airbnb

Group 14

Vidya Balakrishnan, Umesh Indla, Shankar Mhamane, and Curtis Bell

IS6420 – Database Theory and Design

Table of Contents

Executive Summary

Airbnb has changed the way the world interacts with the hospitality industry forever. Traditional hotels now have a whole new competitive landscape to consider, and consumers have more exciting choices than ever before. All this is thanks to a few friends who started renting an air mattress in their living room 14 years ago. The internet has dramatically shaped the way many businesses operate, and the hospitality industry is no exception.

In the project described below we choose to limit our scope to Airbnb's property selection, amenities selection, property reservation, and payment processes. This scope has produced nine tables within our database and accurately represents the processes previously described.

To meet the requirements of our scope we developed the following nine tables:

Payment, User_Profile, Card_Details, Reviews, Books, Property, Amenities, Host_Info, and Owner

These tables were populated with data and implemented into a working transactional database. We then successfully ran various sample queries to prove that our database was operational and meet the user requirements of the project.

In summary, this project has been a success. Airbnb is still a growing company, and this project has provided us with new insight into how they operate. As a team we all felt that our learning objectives were met during this project, and we look forward to putting this knowledge to use in our future careers.

General Description of Airbnb

Introduction and History

Airbnb is company in the hospitality industry who specializes in connecting guests with private property owners. They were founded in 2008 by two roommates who thought they could rent an air mattress in their living room to make some extra money. This thought turned out to be correct and soon after a company was born, Airbed & Breakfast. Living in the crowded San Francisco Bay Area they saw an opportunity to create a platform that would allow other property hosts to utilize their spare space for guests and extra income. With hotel space very limited during the Industrial Design Conference in 2008 they were able to successfully book several stays on their platform.

After this proof of concept, they saw an opportunity to scale, but were in desperate need of funds. The founders decided to sell limited edition cereal themed for the 2008 presidential election and were able to raise an astounding $30,000 in short order. Soon after they were able to secure $20,000 from the startup incubator, Y Combinator. At this point they had established a profitable business model and a working website which attracted a cascade of other investors through the next couple of years.

The company continued to grow and develop its platforms, and eventually released an IPO worth $3.5 billion in December of 2020. Today Airbnb has an estimated six million listings in more than 220 countries around the globe and roughly 2.9 million hosts. They continue to be a leader in this new hospitality market segment which they created.

Vision and Objectives

Airbnb’s mission is to create a world where anyone can belong anywhere, and they are focused on creating an end-to-end travel platform that will handle every part of your trip. This is accomplished by making bookings around the world easily accessible to all users. With much of the globe covered by their services and a wide range of pricing available in each area they have accomplished this mission.

Products and Services

Airbnb offers a digital platform that allows those with extra space to connect with those looking for somewhere to stay. Their service is run through a website and mobile application primarily. Property owners are referred to as hosts, while those booking stays are referred to as guests. As the company has grown, they have added some premium services such as the ability to be a super host, and Airbnb Plus which is available to hosts with ratings (4.8+) willing to fully equipped homes so guests need nothing but clothing for their stay.

How does Airbnb use Transactional Databases?

Airbnb uses transactional databases manage the way information flows through their website and application. The primary functions of their databases would be to allow users to search, book, pay, and review properties, and to allow hosts to post and manage attributes of their properties.

Airbnb also collects data on the properties which have the most bookings/ratings and then makes a list of all the amenities provided by these groups of properties. By doing this they tend to increase customer satisfaction which is directly proportional to more bookings.

Prioritized Requirements

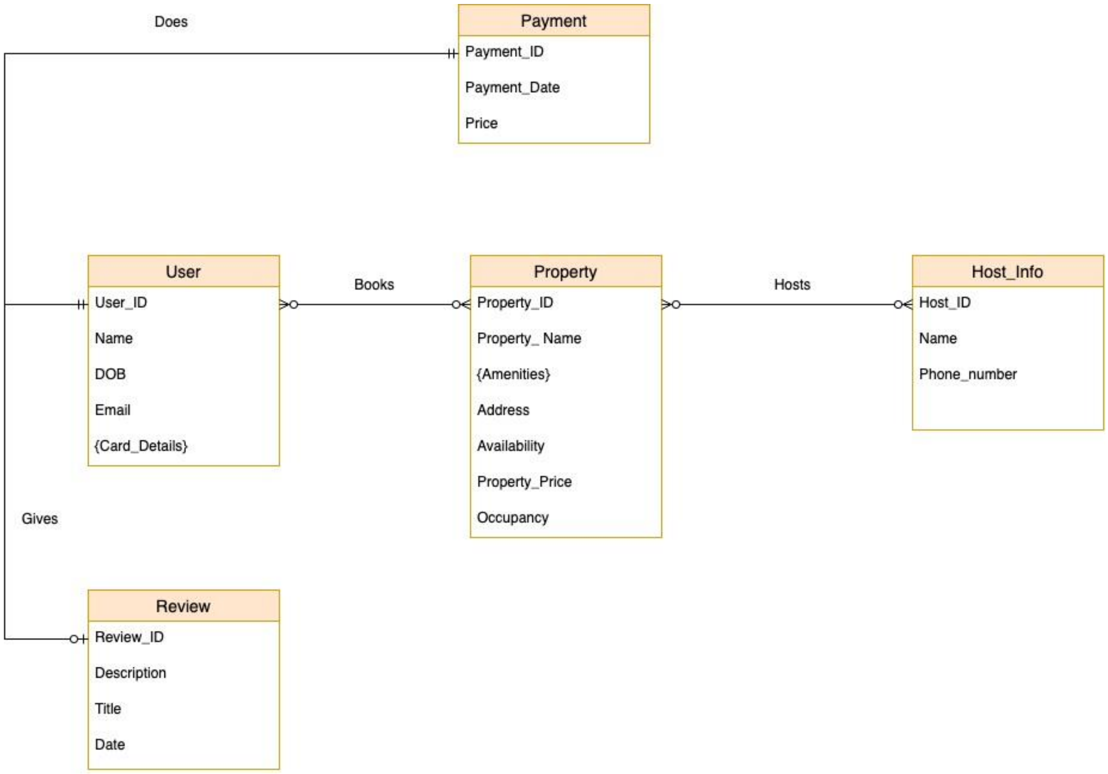
For the scope of our project, we limited our prioritized requirements. We determined below requirements for go live which are the following:

- Users can host the properties
- Users can reserve the properties
- Users can Search properties based on various attributes
- Users can be able to leave reviews and rate on the properties

To meet these requirements, we developed a transactional database model with nine tables. The tables included are listed below.

- Payment
- User_Profile
- Card_Details
- Reviews
- Reserves
- Property
- Amenities
- Host_Info
- Owner

Conceptual Model



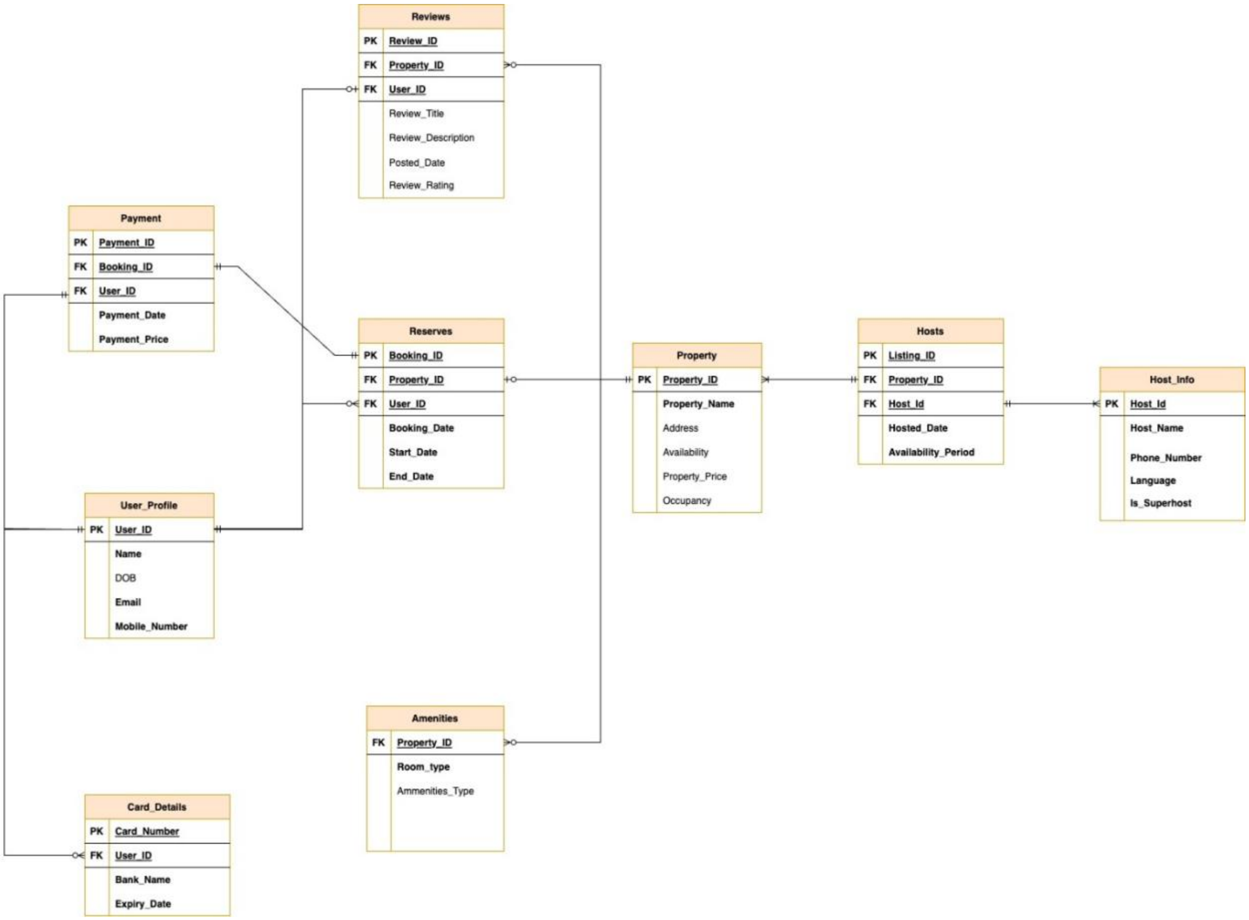
Conceptual Model Description

As we all know, conceptual model is a high-level model of a database which is used for better understanding of requirements and needs. The above figure represents a high-level model of the database that we as a team created for Airbnb. The main characteristics that we wanted to highlight from the Airbnb website are the information of host, different properties that are available in Airbnb website which have been hosted by hosts, how a user can complete payment and review each property. The conceptual model also gives out useful information on relationships between different entities.

- Each host can host multiple properties
- Each property can have multiple hosts (example: a host and a sub host)
- Similarly, a user can book multiple properties
- A property can be booked by multiple users
- A user has only one payment
- Each payment must have only one user
- Also, a user may or may not give reviews
- Each review can be linked only to a single user

All such useful information already gives out an overview of business requirements of Airbnb.

Logical Model

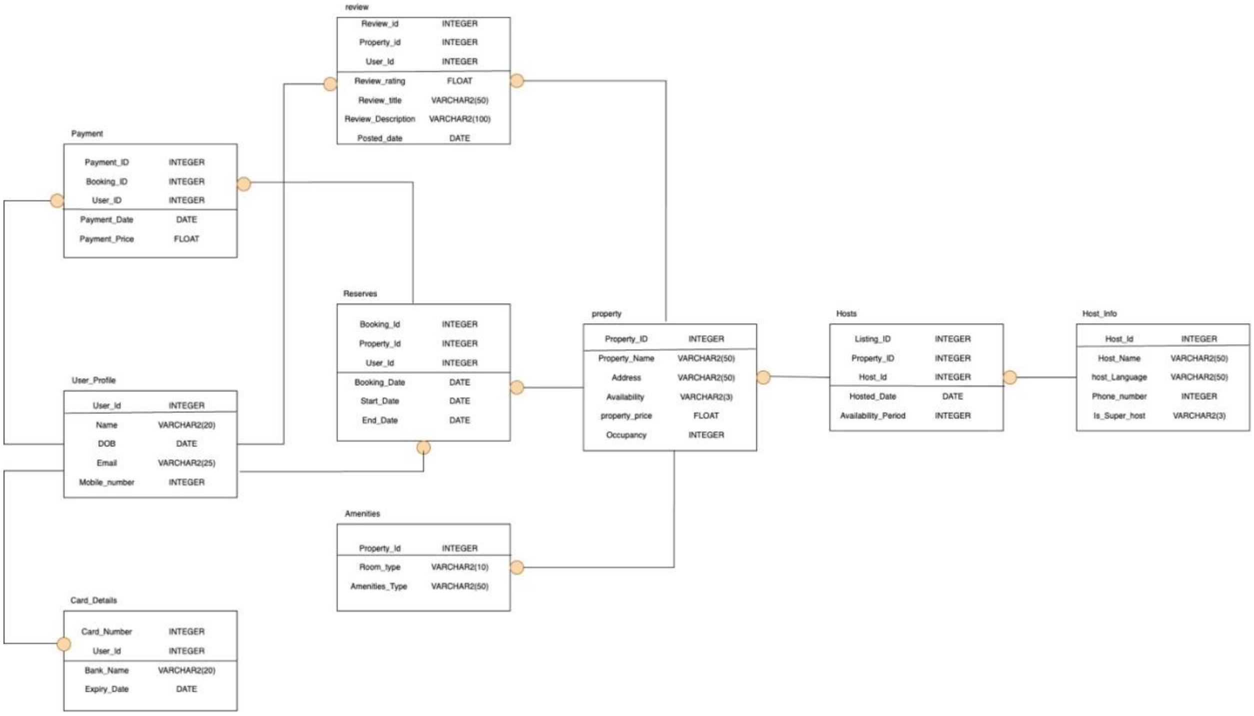


Logical Model Description

The Logical model is a much more sophisticated model which gives information on how the database should be constructed and its technicalities. Every user, property, host, transactional payments are assigned unique IDs which is the Primary key to identify the tables and can be used as foreign key to reference the tables.

The logical model for Airbnb gives the actual break up of tables and columns in each table and is much more refined than the conceptual model. For example: the optional many to many relationships between the host and property in the conceptual model is further broken down by introducing the “hosts” table where it contains the host id and property id referenced as foreign keys and listing id as the Primary key.
Similarly, the optional many to many direct relationship is broken down in logical model by introducing the “books” table and making property id and user id as foreign keys, with booking id as primary key.
This logical model of Airbnb is the closest model to the physical model seen in the next section.

Physical Model



Approach Used to Develop the Solution

The approach our team followed was to create accounts in Airbnb as both a normal user who books a property and as a host who hosts properties. We used the Airbnb website to gather maximum information for our model. We took a note of all the user information Airbnb collects from its users, as users are the main participants.

An interesting thing that we came across about Airbnb is that it allows the user to be a host and host any space, be it a single room and an entire home. The host needs to enter all information about the property he/she is going to host such as location, price, occupancy, special amenities if any, rent and so on. We wanted to make sure we do not miss that information. If a host has good years of experience and best reviews, the host would become a superhost and this was one keyword that was solely unique that we did not see in other websites and captured in Airbnb's website.

As a normal user(not host), we looked at properties that can be booked and its availability by filtering check in and check out dates, locality, amenities and so on. It was fun to discover a lot of new and interesting places just in Utah. We also made sure that we captured the payment options and the method.

For online booking websites like Airbnb, reviews play a significant role. Mimicking the website, we made sure to capture the ratings, review posted dates, users who posted reviews and the description for future users to choose better and not miss anything.

Description of the Sample Data

The sample data includes a few best rated properties from Airbnb website and mostly includes properties and spaces in and around Utah since our audience (including our team) live in Utah and they could use this information for fall/winter break.

We listed a reasonable amount of standout amenities which could be an attraction for users to book. Also made sure to include review rating and descriptions for best places in the Database.

Future Features

Airbnb has unveiled a roadmap for its core business that is designed to bring key benefits for personal, local and authentic travel across the globe. The roadmaps list down the following:

- **Inclusion of new property types** such as Vacation homes, B&B and Boutiques in addition to private room, entire home, and shared spaces. This will give increased choices for guests, makes sure better transparency over the amenities available and gives opportunities for hosts to showcase their unique and best features.
- **New tires:** AirBnb plus & Beyond which supports special features and offers. This includes verified spaces for quality and comfort. Plans to open more spaces in 13 more cities. Airbnb plus homes inspection and personal verification with over 100+ checklist points.
- **Airbnb Collections:** Homes that are well suited for every possible occasion which includes 2 plans - Airbnb for Family and Airbnb for work which has spaces fitting any occasions such as weddings, group getaways, social stays etc
- **Community investments** which include revamped Superhost program (hosts with experience and better ratings) recognizing most loved hosts and also a new guest membership program which will recognize best guests and add more benefits in their membership programs.

Compatibility of future features with current model:

The current model for Airbnb will support inclusion of new property and can be done by inserting new property into the list of properties. Any new and unique amenities that are going to be supported can also be mentioned. Once Airbnb Plus & Beyond decides upon the features, it can be added to the database as the current database does not support unique features for hosts and users yet.

Requirements Review

To meet are design requirements, we went through and started analyzing the organizational requirements and we were able to narrow down our priorities to nine specific tables which are as follows:

Reviews

Complete – The reviews table has the primary key Review_ID and foreign keys Property_ID and User_ID. This key structure allows reviews to be submitted by a user and linked to a property which links to a host.

Reserves

Complete – The reserves table has the primary key Booking_ID and foreign keys Property_ID and User_ID. This key structure allows reservations to be booked by a user and then linked to the property which links to the host.

Payment

Complete – The payment table has the primary key Payment_ID and foreign keys Booking_ID and User_ID. This key structure allows payments to associated with the unique user and unique property that user booked.

User_Profile

Complete – The user_profile table has the primary key User_ID. As users are central to the processes within our data base User_ID is used as a foreign key in four other tables surrounding the user.

Card_Details

Complete – The card_details table has the primary key Card_Number and the foreign key User_ID. This allows a credit card to be linked to its user and the payment table.

Property

Complete – The property table has the primary key Property_ID. This table is linked to the reserves and hosts tables through foreign keys.

Hosts

Complete – The hosts table has the primary key Listing_ID and foreign keys Property_ID and Host_ID. This key structure allows hosts to be linked to their unique information and their properties.

Host_Info

Complete – The owner table has the primary key Host_ID. This table is linked to the hosts table by a foreign key structure.

Amenities

Complete – The amenities table has the foreign key Property_ID. This key structure allows specific amenities to be linked to reviews.

Conclusion:

Airbnb is a very promising online booking website which has been a contributor to making every traveler's life better with best-in-class amenities, offers and customer support. Airbnb's success can be measured from their future plans to expand into 13+ new cities with more than over 2000+ spaces to share with guests. With transparency (about the property and user), guests can pick the best spaces to stay comfortably and peacefully in any corner of the world.

References

Most of the information for this project was obtained from airbnb.com.

Research on company statistics and history was obtain from:

https://www.investopedia.com/articles/personal-finance/032814/pros-and-cons-using-airbnb.asp

https://en.wikipedia.org/wiki/Airbnb

https://finance.yahoo.com/news/airbnb-sold-30-000-obama-204034867.html?guccounter=1&guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_sig=AQAAABgQrd9-VkeixAM1__JJ-w2MQRcaCjTTll9FE7BKFPnFNRYjQMI4DzM9Ra1jAx9b6xKJBAnGSppFHu9WOBKdxFIi1Jo3dj6MA6KrLSXB7BmlWjTB55Df0YOlUCq7LSC9u3JW4YY022RgrZOuH0Hy882THHORh0rXauJ16vmlZx5D#:~:text=Many%20people%20know%20the%20story,bring%20in%20some%20needed%20cash.

https://techcrunch.com/2008/08/11/airbed-and-breakfast-takes-pad-crashing-to-a-whole-new-level/

https://www.stratosjets.com/blog/airbnb-statistics/

Appendix

SQL Statements

Create Tables

```
drop table if exists payment;

drop table if exists card_details;

drop table if exists ammenities;

drop table if exists hosts;

drop table if exists review;

drop table if exists reserves;

drop table if exists host_info;

drop table if exists user_profile;

drop table if exists property;


CREATE TABLE user_profile (

user_id INTEGER NOT NULL,

username VARCHAR(255) NOT NULL,

dob DATE,

email VARCHAR(255),

mobile_number VARCHAR(20) NOT NULL,

PRIMARY KEY(user_id)

);


create table property (

property_id integer,

property_name varchar(255),

address varchar(255),

availability VARCHAR(255),

property_price money,

occupancy integer,

PRIMARY KEY(property_id)

);


create table reserves (
booking_id INTEGER,
property_id INTEGER,
user_id INTEGER,
start_date DATE,
end_date DATE,
booking_date DATE,
PRIMARY KEY(booking_id),
CONSTRAINT reserves_fkey_user_id FOREIGN KEY (user_id) REFERENCES user_profile (user_id),
CONSTRAINT reserves_fkey_property_id FOREIGN KEY (property_id) REFERENCES property
(property_id)
);

create table payment (
payment_id INTEGER,
booking_id INTEGER,
user_id INTEGER,
payment_date date,
payment_price money,
PRIMARY KEY(payment_id),
CONSTRAINT payment_fkey_booking_id FOREIGN KEY (booking_id) REFERENCES reserves
(booking_id),
CONSTRAINT payment_fkey_user_id FOREIGN KEY (user_id) REFERENCES user_profile (user_id)
);

create table card_details (
card_number VARCHAR(255),
user_id INTEGER,
expiry_date VARCHAR(20),
bank_name VARCHAR(255),
PRIMARY KEY(card_number),
CONSTRAINT card_details_fkey_user_id FOREIGN KEY (user_id) REFERENCES user_profile (user_id)
);
```

```
create table ammenities (

property_id integer,

room_type varchar(255),

ammenities_type varchar(255),

primary key (property_id),

CONSTRAINT ammenities_fkey_user_id FOREIGN KEY (property_id) REFERENCES property
(property_id)

);


create table host_info (

host_id integer not null,

host_name varchar(255),

host_language varchar(255),

phone_number varchar(255) not null,

is_super_host varchar(10),

primary key(host_id)

);


create table hosts (

listing_id integer not null,

property_id integer,

host_id integer,

host_date date,

availability_period varchar(255),

primary key(listing_id),

CONSTRAINT hosting_fkey_user_id FOREIGN KEY (property_id) REFERENCES property (property_id),

CONSTRAINT hosting_fkey_host_id FOREIGN KEY (host_id) REFERENCES host_info (host_id)

);


create table review (

review_id integer,

booking_id integer,

user_id integer,
review_rating decimal,
review_title varchar(255),
review_description varchar(512),
posted_date date,
primary key(review_id),
CONSTRAINT review_fkey_booking_id FOREIGN KEY (booking_id) REFERENCES reserves (booking_id),
CONSTRAINT review_fkey_user_id FOREIGN KEY (user_id) REFERENCES user_profile (user_id)
);
```

Insert Statements

```
INSERT INTO user_profile VALUES (1001,'David Kruger','10-17-1994', 'david.kruger@gmail.com', '206-677-8979');
INSERT INTO user_profile VALUES (1002,'Stanford Pugsley','06-15-1980', 'stan.pugsley@gmail.com', '123-345-6787');
INSERT INTO user_profile VALUES (1003,'Braddy Nike','10-17-1994', 'braddy.nike@gmail.com', '321-455-6321');
INSERT INTO user_profile VALUES (1004,'Lucifer Bells','10-17-2003', 'lucifer.bells@gmail.com', '890-221-3434');
INSERT INTO user_profile VALUES (1005,'Thomsan Smith','10-17-2001', 'smith.thomsan@gmail.com', '801-565-7345');
INSERT INTO user_profile VALUES (1006,'Rutherford Barry','10-17-2004', 'barry.rutherford@gmail.com', '277-344-5666');
INSERT INTO user_profile VALUES (1007,'Curtus Dani','10-17-2006', 'curtus.danithomsan@gmail.com', '608-405-5678');
INSERT INTO user_profile VALUES (1008,'Shiny Wood','10-17-2001', 'shiny.wood@gmail.com', '104-688-9080');
INSERT INTO user_profile VALUES (1009,'Kelin Smith','10-17-2002', 'kelin.smith@gmail.com', '201-699-9061');
INSERT INTO user_profile VALUES (1010,'Wonder Don','10-17-2001', 'don.wonder@gmail.com', '455-282-7643');

select * from user_profile;
```

```
INSERT INTO property values (1011,'The Domes at Canyonlands', 'Monticello, Utah', 'yes', 225, 5);
INSERT INTO property values (1015,'Dreamy Treehouse', 'Parkcity, Utah', 'yes', 60, 3);
INSERT INTO property values (1020,'Zion EcoCabin', 'Hildale, Utah', 'yes', 200, 4);
INSERT INTO property values (1003,'Crown Heights Garden Apt', 'Crown Heights, Brooklyn', 'no', 400, 8);
INSERT INTO property values (1000,'ENJOY Downtown NYC!!', 'Manhattan, East Village', 'yes', 120, 3);
INSERT INTO property values (1999,'Cliffside Tiny Homes', 'Orderville, Utah', 'no', 500, 10);
INSERT INTO property values (1200,'Columbia Castle in Brooklyn Heights', 'Brooklyn Heights, Brooklyn', 'yes', 350, 5);
INSERT INTO property values (1007,'Carriage House on the Stream', 'Sundance, Utah', 'yes', 100, 3);
INSERT INTO property values (1213,'Historic House Boerum Hill', 'Boerum Hill, Brooklyn', 'no', 300, 4);
INSERT INTO property values (1314,'The Brooklyn Waverly', 'Clinton Hill, Brooklyn', 'no', 250, 2);

select * from property;


INSERT INTO reserves values (1111, 1015, 1001, '2021-12-04', '2021-12-10', '2021-08-03');
INSERT INTO reserves values (1234, 1020, 1002, '2021-12-01', '2021-12-05', '2021-09-01');
INSERT INTO reserves values (2222, 1314, 1008, '2022-05-03', '2022-05-08', '2022-04-01');
INSERT INTO reserves values (2345, 1003, 1004, '2022-03-10', '2022-03-12', '2022-01-08');
INSERT INTO reserves values (3333, 1999, 1010, '2020-04-04', '2020-04-12', '2020-03-02');
INSERT INTO reserves values (3456, 1000, 1007, '2019-01-11', '2019-01-15', '2018-09-09');
INSERT INTO reserves values (4444, 1213, 1009, '2020-10-01', '2020-10-08', '2020-08-01');
INSERT INTO reserves values (4567, 1007, 1006, '2022-09-11', '2022-09-16', '2022-08-11');
INSERT INTO reserves values (5555, 1011, 1003, '2018-01-02', '2018-01-04', '2017-09-01');
INSERT INTO reserves values (5678, 1200, 1005, '2017-03-10', '2017-03-15', '2016-10-04');

select * from reserves;

INSERT INTO payment VALUES (2345, 1111, 1001, '2021-08-04', 70.99);
INSERT INTO payment VALUES (4567, 1234, 1002, '2021-09-01', 200.99);
INSERT INTO payment VALUES (1234, 2222, 1008, '2022-04-03', 270.99);
INSERT INTO payment VALUES (3456, 2345, 1004, '2022-01-10', 430.99);
INSERT INTO payment VALUES (9087, 3333, 1010, '2020-03-04', 524.99);
INSERT INTO payment VALUES (5678, 3456, 1007, '2018-09-11', 134.99);
INSERT INTO payment VALUES (8765, 4444, 1009, '2020-08-01', 333.99);
INSERT INTO payment VALUES (7098, 4567, 1006, '2022-08-11', 200.99);
INSERT INTO payment VALUES (9845, 5555, 1003, '2017-09-02', 236.99);
INSERT INTO payment VALUES (1077, 5678, 1005, '2016-10-10', 422.99);

select * from payment;

INSERT INTO card_details values ('1313242445455656', 1001, '09-24', 'BOA');
INSERT INTO card_details values ('2345678909875643', 1002, '08-26', 'Zions');
INSERT INTO card_details values ('9078567845231456', 1003, '09-23', 'Wells Fargo');
INSERT INTO card_details values ('5678234590678912', 1004, '12-29', 'Key');
INSERT INTO card_details values ('7623124509005609', 1005, '05-24', 'BOA');
INSERT INTO card_details values ('1245321234443322', 1006, '09-24', 'Zions');
INSERT INTO card_details values ('2156568800235686', 1007, '01-25', 'Wells Fargo');
INSERT INTO card_details values ('7634568911094622', 1008, '09-26', 'Key');
INSERT INTO card_details values ('6589235578097612', 1009, '06-25', 'First');
INSERT INTO card_details values ('4906451133890235', 1010, '03-24', 'Chase');

select * from card_details;

INSERT INTO ammenities values (1011, 'Entire home/apt', 'Swimming pool, BBQ, Pet friendly');
INSERT INTO ammenities values (1015, 'Private room', 'WiFi, Speakers, Pet friendly');
INSERT INTO ammenities values (1020, 'Entire home/apt', 'Theater room, BBQ, Jacuzzi, Game room');
INSERT INTO ammenities values (1003, 'Entire home/apt', 'Swimming pool, Fireplace, Pet friendly');
INSERT INTO ammenities values (1000, 'Private room', 'WiFi, Pet friendly');
INSERT INTO ammenities values (1999, 'Entire home/apt', 'Basketball court, Game room, Theater room');
INSERT INTO ammenities values (1200, 'Entire home/apt', 'Table tennis, Billiards, Jacuzzi');
INSERT INTO ammenities values (1007, 'Entire home/apt', 'BBQ, Swimming pool, Jacuzzi, Theater room');
INSERT INTO ammenities values (1213, 'Entire home/apt', 'Fireplace, Pet friendly');
INSERT INTO ammenities values (1314, 'Private room', 'WiFi, Pet friendly, Fully furnished');

select * from ammenities;



INSERT INTO host_info values (1, 'Henry Bell', 'English, French', '304-444-7890', 'yes');
INSERT INTO host_info values (2, 'Clara Garret', 'English, French, German', '445-345-6789', 'no');
INSERT INTO host_info values (3, 'Henry Cavin', 'English, Spanish', '234-345-5678', 'no');
INSERT INTO host_info values (4, 'Ben Foster', 'Spanish', '211-347-9067', 'yes');
INSERT INTO host_info values (5, 'Michelle Wood', 'French, German', '345-233-8980', 'yes');
INSERT INTO host_info values (6, 'Rose Carey', 'English', '890-444-2345', 'yes');
INSERT INTO host_info values (7, 'Donald Specter', 'English Spanish', '445-345-6889', 'yes');
INSERT INTO host_info values (8, 'Harvey Ross', 'English Chinese', '445-345-5599', 'no');
INSERT INTO host_info values (9, 'Christopher Nolan', 'English, Korean', '374-499-7890', 'yes');
INSERT INTO host_info values (10, 'Emily Donald', 'English, German', '154-546-7521', 'yes');

select * from host_info;



INSERT INTO hosts values (9990, 1011, 1, '2023-01-02', '2022-01-01 - 2023-01-01');
INSERT INTO hosts values (9991, 1003, 2, '2022-03-10', '2022-01-01 - 2023-01-01');
INSERT INTO hosts values (9992, 1015, 3, '2023-12-04', '2023-01-01 - 2024-01-01');
INSERT INTO hosts values (9993, 1020, 4, '2022-12-01', '2022-01-01 - 2023-01-01');
INSERT INTO hosts values (9994, 1000, 5, '2023-01-11', '2023-01-01 - 2024-01-01');
INSERT INTO hosts values (9995, 1200, 6, '2022-03-10', '2022-01-01 - 2023-01-01');
INSERT INTO hosts values (9996, 1999, 7, '2023-04-04', '2023-01-01 - 2024-01-01');
INSERT INTO hosts values (9997, 1007, 8, '2022-09-11', '2022-01-01 - 2023-01-01');
INSERT INTO hosts values (9998, 1213, 9, '2023-10-01', '2023-01-01 - 2024-01-01');
INSERT INTO hosts values (9999, 1314, 10, '2023-05-03', '2023-01-01 - 2024-01-01');

select * from hosts;

INSERT INTO review values (90, 1111, 1001, 4.5, 'Very Good', 'Loved the place', '2022-01-02');
INSERT INTO review values (91, 1234, 1002, 4, 'Good', 'Enjoyed the stay, the tables were big enough', '2022-02-03');
INSERT INTO review values (92, 2222, 1008, 2, 'Bad', 'Disliked the place', '2022-06-05');
INSERT INTO review values (93, 2345, 1004, 4, 'Good', 'A peaceful and calm surrounding', '2022-06-07');
INSERT INTO review values (94, 3333, 1010, 3.5, 'Not Bad', 'Could have maintained better', '2020-08-09');
INSERT INTO review values (95, 3456, 1007, 1, 'Very Bad', 'Very poor maintenance', '2019-10-11');
INSERT INTO review values (96, 4444, 1009, 5, 'Excellent', 'Enjoyed the place and food', '2020-11-12');
INSERT INTO review values (97, 4567, 1006, 4.5, 'Very Good', 'Highly recommended', '2022-10-04');
INSERT INTO review values (98, 5555, 1003, 4, 'Good', 'Had a good stay', '2018-03-06');
INSERT INTO review values (99, 5678, 1005, 3, 'Not Bad', 'Poor food', '2017-08-12');

select * from review;
```

Sample Queries

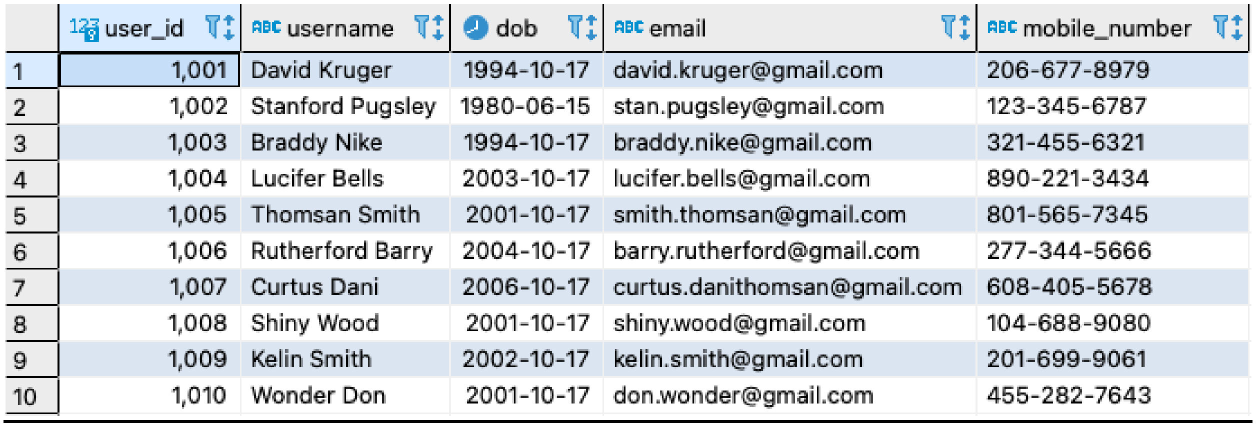
```
select username
from user_profile;

select up.username, r.start_date, r.end_date, p.payment_date, re.review_rating, re.review_title, re.posted_date
from user_profile up
inner join reserves r on up.user_id = r.user_id
inner join payment p on r.user_id = p.user_id
inner join review re on p.user_id = re.user_id;

select u.username, p.property_name, p.address, a.room_type, a.ammenities_type, p.property_price, p.occupancy, r.start_date, r.end_date
from user_profile u
join reserves r on u.user_id = r.user_id
join property p on r.property_id = p.property_id
join ammenities a on p.property_id = a.property_id
where u.user_id = 1002;
```

```
select u.username, p.property_name, re.posted_date, re.review_rating, re.review_title,
re.review_description
from user_profile u
join review re on u.user_id = re.user_id
join reserves r on re.user_id = r.user_id
join property p on r.property_id = p.property_id
where u.user_id = 1002;

select p.property_id, p.property_name, p.address, re.review_rating
from property p
join reserves r on p.property_id = r.property_id
join review re on re.user_id = r.user_id
where re.review_rating >= 3;

update property
set property_name = 'Zions EcoCabin'
where property_id = 1020;

select property_id, property_name, address
from property
where address like '%Uta%h';

alter table property
drop column property_price;

select * from property;
```

Example Database Tables

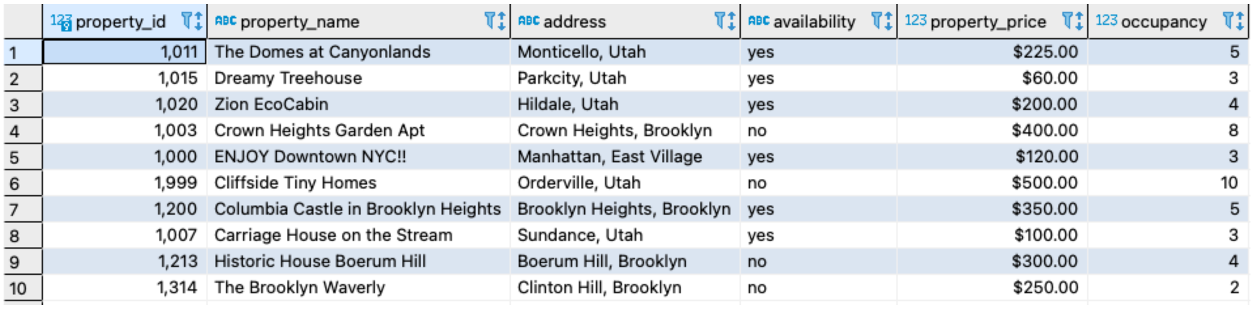
User_profile Table - SQL

airbnb



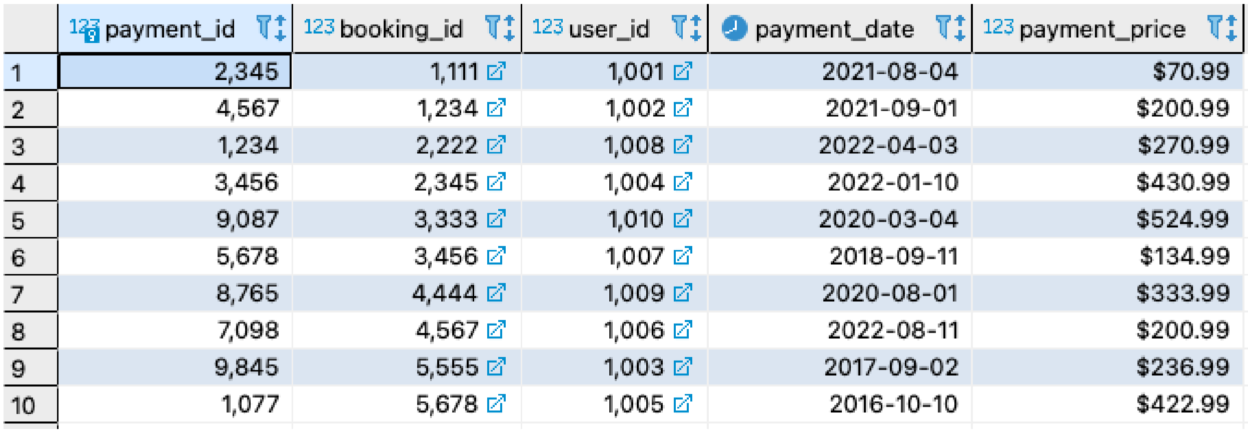
	user_id	username	dob	email	mobile_number
1	1,001	David Kruger	1994-10-17	david.kruger@gmail.com	206-677-8979
2	1,002	Stanford Pugsley	1980-06-15	stan.pugsley@gmail.com	123-345-6787
3	1,003	Braddy Nike	1994-10-17	braddy.nike@gmail.com	321-455-6321
4	1,004	Lucifer Bells	2003-10-17	lucifer.bells@gmail.com	890-221-3434
5	1,005	Thomsan Smith	2001-10-17	smith.thomsan@gmail.com	801-565-7345
6	1,006	Rutherford Barry	2004-10-17	barry.rutherford@gmail.com	277-344-5666
7	1,007	Curtus Dani	2006-10-17	curtus.danithomsan@gmail.com	608-405-5678
8	1,008	Shiny Wood	2001-10-17	shiny.wood@gmail.com	104-688-9080
9	1,009	Kelin Smith	2002-10-17	kelin.smith@gmail.com	201-699-9061
10	1,010	Wonder Don	2001-10-17	don.wonder@gmail.com	455-282-7643

Property Table – SQL



	property_id	property_name	address	availability	property_price	occupancy
1	1,011	The Domes at Canyonlands	Monticello, Utah	yes	$225.00	5
2	1,015	Dreamy Treehouse	Parkcity, Utah	yes	$60.00	3
3	1,020	Zion EcoCabin	Hildale, Utah	yes	$200.00	4
4	1,003	Crown Heights Garden Apt	Crown Heights, Brooklyn	no	$400.00	8
5	1,000	ENJOY Downtown NYC!!	Manhattan, East Village	yes	$120.00	3
6	1,999	Cliffside Tiny Homes	Orderville, Utah	no	$500.00	10
7	1,200	Columbia Castle in Brooklyn Heights	Brooklyn Heights, Brooklyn	yes	$350.00	5
8	1,007	Carriage House on the Stream	Sundance, Utah	yes	$100.00	3
9	1,213	Historic House Boerum Hill	Boerum Hill, Brooklyn	no	$300.00	4
10	1,314	The Brooklyn Waverly	Clinton Hill, Brooklyn	no	$250.00	2

Payment Table – SQL



	payment_id	booking_id	user_id	payment_date	payment_price
1	2,345	1,111	1,001	2021-08-04	$70.99
2	4,567	1,234	1,002	2021-09-01	$200.99
3	1,234	2,222	1,008	2022-04-03	$270.99
4	3,456	2,345	1,004	2022-01-10	$430.99
5	9,087	3,333	1,010	2020-03-04	$524.99
6	5,678	3,456	1,007	2018-09-11	$134.99
7	8,765	4,444	1,009	2020-08-01	$333.99
8	7,098	4,567	1,006	2022-08-11	$200.99
9	9,845	5,555	1,003	2017-09-02	$236.99
10	1,077	5,678	1,005	2016-10-10	$422.99

Group Contribution Table

Team Member	Hours Spent	Activity
Everyone	4	Initial meeting to plan project logistics, discussion of relational and logical diagram designs, refinement of power point
Curtis	8.5	Company research, design of presentation, creation of slides, formatting project summary report, writing project summary report
Umesh	6	Create statements, Insert statements, ppt, project summary report preparation
Shankar	7	Creation of conceptual, logical, and physical diagrams
Vidya	7	Framed the initial conceptual model, employed DBeaver to create and populate database, tables and columns, worked on queries, contributed to project summary report