Project Title: Airline Reservation System

Introduction

This project implements a simplified Airline Reservation System in SQL. It manages flight schedules, customer data, bookings, payments, and waitlists. The goal is to simulate real-world airline operations using SQL queries, triggers, and views.

Abstract

This project demonstrates relational database design for airline booking management by building a complete SQL-based Airline Reservation System. The system includes schema creation, normalization up to Third Normal Form, and insertion of realistic sample data for customers, flights, seats, bookings, payments, and waitlists. Core business logic is implemented through SQL queries for flight search, available seat tracking, booking summaries, and revenue analysis, while automation is handled using triggers that mark seats as booked on reservation, free them on cancellation, promote waitlisted passengers fairly based on priority and request time, generate payments automatically on booking, and issue refunds upon cancellations. Views such as Flight Availability, Booking Summary, and Revenue Report provide analytical insights in real time. Data consistency is maintained through constraints like primary keys, foreign keys, unique columns, and ENUM fields, while normalization ensures reduced redundancy and better integrity. Testing involved multiple scenarios including new bookings, cancellations, overbooking with waitlists, and refunds, all of which validated the system's correctness and reliability. The database design highlights scalability, integrity, and automation, closely mimicking real-world airline workflows, and demonstrates advanced SQL skills in schema design, triggers, and reporting.
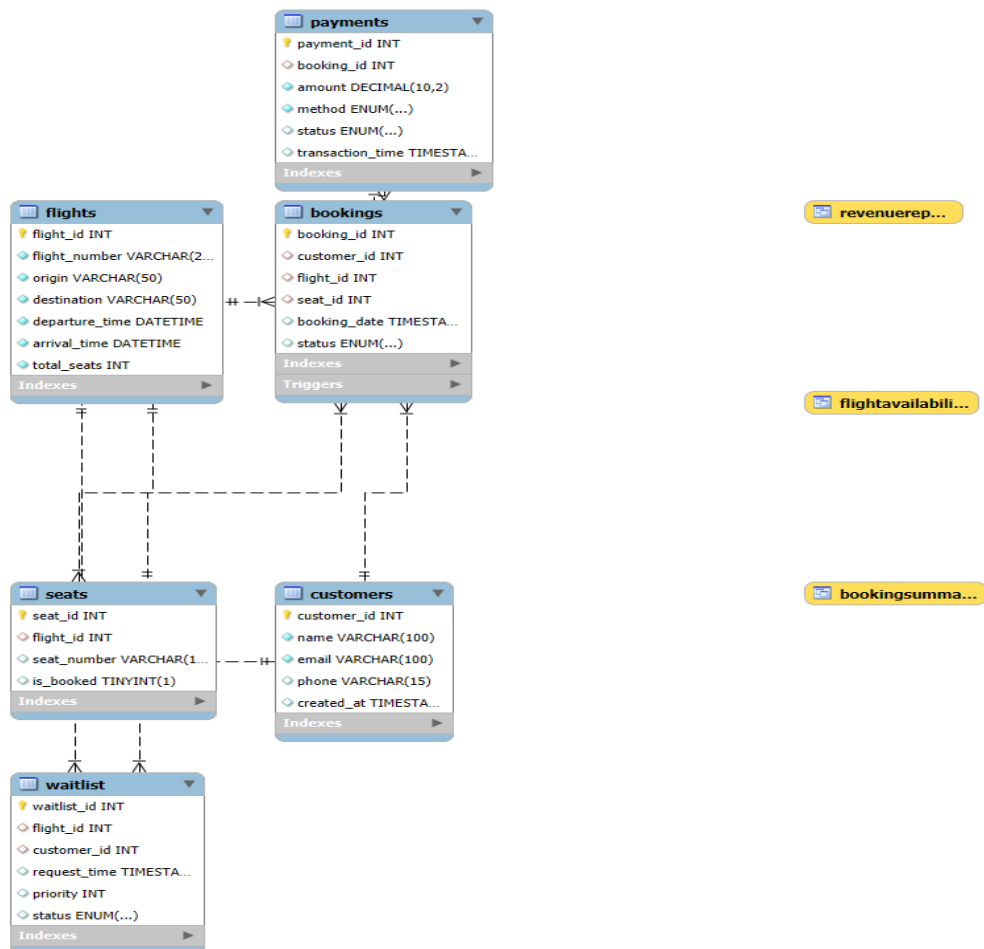
Tools Used

1. MySQL Workbench -SQL (DDL, DML, Views, Triggers)
2. Dataset with sample flights, customers, bookings, payments.

Steps Involved in Building the Project

1. Schema Design: Designed tables for Flights, Customers, Bookings, Seats, Payments, and Waitlist.
2. Normalization: Applied 3NF and defined primary/foreign keys and constraints.
3. Data Insertion: Inserted realistic sample data for customers, flights, and bookings.
4. Queries: Wrote SQL queries for flight search, seat availability, and booking reports.
5. Triggers: Added automation for seat allocation, cancellations, waitlist handling, payments, and refunds.
6. Views: Built views for flight availability, booking summary, and revenue reports.
7. Testing: Verified triggers and queries with different booking and cancellation scenarios.

ER Diagram

payments
- payment_id INT
- booking_id INT
- amount DECIMAL(10,2)
- method ENUM(...)
- status ENUM(...)
- transaction_time TIMESTA...
- Indexes

flights
- flight_id INT
- flight_number VARCHAR(2...
- origin VARCHAR(50)
- destination VARCHAR(50)
- departure_time DATETIME
- arrival_time DATETIME
- total_seats INT
- Indexes

bookings
- booking_id INT
- customer_id INT
- flight_id INT
- seat_id INT
- booking_date TIMESTA...
- status ENUM(...)
- Indexes
- Triggers

revenuerep...

flightavailabili...

seats
- seat_id INT
- flight_id INT
- seat_number VARCHAR(1...
- is_booked TINYINT(1)
- Indexes

customers
- customer_id INT
- name VARCHAR(100)
- email VARCHAR(100)
- phone VARCHAR(15)
- created_at TIMESTA...
- Indexes

bookingsumma...

waitlist
- waitlist_id INT
- flight_id INT
- customer_id INT
- request_time TIMESTA...
- priority INT
- status ENUM(...)
- Indexes

Conclusion

The Airline Reservation System project demonstrates how SQL can effectively solve real-world business problems by simulating the essential operations of airline management. Through careful schema design, normalization, and the use of constraints, the project ensures data integrity and reliability. Automation with triggers streamlines tasks such as seat allocation, cancellations, waitlist handling, payments, and refunds, reducing manual intervention and errors. Views provide meaningful insights into seat availability, booking summaries, and revenue reports, supporting decision-making and analysis. Overall, this project strengthens practical SQL skills in database design, query optimization, and business logic implementation, making it valuable for professional applications.