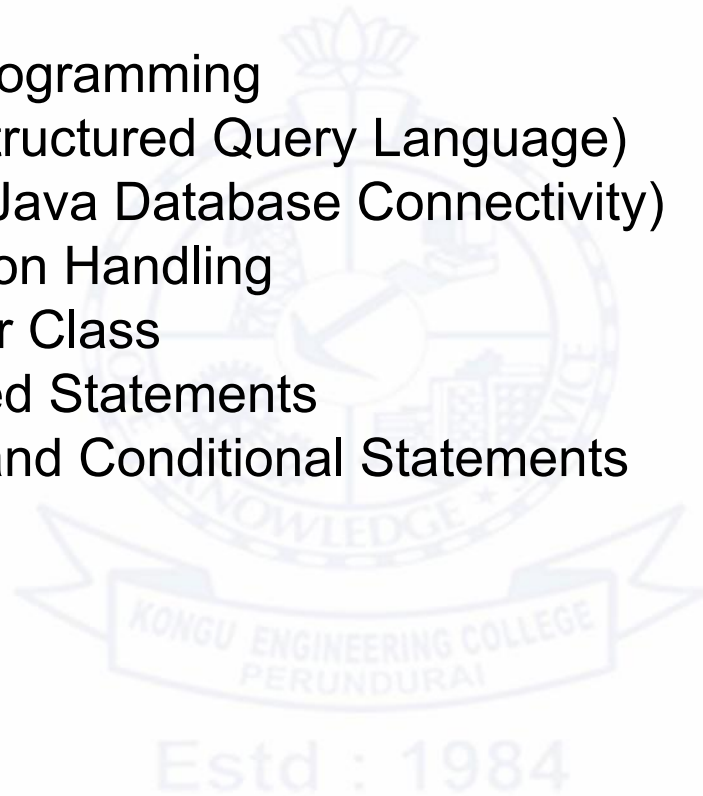
Estd : 1984

OBJECTIVES

- **Manage Flights:** Add and update flight details like flight number, origin, and available seats.
- **Simplify Bookings:** Allow passengers to book tickets with real-time seat updates.
- **View Bookings:** Display all bookings with passenger and flight details.
- **Centralized Data:** Store all data in a MySQL database for easy access and consistency.
- **User-Friendly:** Provide an intuitive interface for smooth interaction.
- **Boost Efficiency:** Automate processes to improve airline operations.

LANGUAGE CONCEPT USED

- Java Programming
- SQL (Structured Query Language)
- JDBC (Java Database Connectivity)
- Exception Handling
- Scanner Class
- Prepared Statements
- Loops and Conditional Statements



DATABASE DESIGN

CREATING TABLE FOR DATABASE Airline Traffic Management System

ch

nce MySQL80 x

Query Database Server Tools Scripting Help

Query 1 x

Limit to 1000 rows

```
1 • CREATE DATABASE AirlineDB1;
2
3  USE AirlineDB1;
4
5 • CREATE TABLE flights (
6     flight_number VARCHAR(10) PRIMARY KEY,
7     origin VARCHAR(50),
8     destination VARCHAR(50),
9     available_seats INT
10 );
11
12 • CREATE TABLE bookings (
13     booking_id INT AUTO_INCREMENT PRIMARY KEY,
14     passenger_name VARCHAR(50),
15     flight_number VARCHAR(10),
16     FOREIGN KEY (flight_number) REFERENCES flights(flight_number)
17 );
```

Automatic context help is
Use the toolbar to manually
for the current caret position
toggle automatic help

Context Help Snippets

Output

Action Output

#	Time	Action	Message
✓ 1	20:41:20	CREATE DATABASE AirlineDB1	1 row(s) affected
✓ 2	20:41:20	USE AirlineDB1	0 row(s) affected
✓ 3	20:41:20	CREATE TABLE flights (flight_number VARCHAR(10) PRIMARY KEY, origin VARCHAR(50), destination ...	0 row(s) affected
✓ 4	20:41:20	CREATE TABLE bookings (booking_id INT AUTO_INCREMENT PRIMARY KEY, passenger_name VARC...	0 row(s) affected

DATABASE DESIGN

COLUMNS IN THE TABLE:

- **Flights Table:**

- 1.flight_number(VARCHAR(10))
- 2.origin(VARCHAR(50))
- 3.destination (VARCHAR(50))
- 4.available_seats (INT)

- **Bookings Table:**

- 1.booking_id (INT, AUTO_INCREMENT)
- 2.passenger_name (VARCHAR(50))
- 3.flight_number (VARCHAR(10))

Estd : 1984

OUTPUT

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\java> javac AirlineTrafficManagementSystem.java
PS C:\java> java -cp ".;C:\Users\Megala\Downloads\mysql-connector-j-9.1.0\mysql-connector-j-9.1.0\mysql-connector-j-9.1.0.jar" AirlineTrafficManagementSystem

--- Airline Traffic Management System ---
1. Add Flight
2. View Flights
3. Book Ticket
4. View Bookings
5. Exit
Enter your choice: 1
Enter flight number: DD101
Enter origin: US
Enter destination: UK
Enter available seats: 10
Flight added successfully.
```


OUTPUT

--- Airline Traffic Management System ---

1. Add Flight
2. View Flights
3. Book Ticket
4. View Bookings
5. Exit

Enter your choice: 2

Flight Number: AA101

Origin: india

Destination: paris

Available Seats: 4

Flight Number: AZ202

Origin: london

Destination: newyork

Available Seats: 5

Flight Number: BA202

Origin: london

Destination: new york

Available Seats: 5

Flight Number: CA101

Origin: russia

Destination: britain

Available Seats: 2

Flight Number: DD101

Origin: US

Destination: UK

Available Seats: 10

Flight Number: FF103

Origin: India

Destination: Paris

Available Seats: 14

--- Airline Traffic Management System ---

1. Add Flight
2. View Flights
3. Book Ticket
4. View Bookings
5. Exit

Enter your choice: 3

Enter passenger name: Giri

Enter flight number: DD101

Booking successful!

OUTPUT

--- Airline Traffic Management System ---

1. Add Flight
2. View Flights
3. Book Ticket
4. View Bookings
5. Exit

Enter your choice: 4

Booking ID: 1

Passenger Name: abc

Flight Number: BA202

Booking ID: 2

Passenger Name: kova

Flight Number: AA101

Booking ID: 3

Passenger Name: keerthi

Flight Number: CA101

Booking ID: 4

Passenger Name: def

Flight Number: BA202

Booking ID: 5

Passenger Name: def

Flight Number: BA202

Booking ID: 6

Passenger Name: Megala

Flight Number: FF103

Booking ID: 7

Passenger Name: Giri

Flight Number: DD101

--- Airline Traffic Management System ---

1. Add Flight
2. View Flights
3. Book Ticket
4. View Bookings
5. Exit

Enter your choice: 5

Exiting... Goodbye!

PS C:\java> |

1984

OUTPUT

Table: Bookings

The screenshot displays the MySQL Workbench interface for a local instance of MySQL 8.0. The 'Query 1' tab is active, showing the following SQL query:

```
1 • USE AirlineDB5;  
2 • select * FROM bookings4;  
3  
4  
5
```

The 'Result Grid' shows the output of the query, displaying 7 rows of data from the 'bookings4' table. The columns are 'booking_id', 'passenger_name', and 'flight_number'.

booking_id	passenger_name	flight_number
1	abc	BA202
2	kova	AA101
3	keerthi	CA101
4	def	BA202
5	def	BA202
6	Megala	FF103
7	Giri	DD101

The 'Output' tab at the bottom shows the execution log, indicating that the query was executed successfully and returned 7 rows.

#	Time	Action	Message
7	08:34:12	USE AirlineDB5	0 row(s) affected
8	08:34:12	select * FROM flights4 LIMIT 0, 1000	6 row(s) returned
9	08:35:02	USE AirlineDB5	0 row(s) affected
10	08:35:02	select * FROM flights4 LIMIT 0, 1000	6 row(s) returned
11	08:35:13	USE AirlineDB5	0 row(s) affected
12	08:35:13	select * FROM bookings4 LIMIT 0, 1000	7 row(s) returned

OUTPUT

Table: **Flights**

The screenshot displays the MySQL Workbench interface for a local instance of MySQL 8.0. The left sidebar contains a 'Navigator' pane with sections for 'MANAGEMENT' (Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, Data Import/Restore), 'INSTANCE' (Startup / Shutdown, Server Logs, Options File), and 'PERFORMANCE' (Dashboard, Performance Reports, Performance Schema Setup). Below this are tabs for 'Administration' and 'Schemas', and an 'Information' section showing 'No object selected'.

The main workspace is titled 'Query 1' and contains the following SQL script:

```
1 • USE AirlineDB5;  
2 • select * FROM flights4;  
3  
4  
5
```

The 'Result Grid' tab is active, showing the output of the query. The results are displayed in a table with the following columns: flight_number, origin, destination, and available_seats.

flight_number	origin	destination	available_seats
AA101	india	paris	4
AZ202	london	newyork	5
BA202	london	new york	5
CA101	ruussia	britain	2
DD101	US	UK	9
FF103	India	Paris	14
NULL	NULL	NULL	NULL

The bottom pane shows the 'Output' tab with a table of execution actions and their results:

#	Time	Action	Message
9	08:35:02	USE AirlineDB5	0 row(s) affected
10	08:35:02	select * FROM flights4 LIMIT 0, 1000	6 row(s) returned
11	08:35:13	USE AirlineDB5	0 row(s) affected
12	08:35:13	select * FROM bookings4 LIMIT 0, 1000	7 row(s) returned
13	08:56:17	USE AirlineDB5	0 row(s) affected
14	08:56:17	select * FROM flights4 LIMIT 0, 1000	6 row(s) returned

CONCLUSION

- In conclusion, the **Airline Traffic Management System** efficiently handles flight and booking management by using core programming concepts like classes, encapsulation, and abstraction.
- It connects to a MySQL database to store and retrieve flight details and bookings..
- This approach makes the system scalable, maintainable, and secure for managing airline operations.

Estd : 1984

*THANK
YOU*



Estd : 1984