

AIRLINE RESERVATION SYSTEM

INTRODUCTION:

The Airline Reservation System is a database management system designed to facilitate the process of booking airline tickets, managing flight schedules, storing passenger records, and processing payments. The increasing complexity and volume of air travel demand a reliable and efficient system to handle real-time operations with minimal human intervention. This project focuses on the back-end database functionalities of an airline reservation system using Structured Query Language (SQL).

ABSTRACT:

This project aims to develop the database for an Airline Reservation System using SQL, enabling users to book flights, cancel reservations, view flight schedules, and retrieve passenger information efficiently. The database system ensures data integrity, security, and consistency, and is designed with relational database principles. It includes the creation of multiple interrelated tables such as Flights, Passengers, Reservations, Payments, and Airports. Queries and stored procedures are used to manage operations like booking, cancellations, and availability checks. The goal is to demonstrate how SQL can be used to support critical operations in a real-world airline system.

TOOLS USED:

Database: SQL server/ SQLite (any relational DBMS)

SQL: Structured Query Language (DDL, DML, DCL, and TCL)

ER Diagram Tool: dbdiagram.io

Operating System: Windows

Steps Involved in Building the Project:

Day 1 & 2: Planning and Database Design

Requirement Analysis

View flight schedules

Passenger and flight data management

Design Entity-Relationship (ER) Diagram

Key Entities:

Flights

Customers

Seats

Payments

Airports

Design the Database Schema

Day 3&4: Implementation and Testing

Write SQL Scripts

Use DDL to create tables with keys and constraints

Use DML to insert sample data

Use SELECT queries to view data

Use JOINS to fetch relational data

Day 5&6: Implement Core Operations

Query to book a flight (INSERT into Reservations)

Cancel booking (UPDATE status in Reservations)

Check flight availability (SELECT with conditions)

Payment record insertion (INSERT into Payments)

Day 7&8: Testing

Testing & Run queries to simulate booking and cancellation

Test foreign key constraints and data integrity

Handle edge cases like overbooking

Day 9 & 10: Optimization & Finalization

Index frequently searched columns

Create views for common queries (e.g., upcoming flights)

Create stored procedures for booking logic

Conclusion:

The Airline Reservation System using SQL demonstrates how relational databases can effectively manage complex data relationships and real-time operations. By separating concerns into multiple tables and using SQL for querying and transactions, the system provides a robust foundation for further integration with front-end applications. This project lays the groundwork for developing a complete airline management system and shows the significance of database design and operations in modern software systems.