DATA WAREHOUSING PROJECT

**Booking Brilliance in Data Accommodation**



**Submitted by:**

1.Akash Thallada

2.Dharma Rakshak Tadi

3.Ravindra Kumar Velidandi

4.Sai Mithil Sagi

5.Sai Charan Dasari

6.Maruthi Sai Vishnu Varma Vanapala

**Executive Summary**

The Hotel Booking System in Data Warehousing, known as "DataStay," is a groundbreaking platform designed to transform the hospitality sector. By harnessing the capabilities of data warehousing technology, DataStay integrates and analyzes vast amounts of information to elevate every facet of hotel booking, management, and customer experience. Through data-driven decision-making, the system provides hotel owners with actionable insights for optimizing revenue and operational efficiency. With a focus on customer-centricity, DataStay tailors each guest's experience through personalized recommendations, targeted promotions, and a streamlined booking process. The platform also optimizes inventory management, ensuring hotels can maximize occupancy rates and dynamically adjust pricing. Real-time updates on availability and pricing, coupled with robust fraud prevention measures, enhance transparency and security. DataStay's scalability, seamless integration, and benefits such as increased revenue, operational efficiency, competitive advantage, and customer retention collectively position it as a pioneering force in reshaping the landscape of hotel booking systems.

**Problem Statement**

In the realm of contemporary hotel management, the absence of a robust data warehousing solution has led to significant operational inefficiencies and missed opportunities for revenue optimization. Traditional systems lack the capacity to harness and analyze vast datasets, resulting in suboptimal decision-making processes. The hospitality industry is grappling with the challenge of offering personalized guest experiences without the comprehensive insights that data warehousing can provide. Current inventory management practices often fall short in dynamically adjusting to demand fluctuations, leading to underutilization of resources and potential revenue loss. Additionally, the lack of real-time updates on availability and pricing poses a hurdle to maintaining transparency in the booking process. Fraudulent activities pose a persistent threat, necessitating a more sophisticated approach to security. This problem statement underscores the pressing need for a data warehousing solution like DataStay to address these challenges and propel the hotel industry into a new era of efficiency, revenue maximization, and personalized customer service.

**Literature Review:**

Our project's main goal was to create a data warehouse model that was especially designed to be used in the construction of an extensive hotel booking database. To successfully combine the relevant data, this project required integrating the Bookings, Hotel, Location, and Room tables. A star join operation was the main method used to do this. The approaches used in current warehousing initiatives are examined in this literature study, which also highlights the need of carefully linking tables to improve the general efficacy and efficiency of data administration in the hospitality industry.

1. Which Hotels have the best average rating?
2. Which Hotels have highest number of rooms?
3. To observe the trend of total price spent on hotel bookings from 2015 to 2022.
4. Which type of rooms generates more revenue for the Hotels?
5. Which state has the best Average booking price?
6. Total number of Booking by State from the year 2015 to 2022?

**Data collection and Preparation:**

The data was gathered from various sources, including Git Hub, Kaggle, and booking data sets. It was then extracted, cleaned, preprocessed, and analyzed. To address the current issue, the data was formatted, imported into Oracle SQL, tables were created, and several queries were run to gain insight. The following are the data sets that we have extracted:

Hotels data set: This data set consists of information such as hotel id, hotel name, ratings, address, rooms available and location id.

Location data: This has information of the locations of the hotels all over India which include information such as, location id, state name, city of the hotel and Pin-code.

Customer data: This data set has information about individual customers which include customer id, first name, last name, email id, phone number, payment details.

Rooms dataset: this has information such as the room id, and room type.

**Schema:**

A yellow and green diagram

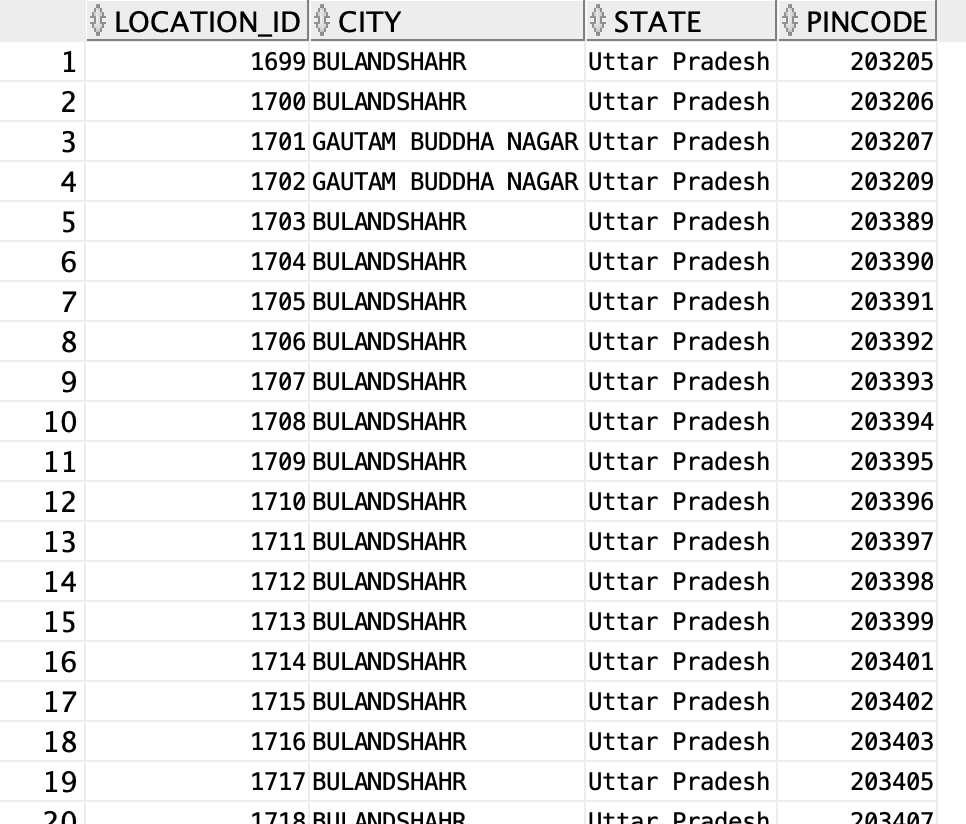
Description automatically generated with medium confidence

Below are the queries that are used to create the above tables in the oracle database:

**Tables & Structures:**

1. **Location Table:**

CREATE TABLE Location (  
    location\_id INT PRIMARY KEY,  
    city VARCHAR(255),  
    State VARCHAR(255),  
    pincode INT UNIQUE  
);



**2. Hotels table:**

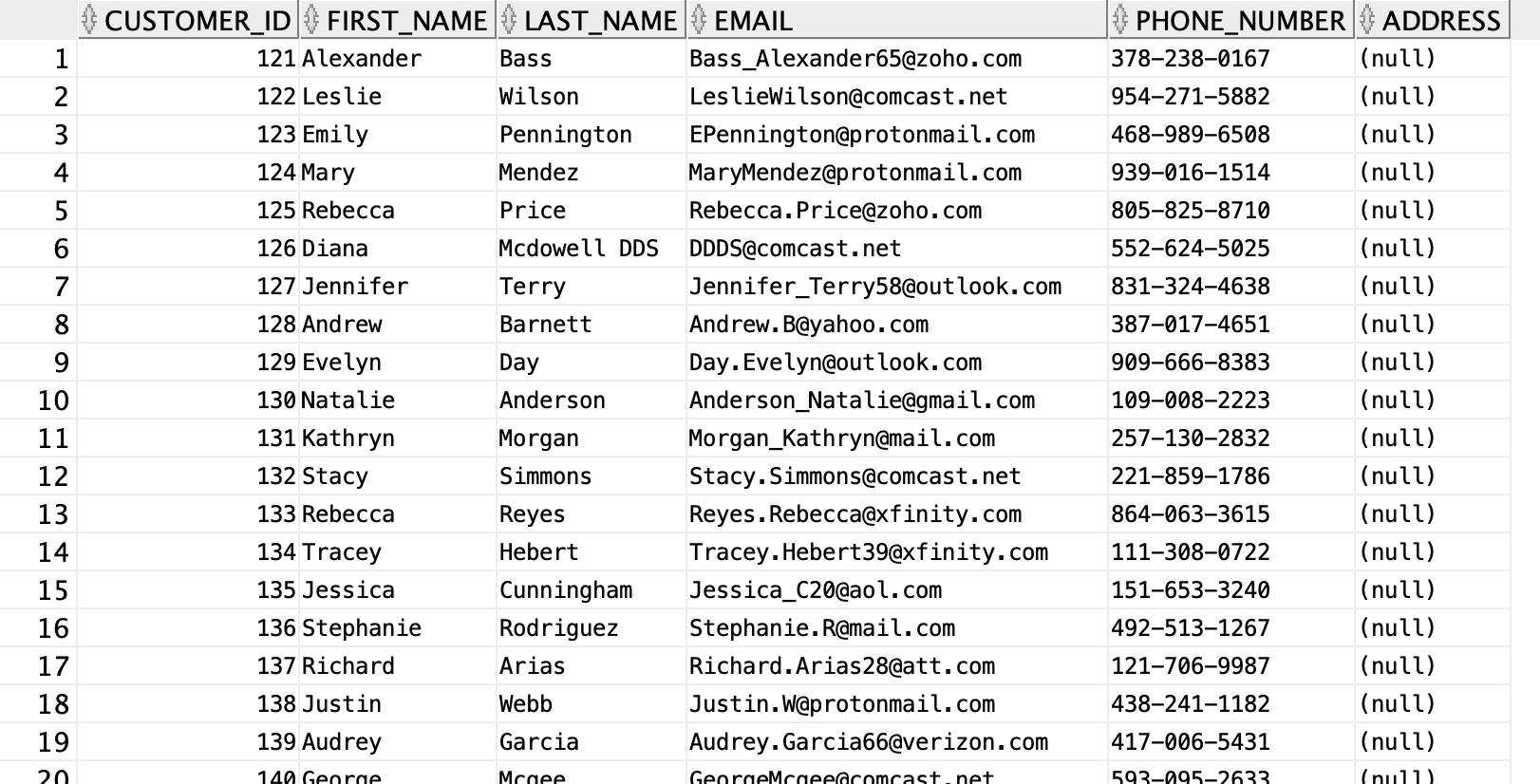
 CREATE TABLE Hotels (  
  hotel\_id INT PRIMARY KEY,  
  hotel\_name VARCHAR(255),  
  location\_id INT,   
  address VARCHAR(255),  
  phone VARCHAR(20),  
  number\_of\_rooms INT,  
  Rating INT,  
  FOREIGN KEY (location\_id) REFERENCES Location(location\_id)  
);

A screenshot of a computer

Description automatically generated

**3.Customers Table:**

 CREATE TABLE Customers (  
  customer\_id INT PRIMARY KEY,  
  first\_name VARCHAR(255),  
  last\_name VARCHAR(255),  
  email VARCHAR(255),  
  phone\_number VARCHAR(20),  
  address VARCHAR(255)  
);



**4.Rooms Table:**

  CREATE TABLE Room\_Types (

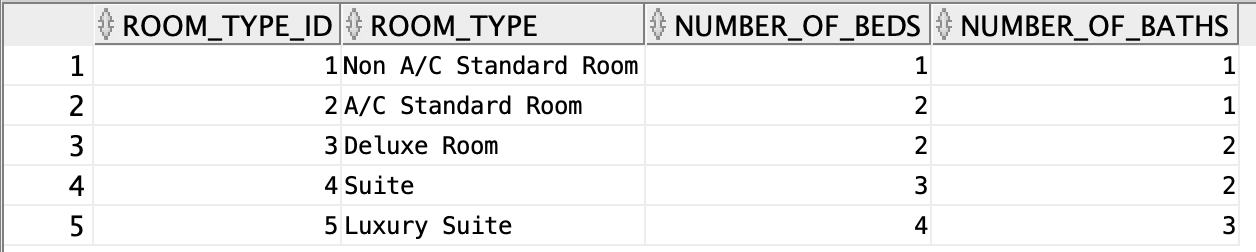
    Room\_Type\_id INT PRIMARY KEY,

    Room\_Type VARCHAR(50),

    Number\_of\_Beds INT,

    Number\_of\_Baths INT

);



**5. Bookings Table:**

CREATE TABLE Bookings (  
  booking\_id INT PRIMARY KEY,   
  customer\_id INT,  
  hotel\_id INT,  
  room\_type\_id INT,  
  booking\_date DATE,  
  check\_in\_date DATE,   
  check\_out\_date DATE,  
  total\_price DECIMAL(10,2),  
  
  FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id),  
  FOREIGN KEY (hotel\_id) REFERENCES Hotels(hotel\_id),  
  FOREIGN KEY (room\_type\_id) REFERENCES Room\_Types(room\_type\_id)  
);

A screenshot of a computer

Description automatically generated

Here are the primary keys and foreign keys used to connect the tables in the provided SQL queries:

**Location Table:**

Primary Key: location\_id

Foreign Key(s): None

**Hotels Table:**

Primary Key: hotel\_id

Foreign Key(s): location\_id (references Location table's location\_id)

**Customers Table:**

Primary Key: customer\_id

Foreign Key(s): None

**Room\_Types Table:**

Primary Key: Room\_Type\_id

Foreign Key(s): None

**Bookings Table:**

Primary Key: booking\_id

Foreign Key(s):

customer\_id (references Customers table's customer\_id)

hotel\_id (references Hotels table's hotel\_id)

room\_type\_id (references Room\_Types table's Room\_Type\_id)

In the SQL schema for the project, the Bookings table can be identified as the fact table, while the Location, Hotels, Customers, and Room\_Types tables serve as dimension tables. The fact table, Bookings, contains the primary keys (booking\_id) and foreign keys (customer\_id, hotel\_id, room\_type\_id) that establish relationships with the dimension tables. These foreign keys connect to the respective primary keys in the dimension tables, enabling the aggregation of information related to hotel bookings. The dimension tables, namely Location, Hotels, Customers, and Room\_Types, provide additional descriptive attributes about the entities involved in the bookings, such as customer details, hotel information, room types, and geographical locations. This fact-dimension model enhances data organization and analysis by separating the core booking information into a central fact table while associating relevant details through dimension tables.

**Indexing:**

**Creating a simple bitmap index on the HOTEL\_ID column:**

CREATE BITMAP INDEX idx\_hotel\_id ON BOOKINGS(HOTEL\_ID).

The creation of the "idx\_hotel\_id" bitmap index on the HOTEL\_ID column in the BOOKINGS table aims to boost query speed. This type of index is useful when there are a few unique hotel IDs. It works like a map, associating each ID with a set of bits. These speeds up searches and filters related to hotels in the BOOKINGS table because the database can quickly check the bits instead of going through each row individually. In simpler terms, it's like having a cheat sheet that makes finding and using hotel information much faster.

**Exploratory Data Analysis:**

**1. Retrieve the customers who made the highest total booking amount:**

SELECT

c.CUSTOMER\_ID,

c.FIRST\_NAME,

c.LAST\_NAME,

EXTRACT(

YEAR

FROM

b.BOOKING\_DATE

) AS BOOKING\_YEAR,

SUM(b.TOTAL\_PRICE) AS TOTAL\_BOOKING\_AMOUNT,

l.CITY

FROM

customers c

JOIN bookings b ON c.CUSTOMER\_ID = b.CUSTOMER\_ID

JOIN HOTELS h ON b.HOTEL\_ID = h.HOTEL\_ID

JOIN LOCATION l ON h.LOCATION\_ID = l.LOCATION\_ID

GROUP BY

c.CUSTOMER\_ID,

c.FIRST\_NAME,

c.LAST\_NAME,

EXTRACT(

YEAR

FROM

b.BOOKING\_DATE

),

l.CITY

ORDER BY TOTAL\_BOOKING\_AMOUNT DESC

A screenshot of a computer

Description automatically generated

**Explanation**: The above query is an aggregation query that retrieves the top customers who made the highest total booking amount over the years 2015 -2022. From the above query, it can be observed that the customer from the top row “Diana Rose” made booking of amount Rs 11816.31 in the year 2018 in the specified location.

**2. Fetches average room price per night of specific room type in each city**

SELECT

r.room\_type,

l.city,

ROUND(

AVG(

b.total\_price / (

TO\_NUMBER(

b.check\_out\_date - b.check\_in\_date

)

)

),

2

) AS avg\_price\_per\_night

FROM

Bookings b

JOIN room\_types r ON b.room\_type\_id = r.room\_type\_id

JOIN Hotels h ON b.hotel\_id = h.hotel\_id

JOIN Location l ON h.location\_id = l.location\_id

GROUP BY

CUBE (r.room\_type, l.city)

HAVING

r.room\_type IS NOT NULL

AND l.city IS NOT NULL

ORDER BY

l.city,

avg\_price\_per\_night

A screenshot of a hotel room

Description automatically generated

**Explanation**: The above query is also an aggregation query fetches the Average room prices per night for each room type in different cities in India. From the results above, we can see that the city “Agra” has different room types and their average pricing accordingly.

**3. Ranking Based on Ratings in Hotels in each city:**

SELECT

hotel\_id,

hotel\_name,

rating,

state,

city,

RANK() OVER (

ORDER BY

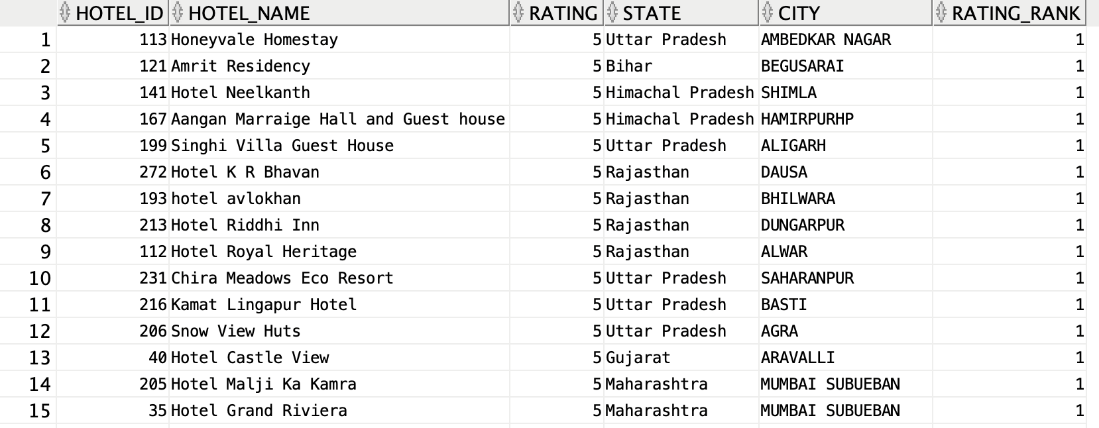
rating DESC

) AS rating\_rank

FROM

hotels h

JOIN location l ON h.location\_id = l.location\_id;



**Explanation**: The above query essentially retrieves information about hotels and assigns a rank to each hotel based on its rating, with the highest-rated hotel receiving a rank of 1, the second highest receiving a rank of 2, and so on. The information is retrieved by joining the hotels and location tables on the common location\_id column. From the results we can observe that the Aagan Marraige Hall has the 1st rank among with other Hotels.

1. **Total Number of Booking by each location:**

CREATE MATERIALIZED VIEW mv\_total\_bookings\_by\_location BUILD IMMEDIATE REFRESH ON DEMAND AS

SELECT

h.HOTEL\_ID,

h.HOTEL\_NAME,

l.CITY,

l.STATE,

COUNT(b.BOOKING\_ID) AS TOTAL\_BOOKINGS

FROM

HOTELS h

JOIN LOCATION l ON h.LOCATION\_ID = l.LOCATION\_ID

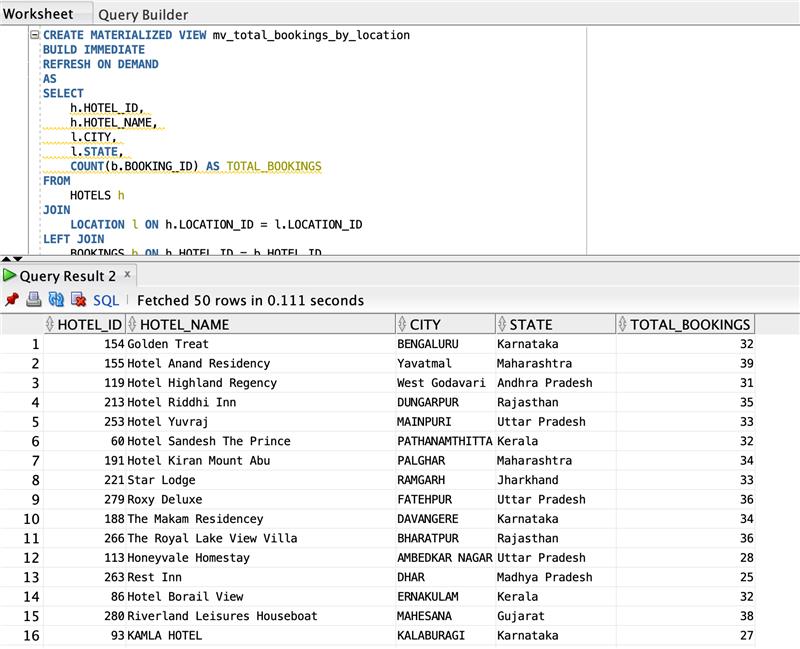
LEFT JOIN BOOKINGS b ON h.HOTEL\_ID = b.HOTEL\_ID

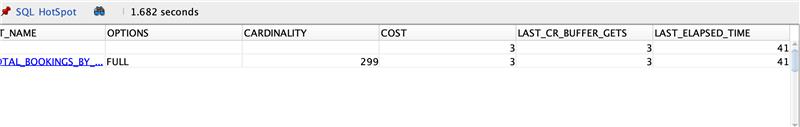
GROUP BY

h.HOTEL\_ID,

h.HOTEL\_NAME,

l.CITY, l.STATE;

  
 select \* from dw138.mv\_total\_bookings\_by\_location;



**Explanation** : The provided query calculates the cumulative number of bookings made by each hotel between 2015 and 2022. As an illustration, the hotel named "Golden Treat" accumulated a total of 32 bookings. To enhance query performance, the results of this query have been stored in a materialized view called "mv\_total\_bookings\_by\_location." This strategic use of a materialized view has significantly reduced the time required for data retrieval, with the elapsed time now at 41 milliseconds, indicating faster access to information from the materialized view compared to querying the base tables directly.

1. **Fetching the ranks based on total bookings:**

SELECT

state,

hotel\_name,

total\_bookings,

booking\_rank

FROM

(

SELECT

loc.state,

h.hotel\_name,

COUNT(b.booking\_id) AS total\_bookings,

RANK() OVER (

PARTITION BY loc.state

ORDER BY

COUNT(b.booking\_id) DESC

) AS booking\_rank

FROM

Bookings b

JOIN Hotels h ON b.hotel\_id = h.hotel\_id

JOIN Location loc ON h.location\_id = loc.location\_id

GROUP BY

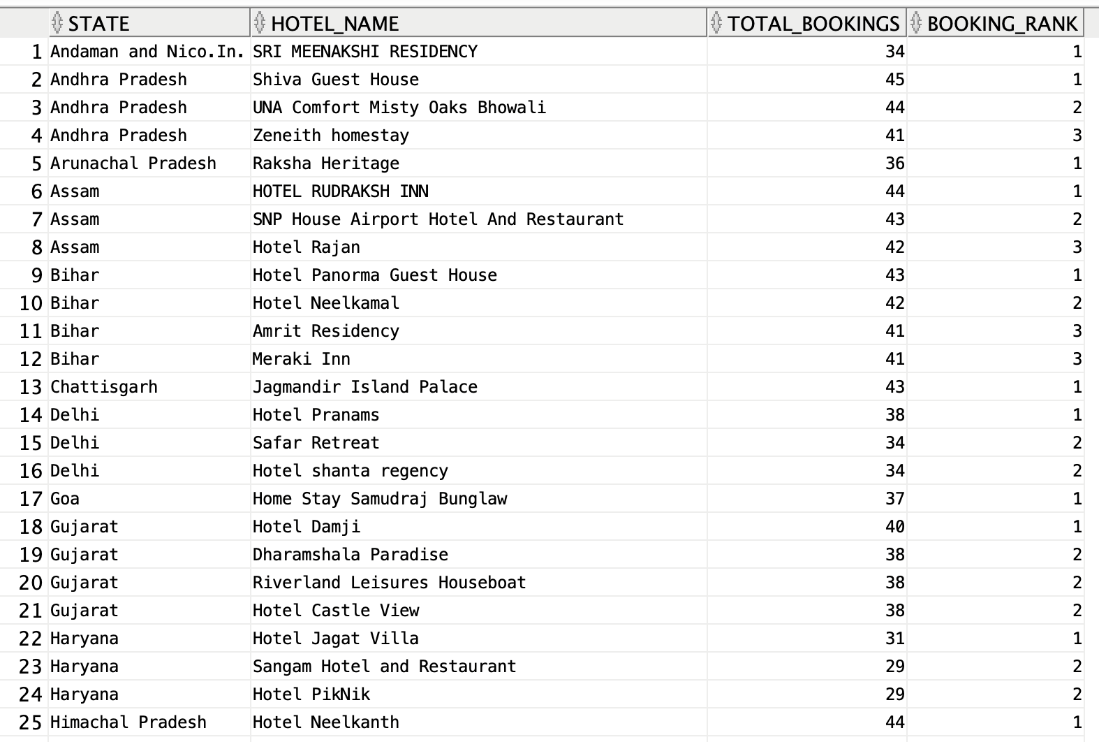
loc.state,

h.hotel\_name

) ranked

WHERE

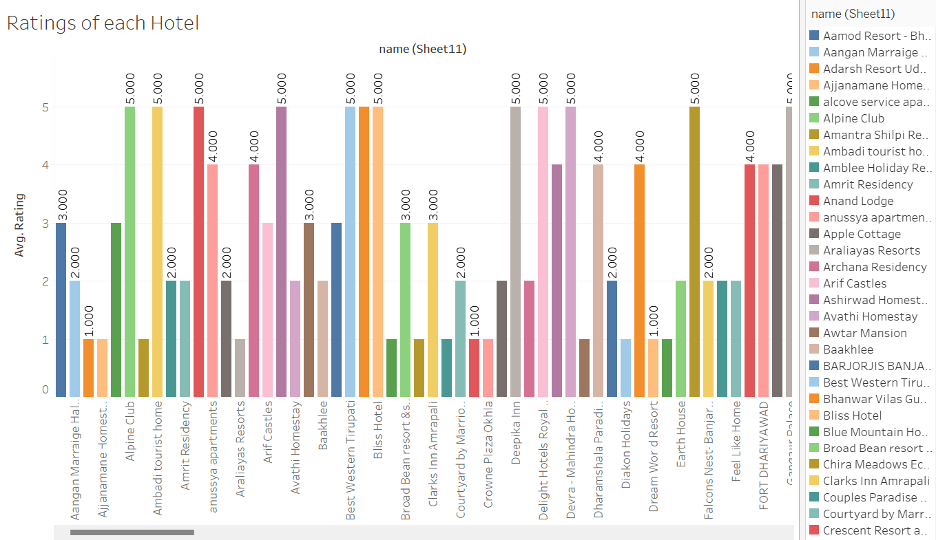
booking\_rank <= 3;



**Explanation:** The Query displays the top 3 hotels as per hotel bookings in each state. This query retrieves information about the top three hotels with the highest total bookings in each state. It selects the state, hotel name, total bookings, and a ranking of hotels based on booking counts. The inner query uses JOIN operations to connect the Bookings, Hotels, and Location tables, grouping the data by state and hotel name while counting the number of bookings. The RANK() window function assigns a rank to each hotel within its respective state, ordered by the descending count of bookings. The outer query then filters the results to include only those with a booking rank up to 3, providing a concise list of the top-performing hotels in each state.

**Reporting, Modeling and Storytelling:**

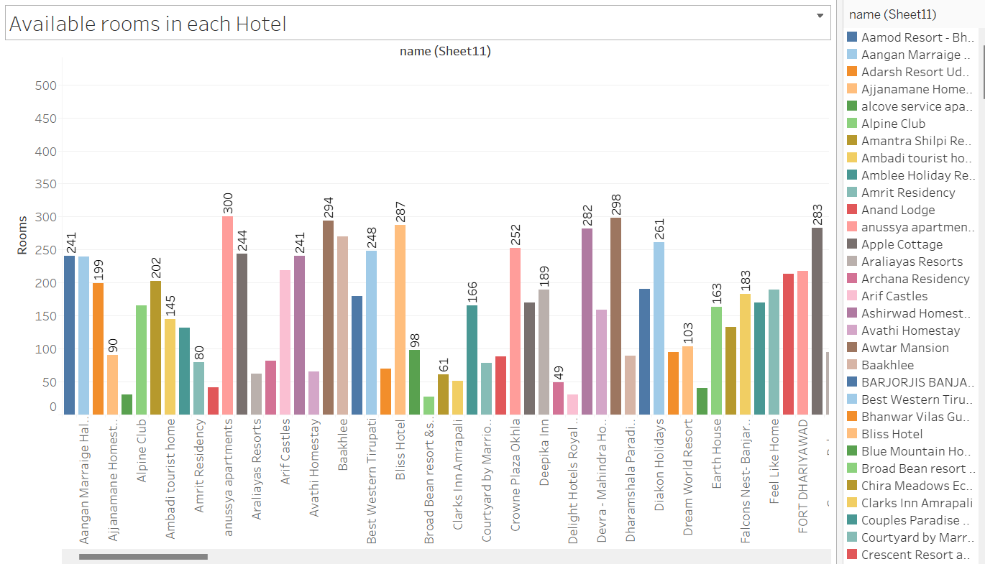
1. **Which Hotels have the best average rating?**



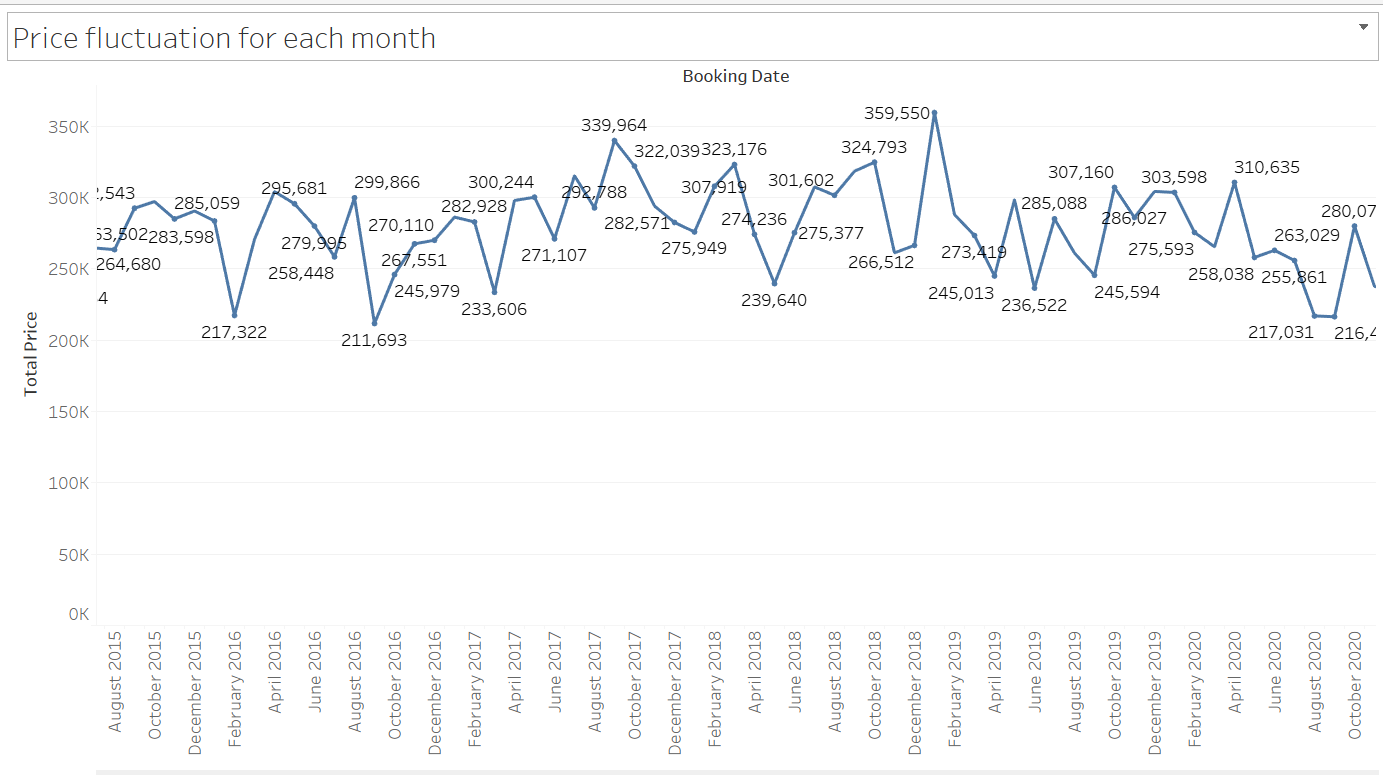
The bar graph that is displayed shows the average ratings that different hotels received between 2015 and 2022. The Alpine Club is notable for having the highest average rating, outperforming a number of other well-known hotels. This information highlights how well it performed in sustaining high levels of customer satisfaction over the given time frame.

1. **Which Hotels have the Highest number of rooms?**

The provided bar graph illustrates the total number of available rooms for each hotel, notably showcasing Anussya Apartments with the highest room availability, totaling 300 rooms. This insight emphasizes the substantial room capacity offered by Anussya Apartments compared to other hotels.

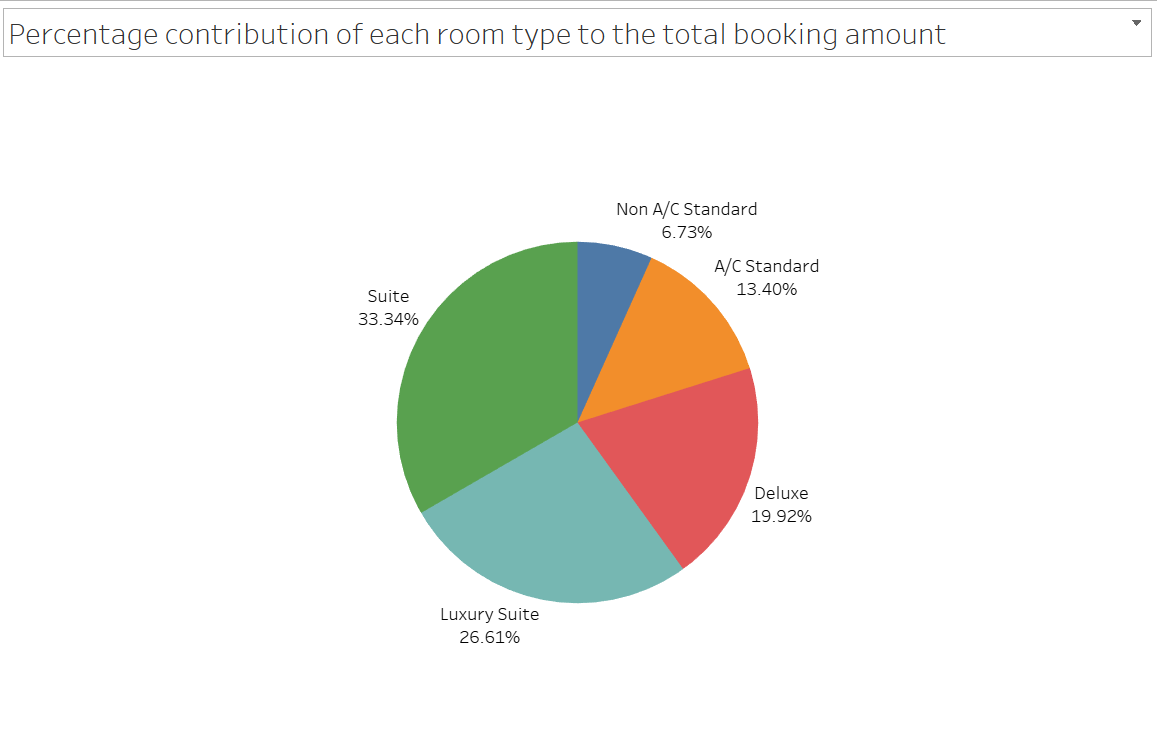


1. **Observing the trend of total price spent on hotel bookings from 2015 to 2022.**



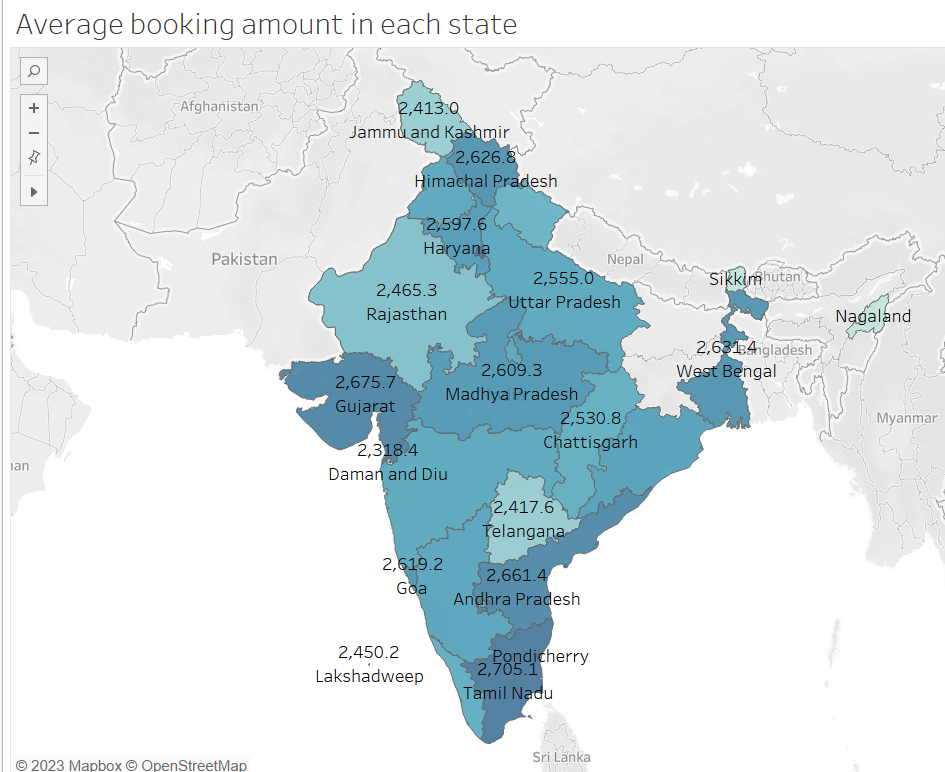
A visual depiction of hotel pricing variations from 2015 to 2022 is given by the above line graph. Notably, regular price increases have been noted in August and December, which correspond with the busiest travel seasons. It appears from this pattern that costs typically go up during these well-liked holiday times.

1. **Which type of rooms generates more revenue for the Hotels?**



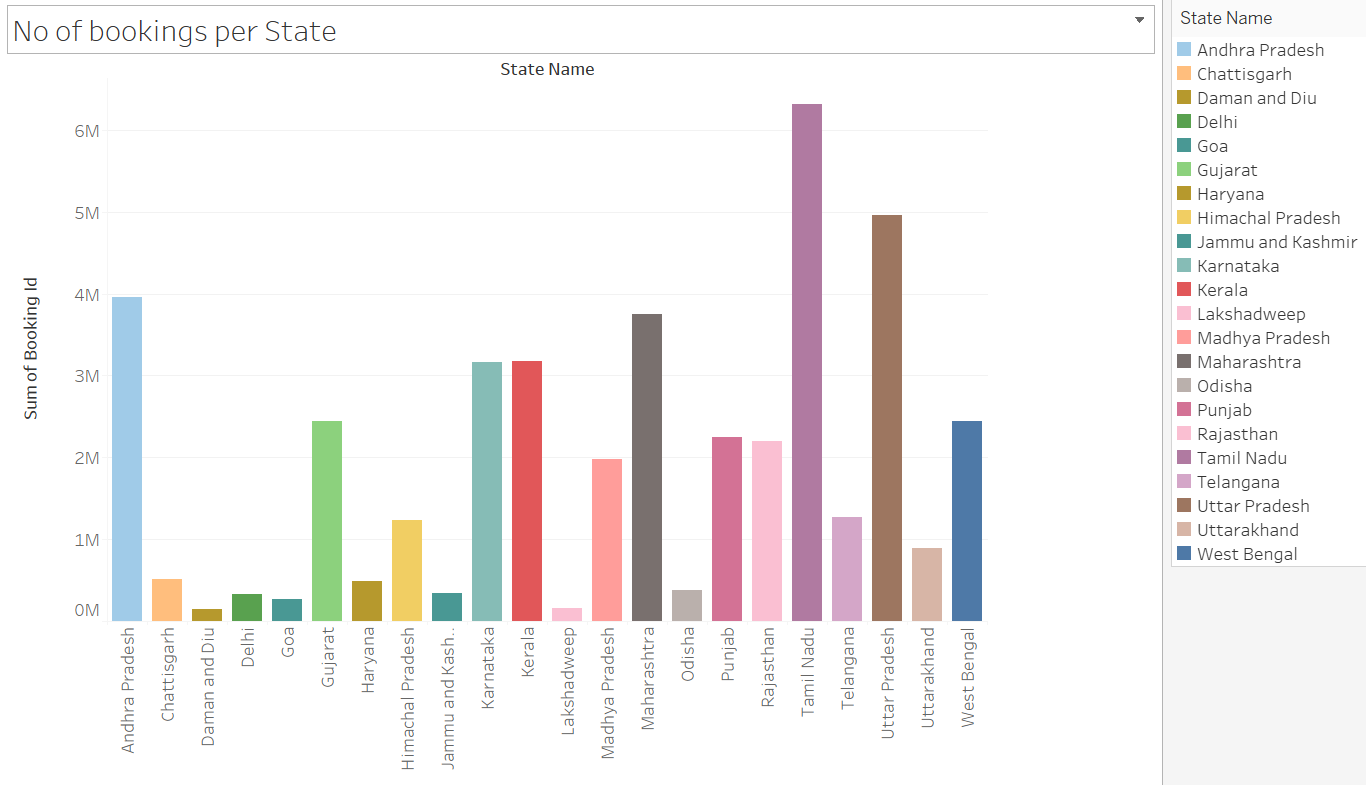
The above pie chart shows the percentage of total price of each type of Rooms. Here we can observe that the Suites generates more revenue with 33.34% than all other room types and followed by luxury suites with 26.61%, this shows that that every though luxury suite is high end the general suites generate more revenue.

1. **Which state has the best Average booking price?**



The map visualizes the variation in average room prices per hotel across states in India. It reveals that Haryana boasts the highest average price at 2754.1, indicative of high demand and elevated prices. In contrast, Daman & Diu reflects the lowest average price at 2318 rupees, potentially signaling lower demand or a more competitive pricing environment.

1. **Total number of Booking from the year 2015 to 2022?**



The bar chart illustrates the total number of bookings registered by each state. Notably, Tamil Nadu demonstrates the highest number of bookings, totaling over 6 million from 2015 to 2022. In contrast, Daman and Diu have the lowest number of bookings at 150,000.

**Conclusion:**

The analysis of hotel-related data spanning 2015 to 2022 provides comprehensive insights into various facets of the hospitality industry. The review encompasses significant elements, including hotel ratings, room availability, price fluctuations, and booking distributions across different states. Notably, it showcases a dynamic landscape: while Alphine Club excels in hotel ratings, Anussya Apartments leads in room availability. Price fluctuations over the years hint at seasonal trends, indicating price peaks around December and August. Further, the geographical distribution of room prices across states demonstrates disparities, with Haryana marking the highest average price and Daman & Diu the lowest, shaping potential insights into demand and pricing strategies. Tamil Nadu emerges as a significant hub, with the highest number of bookings. These analyses collectively offer nuanced perspectives and valuable directions for strategic planning and decision-making in the hotel industry.

**References:**

**Datasets:**

Kaggle, Hotel booking Dataset, retrieved February 2023

[**https://www.kaggle.com/datasets/abdulrahmankhaled1/hotel-booking-dataset/**](https://www.kaggle.com/datasets/abdulrahmankhaled1/hotel-booking-dataset/)

Data world, Hotel Bookings Dataset, Retrieved January 2023

[**https://data.world/mesum/hotel-bookings-dataset**](https://data.world/mesum/hotel-bookings-dataset)

Github, HOTEL-MANAGEMENT-SYSTEM-DATABASE-PROJECT, Retrieved May 2019

[**https://github.com/swetababu/HOTEL-MANAGEMENT-SYSTEM-DATABASE-PROJECT**](https://github.com/swetababu/HOTEL-MANAGEMENT-SYSTEM-DATABASE-PROJECT)