STEP 1: Topic/Domain Selection

Selected Topic: Hotel Booking Management System

Introduction:

The Hotel Booking Management System is a comprehensive database project designed to streamline and enhance the management of customer, accounts, property information, reservations, transactions, and rewards within the hotel industry. This system caters to the dynamic needs of hotel management by integrating various aspects of customer interaction and rewards programs. The database encompasses detailed customer profiles, capturing essential information such as personal details, preferences, and historical interactions with the hotel. It incorporates a robust account management module, enabling customers to create and maintain profiles, along with tracking subscription details and expiration dates.

The reservation and transaction components facilitate seamless booking processes, capturing reservation details, check-in/check-out dates, and financial transactions. Furthermore, the system incorporates a rewards module that tracks and manages customer loyalty programs, offering insights into earned rewards, redemption history, and overall engagement. This Hotel Booking Management System optimizes operations by providing a holistic view of customer interactions, allowing hotel administrators to tailor services, improve customer satisfaction, and strategically enhance loyalty programs. With its user-friendly interface and comprehensive data organization, this database project stands as a powerful tool for effective hotel management in the ever-evolving hospitality industry.

Advantages:

The advantages of implementing the Hotel Booking Management System database model are multifaceted, offering valuable insights and strategic benefits for decision-makers within the hotel industry. The database allows decision-makers to analyze customer reservation data, identifying popular and unpopular Properties. This insight enables the hotel management to refine and enhance customer experience and business model, tailoring them to customer preferences and increasing overall subscription sales.

The centralized database ensures data accuracy and consistency. This helps prevent errors or inconsistencies in customer profiles, reservations, and rewards data. Accurate and reliable information is crucial for delivering seamless customer experiences and maintaining operational efficiency. In summary, the Hotel Booking Management System's database model empowers decision-makers in the hotel industry by providing valuable insights into customer behavior, subscription preferences, and property popularity. The advantages encompass strategic decision-making, improved customer satisfaction, and streamlined operational efficiency, contributing to the overall success of hotel management.

Use Cases

The Hotel Booking Management System serves as a sophisticated solution for the hotel industry, akin to popular booking platforms like Hilton or Marriott in its approach to customer-centric services. Similar to hotel booking platforms' membership structures, this database centralizes customer accounts, preferences, and transaction histories, offering a comprehensive view of customer interactions.

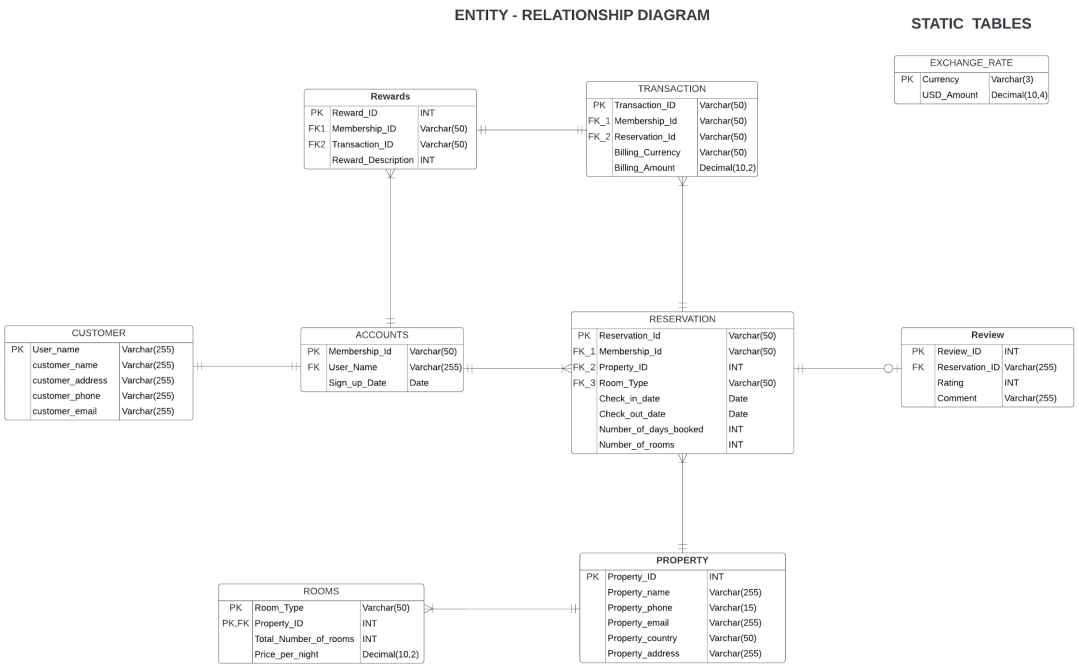
The system allows for efficient reservation management, tracking check-in/check-out dates and financial transactions, mirroring the content delivery models of streaming platforms. The inclusion of a rewards module aligns with the loyalty programs providing insights into earned rewards and redemption history. This adaptable database model extends beyond hotels, finding relevance in physical stores. Overall, the Hotel Booking Management System stands as a powerful tool for optimizing customer relations and loyalty programs in the dynamic hospitality landscape.

STEP 2: Conceptual Data Modeling and Database Design

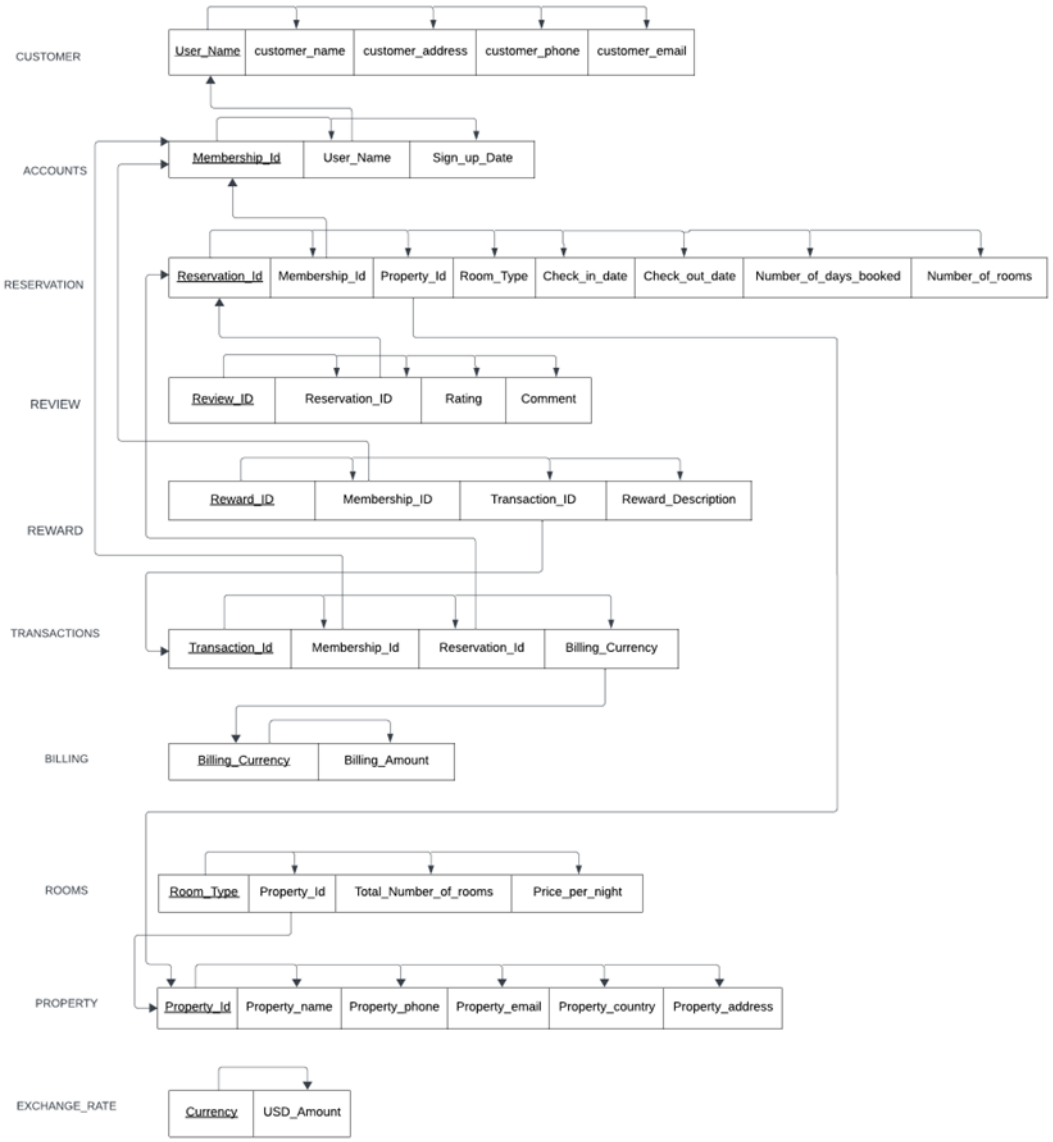
Business Logic:

1. Each Customer can have multiple Accounts
2. Each account contains a Membership Id.
3. A Customer should have a Valid account and membership Id to be eligible for rewards Program.
4. A Customer can view into his account and look for analytical details such as Number of reservations, Total Amount spent on transactions.
5. Each Customer is classified into Platinum, Gold and silver members based on his spendings.
6. A Customer can reserve multiple rooms at a time.
7. A Customer can Pay for reservation in any currency of their choice.
8. A Property has multiple rooms and different room types.
9. A Customer earns One reward point for every 50 dollars transactions made.

ER/EER DIAGRAM:



Relational Model:



STEP 3: Database Implementation

SQL Commands to Create Tables for Database:

Below are the SQL commands that were used to create tables for the database The SQL commands used to create the tables for the database can also be found in an attached word file called EDM_GROUP2_BASE_TABLES_QUERIES.docx

CUSTOMER TABLE

```
CREATE TABLE `customer` (

  `User_name` varchar(255) NOT NULL,
```

```
  `customer_name` varchar(255) NOT NULL,
  `customer_address` varchar(255) NOT NULL,
  `customer_phone` varchar(15) NOT NULL,
  `customer_email` varchar(255) NOT NULL,
  PRIMARY KEY (`User_name`)
);
```

ACCOUNTS TABLE

```
CREATE TABLE `accounts` (
  `Membership_Id` varchar(50) NOT NULL,
  `User_Name` varchar(255) DEFAULT NULL,
  `Sign_up_Date` date DEFAULT NULL,
  PRIMARY KEY (`Membership_Id`)
);
```

PROPERTY TABLE

```
CREATE TABLE `property` (
  `Property_ID` int NOT NULL,
  `Property_name` varchar(255) DEFAULT NULL,
  `Property_phone` varchar(15) DEFAULT NULL,
  `Property_email` varchar(255) DEFAULT NULL,
  `Property_country` varchar(50) DEFAULT NULL,
  `Property_address` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`Property_ID`)
);
```

ROOMS TABLE

```
CREATE TABLE `rooms` (
  `Room_Type` varchar(50) NOT NULL,
```

```
  `Property_ID` int NOT NULL,
  `Total_Number_of_rooms` int NOT NULL,
  `Price_per_night` decimal(10,2) DEFAULT NULL,
  PRIMARY KEY (`Room_Type`,`Property_ID`),
  KEY `Property_ID` (`Property_ID`),
  CONSTRAINT `rooms_ibfk_1` FOREIGN KEY (`Property_ID`) REFERENCES
`property` (`Property_ID`)
);
```

RESERVATION TABLE

```
CREATE TABLE `reservation` (
  `Reservation_Id` varchar(50) NOT NULL,
  `Membership_Id` varchar(50) DEFAULT NULL,
  `Property_ID` int DEFAULT NULL,
  `Check_in_date` date DEFAULT NULL,
  `Check_out_date` date DEFAULT NULL,
  `Number_of_days_booked` int GENERATED ALWAYS AS ((to_days(`Check_out_date`) -
to_days(`Check_in_date`))) STORED,
  `Number_of_rooms` int DEFAULT NULL,
  `Room_Type` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`Reservation_Id`),
  KEY `Membership_Id` (`Membership_Id`),
  KEY `Property_ID` (`Property_ID`),
  CONSTRAINT `reservation_ibfk_1` FOREIGN KEY (`Membership_Id`) REFERENCES
`accounts` (`Membership_Id`),
  CONSTRAINT `reservation_ibfk_2` FOREIGN KEY (`Property_ID`) REFERENCES
`property` (`Property_ID`)
);
```

TRANSACTIONS TABLE

```
CREATE TABLE `transactions` (
  `Transaction_ID` varchar(50) NOT NULL,
  `Membership_Id` varchar(50) DEFAULT NULL,
  `Reservation_Id` varchar(50) DEFAULT NULL,
  `Billing_Currency` varchar(50) DEFAULT NULL,
  `Billing_Amount` decimal(10,2) DEFAULT NULL,
  PRIMARY KEY (`Transaction_ID`),
  KEY `Membership_Id` (`Membership_Id`),
  CONSTRAINT `transactions_ibfk_1` FOREIGN KEY (`Membership_Id`) REFERENCES
`reservation` (`Membership_Id`)
);
```

REVIEW TABLE

```
CREATE TABLE `review` (
  `Review_ID` int NOT NULL,
  `Reservation_ID` varchar(50) DEFAULT NULL,
  `Rating` int DEFAULT NULL,
  `Comments` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`Review_ID`),
  KEY `Reservation_ID` (`Reservation_ID`),
  CONSTRAINT `review_ibfk_1` FOREIGN KEY (`Reservation_ID`) REFERENCES
`reservation` (`Reservation_Id`)
);
```

REWARDS TABLE

```
CREATE TABLE `rewards` (
  `Reward_ID` int NOT NULL,
  `Membership_Id` varchar(50) DEFAULT NULL,
```

```
  `Transaction_Id` varchar(50) DEFAULT NULL,

  `Reward_Description` varchar(255) DEFAULT 'Membership Rewards',

  PRIMARY KEY (`Reward_ID`),

  KEY `Membership_Id` (`Membership_Id`),

  KEY `Transaction_Id` (`Transaction_Id`),

  CONSTRAINT `rewards_ibfk_1` FOREIGN KEY (`Membership_Id`) REFERENCES
`accounts` (`Membership_Id`),

  CONSTRAINT `rewards_ibfk_2` FOREIGN KEY (`Transaction_Id`) REFERENCES
`transactions` (`Transaction_ID`)

);
```

EXCHANGE RATE TABLE

```
CREATE TABLE exchange_rate (

  Currency varchar(3) NOT NULL,

  USD_Amount decimal(10,4) DEFAULT NULL,

  PRIMARY KEY (Currency)

);
```

SQL Commands to Insert Data Into Database:

A fake data set with a total of 180 rows was created using Excel Sheet (MIS 686 Hotel Rewards Management)

(https://1drv.ms/x/s!AsnQHN524fFshmgG5zllOAdY6zvV?e=fDLIoV) . The data consists of data for the tables of Customer, Account, Property, Rooms, Reservation, Transaction, Rewards, Review and Exchange Rate. The corresponding INSERT commands can be found in the Word Document. These INSERT INTO commands were then used to populate the database on MYSQL Workbench. A link to the Word document with all the insert commands is provided with the report https://1drv.ms/w/s!AsnQHN524fFshmxTV8ts8E_eFez8?e=94FRGv.

VIEWS AND TRIGGERS:

Account View

```
CREATE VIEW `account_view` AS

select `a`.`Membership_Id` AS `Membership_Id`,
```

`a`.`User_Name` AS `User_Name`,

`a`.`Sign_up_Date` AS `Sign_up_Date`,

count(distinct `r`.`Reservation_Id`) AS `Number_of_Reservations`,

coalesce(sum(`t`.`Adjusted_Amount_USD`),0) AS `Total_Transaction_Amount`,

(case when (coalesce(sum(`t`.`Adjusted_Amount_USD`),0) >= 1000) then 'Platinum' when (coalesce(sum(`t`.`Adjusted_Amount_USD`),0) between 750 and 999.99) then 'Gold' else 'Silver' end) AS `Member_Level`,

coalesce(sum(`rw`.`Reward_Points`),0) AS `Total_Reward_Points`

from (((`accounts` `a` left join `reservation` `r` on((`a`.`Membership_Id` = `r`.`Membership_Id`))) left join `transactions_view` `t` on((`r`.`Reservation_Id` = `t`.`Reservation_Id`))) left join `rewards_view` `rw` on((`r`.`Reservation_Id` = `rw`.`Reservation_Id`))) group by `a`.`Membership_Id`;

Reservation View

CREATE VIEW `reservation_view` AS

select `r`.`Reservation_Id` AS `Reservation_Id`,

`r`.`Membership_Id` AS `Membership_Id`,

`r`.`Property_ID` AS `Property_ID`,

`r`.`Check_in_date` AS `Check_in_date`,

`r`.`Check_out_date` AS `Check_out_date`,

`r`.`Number_of_days_booked` AS `Number_of_days_booked`,

`r`.`Number_of_rooms` AS `Number_of_rooms`,

`r`.`Room_Type` AS `Room_Type`,

((`r`.`Number_of_days_booked` * `r`.`Number_of_rooms`) * `rooms`.`Price_per_night`) AS `Reservation_Amount`

from (`reservation` `r` join `rooms` on(((`r`.`Property_ID` = `rooms`.`Property_ID`) and (`r`.`Room_Type` = `rooms`.`Room_Type`))))

Rewards View

CREATE VIEW `rewards_view` AS

select `t`.`Transaction_ID` AS `Transaction_ID`,

`a`.`Reservation_Id` AS `Reservation_Id`,

`t`.`Membership_Id` AS `Membership_Id`,

cast((`t`.`Adjusted_Amount_USD` / 50.00) as decimal(5,2)) AS `Reward_Points`,

'Memberlevel Rewards' AS `Rewards_Description`

from (`transactions_view` `t` join `reservation` `a` on((`t`.`Reservation_Id` = `a`.`Reservation_Id`)))

Rooms View

CREATE VIEW `rooms_view` AS

select distinct `r`.`Room_Type` AS `Room_Type`,

`r`.`Property_ID` AS `Property_ID`,

`r`.`Total_Number_of_rooms` AS `Total_Number_of_rooms`,

`re`.`Number_of_rooms` AS `Number_of_rooms_Booked`,

(`r`.`Total_Number_of_rooms` - `re`.`Number_of_rooms`) AS `Number_of_rooms_available`,

`r`.`Price_per_night` AS `Price_per_night`

from (`rooms` `r` join `reservation` `re` on((`re`.`Room_Type` = `r`.`Room_Type`)))

Transaction View

CREATE VIEW `transactions_view` AS

select `t`.`Transaction_ID` AS `Transaction_ID`,

`t`.`Reservation_Id` AS `Reservation_Id`,

`t`.`Membership_Id` AS `Membership_Id`,

`t`.`Billing_Currency` AS `Billing_Currency`,

`t`.`Billing_Amount` AS `Billing_Amount`,

truncate((`t`.`Billing_Amount` * `e`.`USD_Amount`),2) AS `Adjusted_Amount_USD`

from (`transactions` `t` left join `exchange_rate` `e` on((`t`.`Billing_Currency` = `e`.`Currency`)))

Trigger

Trigger to Calculate Reward Points after every Transaction:

```
DELIMITER //

CREATE TRIGGER calculate_reward_points

BEFORE INSERT ON REWARDS

FOR EACH ROW

BEGIN

    DECLARE adjusted_amount DECIMAL(10, 2);

    -- Fetch the adjusted amount from the associated transaction

    SELECT Adjusted_Amount_USD INTO adjusted_amount

    FROM transactions

    WHERE Transaction_ID = NEW.Transaction_Id;

    -- Calculate and set the Reward_Points based on the adjusted amount

    SET NEW.Reward_Points = ROUND(adjusted_amount / 50);

END;

//

DELIMITER ;
```

Analytical Questions and SQL Queries:

1. Which Property has Performed well in terms of number of reservations?

```
SELECT p.`Property_name`, COUNT(r.`Reservation_Id`) AS
`Number_of_Reservations`
FROM `property` p
JOIN `reservation` r ON p.`Property_ID` = r.`Property_ID`
GROUP BY p.`Property_name`
ORDER BY `Number_of_Reservations` DESC
LIMIT 1;
```

```
1 •  SELECT p.`Property_name`, COUNT(r.`Reservation_Id`) AS `Number_of_Reservations`
2    FROM `property` p
3    JOIN `reservation` r ON p.`Property_ID` = r.`Property_ID`
4    GROUP BY p.`Property_name`
5    ORDER BY `Number_of_Reservations` DESC
6    LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

Property_name	Number_of_Reservations
Cosy Lodge	4

2. Which Room type has the highest number of Bookings across all Properties?

```
SELECT `Room_Type`, COUNT(`Reservation_Id`) AS `Number_of_Bookings`
FROM `reservation`
GROUP BY `Room_Type`
ORDER BY `Number_of_Bookings` DESC
LIMIT 1;
```

```
8 •  SELECT `Room_Type`, COUNT(`Reservation_Id`) AS `Number_of_Bookings`
9    FROM `reservation`
10   GROUP BY `Room_Type`
11   ORDER BY `Number_of_Bookings` DESC
12   LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

Room_Type	Number_of_Bookings
Standard Room	4

3. Which currencies were used to pay for reservations across all Properties?

```
SELECT DISTINCT `Billing_Currency`
FROM `transactions`;
```

```
14 •  SELECT DISTINCT `Billing_Currency`
15    FROM `transactions`;
16
17
```

Result Grid | Filter Rows: | Export:

Billing_Currency
USD
EUR
GBP
JPY
CAD
INR

4. Which Customer Spends the Most on reservations?

```
SELECT a.`User_Name`, SUM(t.`Adjusted_Amount_USD`) AS `Total_Spending`
FROM `accounts` a
JOIN `reservation` r ON a.`Membership_Id` = r.`Membership_Id`
JOIN `transactions_view` t ON r.`Reservation_Id` = t.`Reservation_Id`
GROUP BY a.`User_Name`
ORDER BY `Total_Spending` DESC
LIMIT 1;
```

```
19 •  SELECT a.`User_Name`, SUM(t.`Adjusted_Amount_USD`) AS `Total_Spending`
20    FROM `accounts` a
21    JOIN `reservation` r ON a.`Membership_Id` = r.`Membership_Id`
22    JOIN `transactions_view` t ON r.`Reservation_Id` = t.`Reservation_Id`
23    GROUP BY a.`User_Name`
24    ORDER BY `Total_Spending` DESC
25    LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

User_Name	Total_Spending
john_doe	3531.50

5. List of Top three customers who spent most on reservations?

```
SELECT a.`User_Name`, SUM(t.`Adjusted_Amount_USD`) AS `Total_Spending`
FROM `accounts` a
JOIN `reservation` r ON a.`Membership_Id` = r.`Membership_Id`
JOIN `transactions_view` t ON r.`Reservation_Id` = t.`Reservation_Id`
GROUP BY a.`User_Name`
ORDER BY `Total_Spending` DESC
LIMIT 3;
```

```
27 •  SELECT a.`User_Name`, SUM(t.`Adjusted_Amount_USD`) AS `Total_Spending`
28    FROM `accounts` a
29    JOIN `reservation` r ON a.`Membership_Id` = r.`Membership_Id`
30    JOIN `transactions_view` t ON r.`Reservation_Id` = t.`Reservation_Id`
31    GROUP BY a.`User_Name`
32    ORDER BY `Total_Spending` DESC
33    LIMIT 3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

User_Name	Total_Spending
john_doe	3531.50
emma_wilson	3456.00
michael_brown	2871.50

6. Which Property has the highest total revenue from reservations?

```
SELECT p.`Property_name`, SUM(t.`Adjusted_Amount_USD`) AS `Total_Revenue`
FROM `property` p
JOIN `reservation` r ON p.`Property_ID` = r.`Property_ID`
JOIN `transactions_view` t ON r.`Reservation_Id` = t.`Reservation_Id`
GROUP BY p.`Property_name`
ORDER BY `Total_Revenue` DESC
LIMIT 1;
```

```
35 •  SELECT p.`Property_name`, SUM(t.`Adjusted_Amount_USD`) AS `Total_Revenue`
36    FROM `property` p
37    JOIN `reservation` r ON p.`Property_ID` = r.`Property_ID`
38    JOIN `transactions_view` t ON r.`Reservation_Id` = t.`Reservation_Id`
39    GROUP BY p.`Property_name`
40    ORDER BY `Total_Revenue` DESC
41    LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

Property_name	Total_Revenue
Riverside Retreat	4356.00

7. What is the average duration of stay for each Room Type?

```
SELECT `Room_Type`, AVG(`Number_of_days_booked`) AS
`Average_Stay_Duration`
FROM `reservation`
GROUP BY `Room_Type`;
```

```
43 •  SELECT `Room_Type`, AVG(`Number_of_days_booked`) AS `Average_Stay_Duration`
44    FROM `reservation`
45    GROUP BY `Room_Type`;
46
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Room_Type	Average_Stay_Duration
Deluxe Suite	6.0000
Ocean View Room	5.0000
Mountain Cabin	5.0000
City View Suite	7.0000
Rural Cottage	4.0000
Standard Room	5.5000
Lake View Suite	5.0000
Cityscape Room	5.0000
Riverside Cabin	6.5000
Tranquility Suite	5.0000
Executive Suite	6.0000
Beachfront Villa	5.0000
Forest Cabin	5.0000
Mountain View S...	7.0000
Garden Cottage	4.0000

8. Which month had the highest number of reservations?

```
SELECT MONTH(`Check_in_date`) AS `Reservation_Month`,
COUNT(`Reservation_Id`) AS `Number_of_Reservations`
FROM `reservation`
GROUP BY `Reservation_Month`
ORDER BY `Number_of_Reservations` DESC
LIMIT 1;
```

```
50 •  SELECT MONTH(`Check_in_date`) AS `Reservation_Month`, COUNT(`Reservation_Id`) AS `Number_of_Reservations`
51    FROM `reservation`
52    GROUP BY `Reservation_Month`
53    ORDER BY `Number_of_Reservations` DESC
54    LIMIT 1;
55
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

Reservation_Month	Number_of_Reservations
6	3

9. What is the total revenue from reservations for each currency?

```
SELECT `Billing_Currency`, SUM(`Adjusted_Amount_USD`) AS `Total_Revenue`
```

```
FROM `transactions_view`
GROUP BY `Billing_Currency`;
```

```
57 •  SELECT `Billing_Currency`, SUM(`Adjusted_Amount_USD`) AS `Total_Revenue`
58    FROM `transactions_view`
59    GROUP BY `Billing_Currency`;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Billing_Currency	Total_Revenue
USD	14330.00
EUR	2912.00
GBP	3325.00
JPY	1662.50
CAD	1794.00
INR	105.98

10. Which Customer made the earliest reservation?

```
SELECT a.`User_Name`, MIN(r.`Check_in_date`) AS `Earliest_Reservation_Date`
FROM `accounts` a
JOIN `reservation` r ON a.`Membership_Id` = r.`Membership_Id`
GROUP BY a.`User_Name`
ORDER BY `Earliest_Reservation_Date`
LIMIT 1;
```

```
61 •  SELECT a.`User_Name`, MIN(r.`Check_in_date`) AS `Earliest_Reservation_Date`
62    FROM `accounts` a
63    JOIN `reservation` r ON a.`Membership_Id` = r.`Membership_Id`
64    GROUP BY a.`User_Name`
65    ORDER BY `Earliest_Reservation_Date`
66    LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

User_Name	Earliest_Reservation_Date
john_doe	2022-06-01

11. How many customers are Platinum Members?

```
SELECT COUNT(Membership_Id) AS Platinum_Members_Count
FROM account_view
WHERE Member_Level = 'Platinum';
```

```
81 •  SELECT COUNT(Membership_Id) AS Platinum_Members_Count
82    FROM account_view
83    WHERE Member_Level = 'Platinum';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Platinum_Members_Count
10

12. What is the average number of reward points earned per reservation?

```
SELECT AVG(Reward_Points) AS Avg_Reward_Points
FROM rewards_view;
```

```
85 •  SELECT AVG(Reward_Points) AS Avg_Reward_Points
86    FROM rewards_view;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Avg_Reward_Points
16.086333

13. How many reservations were made each month?

```
SELECT MONTHNAME(Check_in_date) AS Reservation_Month,
COUNT(Reservation_Id) AS Number_of_Reservations
FROM reservation_view
GROUP BY Reservation_Month;
```

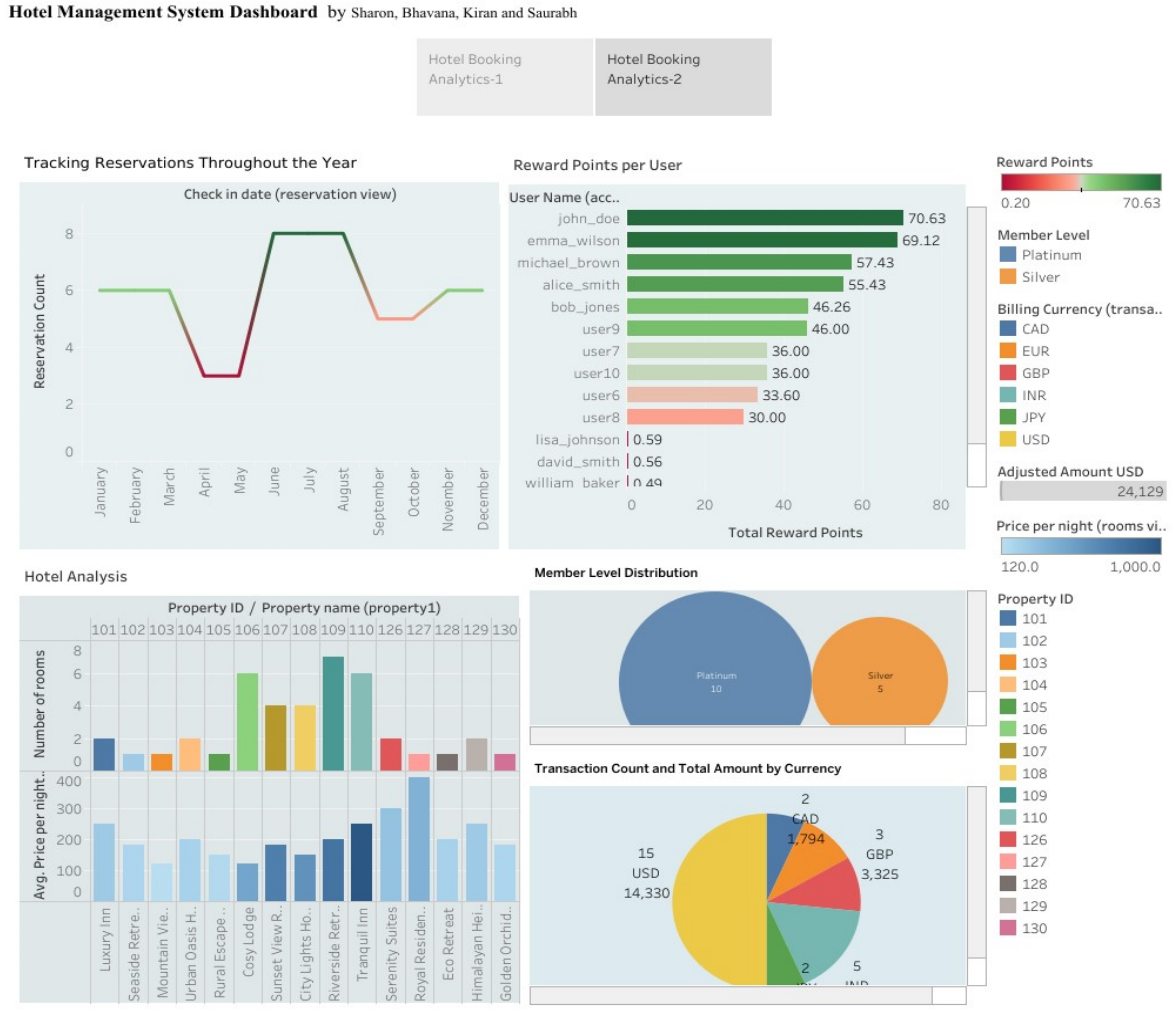
```
87 •  SELECT MONTHNAME(Check_in_date) AS Reservation_Month, COUNT(Reservation_Id) AS Number_of_Reservations
88    FROM reservation_view
89    GROUP BY Reservation_Month;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Reservation_Month	Number_of_Reservations
June	3
July	3
August	3
September	2
October	2
November	3
April	1
December	3
May	1
January	3
February	3
March	3

STEP 4: Enterprise (web) Database Dashboard

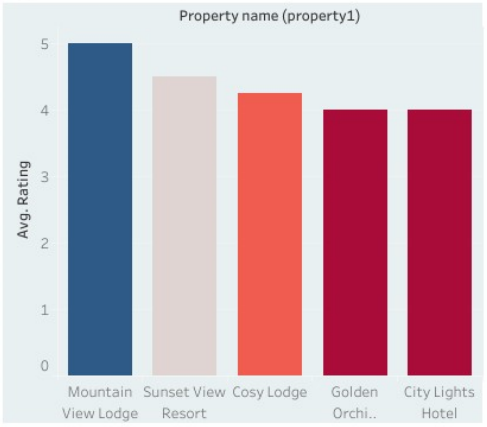
The database dashboard can be found under the following link:
[Hotel Booking Management System | Tableau Public](https://public.tableau.com/app/profile/saurabh.kumar3759/viz/HotelRewardsManagementSystem/Story2?publish=yes)
(https://public.tableau.com/app/profile/saurabh.kumar3759/viz/HotelRewardsManagementSystem/Story2?publish=yes)



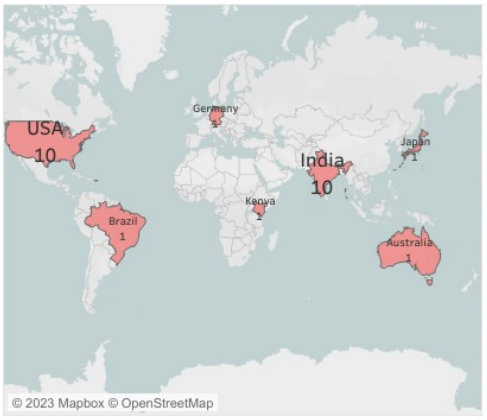
Hotel Management System Dashboard by Sharon, Bhavana, Kiran and Saurabh

Hotel Booking Analytics-1 | Hotel Booking Analytics-2

Top 5 Highest-Rated Properties



Hotels per Country

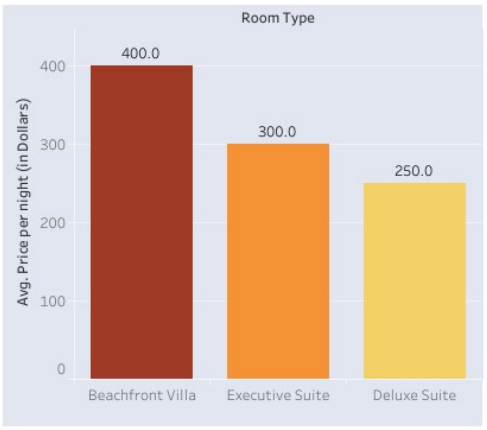


Avg. Rating
4.000 5.000

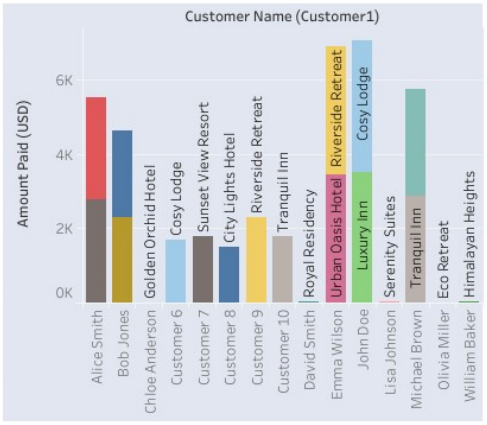
Price per night (rooms vie..
250.0 400.0

Property name (property1)
City Lights Hotel
Cosy Lodge
Eco Retreat
Golden Orchid Hotel
Himalayan Heights
Luxury Inn
Mountain View Lodge
Riverside Retreat
Royal Residency
Rural Escape Resort
Seaside Retreat
Serenity Suites
Sunset View Resort
Tranquil Inn
Urban Oasis Hotel

Top 3 Most Luxurious Room Types



Customer Booking Behavior and Expenditure Across Hotels



Pages

Columns: Customer Name (Cu..

Rows: SUM(Adjusted Amou..

Filters

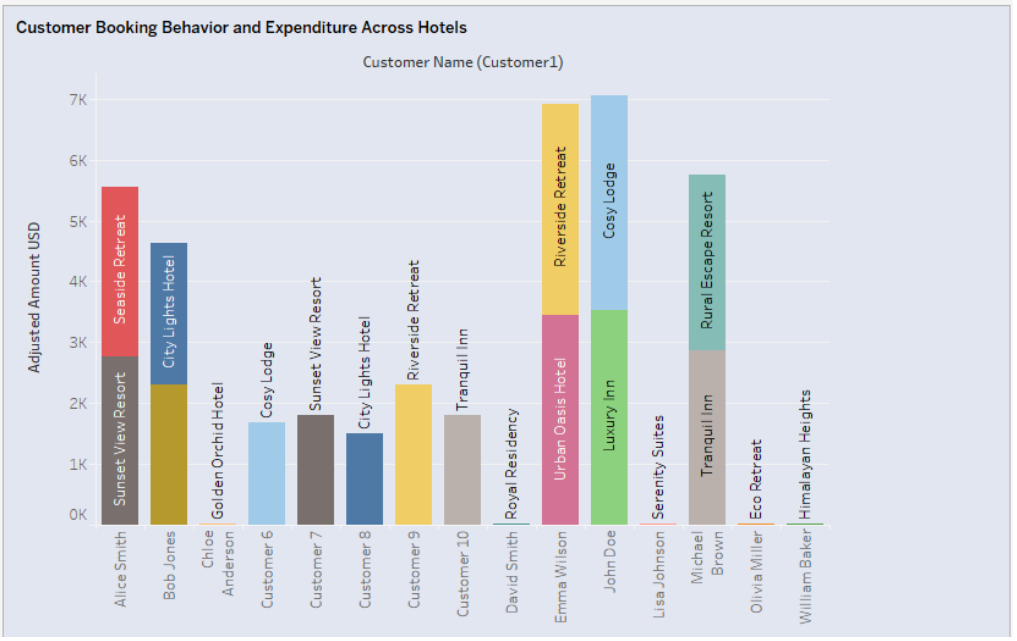
Marks

Automatic

Color | Size | Label | Detail | Tooltip

Property name..

Property name..



Pages

Columns: Property ID | Property name (prop..

Rows: SUM(Number of roo.. | AVG(Price per night (..

Filters

Marks

All

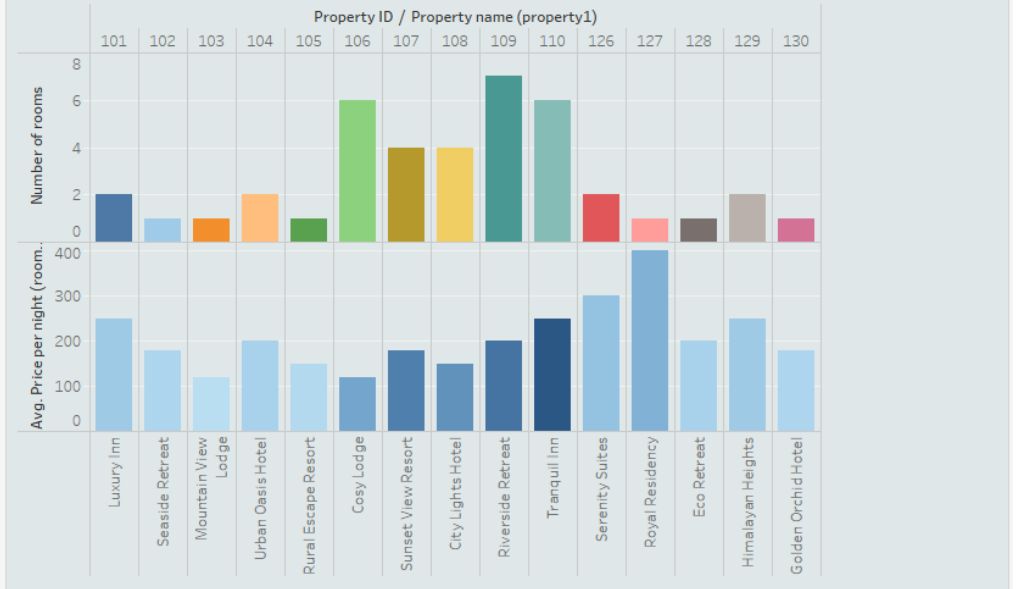
SUM(Number o...

AVG(Price per ...

Automatic

Color | Size | Label | Detail | Tooltip

SUM(Price per..



Pages

Columns Longitude (generated)

Rows Latitude (generated)

Filters

Marks

Map

Color Size Label

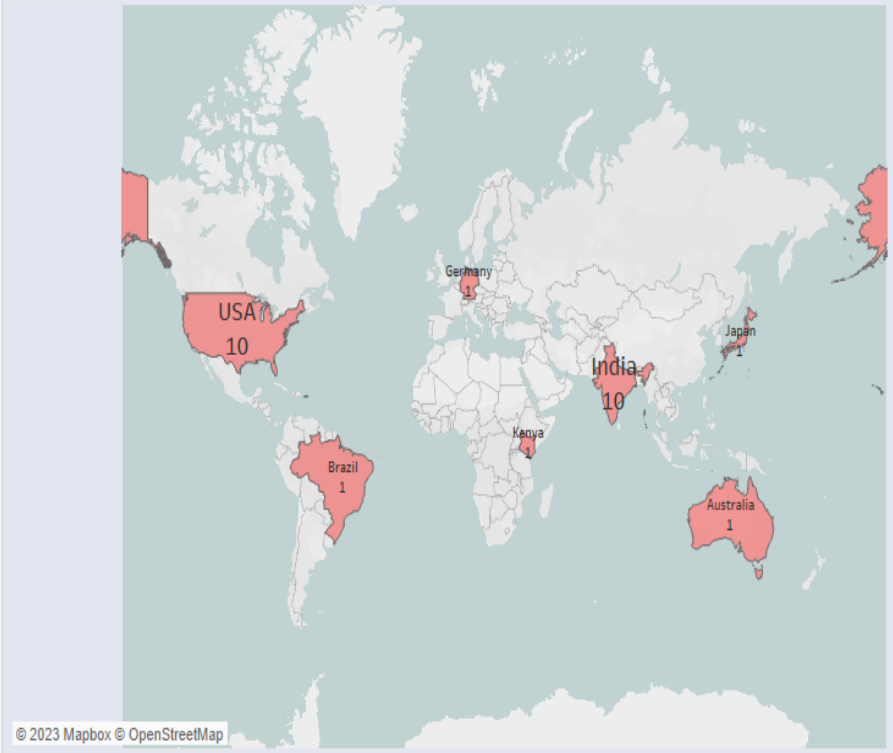
Detail Tooltip

Property coun..

CNT(property1..

CNT(property1..

Hotels per Country



Filters

Marks

Circle

Color Size Label

Detail Tooltip

CNT(account_..

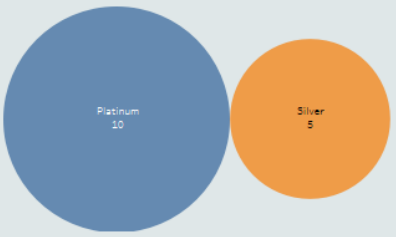
Member Level

CNT(account_..

Member Level

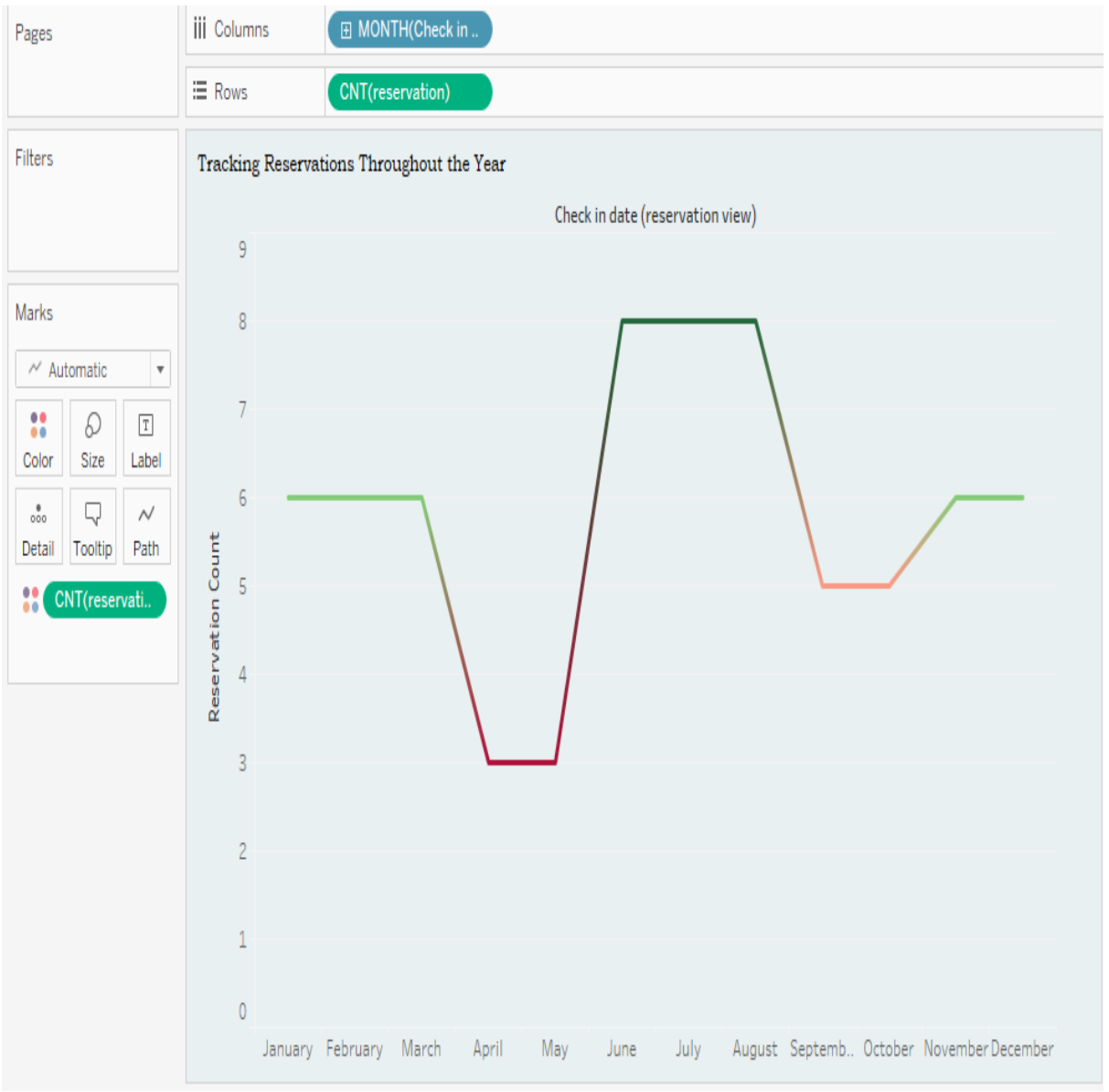
SUM(Total Re..

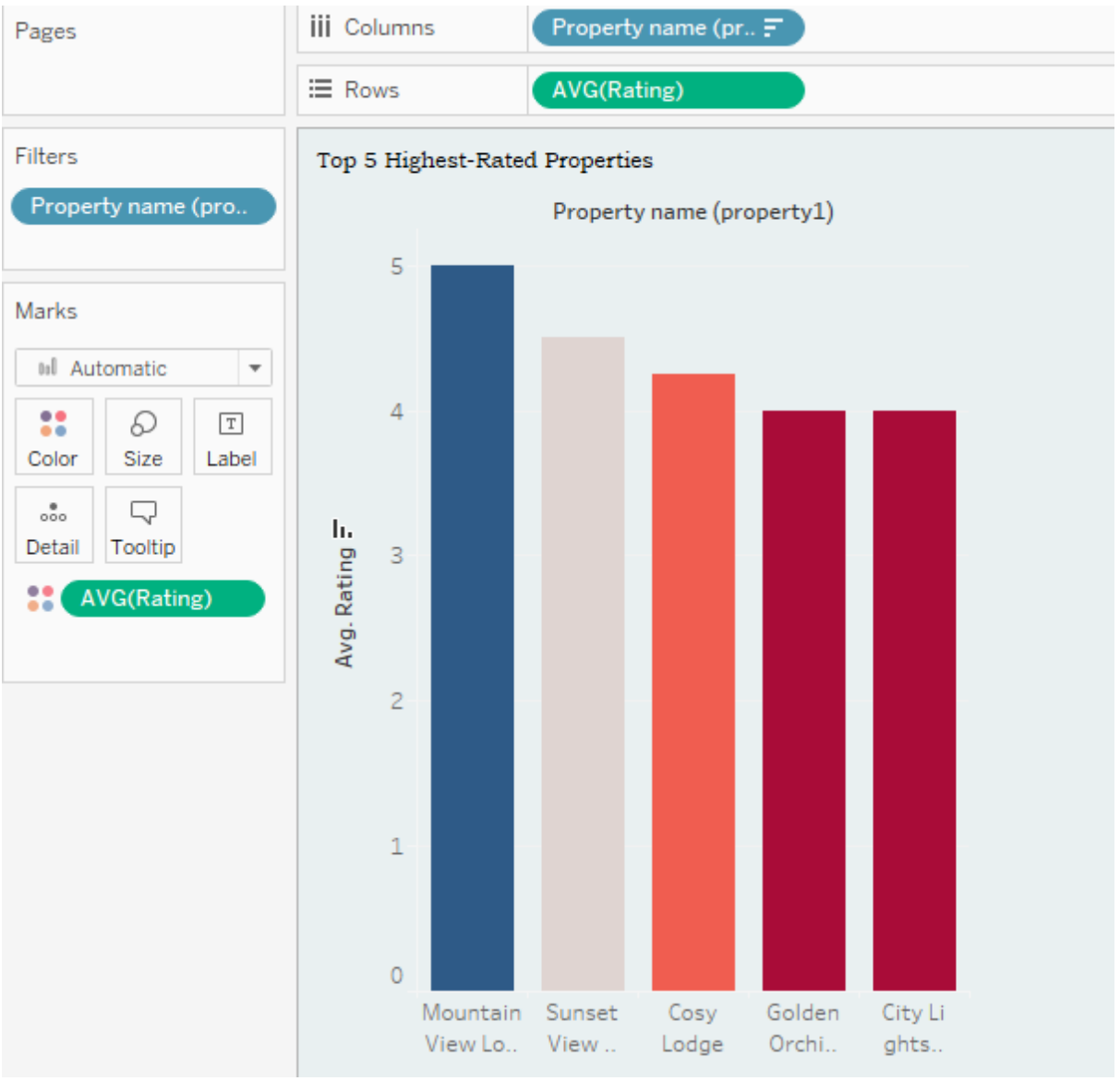
Member Level Distribution




Pages
Columns
SUM(Total Reward P..
Rows
User Name (account ..
Filters
Marks
Automatic
Color
Size
Label
Detail
Tooltip
SUM(Reward ..
SUM(Total Re..
Reward Points per User
User Name (acc..
alice_smith 55.43
bob_jones 46.26
chloe_anderson 0.20
david_smith 0.56
emma_wilson 69.12
john_doe 70.63
lisa_johnson 0.59
michael_brown 57.43
olivia_miller 0.28
user6 33.60
user7 36.00
user8 30.00
user9 46.00
user10 36.00
william_baker 0.49
0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75
Total Reward Points
SUM(Reward Points)
0.20
70.63


Pages
Columns
Room Type
Rows
AVG(Price per night (..
Filters
AVG(Price per night ..
Room Type
Marks
Bar
Color
Size
Label
Detail
Tooltip
SUM(Price per..
AVG(Price per ..
Top 3 Most Luxurious Room Types
Room Type
Avg. Price per night (in Dollars)
400.0
300.0
250.0
0
50
100
150
200
250
300
350
400
Beachfront Villa
Executive Suite
Deluxe Suite


Pages
Columns
MONTH(Check in ..
Rows
CNT(reservation)
Filters
Marks
Automatic
Color
Size
Label
Detail
Tooltip
Path
CNT(reservati..
Tracking Reservations Throughout the Year
Check in date (reservation view)
Reservation Count
9
8
7
6
5
4
3
2
1
0
January
February
March
April
May
June
July
August
Septemb..
October
November
December


Pages
Columns
Property name (pr..
Rows
AVG(Rating)
Filters
Property name (pro..
Marks
Automatic
Color
Size
Label
Detail
Tooltip
AVG(Rating)
Top 5 Highest-Rated Properties
Property name (property1)
Avg. Rating
5
4
3
2
1
0
Mountain View Lo..
Sunset View ..
Cosy Lodge
Golden Orchi..
City Li ghts..