Jaypee Institute of Information Technology, Noida

PROJECT SYNOPSIS


# Travel Booking and Management System

Submitted to:

Prof. Sumeshwar Singh

Dr. Sonal

Dr. Anuja Arora

Submitted By:

Aditya Pandey, 2401030027

Shreyansh Rajat, 2401030020

Tanay Saxena, 2401030150

Course Name: Database Managemnt Lab

Program: BTech CSE

3rd Semester, 2nd Year

# Contents

# 1 Title of the Project

**Travel Booking System: A Comprehensive Database Management Solution for Integrated Flight and Hotel Reservations**

This project focuses on developing a robust relational database system for managing travel bookings, emphasizing SQL queries for efficient data handling and analysis. It integrates real-world datasets to simulate a practical travel agency environment, aligning with the DBMS Lab curriculum.

# 2 Introduction & Problem Statement

## 2.1 Introduction

In the modern era of globalization and digital transformation, the travel and tourism industry has experienced exponential growth. With millions of people traveling for business, leisure, or personal reasons annually, the demand for efficient booking systems has surged. A Travel Booking System (TBS) is a computerized platform that allows users to search, book, and manage travel services like flights, hotels, and packages seamlessly.

The system leverages Database Management Systems (DBMS) to store, retrieve, and analyze vast amounts of data related to users, itineraries, availability, and payments. By employing relational databases, TBS ensures data integrity, reduces redundancy, and enables quick queries for real-time decision-making. This project draws inspiration from real-world applications, such as those used by online travel agencies (OTAs) like Booking.com or Expedia, but focuses primarily on the backend database layer to demonstrate SQL proficiency.

The tourism sector contributes significantly to global GDP, with projections estimating it to reach $11 trillion by 2025 (**?** ). However, the backbone of these systems is efficient data management, which this project aims to explore through hands-on implementation.

## 2.2 Problem Statement

Despite advancements in technology, many travel booking processes still suffer from inefficiencies. Manual systems are prone to human errors, such as double bookings or incorrect availability checks, leading to customer dissatisfaction and financial losses. Key challenges include:

- **Data Fragmentation**: Information on flights, hotels, and users is often stored in silos, making integrated queries difficult.
- **Scalability Issues**: Handling peak-season bookings requires efficient indexing and optimization, which legacy systems lack.

- **Integrity and Security**: Ensuring referential integrity (e.g., no orphan bookings) and preventing data anomalies is crucial.
- **Analytical Gaps**: Travel agencies need insights like pricing trends or cancellation patterns, but manual analysis is time-consuming.
- **Real-Time Processing**: Delays in updating availability can result in overbookings, as seen in industry reports where cancellation rates hover around 20-30% due to poor management (4).

In academic contexts, such as this DBMS Lab, the problem extends to applying theoretical concepts like normalization, joins, and subqueries to practical scenarios. Existing systems often overlook advanced SQL features for optimization, leading to suboptimal performance in large datasets.

This project addresses these by designing a SQL-centric database that simulates a travel booking ecosystem, using datasets with thousands of records to test real-world applicability.

# 3 Objectives of the Project

The primary goal is to create a functional database prototype that demonstrates mastery of DBMS concepts covered in the lab calendar, from CRUD operations (Week 1) to PL/SQL triggers (Week 12).

Specific objectives include:

- **Database Design**: Develop an Entity-Relationship (ER) model and normalize the schema to at least 3NF to minimize redundancy.
- **Data Integrity Enforcement**: Implement keys, constraints (entity and referential integrity, Week 2), and triggers to maintain data consistency.
- **Query Implementation**: Write advanced SQL queries using SELECT with WHERE, aggregates, joins (Weeks 3-4), subqueries (Weeks 6,9), grouping/sorting (Week 7), and set operators (Week 9).
- **Automation with PL/SQL**: Incorporate procedures, functions, cursors (Week 10), and triggers (Week 12) to automate tasks like availability checks and reports.
- **Analytical Features**: Enable data analysis for trends, such as average booking costs or cancellation rates, using window functions and aggregates.
- **Performance Optimization**: Use indexes, views, and explain plans to handle large datasets efficiently.
- **Integration of Real Data**: Import and query public datasets to simulate realistic scenarios, ensuring the system can handle 1000+ records.
- **Documentation and Evaluation**: Prepare scripts and reports for lab tests (Weeks 8,13) and PBL evaluation (Week 14).

These objectives align with the course's progression, building from basic SQL to advanced PL/SQL, and aim to provide a scalable solution for travel management.

# 4 Proposed System / Methodology

## 4.1 System Overview

The proposed TBS is a backend-focused database system without a full frontend, emphasizing SQL for all operations. It models entities like Users, Flights, Hotels, Bookings, and Payments, interconnected via relationships.

## 4.2 Methodology

### 4.2.1 Requirement Analysis

Gather requirements from industry standards, such as handling user profiles, searching availabilities, and generating reports (1). Analyze datasets like Hotel Booking Demand (Kaggle) for structure (2).

### 4.2.2 Database Design

- **ER Diagram**: Entities include User (ID, Name, Email), Flight (ID, Origin, Destination, Date, Seats), Hotel (ID, Location, Rooms, Price), Booking (ID, UserID, FlightID/HotelID, Status), Payment (ID, BookingID, Amount). - **Normalization**: Apply 1NF-3NF to eliminate anomalies, e.g., separate addresses into a Location table. - **Constraints**: Primary keys, foreign keys, CHECK for positive prices.

### 4.2.3 Implementation

- **DDL Scripts**: Create tables with integrity constraints (Week 2-3). - **DML Operations**: Bulk inserts from CSV datasets (e.g., 119,390 hotel records) (2). - **Queries**: - Basic: SELECT flights WHERE destination = 'Delhi' ORDER BY price. - Advanced: Nested subqueries for available hotels (NOT EXISTS in bookings), joins for user itineraries. - Aggregates: AVG(price) GROUP BY month HAVING COUNT > 10. - Set Operators: UNION for combined flight/hotel searches. - Window Functions: RANK() OVER (PARTITION BY destination ORDER BY rating DESC) for recommendations. - **PL/SQL**: - Procedure: To process bookings with transaction control. -

Function: Calculate trip costs using scalar functions. - Cursor: For batch reporting on cancellations. - Trigger: Update seats/rooms on INSERT/DELETE in bookings.

### 4.2.4 Testing and Optimization

- Test with sample data for edge cases (e.g., overbookings). - Optimize using indexes on frequent columns (e.g., dates) and views for complex queries. - Evaluate performance with EXPLAIN PLAN.

This methodology follows a waterfall approach, from design to testing, tailored to the lab timeline.

# 5 Expected Outcomes

Upon completion, the project will yield:

- A fully functional database schema with scripts for creation, population, and querying.
- Demonstrated efficiency in handling large datasets, e.g., quick response times for availability checks.
- Analytical insights, such as reports on booking trends (e.g., peak seasons via GROUP BY).
- Robust error handling through constraints and triggers, reducing invalid data by 100%.
- Documentation including ER diagrams, query examples, and performance metrics.
- Alignment with lab evaluations: Ready for Lab Test-1 (Week 8) on subqueries and Lab Test-2 (Week 13) on PL/SQL.
- Potential extensions: Integration with simple scripts for simulated user interfaces.

Overall, this will enhance understanding of DBMS in real applications, potentially reducing booking errors in simulated scenarios by 50% compared to manual methods (5).

# 6 Tools & Technologies to be Used

- **DBMS Software**: Oracle Database or PostgreSQL for relational management and PL/SQL support. - **Development Tools**: SQL Developer or DBeaver for scripting and execution. - **Datasets**: Kaggle Hotel Booking Demand (119,390 records) (2) and adapted flight data from similar sources. - **Languages**: SQL (core), PL/SQL (automation). - **Other**: CSV importers for data loading, EXPLAIN for optimization.

No additional hardware required; standard lab setup suffices.

# 7 References

# References

[1] Synopsis of Tour and Travel Booking System. Scribd.

[2] Hotel Booking Demand Dataset. Kaggle.

[3] Travel Booking System. ResearchGate.

[4] Data Article: Hotel Booking Demand Datasets. ScienceDirect.

[5] Tourism Database Management System. Scribd.

[6] The Design of Relational Database and NoSQL in Travel Agency. ResearchGate.