

**IS380 Database Management Project**

**VRBO Management Solution**

By

Leonard Reyes, Ken Tran, Stephanie Santiago Dorado, Tasnime Mekouar, Olga Dashuta

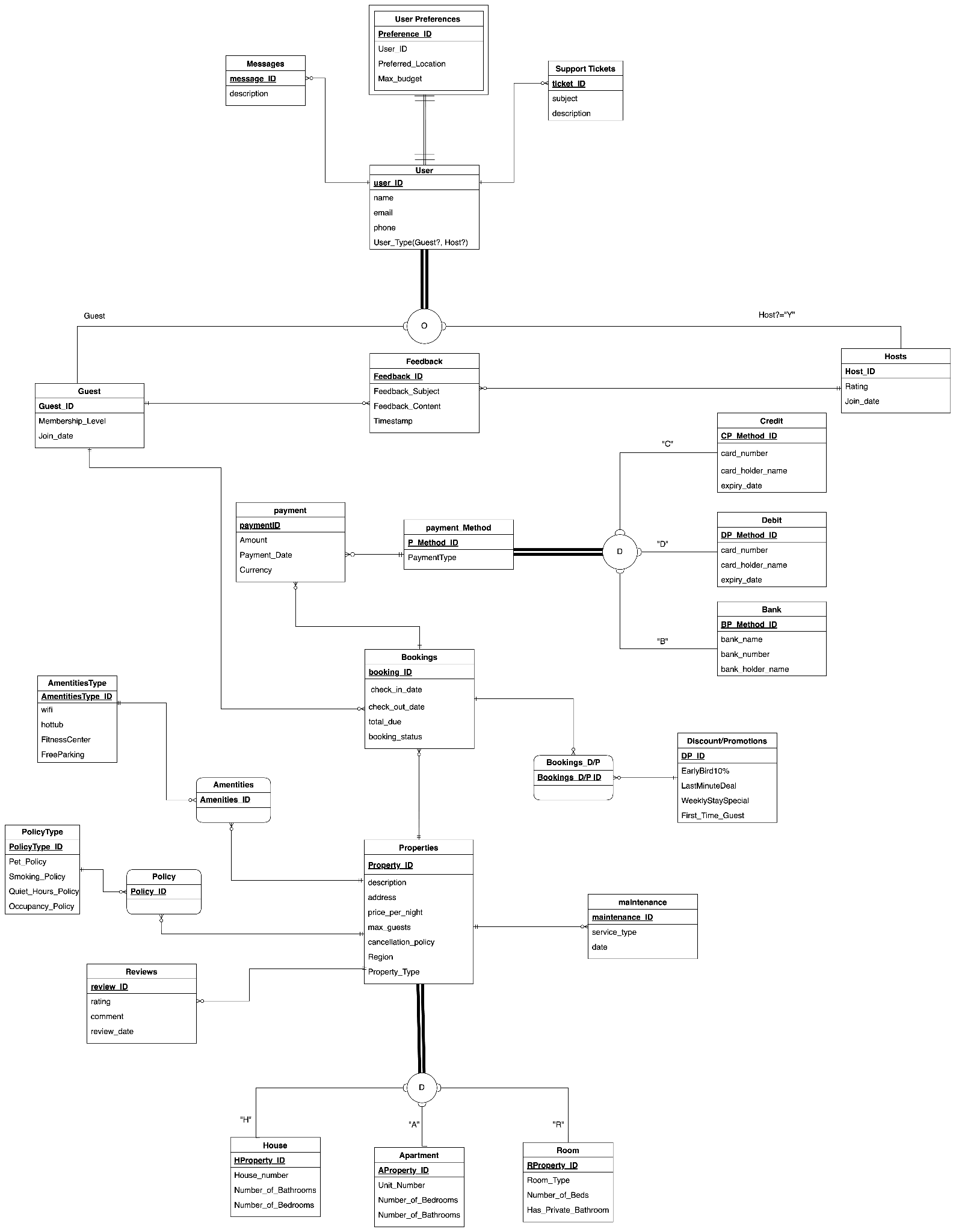
**INTRODUCTION**

*VRBO* is a home-sharing service that has transformed the vacation rental industry. It has changed the way people travel and find accommodations by offering customized vacation rental options. Through VRBO, homeowners can list their properties, which may include shared accommodations or entire homes, which can give travelers a wide range of choices. This platform also allows homeowners to earn additional income by renting out their properties.

However, the rental industry faces challenges, particularly regarding the large amount of data generated by listings. VRBO must provide an efficient search and booking process for guests. Additionally, it needs to offer tools for property owners, such as the ability to update occupancy rates and pricing. Robust customer support services and secure payment processing systems are also essential.

We will create a database management system focused on effective data retrieval and storage. Our design will include a conceptual model that illustrates entities and relationships, a logical model that demonstrates data normalization, and a comprehensive data dictionary. The database will be optimized for performance efficiency and will encompass user profile management, bookings, properties, and reviews. Ultimately, the database we build for VRBO will enhance user satisfaction and improve operational efficiency.

**CONCEPTUAL MODEL**



The conceptual model we will present showcases twenty-five entities with their appropriate relationships to display the key components of the system. The core entities represented in the model are Users, Guests, Hosts, Properties, Bookings, Payments, Reviews, and Support Tickets.

The entities are connected through specific relationships that use various types of cardinalities that accurately represent the interactions within the system, such as Optional One, Optional Many, Mandatory One, and Mandatory Many.

Most entities are linked to the Property entity, and it serves as a core part of the system. For example, in a one-to-many relationship, a single property can have multiple bookings over time, while each booking is linked to only one specific property. Similarly, each guest can make several bookings, but each individual booking is associated with just one guest.

We have also implemented Supertype-Subtype Identity transformations to establish identifying relationships between supertype entities and their corresponding subtypes, which improves the model's clarity and query performance. For example, the Property entity includes three subtypes: House, Apartment, and Room. This structure enables the storage of properties specific to each subtype while retaining general information at the supertype level.

There are two types of constraints in the supertype/subtype relationships within the model. The first is the Completeness Constraint, which ensures that every property must belong to one of the subtypes, such as House, Apartment, or Room. This guarantees that no property remains undefined. The second constraint is the Disjointness Constraint, which specifies that a property instance must belong to exactly one subtype and cannot be assigned to multiple subtypes simultaneously.

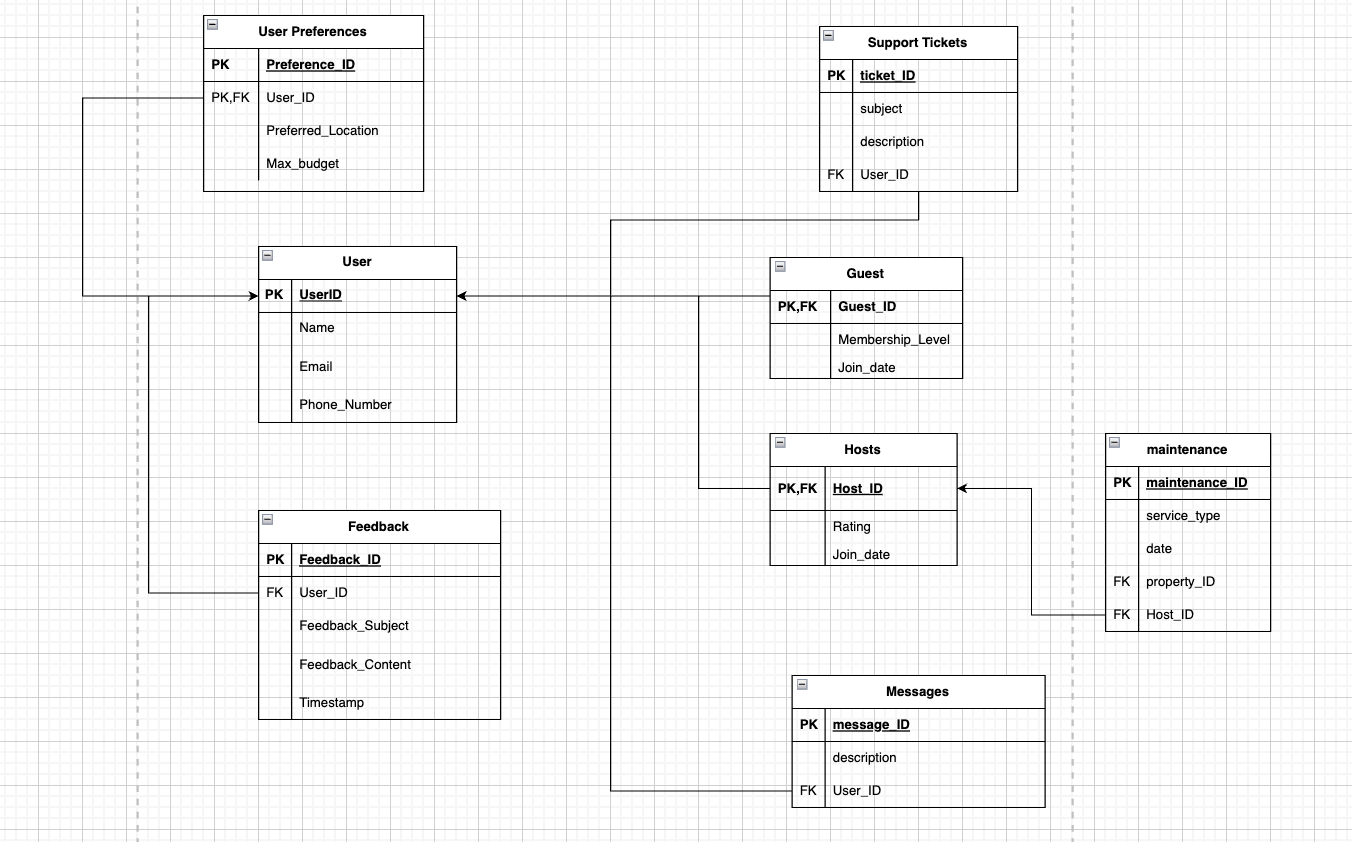
Another entity that utilizes a supertype-subtype structure is the Payment\_Method, which can be either Credit, Debit, or Bank. A payment must be made using a specific method, ensuring disjoint and total participation.

In the model, there are several supporting entities, such as Feedback, which manages guest reviews for properties; Support\_Ticket, which addresses customer service issues; User\_Preferences; and DiscountsPromotions. With these supporting entities, it can help ensure that the database can support a vacation rental application that encompasses necessary business processes such as customer support, amenities management, payments, insurance policies, and information about attractions near the properties.

**LOGICAL MODEL**

In order to create an actual database, we must first transform our conceptual model into a form that preserves data integrity and ensures consistency. Therefore, we converted the components of the conceptual model into a logical model that can be implemented. More than 25 entities from our EER diagram have been transferred into the logical model, which includes clearly defined primary and foreign keys while adhering to domain constraints, entity integrity, and referential integrity.

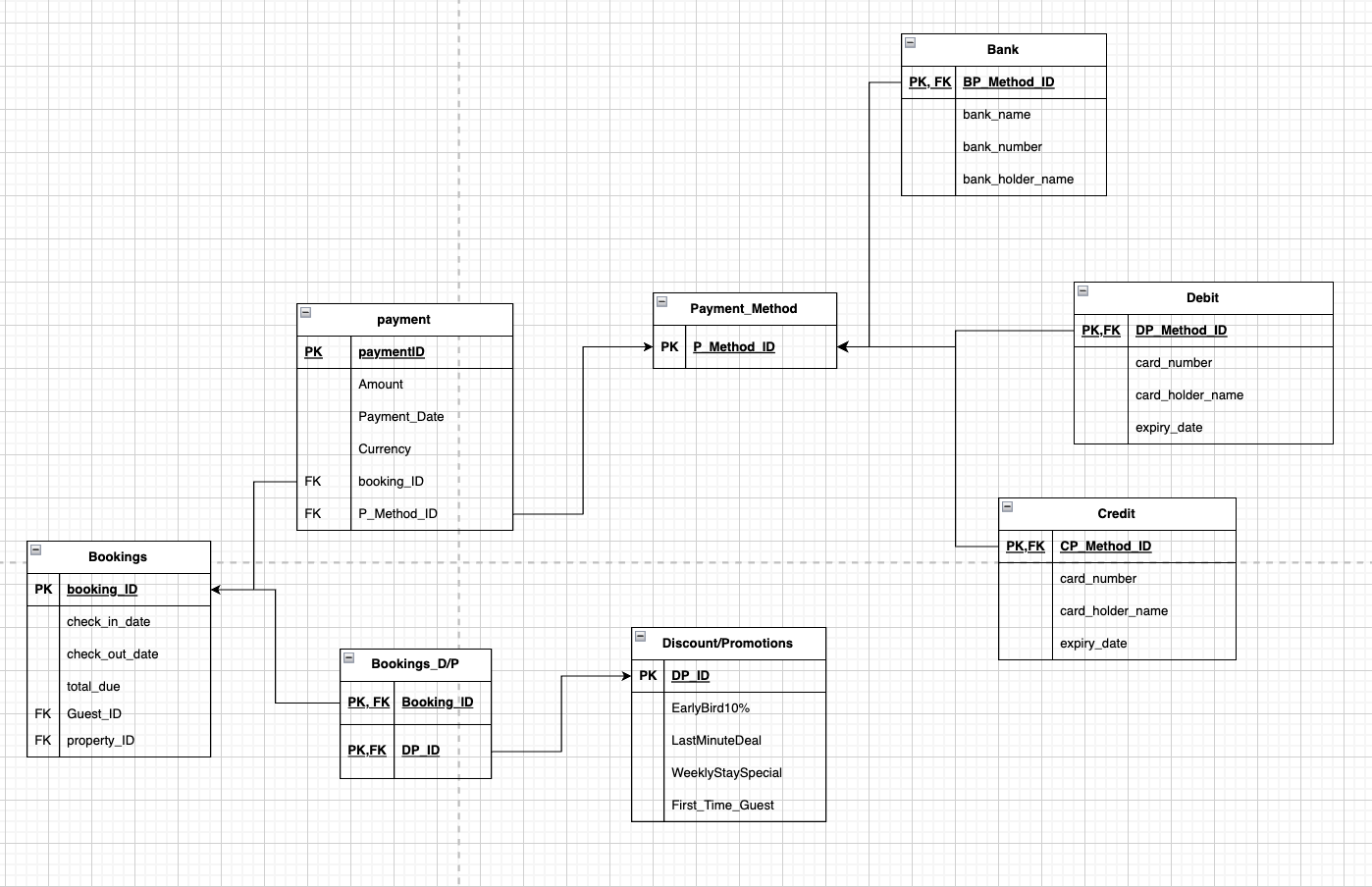
We divided our logical model into 3 clusters for clarity: user cluster, booking cluster, and property cluster.



**Uer Cluster**

The User cluster is responsible for managing user data and interactions, storing details such as the user's name, phone number, and email.

* The User entity is divided into two subtypes: Guest and Host. Guests are users who book properties, while Hosts are users who list and manage those properties.
* To enhance user personalization, there is a User Preference entity that captures individual preferences, such as preferred location and maximum budget.
* When a user encounters an issue, they can generate Support Tickets that are linked to their User ID.
* To facilitate user feedback, the Feedback entity was created, which includes a subject, content, and a timestamp.
* The Message entity enables communication between users, while the Maintenance entity records maintenance tasks required for properties. This entity is linked to Hosts and the properties they manage.
* The cluster efficiently manages all types of users, tracking and organizing support requests, communications, and preferences.

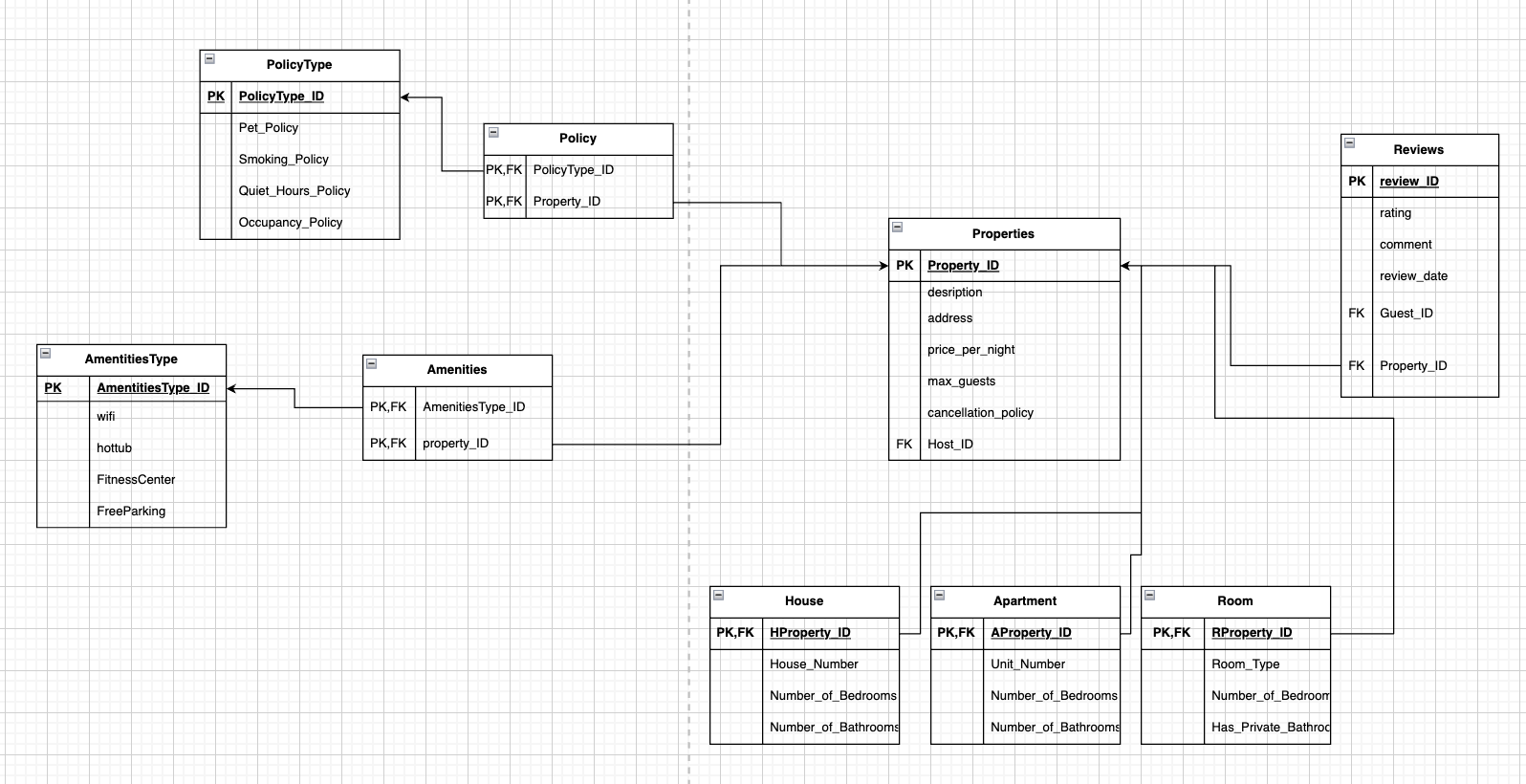


**Bookings Cluster**

The Booking entity above is the central component of the cluster. This cluster will handle the financial and transaction operations of the system. The cluster will include details for each reservation, like the check-in and check-out dates, the total amount due, and the information about the guest and the property involved.

* The Payment entity records transactions related to bookings, including the payment date, amount, and currency.
* The Payment Method serves as a supertype for three subtypes: Credit, Debit, and Bank. Payments are processed through this entity, and each subtype includes specific details such as card\_numbers or bank\_holder names.
* Special offers, like early bird discounts and last-minute deals, are managed through the Discount or Promotions entity, which helps to enhance user engagement.
* An association between bookings and promotions is established by the Bookings\_D/P entity. This allows us to understand which promotions can be applied to specific properties.

With this cluster, we can ensure that booking transactions are securely stored, and well-managed and that promotions are properly placed and applied.



**Properties Cluster**

The Property cluster will manage property listings, features, customer feedback, and policies. Property entity will also hold essential details for rental property listings, including the address, maximum number of guests allowed, and the price per night.

* The Property entity consists of three subtypes: House, Apartment, and Room. Each property will be categorized into one of these subtypes, which have specific attributes, such as whether a room has a private bathroom and the number of beds.
* The AmenitiesType offers features like hot tubs, fitness centers, Wi-Fi, and free parking. These are the Amenities that the property will offer.
* The Policy entity outlines the rules related to the property, with categories such as pet policy and smoking policy.
* The Review entity gathers the customer’s feedback regarding a property they rented. The Review gathers and showcases the customer’s comments, ratings, and the date it was posted.

This cluster will help guests make better decisions to pick which property is best suited for them by listing the property's amenities, attributes, reviews, and property policy.

**DATA DICTIONARY**

**Entity: User**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entity** | **Attribute** | **Data Type** | **Range** | **Key** | **Description** |
| **User** | user\_ID | INT | 1001-1020 | PK | Unique user ID |
|  | name | VARCHAR2(255) |  |  | Full name of the user |
|  | email | VARCHAR2(255) |  |  | User's email address |
|  | phone | VARCHAR2(20) |  |  | User's phone number |
|  | user\_type | VARCHAR2(20) |  |  | Guest or Host |
| **Guest** | guest\_ID | INT | 1001-1020 | PK, FK | Links to user\_ID |
|  | membership\_level | VARCHAR2(50) |  |  | Membership status (e.g., Silver, Gold) |
|  | join\_date | DATE |  |  | When the guest registered |
| **Host** | host\_ID | INT | 1001-1020 | PK, FK | Links to user\_ID |
|  | rating | NUMBER(2,1) | 0.0 - 5.0 |  | Average rating score |
|  | join\_date | DATE |  |  | Date joined as a host |
| **Properties** | property\_ID | INT | 3001-3010 | PK | Unique property ID |
|  | description | CLOB |  |  | Property description |
|  | address | VARCHAR2(255) |  |  | Property full address |
|  | price\_per\_night | NUMBER(10,2) |  |  | Nightly rate |
|  | max\_guests | INT | 1-10 |  | Maximum number of guests allowed |
|  | cancellation\_policy | VARCHAR2(100) |  | FK | Policy names. Linked to PolicyType |
|  | region | VARCHAR2(255) |  |  | Region or area name |
|  | property\_type | VARCHAR2(50) |  | FK | (Apartment, House, Room) |
| **Bookings** | booking\_ID | INT | 4001-4040 | PK | Unique booking ID |
|  | check\_in\_date | DATE |  |  | Check-in date |
|  | check\_out\_date | DATE |  |  | Check-out date |
|  | total\_due | NUMBER(10,2) | 0.00+ |  | Total amount for booking |
|  | booking\_status | VARCHAR2(50) |  |  | Status (confirmed, cancelled) |
|  | guest\_ID | INT | 1001-1020 | FK | Guest who made the booking |
|  | property\_ID | INT | 3001-3010 | FK | Property booked |
| **Reviews** | review\_ID | INT | 5001-5040 | PK | Unique review ID |
|  | rating | NUMBER(2,1) | 0.0 - 5.0 |  | Star rating |
|  | comment | CLOB |  |  | Review comment text |
|  | review\_date | DATE |  |  | Date of review |
|  | guest\_ID | INT | 1001-1020 | FK | Reviewer (Guest) |
|  | property\_ID | INT | 3001-3010 | FK | Reviewed property |
| **Payment** | payment\_ID | INT | 6001-6040 | PK | Unique payment ID |
|  | amount | NUMBER(10,2) |  |  | Payment amount |
|  | payment\_date | DATE |  |  | Date of payment |
|  | currency | VARCHAR2(3) |  |  | Currency code (USD, etc.) |
|  | booking\_ID | INT | 4001-4040 | FK | Payment linked to a booking |
| **Messages** | message\_ID | INT |  | PK | Unique message ID |
|  | description | CLOB |  |  | Content of message |
|  | user\_ID | INT | 1001-1020 | FK | Linked to sender user |
| **PolicyType** | policytype\_ID | INT |  | PK | Unique policy ID |
|  | pet\_policy | VARCHAR2(50) |  |  | Rules for pets. ‘Allowed’, ‘Not Allowed’. |
|  | smoking\_policy | VARCHAR2(50) |  |  | Smoking allowed? ‘Allowed’, ‘Not Allowed’. |
|  | quiet\_hours\_policy | VARCHAR2(255) |  |  | Quiet time restrictions |
|  | occupancy\_policy | VARCHAR2(255) |  |  | Maximum occupancy rules |
| **Apartment** | Aproperty\_ID | INT | 3001-3010 | PK, FK | FK to Properties |
|  | unit\_number | VARCHAR2(50) |  |  | Apartment-specific detail |
|  | number\_of\_bedrooms | INT | 0-10 |  | Bedrooms count |
|  | number\_of\_bathrooms | INT | 0-10 |  | Bathrooms count |
| **House** | Hproperty\_ID | INT | 3001-3010 | PK, FK | FK to Properties |
|  | house\_number | VARCHAR2(50) |  |  | House-specific detail |
|  | number\_of\_bedrooms | INT | 0-10 |  | Bedrooms count |
|  | number\_of\_bathrooms | INT | 0-10 |  | Bathrooms count |
| **Room** | Rproperty\_ID | INT | 3001-3010 | PK, FK | FK to Properties |
|  | room\_type | VARCHAR2(50) |  |  | Private/shared room |
|  | number\_of\_beds | INT | 1-10 |  | Beds available |
|  | has\_private\_bathroom | CHAR(1) |  |  | Yes/No private bathroom. ‘Y’/’N’ |

**PHYSICAL DESIGN**

1. **Whether some tables will be partitioned, or denormalized and why.**

Whether we’ll be using partitioning or denormalization to support scalability and to provide a strong performance ultimately depends on where it is appropriate. Because most search results are mainly focused on recent reservations, Bookings will be divided by date, such as monthly or quarterly. By using this process, archiving old data is simplified and helps speed queries for bookings. To help reduce data redundancy, database design uses normalization principles, and for faster performance, denormalization will be applied. This helps avoid the need for repeated joins within the Review table by instead storing the ratings directly in the Properties table.

1. **Which entities (subjects or objects) will have access to some functions of the database and which would those be (insert, read, modify, delete)? Are there constraints for them to maintain security?**

Users can have different levels of access within the system. But in order for the user to have access to these levels, it will ultimately depend on their role. An example is that a Host can put their properties up for rent, update property information, and manage their listings. The Host can also chat with their guests if they have any questions or concerns. Guests can review properties, manage their profiles, and look up and book properties. To have access to everything, you will need an administrative authority. The administrators will have access to payments, user management, and maintaining the system. So, each user will have limited access depending on their role. To ensure this and keep the systems secured, we will have the users access the system with their login credentials. We will also have records of any changes within the system, such as if there are any bugs that need to be resolved, account changes, payment changes, or if there is any suspicious activity going on.

1. **What data type will each attribute possess? Why?**

For the entities that have an ID, we will be using INT. Some examples are userID and bookingID. This helps with sorting and looking them up quickly. We will be using VARCHAR for emails, names, and addresses. DECIMAL will be used for currency, such as price and budget. DATE will be used for dates because this will help with the check-in and check-out times. BOOLEAN will be used for Yes/No fields, for example, we can use BOOLEAN to help check if a property has certain amenities.

1. **How are you organizing files (Heap, Sequential order, Indexed, or Hashed) and why?**

There will be different methods of organizing data files depending on how each table is used. For example, Messages and Support\_Tickets will use heap storage because heap storage is good for tables that get a lot of random inserts. Whereas tables like Users, Properties, and Bookings will be indexed so we can quickly find data by user ID, property ID, etc. Tables such as Reviews, and Maintenance where we usually look up entries by date will be organized using sequential order storage. For quick searches by exact ID, hashed storage can give us faster results. This mix of ways to organize files ensures that we're balancing speed and flexibility across the system.

1. **Please add a brief description of our database recovery policy (are you including backups, journalizing, etc.? How will you restore the database in case of a disaster (disk mirroring, restore, Rollback, or roll forward)?**

Every night, there will be a full backup of the entire database, combined with smaller incremental backups throughout the day to capture changes. In addition, each transaction will be logged in real-time making sure that if something crashes we can still roll back incomplete actions or roll forward to the most recent stable point. On the hardware side of things, we will implement disk mirroring to make sure that there is always a live copy ready to go even if one disk fails. By doing so we can recover quickly from any crisis and minimize downtime for our users.

**SQL QUERIES**

To support VRBO’s platform goals of enhancing guest satisfaction, host success, and operational efficiency, we designed a series of SQL queries that aim to extract meaningful insights from the platform’s data. We will create queries that will help demonstrate user behavior, host insights, property performance, etc. By analyzing these areas, the platform can make better-informed business decisions, improve user experiences, and create new growth opportunities for hosts and guests.

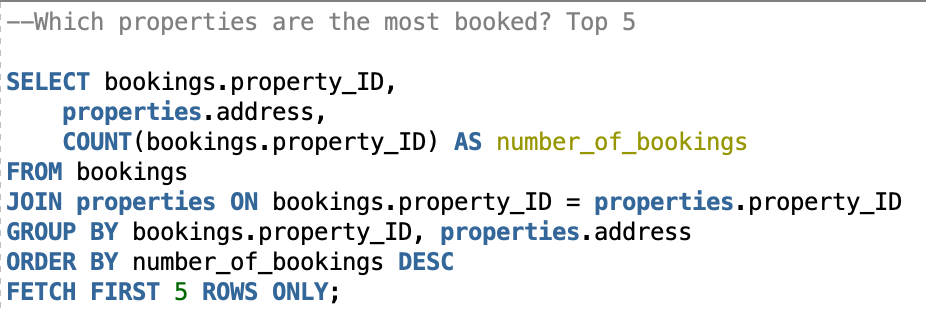
The purpose of this database is to manage and analyze information related to rentals, including users, bookings, payments, and reviews. It is designed to support both operational needs (e.g., managing bookings and payments) and broader strategic analysis (e.g., identifying high-performing hosts or popular regions). The database accepts various inputs while keeping track of the relationships between guests, hosts, and properties. Inputs include user registration details such as name, user type (guest or host), and contact information; property listings with information on location, host, and price; and booking records capturing information such as the guest, check-in and check-out dates, and booking status; It also stores guest ratings and reviews, as well as payment amounts and transaction date. The primary users of this database include: VRBO administrators who need reports on revenue, user behavior, and property performance; Hosts who benefit from feedback and performance insights; and guests whose activity and preferences inform the platform’s personalized recommendations

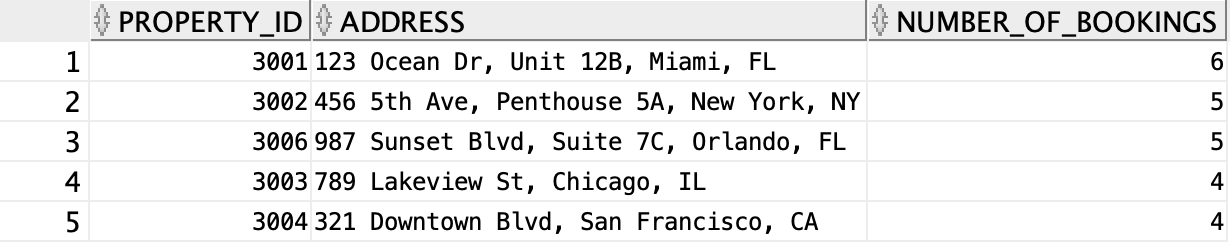
**The queries that will be showcased for this project are the following:**

1. Which properties are the most booked? (Top 5)
2. Which hosts have the best average ratings?
3. What type of property gets booked the most?
4. How does booking length vary by property type? Do guests stay longer in certain types of properties?
5. Which guests book the most properties?
6. Which properties generate the most revenue?

**Query #1:**

**Which properties are the most booked? (Top 5)**

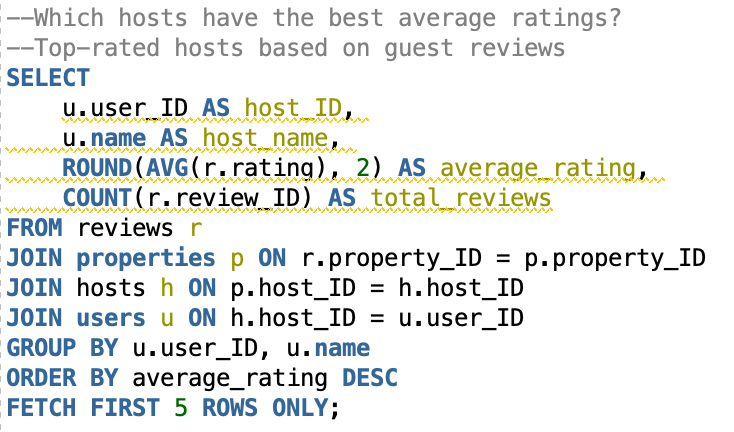
**Query Result:**



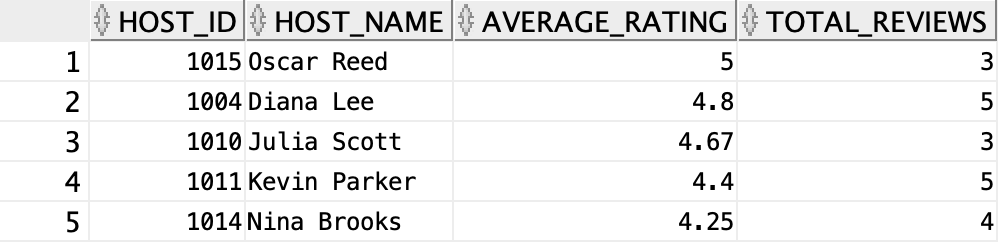
This query fetches us the top 5 properties that are most booked. As you can see, the results return the property ID, address, and the number of bookings per property. This query can be useful for identifying the high-demand listings that could be featured and promoted in marketing campaigns or used as examples for host onboarding.

**Query #2:**

**Which hosts have the best average ratings?**

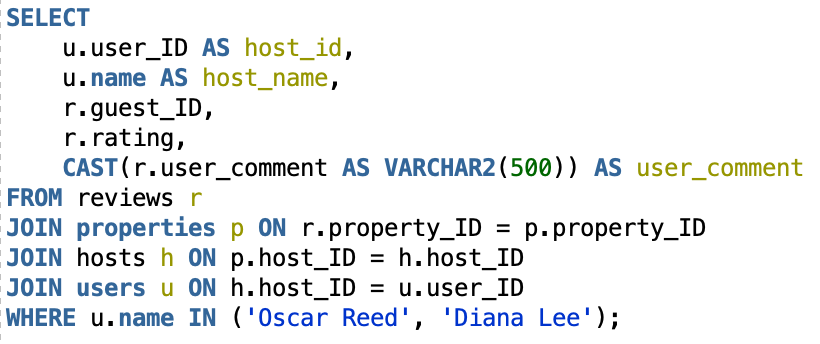


**Query Result:**

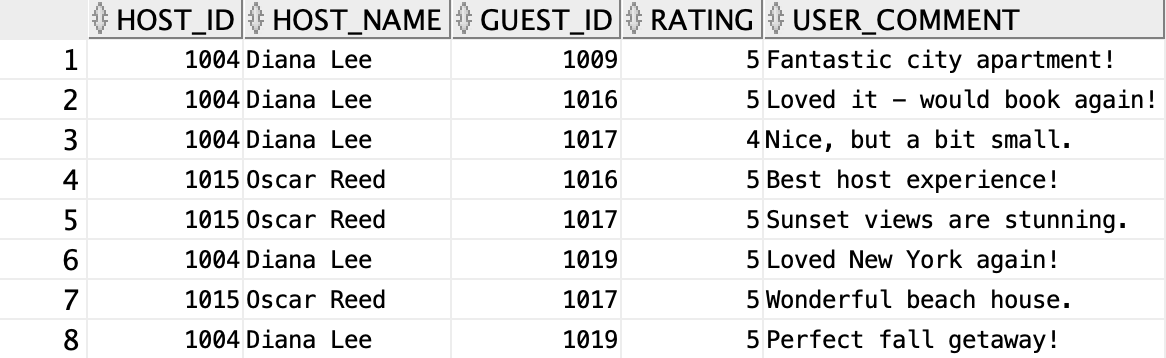


This query finds hosts with the highest average review scores and total review counts. This enables VRBO to highlight and reward top-performing hosts and use top hosts as a guide to train others. It also supports quality assurance and trust-building among users.

**Reviews for Specific Hosts:**



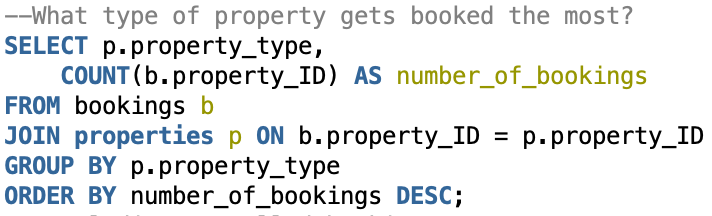
**Query Result:**



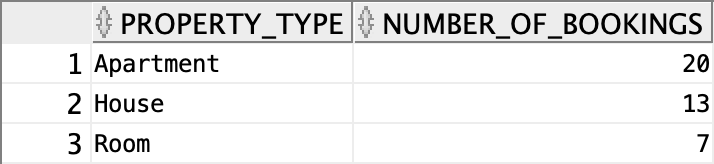
This query contains reviews for specific highly rated hosts, which can be useful in figuring out why particular hosts are top-rated. As we can see from query two, Oscar Reed and Diana Lee were the top two in the best average rating. So, if one wants to find out how these hosts separate from other hosts, we can refer to this query, which contains each host's rating and review. Similarly, if we wanted to find out why specific hosts have low ratings, we would have to first find the lowest average rating. We could use query #2 and filter by ascending order in the ORDER BY statement. Then, using that information, we could use the query for finding reviews of specific hosts and replace the names in the WHERE statement with the ones who had the lowest rating.

**Query #3:**

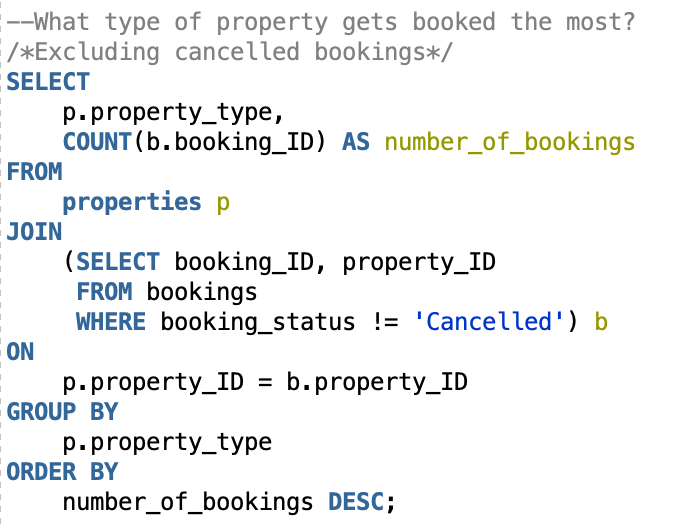
**What type of property gets booked the most?**



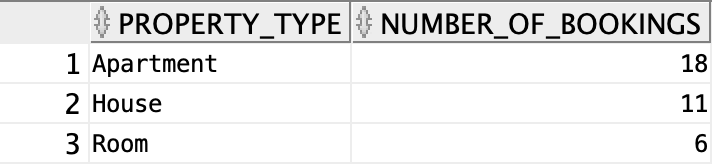
**Query Result:**



**Subquery – Bookings per property, excluding cancellations**



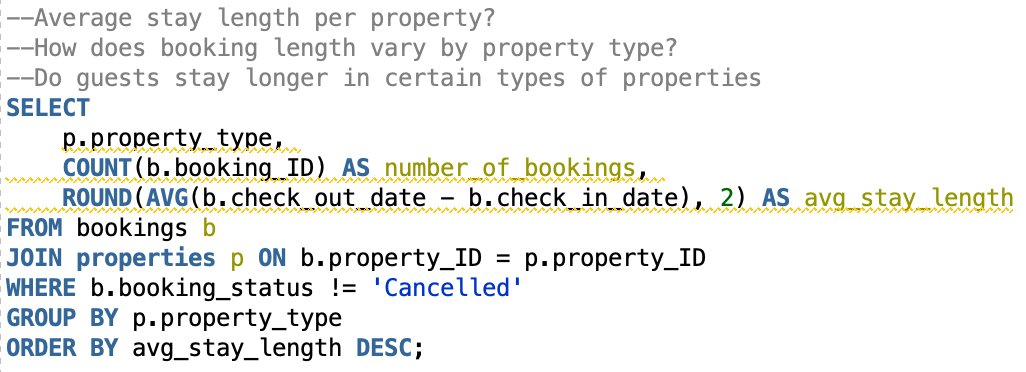
**Query Result:**



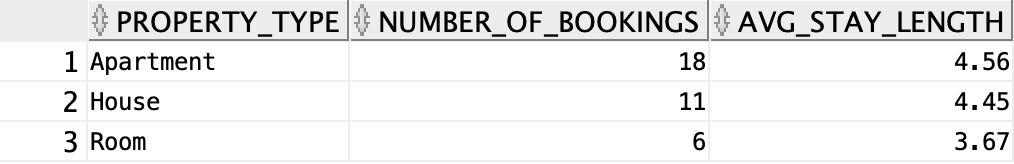
In this query, we compared booking volumes across apartments, houses, and rooms. This guides hosts on what kinds of properties are most profitable to list.

**Query #4:**

**How does booking length vary by property type? Do guests stay longer in certain types of properties?**



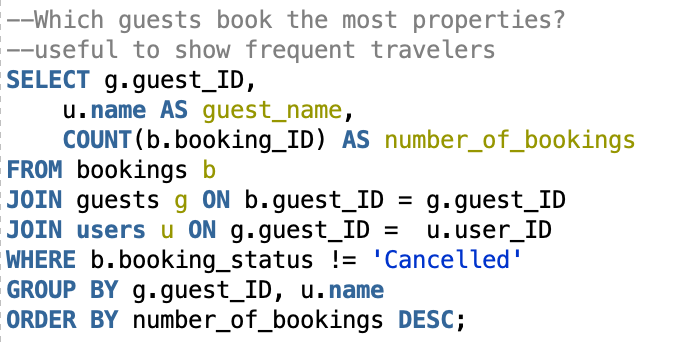
**Query Result:**



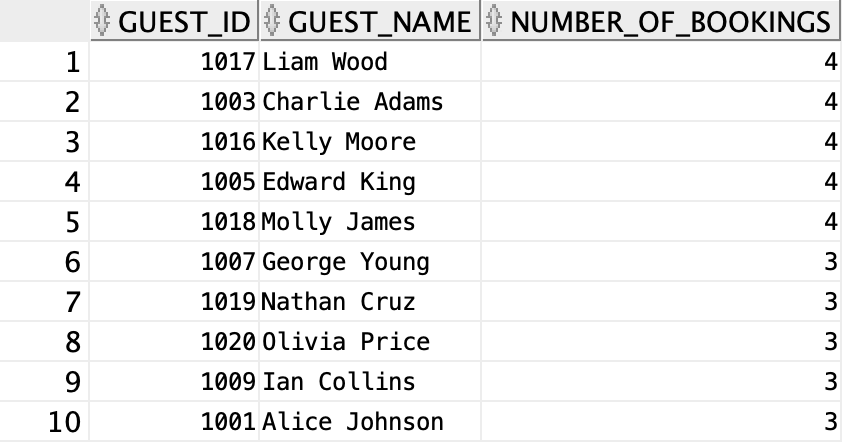
Suppose one wanted to figure out how booking length varies by property type. To find this, we created a query that calculated the average stay length per property type. This can help offer insight into guest behavior and inform hosts how to price and plan. With the results shown, we can assume that people book rooms for short-term stays, and with longer vacations, apartments and houses are the most popular.

**Query #5:**

**Which guests book the most properties?**



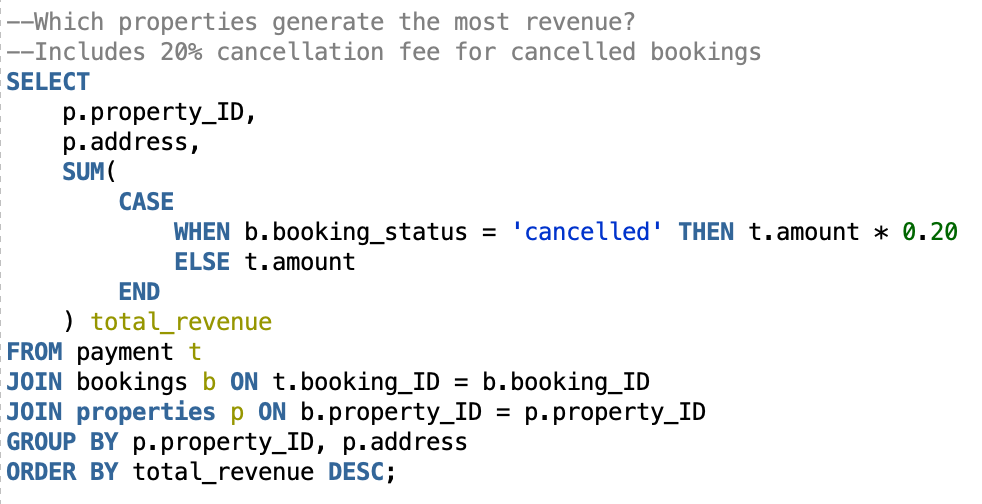
**Query Result:**



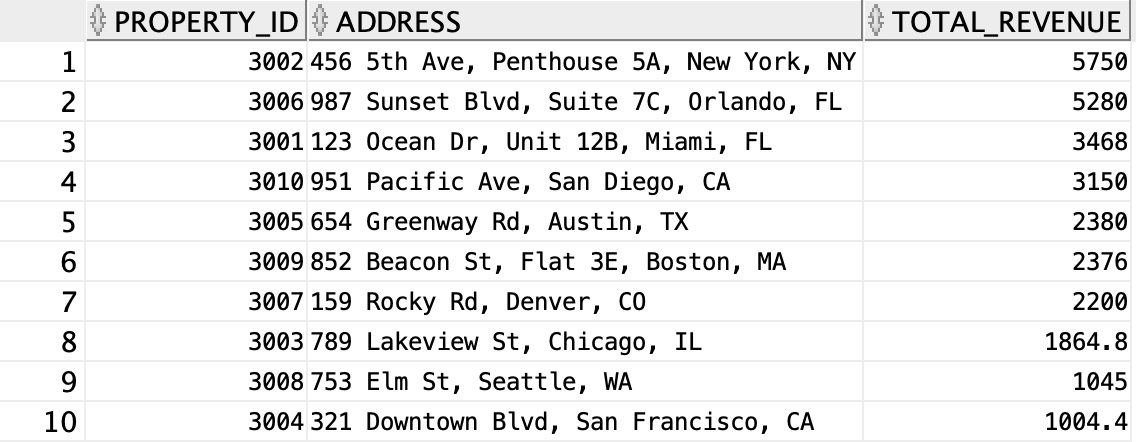
This query lists guests with the most bookings, which can help identify the most active and loyal guests. Helps VRBO target high-value customers with loyalty programs, early access deals, or personalized service to increase platform engagement.

**Query #6:**

**Which properties generate the most revenue?**



**Query Result:**



This query highlights listings that contribute the most to platform earnings. Note that we included the number of bookings, including the cancellations that were charged only 20% of the original booking.

**SCHEMA**

-- Parent Tables CREATE TABLE Users ( user\_ID INT PRIMARY KEY, name VARCHAR2(255), email VARCHAR2(255), phone VARCHAR2(20), user\_type VARCHAR2(20) ); CREATE TABLE PolicyType ( policytype\_ID INT PRIMARY KEY, pet\_policy VARCHAR2(50), smoking\_policy VARCHAR2(50), quiet\_hours\_policy VARCHAR2(255), occupancy\_policy VARCHAR2(255) ); CREATE TABLE Hosts ( host\_ID INT PRIMARY KEY, rating NUMBER(2,1), join\_date DATE, FOREIGN KEY (host\_ID) REFERENCES Users(user\_ID) ); CREATE TABLE Properties ( property\_ID INT PRIMARY KEY, description CLOB, address VARCHAR2(255), price\_per\_night NUMBER(10,2), max\_guests INT, cancellation\_policy VARCHAR2(100), region VARCHAR2(255), property\_type VARCHAR2(50), host\_ID INT, FOREIGN KEY (host\_ID) REFERENCES Hosts(host\_ID) ); -- Child Tables CREATE TABLE Guests ( guest\_ID INT PRIMARY KEY, membership\_level VARCHAR2(50), join\_date DATE, FOREIGN KEY (guest\_ID) REFERENCES Users(user\_ID) ); CREATE TABLE Bookings ( booking\_ID INT PRIMARY KEY, check\_in\_date DATE, check\_out\_date DATE, total\_due NUMBER(10,2), booking\_status VARCHAR2(50), guest\_ID INT, property\_ID INT, FOREIGN KEY (guest\_ID) REFERENCES Guests(guest\_ID), FOREIGN KEY (property\_ID) REFERENCES Properties(property\_ID) ); CREATE TABLE Reviews ( review\_ID INT PRIMARY KEY, rating NUMBER(2,1), comment CLOB, review\_date DATE, guest\_ID INT, property\_ID INT, FOREIGN KEY (guest\_ID) REFERENCES Guests(guest\_ID), FOREIGN KEY (property\_ID) REFERENCES Properties(property\_ID) ); CREATE TABLE Payment ( payment\_ID INT PRIMARY KEY, amount NUMBER(10,2), payment\_date DATE, currency VARCHAR2(3), booking\_ID INT, FOREIGN KEY (booking\_ID) REFERENCES Bookings(booking\_ID) ); CREATE TABLE Messages ( message\_ID INT PRIMARY KEY, description CLOB, user\_ID INT, FOREIGN KEY (user\_ID) REFERENCES Users(user\_ID) ); -- Specialized Property Types CREATE TABLE Apartment ( Aproperty\_ID INT PRIMARY KEY, unit\_number VARCHAR2(50), number\_of\_bedrooms INT, number\_of\_bathrooms INT, FOREIGN KEY (Aproperty\_ID) REFERENCES Properties(property\_ID) ); CREATE TABLE House ( Hproperty\_ID INT PRIMARY KEY, house\_number VARCHAR2(50), number\_of\_bedrooms INT, number\_of\_bathrooms INT, FOREIGN KEY (Hproperty\_ID) REFERENCES Properties(property\_ID) ); CREATE TABLE Room ( Rproperty\_ID INT PRIMARY KEY, room\_type VARCHAR2(50), number\_of\_beds INT, has\_private\_bathroom CHAR(1), FOREIGN KEY (Rproperty\_ID) REFERENCES Properties(property\_ID) );

-- Insert into Users

INSERT INTO Users (user\_ID, name, email, phone, user\_type) VALUES

(1001, 'Alice Johnson', 'alice.johnson@example.com', '5551234567', 'Guest');

INSERT INTO Users (user\_ID, name, email, phone, user\_type) VALUES

(1002, 'Bob Smith', 'bob.smith@example.com', '5559876543', 'Host');

INSERT INTO Users (user\_ID, name, email, phone, user\_type) VALUES

(1003, 'Charlie Adams', 'charlie.adams@example.com', '5552223333', 'Guest');

INSERT INTO Users (user\_ID, name, email, phone, user\_type) VALUES

(1004, 'Diana Lee', 'diana.lee@example.com', '5554445555', 'Host');

INSERT INTO Users (user\_ID, name, email, phone, user\_type) VALUES

(1005, 'Edward King', 'edward.king@example.com', '5556667777', 'Guest');

INSERT INTO Users (user\_ID, name, email, phone, user\_type) VALUES

(1006, 'Fiona Baker', 'fiona.baker@example.com', '5558889999', 'Host');

INSERT INTO Users (user\_ID, name, email, phone, user\_type) VALUES

(1007, 'George Young', 'george.young@example.com', '5553332222', 'Guest');

INSERT INTO Users (user\_ID, name, email, phone, user\_type) VALUES

(1008, 'Hannah Wright', 'hannah.wright@example.com', '5554448888', 'Host');

INSERT INTO Users (user\_ID, name, email, phone, user\_type) VALUES

(1009, 'Ian Collins', 'ian.collins@example.com', '5551117777', 'Guest');

INSERT INTO Users (user\_ID, name, email, phone, user\_type) VALUES

(1010, 'Julia Scott', 'julia.scott@example.com', '5559990000', 'Host');

INSERT INTO Users (user\_ID, name, email, phone, user\_type) VALUES

(1011, 'Kevin Parker', 'kevin.parker@example.com', '5553331111', 'Host');

INSERT INTO Users (user\_ID, name, email, phone, user\_type) VALUES

(1012, 'Laura Davis', 'laura.davis@example.com', '5552226666', 'Host');

INSERT INTO Users (user\_ID, name, email, phone, user\_type) VALUES

(1013, 'Mike Turner', 'mike.turner@example.com', '5557771111', 'Host');

INSERT INTO Users (user\_ID, name, email, phone, user\_type) VALUES

(1014, 'Nina Brooks', 'nina.brooks@example.com', '5554443333', 'Host');

INSERT INTO Users (user\_ID, name, email, phone, user\_type) VALUES

(1015, 'Oscar Reed', 'oscar.reed@example.com', '5556662222', 'Host');

INSERT INTO Users (user\_ID, name, email, phone, user\_type) VALUES

(1016, 'Kelly Moore', 'kelly.moore@example.com', '5553334444', 'Guest');

INSERT INTO Users (user\_ID, name, email, phone, user\_type) VALUES

(1017, 'Liam Wood', 'liam.wood@example.com', '5551112222', 'Guest');

INSERT INTO Users (user\_ID, name, email, phone, user\_type) VALUES

(1018, 'Molly James', 'molly.james@example.com', '5559998888', 'Guest');

INSERT INTO Users (user\_ID, name, email, phone, user\_type) VALUES

(1019, 'Nathan Cruz', 'nathan.cruz@example.com', '5557776666', 'Guest');

INSERT INTO Users (user\_ID, name, email, phone, user\_type) VALUES

(1020, 'Olivia Price', 'olivia.price@example.com', '5554443333', 'Guest');

-- Insert into PolicyType

INSERT INTO PolicyType (policytype\_ID, pet\_policy, smoking\_policy, quiet\_hours\_policy, occupancy\_policy) VALUES

(2001, 'Allowed', 'Not Allowed', '10PM-7AM Quiet Hours', 'Maximum 4 guests');

INSERT INTO PolicyType (policytype\_ID, pet\_policy, smoking\_policy, quiet\_hours\_policy, occupancy\_policy) VALUES

(2002, 'Not Allowed', 'Allowed', '11PM-6AM Quiet Hours', 'Maximum 6 guests');

INSERT INTO PolicyType (policytype\_ID, pet\_policy, smoking\_policy, quiet\_hours\_policy, occupancy\_policy) VALUES

(2003, 'Allowed with Fee', 'Not Allowed', '9PM-7AM Quiet Hours', 'Maximum 2 guests');

INSERT INTO PolicyType (policytype\_ID, pet\_policy, smoking\_policy, quiet\_hours\_policy, occupancy\_policy) VALUES

(2004, 'Not Allowed', 'Not Allowed', '10PM-8AM Quiet Hours', 'Maximum 5 guests');

INSERT INTO PolicyType (policytype\_ID, pet\_policy, smoking\_policy, quiet\_hours\_policy, occupancy\_policy) VALUES

(2005, 'Allowed', 'Allowed', 'No enforced quiet hours', 'Maximum 8 guests');

INSERT INTO PolicyType (policytype\_ID, pet\_policy, smoking\_policy, quiet\_hours\_policy, occupancy\_policy) VALUES

(2006, 'Allowed with Restrictions', 'Not Allowed', '8PM-6AM Quiet Hours', 'Maximum 3 guests');

INSERT INTO PolicyType (policytype\_ID, pet\_policy, smoking\_policy, quiet\_hours\_policy, occupancy\_policy) VALUES

(2007, 'Not Allowed', 'Not Allowed', 'Strict 9PM-6AM Quiet Hours', 'Maximum 10 guests');

INSERT INTO PolicyType (policytype\_ID, pet\_policy, smoking\_policy, quiet\_hours\_policy, occupancy\_policy) VALUES

(2008, 'Allowed', 'Allowed', 'Flexible quiet hours', 'Maximum 7 guests');

INSERT INTO PolicyType (policytype\_ID, pet\_policy, smoking\_policy, quiet\_hours\_policy, occupancy\_policy) VALUES

(2009, 'Allowed', 'Not Allowed', 'No noise after 10PM', 'Maximum 2 guests');

INSERT INTO PolicyType (policytype\_ID, pet\_policy, smoking\_policy, quiet\_hours\_policy, occupancy\_policy) VALUES

(2010, 'Not Allowed', 'Allowed', 'Midnight-6AM Quiet Hours', 'Maximum 5 guests');

-- Hosts for existing Users

INSERT INTO Hosts (host\_ID, rating, join\_date) VALUES

(1002, 2, TO\_DATE('2021-06-15', 'YYYY-MM-DD'));

INSERT INTO Hosts (host\_ID, rating, join\_date) VALUES

(1004, 5, TO\_DATE('2020-04-10', 'YYYY-MM-DD'));

INSERT INTO Hosts (host\_ID, rating, join\_date) VALUES

(1006, 3, TO\_DATE('2022-01-20', 'YYYY-MM-DD'));

INSERT INTO Hosts (host\_ID, rating, join\_date) VALUES

(1008, 4, TO\_DATE('2019-11-05', 'YYYY-MM-DD'));

INSERT INTO Hosts (host\_ID, rating, join\_date) VALUES

(1010, 5, TO\_DATE('2021-09-12', 'YYYY-MM-DD'));

INSERT INTO Hosts (host\_ID, rating, join\_date) VALUES

(1011, 2, TO\_DATE('2022-03-18', 'YYYY-MM-DD'));

INSERT INTO Hosts (host\_ID, rating, join\_date) VALUES

(1012, 4, TO\_DATE('2020-12-30', 'YYYY-MM-DD'));

INSERT INTO Hosts (host\_ID, rating, join\_date) VALUES

(1013, 5, TO\_DATE('2021-08-22', 'YYYY-MM-DD'));

INSERT INTO Hosts (host\_ID, rating, join\_date) VALUES

(1014, 3, TO\_DATE('2019-05-01', 'YYYY-MM-DD'));

INSERT INTO Hosts (host\_ID, rating, join\_date) VALUES

(1015, 3, TO\_DATE('2023-01-10', 'YYYY-MM-DD'));

-- Guests for existing Users

INSERT INTO Guests (guest\_ID, membership\_level, join\_date) VALUES

(1001, 'Gold', TO\_DATE('2021-05-01', 'YYYY-MM-DD'));

INSERT INTO Guests (guest\_ID, membership\_level, join\_date) VALUES

(1003, 'Silver', TO\_DATE('2020-08-15', 'YYYY-MM-DD'));

INSERT INTO Guests (guest\_ID, membership\_level, join\_date) VALUES

(1005, 'Gold', TO\_DATE('2022-03-20', 'YYYY-MM-DD'));

INSERT INTO Guests (guest\_ID, membership\_level, join\_date) VALUES

(1007, 'Silver', TO\_DATE('2019-12-05', 'YYYY-MM-DD'));

INSERT INTO Guests (guest\_ID, membership\_level, join\_date) VALUES

(1009, 'Silver', TO\_DATE('2021-07-30', 'YYYY-MM-DD'));

INSERT INTO Guests (guest\_ID, membership\_level, join\_date) VALUES

(1016, 'Gold', TO\_DATE('2022-06-10', 'YYYY-MM-DD'));

INSERT INTO Guests (guest\_ID, membership\_level, join\_date) VALUES

(1017, 'Silver', TO\_DATE('2023-01-05', 'YYYY-MM-DD'));

INSERT INTO Guests (guest\_ID, membership\_level, join\_date) VALUES

(1018, 'Silver', TO\_DATE('2021-10-22', 'YYYY-MM-DD'));

INSERT INTO Guests (guest\_ID, membership\_level, join\_date) VALUES

(1019, 'Gold', TO\_DATE('2020-04-17', 'YYYY-MM-DD'));

INSERT INTO Guests (guest\_ID, membership\_level, join\_date) VALUES

(1020, 'Gold', TO\_DATE('2019-03-09', 'YYYY-MM-DD'));

-- 40 Insert statements for Bookings table

-- Bookings for Property 3001 (Miami, Apartment)

INSERT INTO Bookings VALUES (4001, TO\_DATE('2024-06-10','YYYY-MM-DD'), TO\_DATE('2024-06-15','YYYY-MM-DD'), 750.00, 'Confirmed', 1001, 3001);

INSERT INTO Bookings VALUES (4002, TO\_DATE('2024-07-01','YYYY-MM-DD'), TO\_DATE('2024-07-05','YYYY-MM-DD'), 600.00, 'Completed', 1003, 3001);

INSERT INTO Bookings VALUES (4003, TO\_DATE('2024-08-12','YYYY-MM-DD'), TO\_DATE('2024-08-16','YYYY-MM-DD'), 600.00, 'Pending', 1005, 3001);

INSERT INTO Bookings VALUES (4004, TO\_DATE('2024-05-10','YYYY-MM-DD'), TO\_DATE('2024-05-13','YYYY-MM-DD'), 450.00, 'Cancelled', 1007, 3001);

-- Bookings for Property 3002 (New York, Apartment)

INSERT INTO Bookings VALUES (4005, TO\_DATE('2024-09-15','YYYY-MM-DD'), TO\_DATE('2024-09-20','YYYY-MM-DD'), 1250.00, 'Confirmed', 1009, 3002);

INSERT INTO Bookings VALUES (4006, TO\_DATE('2024-12-01','YYYY-MM-DD'), TO\_DATE('2024-12-06','YYYY-MM-DD'), 1250.00, 'Completed', 1016, 3002);

INSERT INTO Bookings VALUES (4007, TO\_DATE('2024-04-10','YYYY-MM-DD'), TO\_DATE('2024-04-13','YYYY-MM-DD'), 750.00, 'Confirmed', 1017, 3002);

-- Bookings for Property 3003 (Chicago, House)

INSERT INTO Bookings VALUES (4008, TO\_DATE('2024-10-05','YYYY-MM-DD'), TO\_DATE('2024-10-10','YYYY-MM-DD'), 900.00, 'Confirmed', 1018, 3003);

INSERT INTO Bookings VALUES (4009, TO\_DATE('2024-11-15','YYYY-MM-DD'), TO\_DATE('2024-11-19','YYYY-MM-DD'), 720.00, 'Cancelled', 1019, 3003);

-- Bookings for Property 3004 (San Francisco, Room)

INSERT INTO Bookings VALUES (4010, TO\_DATE('2024-07-10','YYYY-MM-DD'), TO\_DATE('2024-07-12','YYYY-MM-DD'), 180.00, 'Confirmed', 1020, 3004);

INSERT INTO Bookings VALUES (4011, TO\_DATE('2024-05-25','YYYY-MM-DD'), TO\_DATE('2024-05-30','YYYY-MM-DD'), 450.00, 'Completed', 1001, 3004);

INSERT INTO Bookings VALUES (4012, TO\_DATE('2024-09-03','YYYY-MM-DD'), TO\_DATE('2024-09-07','YYYY-MM-DD'), 360.00, 'Pending', 1003, 3004);

-- Bookings for Property 3005 (Austin, House)

INSERT INTO Bookings VALUES (4013, TO\_DATE('2024-06-01','YYYY-MM-DD'), TO\_DATE('2024-06-06','YYYY-MM-DD'), 850.00, 'Confirmed', 1005, 3005);

INSERT INTO Bookings VALUES (4014, TO\_DATE('2024-10-20','YYYY-MM-DD'), TO\_DATE('2024-10-24','YYYY-MM-DD'), 680.00, 'Completed', 1007, 3005);

-- Bookings for Property 3006 (Orlando, Apartment)

INSERT INTO Bookings VALUES (4015, TO\_DATE('2024-05-15','YYYY-MM-DD'), TO\_DATE('2024-05-20','YYYY-MM-DD'), 1100.00, 'Confirmed', 1009, 3006);

INSERT INTO Bookings VALUES (4016, TO\_DATE('2024-07-22','YYYY-MM-DD'), TO\_DATE('2024-07-28','YYYY-MM-DD'), 1320.00, 'Completed', 1016, 3006);

INSERT INTO Bookings VALUES (4017, TO\_DATE('2024-09-12','YYYY-MM-DD'), TO\_DATE('2024-09-15','YYYY-MM-DD'), 660.00, 'Pending', 1017, 3006);

INSERT INTO Bookings VALUES (4018, TO\_DATE('2024-12-20','YYYY-MM-DD'), TO\_DATE('2024-12-25','YYYY-MM-DD'), 1100.00, 'Completed', 1018, 3006);

-- Bookings for Property 3007 (Denver, House)

INSERT INTO Bookings VALUES (4019, TO\_DATE('2024-04-12','YYYY-MM-DD'), TO\_DATE('2024-04-16','YYYY-MM-DD'), 800.00, 'Confirmed', 1019, 3007);

INSERT INTO Bookings VALUES (4020, TO\_DATE('2024-08-15','YYYY-MM-DD'), TO\_DATE('2024-08-20','YYYY-MM-DD'), 1000.00, 'Completed', 1020, 3007);

-- Bookings for Property 3008 (Seattle, Room)

INSERT INTO Bookings VALUES (4021, TO\_DATE('2024-06-05','YYYY-MM-DD'), TO\_DATE('2024-06-09','YYYY-MM-DD'), 380.00, 'Confirmed', 1001, 3008);

INSERT INTO Bookings VALUES (4022, TO\_DATE('2024-10-01','YYYY-MM-DD'), TO\_DATE('2024-10-04','YYYY-MM-DD'), 285.00, 'Pending', 1003, 3008);

-- Bookings for Property 3009 (Boston, Apartment)

INSERT INTO Bookings VALUES (4023, TO\_DATE('2024-07-05','YYYY-MM-DD'), TO\_DATE('2024-07-10','YYYY-MM-DD'), 900.00, 'Confirmed', 1005, 3009);

INSERT INTO Bookings VALUES (4024, TO\_DATE('2024-08-01','YYYY-MM-DD'), TO\_DATE('2024-08-05','YYYY-MM-DD'), 720.00, 'Completed', 1007, 3009);

INSERT INTO Bookings VALUES (4025, TO\_DATE('2024-11-10','YYYY-MM-DD'), TO\_DATE('2024-11-15','YYYY-MM-DD'), 900.00, 'Cancelled', 1009, 3009);

-- Bookings for Property 3010 (San Diego, House)

INSERT INTO Bookings VALUES (4026, TO\_DATE('2024-06-15','YYYY-MM-DD'), TO\_DATE('2024-06-20','YYYY-MM-DD'), 1050.00, 'Confirmed', 1016, 3010);

INSERT INTO Bookings VALUES (4027, TO\_DATE('2024-09-25','YYYY-MM-DD'), TO\_DATE('2024-09-30','YYYY-MM-DD'), 1050.00, 'Completed', 1017, 3010);

-- Some extra random bookings to reach 40

INSERT INTO Bookings VALUES (4028, TO\_DATE('2024-05-20','YYYY-MM-DD'), TO\_DATE('2024-05-25','YYYY-MM-DD'), 750.00, 'Pending', 1018, 3001);

INSERT INTO Bookings VALUES (4029, TO\_DATE('2024-07-25','YYYY-MM-DD'), TO\_DATE('2024-07-30','YYYY-MM-DD'), 1250.00, 'Completed', 1019, 3002);

INSERT INTO Bookings VALUES (4030, TO\_DATE('2024-12-05','YYYY-MM-DD'), TO\_DATE('2024-12-10','YYYY-MM-DD'), 900.00, 'Confirmed', 1020, 3003);

INSERT INTO Bookings VALUES (4031, TO\_DATE('2024-04-01','YYYY-MM-DD'), TO\_DATE('2024-04-05','YYYY-MM-DD'), 360.00, 'Cancelled', 1001, 3004);

INSERT INTO Bookings VALUES (4032, TO\_DATE('2024-10-10','YYYY-MM-DD'), TO\_DATE('2024-10-14','YYYY-MM-DD'), 850.00, 'Completed', 1003, 3005);

INSERT INTO Bookings VALUES (4033, TO\_DATE('2024-08-12','YYYY-MM-DD'), TO\_DATE('2024-08-16','YYYY-MM-DD'), 1100.00, 'Confirmed', 1005, 3006);

INSERT INTO Bookings VALUES (4034, TO\_DATE('2024-05-01','YYYY-MM-DD'), TO\_DATE('2024-05-03','YYYY-MM-DD'), 400.00, 'Pending', 1007, 3007);

INSERT INTO Bookings VALUES (4035, TO\_DATE('2024-11-22','YYYY-MM-DD'), TO\_DATE('2024-11-26','YYYY-MM-DD'), 380.00, 'Completed', 1009, 3008);

INSERT INTO Bookings VALUES (4036, TO\_DATE('2024-06-20','YYYY-MM-DD'), TO\_DATE('2024-06-24','YYYY-MM-DD'), 720.00, 'Confirmed', 1016, 3009);

INSERT INTO Bookings VALUES (4037, TO\_DATE('2024-07-15','YYYY-MM-DD'), TO\_DATE('2024-07-20','YYYY-MM-DD'), 1050.00, 'Completed', 1017, 3010);

INSERT INTO Bookings VALUES (4038, TO\_DATE('2024-12-10','YYYY-MM-DD'), TO\_DATE('2024-12-15','YYYY-MM-DD'), 750.00, 'Pending', 1018, 3001);

INSERT INTO Bookings VALUES (4039, TO\_DATE('2024-09-05','YYYY-MM-DD'), TO\_DATE('2024-09-10','YYYY-MM-DD'), 1250.00, 'Confirmed', 1019, 3002);

INSERT INTO Bookings VALUES (4040, TO\_DATE('2024-08-20','YYYY-MM-DD'), TO\_DATE('2024-08-25','YYYY-MM-DD'), 900.00, 'Cancelled', 1020, 3003);

-- Insert Reviews for all 40 Bookings

INSERT INTO Reviews VALUES (5001, 5, 'Amazing stay, very clean and welcoming!', TO\_DATE('2024-06-16','YYYY-MM-DD'), 1001, 3001);

INSERT INTO Reviews VALUES (5002, 4, 'Good apartment, nice location.', TO\_DATE('2024-07-06','YYYY-MM-DD'), 1003, 3001);

INSERT INTO Reviews VALUES (5003, 3, 'Stay was okay, a bit noisy.', TO\_DATE('2024-08-17','YYYY-MM-DD'), 1005, 3001);

INSERT INTO Reviews VALUES (5004, 2, 'Cancelled booking - poor communication.', TO\_DATE('2024-05-14','YYYY-MM-DD'), 1007, 3001);

INSERT INTO Reviews VALUES (5005, 5, 'Fantastic city apartment!', TO\_DATE('2024-09-21','YYYY-MM-DD'), 1009, 3002);

INSERT INTO Reviews VALUES (5006, 5, 'Loved it - would book again!', TO\_DATE('2024-12-07','YYYY-MM-DD'), 1016, 3002);

INSERT INTO Reviews VALUES (5007, 4, 'Nice, but a bit small.', TO\_DATE('2024-04-14','YYYY-MM-DD'), 1017, 3002);

INSERT INTO Reviews VALUES (5008, 5, 'Beautiful house, great amenities.', TO\_DATE('2024-10-11','YYYY-MM-DD'), 1018, 3003);

INSERT INTO Reviews VALUES (5009, 1, 'Cancelled last minute by host.', TO\_DATE('2024-11-20','YYYY-MM-DD'), 1019, 3003);

INSERT INTO Reviews VALUES (5010, 4, 'Room was clean and cozy.', TO\_DATE('2024-07-13','YYYY-MM-DD'), 1020, 3004);

INSERT INTO Reviews VALUES (5011, 5, 'Loved the city view.', TO\_DATE('2024-05-31','YYYY-MM-DD'), 1001, 3004);

INSERT INTO Reviews VALUES (5012, 4, 'Good service, decent room.', TO\_DATE('2024-09-08','YYYY-MM-DD'), 1003, 3004);

INSERT INTO Reviews VALUES (5013, 5, 'Wonderful home, spacious!', TO\_DATE('2024-06-07','YYYY-MM-DD'), 1005, 3005);

INSERT INTO Reviews VALUES (5014, 5, 'Very relaxing stay.', TO\_DATE('2024-10-25','YYYY-MM-DD'), 1007, 3005);

INSERT INTO Reviews VALUES (5015, 4, 'Perfect family trip spot.', TO\_DATE('2024-05-21','YYYY-MM-DD'), 1009, 3006);

INSERT INTO Reviews VALUES (5016, 5, 'Great value for money.', TO\_DATE('2024-07-29','YYYY-MM-DD'), 1016, 3006);

INSERT INTO Reviews VALUES (5017, 3, 'Decent, but could be cleaner.', TO\_DATE('2024-09-16','YYYY-MM-DD'), 1017, 3006);

INSERT INTO Reviews VALUES (5018, 5, 'Best holiday ever!', TO\_DATE('2024-12-26','YYYY-MM-DD'), 1018, 3006);

INSERT INTO Reviews VALUES (5019, 5, 'Loved the mountains!', TO\_DATE('2024-04-17','YYYY-MM-DD'), 1019, 3007);

INSERT INTO Reviews VALUES (5020, 4, 'Nice and quiet.', TO\_DATE('2024-08-21','YYYY-MM-DD'), 1020, 3007);

INSERT INTO Reviews VALUES (5021, 4, 'Room with a view!', TO\_DATE('2024-06-10','YYYY-MM-DD'), 1001, 3008);

INSERT INTO Reviews VALUES (5022, 3, 'Ok stay, bed a bit small.', TO\_DATE('2024-10-05','YYYY-MM-DD'), 1003, 3008);

INSERT INTO Reviews VALUES (5023, 5, 'Amazing apartment downtown.', TO\_DATE('2024-07-11','YYYY-MM-DD'), 1005, 3009);

INSERT INTO Reviews VALUES (5024, 5, 'Excellent for a business trip.', TO\_DATE('2024-08-06','YYYY-MM-DD'), 1007, 3009);

INSERT INTO Reviews VALUES (5025, 2, 'Poor communication.', TO\_DATE('2024-11-16','YYYY-MM-DD'), 1009, 3009);

INSERT INTO Reviews VALUES (5026, 5, 'Best host experience!', TO\_DATE('2024-06-21','YYYY-MM-DD'), 1016, 3010);

INSERT INTO Reviews VALUES (5027, 5, 'Sunset views are stunning.', TO\_DATE('2024-10-01','YYYY-MM-DD'), 1017, 3010);

INSERT INTO Reviews VALUES (5028, 3, 'Apartment was clean.', TO\_DATE('2024-05-26','YYYY-MM-DD'), 1018, 3001);

INSERT INTO Reviews VALUES (5029, 5, 'Loved New York again!', TO\_DATE('2024-07-31','YYYY-MM-DD'), 1019, 3002);

INSERT INTO Reviews VALUES (5030, 5, 'Home was beautiful!', TO\_DATE('2024-12-11','YYYY-MM-DD'), 1020, 3003);

INSERT INTO Reviews VALUES (5031, 1, 'Booking was cancelled, very upset.', TO\_DATE('2024-04-06','YYYY-MM-DD'), 1001, 3004);

INSERT INTO Reviews VALUES (5032, 4, 'Great kitchen and patio.', TO\_DATE('2024-10-15','YYYY-MM-DD'), 1003, 3005);

INSERT INTO Reviews VALUES (5033, 5, 'Super relaxing.', TO\_DATE('2024-08-17','YYYY-MM-DD'), 1005, 3006);

INSERT INTO Reviews VALUES (5034, 3, 'Short but nice.', TO\_DATE('2024-05-04','YYYY-MM-DD'), 1007, 3007);

INSERT INTO Reviews VALUES (5035, 5, 'Fantastic winter vacation.', TO\_DATE('2024-11-27','YYYY-MM-DD'), 1009, 3008);

INSERT INTO Reviews VALUES (5036, 5, 'Easy access to city.', TO\_DATE('2024-06-25','YYYY-MM-DD'), 1016, 3009);

INSERT INTO Reviews VALUES (5037, 5, 'Wonderful beach house.', TO\_DATE('2024-07-21','YYYY-MM-DD'), 1017, 3010);

INSERT INTO Reviews VALUES (5038, 4, 'Good holiday, will return.', TO\_DATE('2024-12-16','YYYY-MM-DD'), 1018, 3001);

INSERT INTO Reviews VALUES (5039, 5, 'Perfect fall getaway!', TO\_DATE('2024-09-11','YYYY-MM-DD'), 1019, 3002);

INSERT INTO Reviews VALUES (5040, 2, 'Cancelled unexpectedly.', TO\_DATE('2024-08-26','YYYY-MM-DD'), 1020, 3003);

-- Insert into Properties

INSERT INTO Properties (property\_ID, description, address, price\_per\_night, max\_guests, cancellation\_policy, region, property\_type, host\_ID)

VALUES (3001, 'Modern oceanfront apartment with stunning views.', '123 Ocean Dr, Unit 12B, Miami, FL', 150.00, 4, 'Flexible Cancellation', 'Miami', 'Apartment', 1002);

INSERT INTO Properties (property\_ID, description, address, price\_per\_night, max\_guests, cancellation\_policy, region, property\_type, host\_ID)

VALUES (3002, 'Luxury high-rise apartment near Central Park.', '456 5th Ave, Penthouse 5A, New York, NY', 250.00, 2, 'Strict Cancellation', 'New York', 'Apartment', 1004);

INSERT INTO Properties (property\_ID, description, address, price\_per\_night, max\_guests, cancellation\_policy, region, property\_type, host\_ID)

VALUES (3003, 'Spacious house perfect for family vacations.', '789 Lakeview St, Chicago, IL', 180.00, 6, 'Moderate Cancellation', 'Chicago', 'House', 1006);

INSERT INTO Properties (property\_ID, description, address, price\_per\_night, max\_guests, cancellation\_policy, region, property\_type, host\_ID)

VALUES (3004, 'Cozy private room in the heart of the city.', '321 Downtown Blvd, San Francisco, CA', 90.00, 1, 'Flexible Cancellation', 'San Francisco', 'Room', 1008);

INSERT INTO Properties (property\_ID, description, address, price\_per\_night, max\_guests, cancellation\_policy, region, property\_type, host\_ID)

VALUES (3005, 'Modern 3-bedroom house with a garden.', '654 Greenway Rd, Austin, TX', 170.00, 5, 'Strict Cancellation', 'Austin', 'House', 1010);

INSERT INTO Properties (property\_ID, description, address, price\_per\_night, max\_guests, cancellation\_policy, region, property\_type, host\_ID)

VALUES (3006, 'Luxury apartment close to theme parks.', '987 Sunset Blvd, Suite 7C, Orlando, FL', 220.00, 4, 'Moderate Cancellation', 'Orlando', 'Apartment', 1011);

INSERT INTO Properties (property\_ID, description, address, price\_per\_night, max\_guests, cancellation\_policy, region, property\_type, host\_ID)

VALUES (3007, 'Cozy mountain house with great views.', '159 Rocky Rd, Denver, CO', 200.00, 6, 'Flexible Cancellation', 'Denver', 'House', 1012);

INSERT INTO Properties (property\_ID, description, address, price\_per\_night, max\_guests, cancellation\_policy, region, property\_type, host\_ID)

VALUES (3008, 'Private room in peaceful suburban area.', '753 Elm St, Seattle, WA', 95.00, 2, 'Strict Cancellation', 'Seattle', 'Room', 1013);

INSERT INTO Properties (property\_ID, description, address, price\_per\_night, max\_guests, cancellation\_policy, region, property\_type, host\_ID)

VALUES (3009, 'Stylish apartment near downtown attractions.', '852 Beacon St, Flat 3E, Boston, MA', 180.00, 3, 'Moderate Cancellation', 'Boston', 'Apartment', 1014);

INSERT INTO Properties (property\_ID, description, address, price\_per\_night, max\_guests, cancellation\_policy, region, property\_type, host\_ID)

VALUES (3010, 'Family-friendly house with a backyard.', '951 Pacific Ave, San Diego, CA', 210.00, 7, 'Flexible Cancellation', 'San Diego', 'House', 1015);

-- Corrected Payment INSERTs based on updated Booking totals

INSERT INTO Payment VALUES (6001, 750.00, TO\_DATE('2024-06-08','YYYY-MM-DD'), 'USD', 4001);

INSERT INTO Payment VALUES (6002, 600.00, TO\_DATE('2024-06-29','YYYY-MM-DD'), 'USD', 4002);

INSERT INTO Payment VALUES (6003, 600.00, TO\_DATE('2024-08-10','YYYY-MM-DD'), 'USD', 4003);

INSERT INTO Payment VALUES (6004, 90.00, TO\_DATE('2024-05-08','YYYY-MM-DD'), 'USD', 4004);

INSERT INTO Payment VALUES (6005, 1250.00, TO\_DATE('2024-09-13','YYYY-MM-DD'), 'USD', 4005);

INSERT INTO Payment VALUES (6006, 1250.00, TO\_DATE('2024-11-29','YYYY-MM-DD'), 'USD', 4006);

INSERT INTO Payment VALUES (6007, 750.00, TO\_DATE('2024-04-08','YYYY-MM-DD'), 'USD', 4007);

INSERT INTO Payment VALUES (6008, 900.00, TO\_DATE('2024-10-03','YYYY-MM-DD'), 'USD', 4008);

INSERT INTO Payment VALUES (6009, 144.00, TO\_DATE('2024-11-13','YYYY-MM-DD'), 'USD', 4009);

INSERT INTO Payment VALUES (6010, 180.00, TO\_DATE('2024-07-08','YYYY-MM-DD'), 'USD', 4010);

INSERT INTO Payment VALUES (6011, 450.00, TO\_DATE('2024-05-23','YYYY-MM-DD'), 'USD', 4011);

INSERT INTO Payment VALUES (6012, 360.00, TO\_DATE('2024-09-01','YYYY-MM-DD'), 'USD', 4012);

INSERT INTO Payment VALUES (6013, 850.00, TO\_DATE('2024-05-28','YYYY-MM-DD'), 'USD', 4013);

INSERT INTO Payment VALUES (6014, 680.00, TO\_DATE('2024-10-17','YYYY-MM-DD'), 'USD', 4014);

INSERT INTO Payment VALUES (6015, 1100.00, TO\_DATE('2024-05-13','YYYY-MM-DD'), 'USD', 4015);

INSERT INTO Payment VALUES (6016, 1320.00, TO\_DATE('2024-07-20','YYYY-MM-DD'), 'USD', 4016);

INSERT INTO Payment VALUES (6017, 660.00, TO\_DATE('2024-09-10','YYYY-MM-DD'), 'USD', 4017);

INSERT INTO Payment VALUES (6018, 1100.00, TO\_DATE('2024-12-18','YYYY-MM-DD'), 'USD', 4018);

INSERT INTO Payment VALUES (6019, 800.00, TO\_DATE('2024-04-10','YYYY-MM-DD'), 'USD', 4019);

INSERT INTO Payment VALUES (6020, 1000.00, TO\_DATE('2024-08-13','YYYY-MM-DD'), 'USD', 4020);

INSERT INTO Payment VALUES (6021, 380.00, TO\_DATE('2024-06-03','YYYY-MM-DD'), 'USD', 4021);

INSERT INTO Payment VALUES (6022, 285.00, TO\_DATE('2024-09-29','YYYY-MM-DD'), 'USD', 4022);

INSERT INTO Payment VALUES (6023, 900.00, TO\_DATE('2024-07-03','YYYY-MM-DD'), 'USD', 4023);

INSERT INTO Payment VALUES (6024, 720.00, TO\_DATE('2024-07-30','YYYY-MM-DD'), 'USD', 4024);

INSERT INTO Payment VALUES (6025, 180.00, TO\_DATE('2024-11-08','YYYY-MM-DD'), 'USD', 4025);

INSERT INTO Payment VALUES (6026, 1050.00, TO\_DATE('2024-06-13','YYYY-MM-DD'), 'USD', 4026);

INSERT INTO Payment VALUES (6027, 1050.00, TO\_DATE('2024-09-23','YYYY-MM-DD'), 'USD', 4027);

INSERT INTO Payment VALUES (6028, 750.00, TO\_DATE('2024-05-18','YYYY-MM-DD'), 'USD', 4028);

INSERT INTO Payment VALUES (6029, 1250.00, TO\_DATE('2024-07-23','YYYY-MM-DD'), 'USD', 4029);

INSERT INTO Payment VALUES (6030, 900.00, TO\_DATE('2024-12-03','YYYY-MM-DD'), 'USD', 4030);

INSERT INTO Payment VALUES (6031, 72.00, TO\_DATE('2024-03-30','YYYY-MM-DD'), 'USD', 4031);

INSERT INTO Payment VALUES (6032, 850.00, TO\_DATE('2024-10-08','YYYY-MM-DD'), 'USD', 4032);

INSERT INTO Payment VALUES (6033, 1100.00, TO\_DATE('2024-08-10','YYYY-MM-DD'), 'USD', 4033);

INSERT INTO Payment VALUES (6034, 400.00, TO\_DATE('2024-04-29','YYYY-MM-DD'), 'USD', 4034);

INSERT INTO Payment VALUES (6035, 380.00, TO\_DATE('2024-11-20','YYYY-MM-DD'), 'USD', 4035);

INSERT INTO Payment VALUES (6036, 720.00, TO\_DATE('2024-06-18','YYYY-MM-DD'), 'USD', 4036);

INSERT INTO Payment VALUES (6037, 1050.00, TO\_DATE('2024-07-13','YYYY-MM-DD'), 'USD', 4037);

INSERT INTO Payment VALUES (6038, 750.00, TO\_DATE('2024-12-08','YYYY-MM-DD'), 'USD', 4038);

INSERT INTO Payment VALUES (6039, 1250.00, TO\_DATE('2024-09-03','YYYY-MM-DD'), 'USD', 4039);

INSERT INTO Payment VALUES (6040, 180.00, TO\_DATE('2024-08-18','YYYY-MM-DD'), 'USD', 4040);

-- Insert Messages

INSERT INTO Messages VALUES (7001, 'Hello, is the apartment available for next weekend?', 1001);

INSERT INTO Messages VALUES (7002, 'Yes, the apartment is available. Feel free to book!', 1002);

INSERT INTO Messages VALUES (7003, 'Can I check in early?', 1003);

INSERT INTO Messages VALUES (7004, 'Early check-in is possible at 1PM.', 1004);

INSERT INTO Messages VALUES (7005, 'What is the cancellation policy?', 1005);

INSERT INTO Messages VALUES (7006, 'Flexible cancellation. Full refund 5 days before check-in.', 1006);

INSERT INTO Messages VALUES (7007, 'Do you allow pets?', 1007);

INSERT INTO Messages VALUES (7008, 'Yes, pets are allowed with a small cleaning fee.', 1008);

INSERT INTO Messages VALUES (7009, 'Is parking included?', 1009);

INSERT INTO Messages VALUES (7010, 'Yes, there is one reserved parking spot.', 1010);

INSERT INTO Messages VALUES (7011, 'Can I extend my stay by 2 nights?', 1016);

INSERT INTO Messages VALUES (7012, 'Extension approved. Please update your booking.', 1011);

INSERT INTO Messages VALUES (7013, 'Is the pool heated?', 1017);

INSERT INTO Messages VALUES (7014, 'Yes, the pool is heated year-round.', 1012);

INSERT INTO Messages VALUES (7015, 'Are there any additional fees?', 1018);

INSERT INTO Messages VALUES (7016, 'No additional fees beyond booking total.', 1013);

INSERT INTO Messages VALUES (7017, 'How far is it from the city center?', 1019);

INSERT INTO Messages VALUES (7018, 'About 10 minutes drive to downtown.', 1014);

INSERT INTO Messages VALUES (7019, 'Is Wi-Fi available?', 1020);

INSERT INTO Messages VALUES (7020, 'High-speed Wi-Fi included for free.', 1015);

-- Insert Apartments (Aproperty\_ID = matching Property\_ID)

INSERT INTO Apartment (Aproperty\_ID, unit\_number, number\_of\_bedrooms, number\_of\_bathrooms)

VALUES (3001, 'Unit 12B', 2, 2);

INSERT INTO Apartment (Aproperty\_ID, unit\_number, number\_of\_bedrooms, number\_of\_bathrooms)

VALUES (3002, 'Penthouse 5A', 1, 1);

INSERT INTO Apartment (Aproperty\_ID, unit\_number, number\_of\_bedrooms, number\_of\_bathrooms)

VALUES (3006, 'Suite 7C', 3, 2);

INSERT INTO Apartment (Aproperty\_ID, unit\_number, number\_of\_bedrooms, number\_of\_bathrooms)

VALUES (3009, 'Flat 3E', 2, 1);

-- Insert Houses (Hproperty\_ID = matching Property\_ID)

INSERT INTO House (Hproperty\_ID, house\_number, number\_of\_bedrooms, number\_of\_bathrooms)

VALUES (3003, '789', 4, 3);

INSERT INTO House (Hproperty\_ID, house\_number, number\_of\_bedrooms, number\_of\_bathrooms)

VALUES (3005, '654', 3, 2);

INSERT INTO House (Hproperty\_ID, house\_number, number\_of\_bedrooms, number\_of\_bathrooms)

VALUES (3007, '159', 5, 3);

INSERT INTO House (Hproperty\_ID, house\_number, number\_of\_bedrooms, number\_of\_bathrooms)

VALUES (3010, '951', 4, 2);

-- Insert Rooms (Rproperty\_ID = matching Property\_ID)

INSERT INTO Room (Rproperty\_ID, room\_type, number\_of\_beds, has\_private\_bathroom)

VALUES (3004, 'Private', 1, 'Y');

INSERT INTO Room (Rproperty\_ID, room\_type, number\_of\_beds, has\_private\_bathroom)

VALUES (3008, 'Private', 2, 'N');

**CONCLUSION**

The database management system provides an end-to-end solution to the vacation rental sector that targets both the strategic and operational needs for property owners and travelers. A conceptual design was created to preserve data integrity and reflect real-world interactions with entities such as Users, Bookings, Properties, Reviews, Payments, and Support Tickets.

Our logical model enhances this scheme next, based on subtype and supertype hierarchies to support the above variability of property types (apartments, homes, rooms) and user types (hosts, guests). By proper normalization, the database optimally fits for storage but also supports key operations such as management of bookings, payment transactions, reviews by customers, and support procedures.

The physical design prioritizes security, scalability, and performance through the utilization of indexed file organization to enable quick retrieval of data and robust backup and recovery policies to safeguard against loss of data. The access controls were also set in place so that sensitive activities like altering listings or viewing payment information are executed by authorized individuals only.

Finally, VRBO uses specific SQL queries such as pinpointing highly rated hosts or spotting popular booking trends to gather insights that help improve the platform. These insights make it easier to enhance the user experience, fine-tune how things run behind the scenes, and keep the company competitive in a constantly evolving short-term rental market.

In summary, the VRBO database management solution successfully addresses the challenges of handling large and dynamic sets of data while preserving the vision of the platform to offer a seamless, secure, and responsive vacation rental experience to customers worldwide.

All in all, the VRBO database management solution effectively addresses the challenges of handling large and dynamic data sets while staying true to the purpose of the platform, to offer a seamless, secure, and responsive vacation rental experience to customers worldwide