**Project Proposal**

**a) Brief Description of the problem and opportunity**

The hospitality business has experienced an increase in peer-to-peer property rentals in the present digital age. Individuals are looking for innovative, economical, and convenient lodging, while property owners are looking for effective methods to monetize their vacant premises. However, there is a gap in platforms that provide a unified experience for both hosts and guests, while also providing trust, flexibility, and user-centric features. Furthermore, there is a potential to implement

2 user data to improve the booking experience, provide personalized suggestions, and build a community of trustworthy hosts and delighted visitors.

**b) How the proposed Database will address the problem/opportunity**

The database will provide a consistent platform for hosts to list their properties and guests to make bookings. This guarantees that the process is as simple as possible for both sides.

**Property Recommendations:** By collecting visitor preferences and feedback, the platform may provide customized property recommendations, boosting the user experience.

**Transparency and trust:** The feedback and rating system will encourage trust. Guests can select hotels based on previous ratings, and hosts can enhance their services in response to comments.

**Loyalty Programs:** By utilizing the Loyalty Points feature for guests, the platform can implement loyalty programs that provide discounts or rewards to repeat users.

**Superhost Recognition:** Using the SuperhostStatusID, the platform will be able to recognize and reward top-performing hosts, encouraging them to keep up their high standards.

**Dynamic Pricing:** The PricePerNight attribute can be changed based on occupancy rate, allowing hosts to create revenue-maximizing dynamic pricing methods.

**Secure Transactions:** The Payment entity makes certain that all transactions are recorded, resulting in a secure and transparent payment process.

**Entities, Attributes, and Relationships**

**User (Superclass):**

* **Attributes:** UserID, Username, Email, DateofJoining, PhoneNumbers.
* **Relationships:** A User can be either a Guest or a Host, but not both.
* **Justification:** The User entity acts as a superclass to capture common attributes and behaviors of both Guests and Hosts. This design promotes data integrity and reduces redundancy.

**a) Guest (Subclass of User):**

* **Attributes:** Preferences, Loyalty Points.
* **Relationships:** A Guest can make multiple Reservations, make payments, and can give a Guest Review.
* **Justification:** By creating a Guest subclass, specific attributes and behaviors related to guests are captured.

**b) Host (Subclass of User):**

* **Attributes:** SuperhostStatus.
* **Relationships:** A Host can list multiple listings/properties and post a HostReview.
* **Justification:** The Host subclass captures attributes and behaviours specific to hosts. The SuperhostStatus can be used to offer special privileges or promotions to superhosts.

**Listing / Property:**

* **Attributes:** PropertyID, Address, RoomType.
* **Relationships:** A property is owned by one Host and each property will have Media.
* **Justification:** The property entity captures all details about properties available for reservation. The UserID as a foreign key ensures that each listing is associated with a host.

**Reservation:**

* **Attributes:** ReservationID, StartDate, EndDate, TotalPrice, Cancellation, DateofBooking.
* **Relationships:** A Reservation is an agreement between a Guest and a Listing.
* **Justification:** The Reservation entity captures the details of a booking. By linking it with UserID and PropertyID, it ensures that each reservation is associated with a guest and a property.

**HostReviews:**

* **Attributes:** HostReviewID, HostRating
* **Justification:** The Host Reviews entity captures feedback specifically about the guest's behaviour during their stay.

**GuestReviews:**

* **Attributes:** GuestReviewID, GuestRating
* **Justification:** The GuestReviews entity allows guests to provide ratings for listings, which can be useful for future guests and for hosts to improve their services.

**Payments:**

* **Attributes:** PaymentID, PaymentDate, Price
* **Justification:** The Payments entity captures transaction details. By linking it with ReservationID and UserID, it ensures that each payment is associated with a reservation.

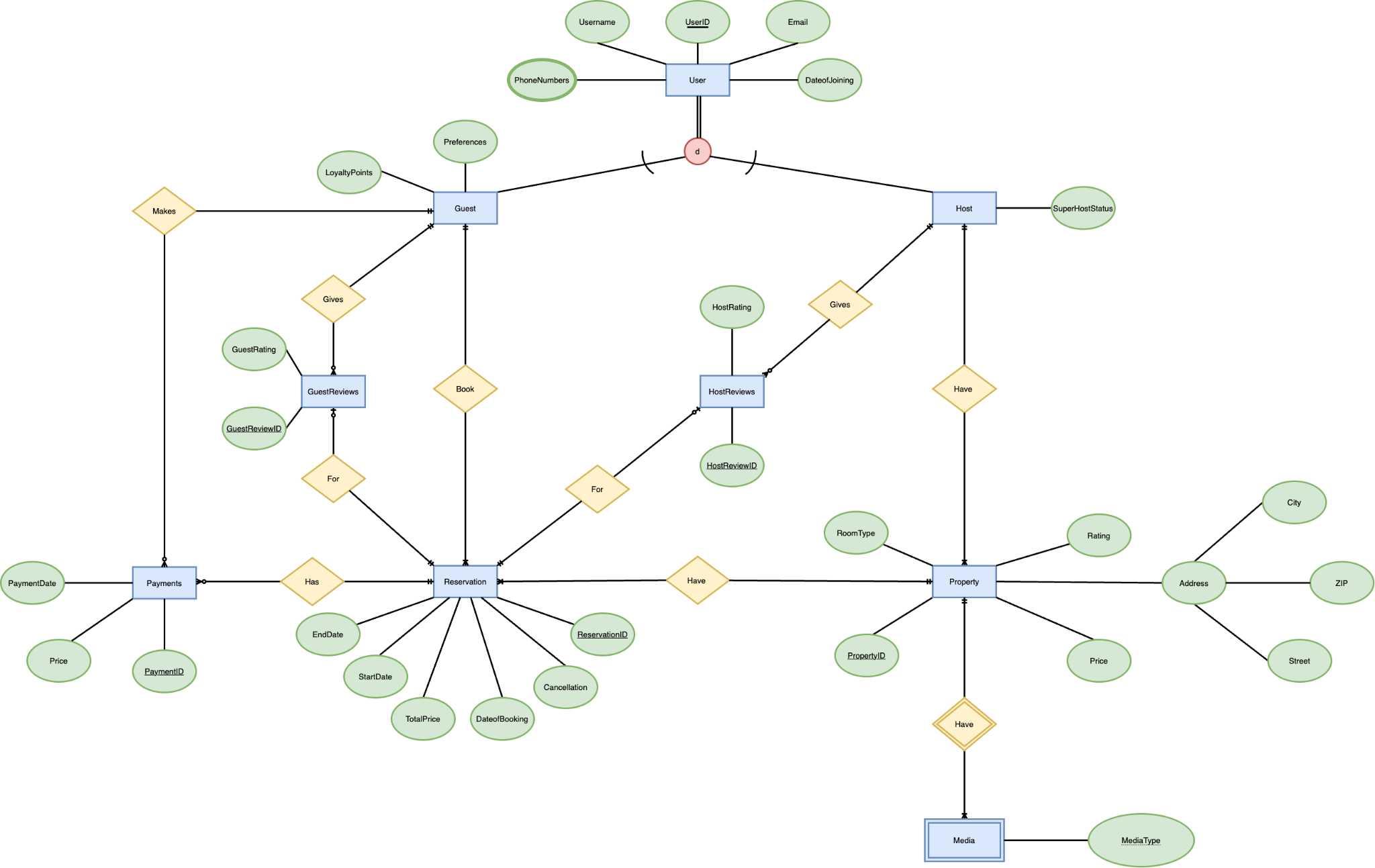
**Media:**

* **Attributes:** Mediatype
* **Justification:** The media entity is a weak entity because it cannot exist without property/listing. It contains photos/videos of the property.

**Project Justification:**

* **Data Integrity:** The design ensures that data is consistent and accurate. For example, a reservation cannot exist without a guest and a listing.
* **Flexibility:** The design allows for easy expansion. New attributes or entities can be added without major changes to the existing structure.
* **User Experience:** By capturing feedback and ratings, the platform can offer better recommendations to future guests.
* **Business Insights:** The design allows for easy analysis. For example, analyzing the feedback can provide insights into the most popular listings or areas of improvement.
* **Operational Efficiency:** With clear relationships and attributes, operations like booking, payment, and feedback can be streamlined.

In conclusion, this design provides a comprehensive structure for a property booking platform, ensuring data integrity, flexibility, and enhanced user experience. It lays a strong foundation for building a robust and scalable system.

****

**Relational Model:**

A.) USER (UserID, Username, Email, Status)

PHONENUMBERS (PhoneNumber, UserID [FK])

B.) GUEST (LoyaltyPoints, Guest\_UserID [FK])

C.) HOST (SuperHostStatus, Host\_UserID [FK])

D.) PROPERTY (PropertyID, RoomType, Rating, Price, Street, City, ZIP, Hosts\_UserID [FK])

E.) MEDIA (MediaURL, PropertyID [FK])

F.) RESERVATION (ReservationID, DateofBooking, StartDate, EndDate, TotalPrice, Cancellation, Guest\_UserID [FK], PropertyID [FK])

G.) HOSTREVIEWS (HostReviewID, HostRating, Host\_UserID [FK], ReservationID [FK])

H.) GUESTREVIEWS (GuestReviewID, GuestRating, Guest\_UserID [FK], ReservationID [FK])

I.) PAYMENTS (PaymentID, PaymentDate, Price, Guest\_UserID [FK], ReservationID [FK])

**What are the average ratings for guests and hosts, and how do they compare to each other?**

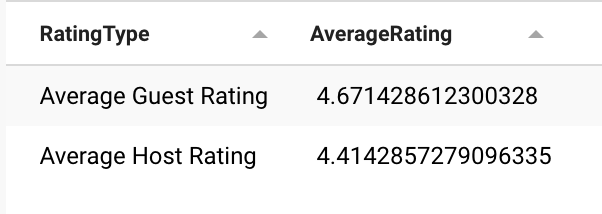
SELECT 'Average Guest Rating' AS RatingType, AVG(GuestRating) AS AverageRating

FROM GuestReviews

UNION

SELECT 'Average Host Rating' AS RatingType, AVG(HostRating) AS AverageRating

FROM HostReviews;

**’**

**Average Ratings for Guests and Hosts:** This query is useful for assessing the overall performance and satisfaction levels of both guests and hosts. Utilizing the GuestReviews and HostReviews entities aligns with your goal of ensuring transparency and trust**.**

**What are the properties with a rating greater than 4.0 that have been reserved by guests, and how are these properties categorized by room type and price per night?**

SELECT Guest\_UserID, PropertyID, RoomType, Rating, PricePerNight

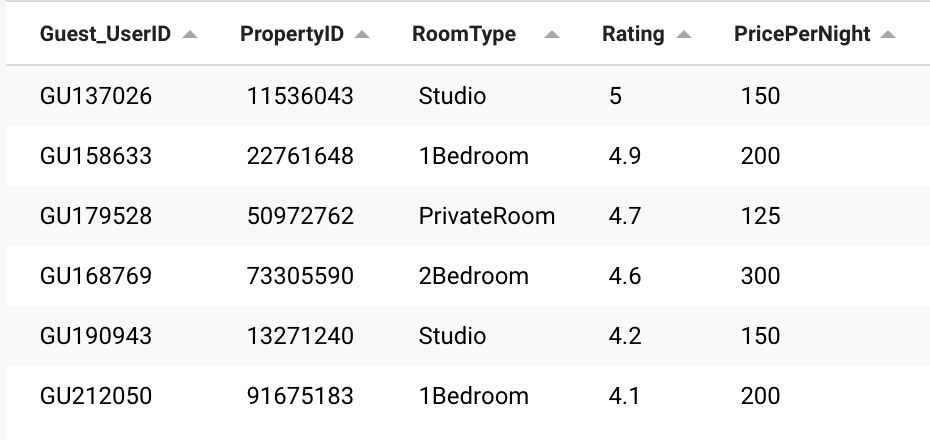
FROM Property

NATURAL LEFT JOIN Reservation

NATURAL LEFT JOIN Guests

WHERE Rating > 4.0

ORDER BY Rating DESC;

****

**Highly Rated Properties:** This query identifies properties with ratings greater than 4.0 and categorizes them according to room type and price per night. This is useful for guests seeking high-quality accommodations and aligns with your goal of making property recommendations based on guest preferences and feedback.

**What are the total amounts spent and the durations of stay for each guest who has booked a property in Metroville, along with the street names of those properties?**

SELECT Guest\_UserID, Street, TotalPrice AS AmountSpent,

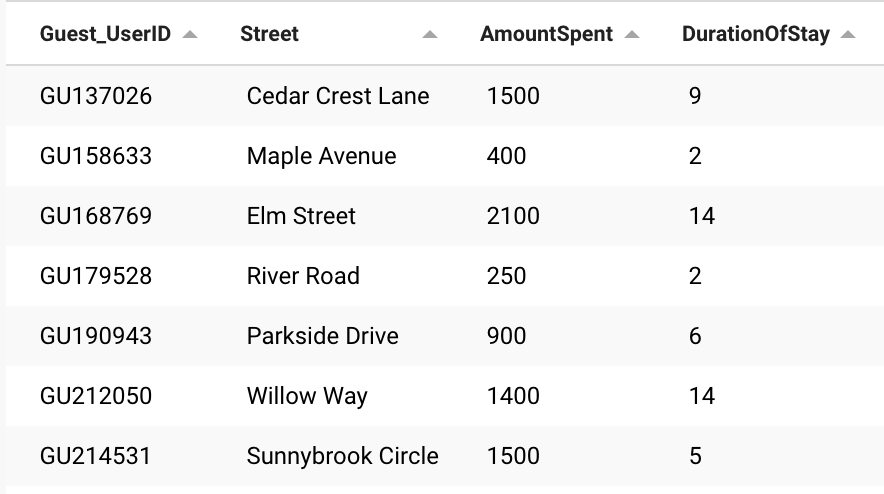
DATEDIFF(EndDate, StartDate) AS DurationOfStay

FROM Reservation

NATURAL LEFT JOIN Property

WHERE City = 'Metroville'

ORDER BY Guest\_UserID;



**Guest Spendings and Durations in Metroville:** This query is excellent for gaining insights into guest behavior in a specific city, Metroville. It provides information on the total amount spent, the duration of stay, and property location details. This aligns with your goal of gaining business insights and understanding user behavior.

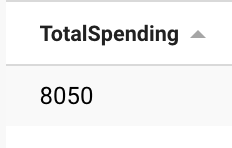
**What is the total amount of money spent by all guests on accommodations in Metroville?**

SELECT SUM(TotalPrice) AS TotalSpending

FROM Reservation

Natural Left JOIN Property

WHERE City = 'Metroville';



**Total Spendings in Metroville**: This query offers a consolidated view of the money spent by visitors in Metroville. It aligns with your objective of understanding market trends and the financial performance in specific areas.

**What are the total earnings of each host and the average loyalty points of guests who have stayed in their properties, and how do these earnings and guest loyalty points vary by host?**

SELECT Host\_UserID, SUM(TotalPrice) AS HostEarnings,

Guest\_UserID, AVG(LoyaltyPoints) AS AverageGuestLoyaltyPoints

FROM Hosts H

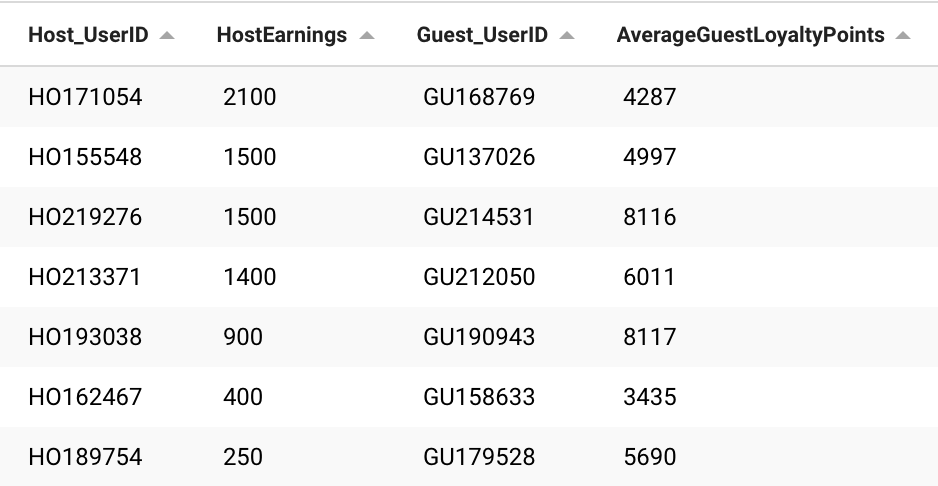
Natural left JOIN Property

Natural left JOIN Reservation

Natural left JOIN Guests

GROUP BY Host\_UserID

ORDER BY HostEarnings DESC;



**Host Earnings and Guest Loyalty Points**: This critical query helps in understanding the financial performance of hosts and the engagement of guests through loyalty points. It directly supports your goals of rewarding top-performing hosts and encouraging guest loyalty.