Hotel Booking Analytics System (Database + AI Integration)

Technical Report

Abstract

This project presents a unified data-driven analytics system designed to help hotel managers understand customer behavior, booking patterns, revenue distribution, and operational performance.
Using a real-world hotel booking dataset, we built a fully normalized MySQL relational database, implemented analytical SQL queries, optimized performance with indexing and SQL Views, and integrated an AI-powered text-to-SQL agent using LangChain and Google Gemini. The system enables both manual SQL-based analysis and natural-language querying, allowing non-technical users to extract insights directly from the database. The final solution provides actionable information about cancellations, seasonality, room popularity, revenue by segment, and pricing efficiency.

Problem & Dataset

Hotels receive thousands of bookings per year, and understanding trends—such as cancellations, seasonality, room demand, and revenue sources—is essential for improving revenue management and customer satisfaction.
However, manually analyzing this data is difficult, especially for managers without SQL knowledge.

To address this challenge, we used a hotel booking dataset containing attributes such as hotel type, customer demographics, stay length, booking channels, room assignments, market segments, meal plans, ADR (average daily rate), cancellations, and arrival dates.
The dataset represents real patterns from both **City Hotel** and **Resort Hotel**, enabling multi-dimensional analysis.

Our project goal was to transform this dataset into a structured relational database and build tools (SQL + AI agent) to extract meaningful, business-oriented insights.

Database Design

We cleaned and normalized the dataset to achieve 3NF, eliminating redundancy and ensuring data integrity.
Core entities include **bookings, customers, hotels, rooms, market_segments, distribution_channels, meals, deposit_types**.

Normalization Summary

- **1NF**: removed repeating groups, ensured atomic values.
- **2NF**: separated lookup tables (meals, rooms, segments, etc.) to remove partial dependencies.
- **3NF**: ensured all non-key attributes depend only on primary keys.

ER Diagram Summary

The central table **bookings** connects to all dimension tables through foreign keys:

- `hotel_id` → hotels
- `customer_id` → customers
- `room_id` → rooms
- `market_segment_id` → market_segments
- `meal_id` → meals
- `distribution_channel_id` → distribution_channels

This star-like schema supports flexible analytical queries and minimizes update anomalies.

Implementation

We implemented the schema in MySQL with explicit primary keys, foreign keys, NOT NULL constraints, CHECK constraints (where appropriate), and consistent data types.
Data was loaded from the CSV file into normalized tables.

Key implementation steps:

1. Created all lookup tables and populated them with distinct values.
2. Loaded the bookings table with references to foreign keys.
3. Applied integrity constraints to ensure valid hotel types, segment codes, and room identifiers.
4. Verified referential integrity using foreign key enforcement.

This structured implementation supports efficient querying and ensures data reliability.

SQL Analysis

We analyzed five core business problems using SQL queries involving joins, grouping, aggregation, filtering, and ordering.

1. Cancellation Rates

We computed cancellation percentages by hotel type.
Result: City Hotel has a significantly higher cancellation rate than Resort Hotel.

2. Monthly Occupancy Trends

We grouped bookings by month and hotel type.
Result: City Hotel peaks in summer months (June–August), while Resort Hotel has more stable seasonal patterns.

3. Popular Room Types

We counted bookings per room code.
Result: Room type “A” is the most frequently booked, followed by “D” and “E”.

4. Revenue by Market Segment

Revenue was calculated using `adr × total nights`.
Result: Online TA generates the highest total revenue by a large margin.

5. Meal Plan & Pricing Analysis

We computed average ADR for each meal plan by hotel type.
Result: HB achieves the highest ADR, while BB is the most commonly used plan.

Optimization

To improve performance and simplify analysis:

1. Indexing

We added indexes on:

- `hotel_id`
- `market_segment_id`
- `arrival_date_month`
- `room_id`
  These improved query execution speed for GROUP BY and JOIN operations.

2. SQL Views

We created analytical SQL Views to pre-aggregate important metrics:

- `view_cancel_rates` – cancellation statistics
- `view_monthly_occupancy` – monthly hotel bookings
- `view_room_popularity` – bookings per room code
- `view_revenue_segments` – revenue per market segment
- `view_meal_plan_adr` – ADR by meal plan and hotel type

Views provided consistent outputs for both SQL analysts and the AI agent.

AI Integration

We integrated a text-to-SQL AI agent using **LangChain** and **Google Gemini**, enabling natural-language interaction with the database.

How it works:

1. User asks a question in English (e.g., *"Which month has the highest bookings?"*).
2. LangChain constructs a prompt with schema + examples.
3. Gemini generates a valid SQL query.
4. Python executes the query using SQLAlchemy.
5. Results are visualized using Pandas and Matplotlib.

Benefits:

- Non-technical users can analyze data without writing SQL.
- The AI agent handles complex queries across multiple tables.
- Views reduce errors and improve generation accuracy.

This integration turns the database into an intelligent analytical system.

Results

Our system produced several valuable insights:

- **City Hotel has the highest cancellation rate** (41.7%).
- **Online TA is the dominant revenue source**, with more than double the revenue of Offline TA/TO.
- **Room type "A" is the most popular**, significantly outperforming others.
- **HB meal plan yields the highest ADR**, indicating strong revenue efficiency.
- **City Hotel occupancy peaks in August**, with the highest monthly volume.

These insights demonstrate the system's usefulness for strategic decision-making.

Conclusion & Future Work

This project successfully combined database engineering, SQL analytics, and AI-driven query generation into one unified hotel analytics system.
We designed a clean relational database, optimized key queries, generated actionable insights, and implemented an intelligent text-to-SQL agent.

Future improvements:

- Deploy the system as a web dashboard for managers.
- Expand the AI agent with more examples for higher accuracy.
- Implement caching and query optimization for faster responses.
- Integrate predictive models (e.g., cancellation prediction or revenue forecasting).

Overall, the project demonstrates how traditional SQL databases and modern AI tools can be combined to enhance business analytics and decision-making.