

PHASE 2

Development Phase

SQL PROJECT: **Database Design** and
Implementation for Airbnb Use Case

By SK SAHIL

Matriculation: 9213036

Updated Test Cases



Phase 2: Developing an Airbnb Database

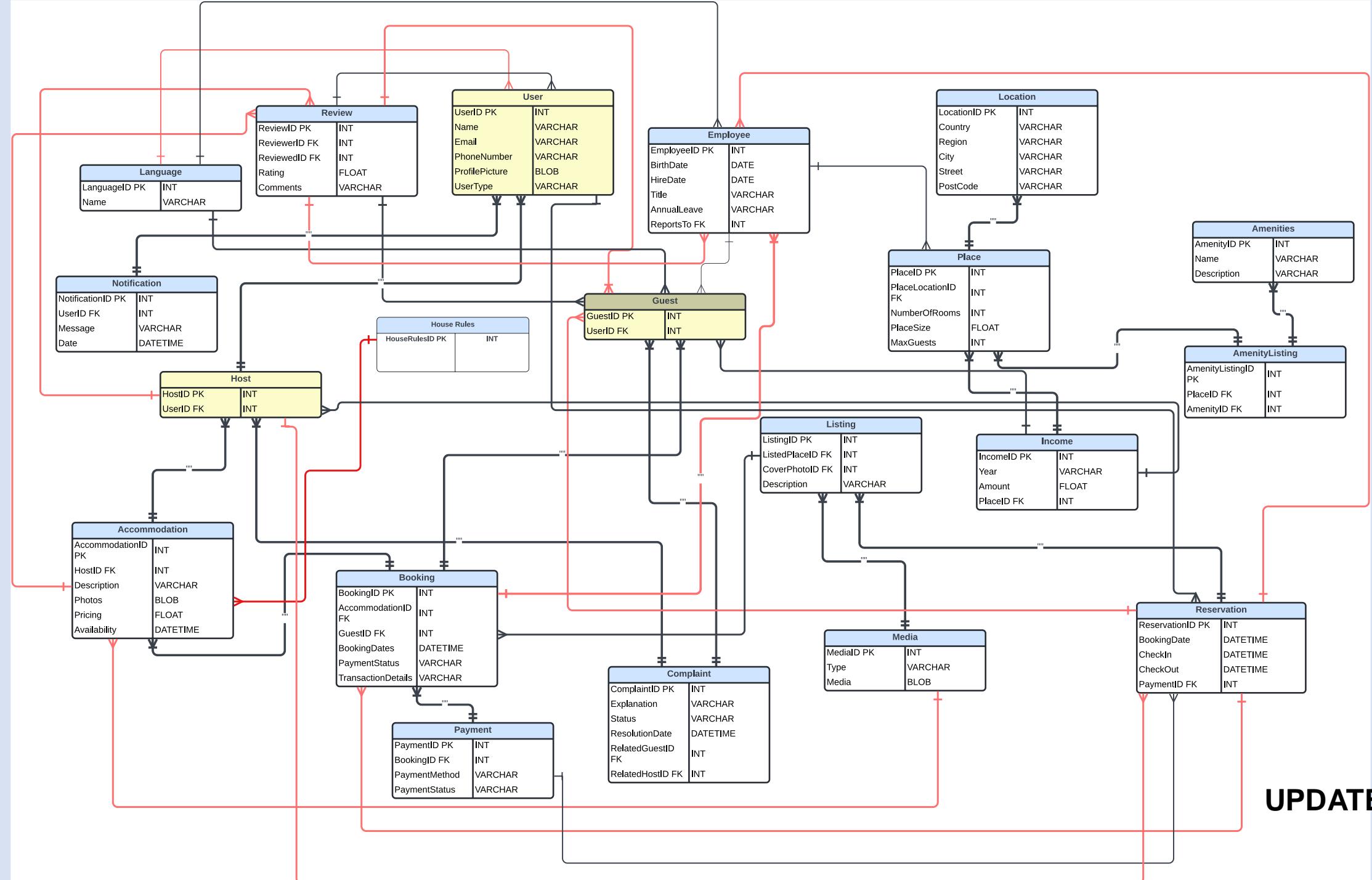


Name: SK Sahil
Program: Bachelor in Computer Science
Matriculation: 9213036
Institute: IU Internationale Hochschule

Summary of the Implementation

This project focuses on creating a relational database tailored for an online booking platform like Airbnb. The development process was carefully structured into several key phases: identifying entities, defining their relationships, determining primary keys, converting the conceptual model into a physical data model, defining and normalizing attributes, establishing consistency conditions, and finally, formulating the necessary transactions. Each step was crucial in building a robust and efficient database system.





UPDATED ERM

User

Record and maintain crucial data.

Create the Database & table:

```
1 • CREATE DATABASE Airbnb;
2
3 • USE Airbnb;
4
5 • CREATE TABLE User (
6     UserID INT PRIMARY KEY,
7     Name VARCHAR(255),
8     Email VARCHAR(255),
9     PhoneNumber VARCHAR(50),
10    ProfilePicture BLOB,
11    UserType VARCHAR(50)
12 );
```

Employ INSERT INTO commands to populate the table with user data.

```
INSERT INTO User (UserID, Name, Email, PhoneNumber, ProfilePicture, UserType) VALUES
(1, 'John Doe', 'john@example.com', '1234567890', NULL, 'Guest'),
(2, 'Jane Smith', 'jane@example.com', '0987654321', NULL, 'Host'),
(3, 'Jim Brown', 'jim@example.com', '1231231234', NULL, 'Guest'),
(4, 'Emily Davis', 'emily@example.com', '3213213210', NULL, 'Host'),
(5, 'Michael Wilson', 'michael@example.com', '2132132130', NULL, 'Guest'),
(6, 'Sarah Johnson', 'sarah@example.com', '2312312310', NULL, 'Host'),
(7, 'David Miller', 'david@example.com', '1321321320', NULL, 'Guest'),
(8, 'Laura Garcia', 'laura@example.com', '3123123120', NULL, 'Host'),
(9, 'James Martinez', 'james@example.com', '1233211234', NULL, 'Guest'),
(10, 'Linda Rodriguez', 'linda@example.com', '3211233210', NULL, 'Host'),
(11, 'Robert Hernandez', 'robert@example.com', '2133212130', NULL, 'Guest'),
(12, 'Patricia Lopez', 'patricia@example.com', '2313212310', NULL, 'Host'),
(13, 'Charles Gonzalez', 'charles@example.com', '1323211320', NULL, 'Guest'),
(14, 'Barbara Wilson', 'barbara@example.com', '3123213120', NULL, 'Host'),
(15, 'Thomas Anderson', 'thomas@example.com', '1232131234', NULL, 'Guest'),
(16, 'Jessica Martinez', 'jessica@example.com', '3212133210', NULL, 'Host'),
(17, 'Christopher Lee', 'christopher@example.com', '2132132130', NULL, 'Guest'),
(18, 'Karen Taylor', 'karen@example.com', '2312312310', NULL, 'Host'),
(19, 'Daniel Moore', 'daniel@example.com', '1321321320', NULL, 'Guest'),
(20, 'Nancy Jackson', 'nancy@example.com', '3123123120', NULL, 'Host');
```

Output:

UserID	Name	Email	PhoneNumber	ProfilePi...	UserType
1	John Doe	john@example.com	1234567890	NULL	Guest
2	Jane Smith	jane@example.com	0987654321	NULL	Host
3	Jim Brown	jim@example.com	1231231234	NULL	Guest
4	Emily Davis	emily@example.com	3213213210	NULL	Host
5	Michael Wilson	michael@example.com	2132132130	NULL	Guest
6	Sarah Johnson	sarah@example.com	2312312310	NULL	Host
7	David Miller	david@example.com	1321321320	NULL	Guest
8	Laura Garcia	laura@example.com	3123123120	NULL	Host
9	James Martinez	james@example.com	1233211234	NULL	Guest
10	Linda Rodriguez	linda@example.com	3211233210	NULL	Host
11	Robert Hernandez	robert@example.com	2133212130	NULL	Guest
12	Patricia Lopez	patricia@example.com	2313212310	NULL	Host
13	Charles Gonzalez	charles@example.com	1323211320	NULL	Guest
14	Barbara Wilson	barbara@example.com	3123213120	NULL	Host
15	Thomas Anderson	thomas@example.com	1232131234	NULL	Guest
16	Jessica Martinez	jessica@example.com	3212133210	NULL	Host
17	Christopher Lee	christopher@example.com	2132132130	NULL	Guest
18	Karen Taylor	karen@example.com	2312312310	NULL	Host
19	Daniel Moore	daniel@example.com	1321321320	NULL	Guest
20	Nancy Jackson	nancy@example.com	3123123120	NULL	Host

Language

Holds details about the languages available in the Airbnb system, enabling multilingual support. It acts as a reference table, with its primary key (id) available for use as a foreign key in other tables that require language information.

Create the Table:

```
USE Airbnb;

CREATE TABLE Language (
    LanguageID INT PRIMARY KEY,
    Name VARCHAR(50)
);
```

Insert the data:

```
INSERT INTO Language (LanguageID, Name) VALUES
(1, 'English'),
(2, 'Spanish'),
(3, 'French'),
(4, 'German'),
(5, 'Chinese'),
(6, 'Japanese'),
(7, 'Russian'),
(8, 'Portuguese'),
(9, 'Italian'),
(10, 'Korean'),
(11, 'Arabic'),
(12, 'Hindi'),
(13, 'Bengali'),
(14, 'Urdu'),
(15, 'Dutch'),
(16, 'Greek'),
(17, 'Turkish'),
(18, 'Vietnamese'),
(19, 'Thai'),
(20, 'Swedish');
```

Output:

LanguageID	Name
1	English
2	Spanish
3	French
4	German
5	Chinese
6	Japanese
7	Russian
8	Portuguese
9	Italian
10	Korean
11	Arabic
12	Hindi
13	Bengali
14	Urdu
15	Dutch
16	Greek
17	Turkish
18	Vietnamese
19	Thai
20	Swedish

Review

Create the table:

```
USE Airbnb;
```

```
) CREATE TABLE Review (
    ReviewID INT PRIMARY KEY,
    ReviewerID INT,
    ReviewedID INT,
    Rating FLOAT,
    Comments VARCHAR(255),
    FOREIGN KEY (ReviewerID) REFERENCES User(UserID),
    FOREIGN KEY (ReviewedID) REFERENCES User(UserID)
);
```

Insert the data:

```
INSERT INTO Review (ReviewID, ReviewerID, ReviewedID, Rating, Comments)
VALUES
(1, 1, 2, 4.5, 'Great place!'),
(2, 3, 4, 3.8, 'Nice host.'),
(3, 5, 6, 4.2, 'Comfortable stay.'),
(4, 7, 8, 4.0, 'Could be better.'),
(5, 9, 10, 5.0, 'Excellent experience!'),
(6, 11, 12, 3.5, 'Not bad.'),
(7, 13, 14, 4.8, 'Highly recommended.'),
(8, 15, 16, 4.1, 'Very good.'),
(9, 17, 18, 3.9, 'Satisfactory.'),
(10, 19, 20, 4.6, 'Wonderful host!'),
(11, 2, 1, 4.3, 'Pleasant stay.'),
(12, 4, 3, 4.7, 'Very nice place.'),
(13, 6, 5, 4.0, 'Good value.'),
(14, 8, 7, 4.4, 'Enjoyed my stay.'),
(15, 10, 9, 4.9, 'Fantastic!'),
(16, 12, 11, 3.7, 'Average experience.'),
(17, 14, 13, 4.5, 'Will visit again.'),
(18, 16, 15, 4.1, 'Nice and cozy.'),
(19, 18, 17, 4.0, 'Decent place.'),
(20, 20, 19, 4.3, 'Had a good time.');
```

Output:

ReviewID	ReviewerID	ReviewedID	Rating	Comments
1	1	2	4.5	Great place!
2	3	4	3.8	Nice host.
3	5	6	4.2	Comfortable stay.
4	7	8	4	Could be better.
5	9	10	5	Excellent experience!
6	11	12	3.5	Not bad.
7	13	14	4.8	Highly recommended.
8	15	16	4.1	Very good.
9	17	18	3.9	Satisfactory.
10	19	20	4.6	Wonderful host!
11	2	1	4.3	Pleasant stay.
12	4	3	4.7	Very nice place.
13	6	5	4	Good value.
14	8	7	4.4	Enjoyed my stay.
15	10	9	4.9	Fantastic!
16	12	11	3.7	Average experience.
17	14	13	4.5	Will visit again.
18	16	15	4.1	Nice and cozy.
19	18	17	4	Decent place.
20	20	19	4.3	Had a good time.

Notification

Create the table:

```
USE Airbnb;

CREATE TABLE Notification (
    NotificationID INT PRIMARY KEY,
    UserID INT,
    Message VARCHAR(255),
    Date DATETIME,
    FOREIGN KEY (UserID) REFERENCES User(UserID)
);
```

Insert elements of the table:

```
INSERT INTO Notification (NotificationID, UserID, Message, Date) VALUES
(1, 1, 'Your booking is confirmed.', '2024-01-01 10:00:00'),
(2, 2, 'New review received.', '2024-01-02 11:00:00'),
(3, 3, 'Payment processed.', '2024-01-03 12:00:00'),
(4, 4, 'Your listing is live.', '2024-01-04 13:00:00'),
(5, 5, 'Booking request received.', '2024-01-05 14:00:00'),
(6, 6, 'New message from guest.', '2024-01-06 15:00:00'),
(7, 7, 'Your booking is canceled.', '2024-01-07 16:00:00'),
(8, 8, 'Profile updated.', '2024-01-08 17:00:00'),
(9, 9, 'Review reminder.', '2024-01-09 18:00:00'),
(10, 10, 'Host response received.', '2024-01-10 19:00:00'),
(11, 11, 'New review received.', '2024-01-11 20:00:00'),
(12, 12, 'Payment failed.', '2024-01-12 21:00:00'),
(13, 13, 'Booking confirmed.', '2024-01-13 22:00:00'),
(14, 14, 'New message from host.', '2024-01-14 23:00:00'),
(15, 15, 'Listing approved.', '2024-01-15 09:00:00'),
(16, 16, 'Booking reminder.', '2024-01-16 08:00:00'),
(17, 17, 'Profile picture updated.', '2024-01-17 07:00:00'),
(18, 18, 'Payment pending.', '2024-01-18 06:00:00'),
(19, 19, 'Booking request received.', '2024-01-19 05:00:00'),
(20, 20, 'Your booking is confirmed.', '2024-01-20 04:00:00');
```

Output:

NotificationID	UserID	Message	Date
1	1	Your booking is confirmed.	2024-01-01 10:00:00
2	2	New review received.	2024-01-02 11:00:00
3	3	Payment processed.	2024-01-03 12:00:00
4	4	Your listing is live.	2024-01-04 13:00:00
5	5	Booking request received.	2024-01-05 14:00:00
6	6	New message from guest.	2024-01-06 15:00:00
7	7	Your booking is canceled.	2024-01-07 16:00:00
8	8	Profile updated.	2024-01-08 17:00:00
9	9	Review reminder.	2024-01-09 18:00:00
10	10	Host response received.	2024-01-10 19:00:00
11	11	New review received.	2024-01-11 20:00:00
12	12	Payment failed.	2024-01-12 21:00:00
13	13	Booking confirmed.	2024-01-13 22:00:00
14	14	New message from host.	2024-01-14 23:00:00
15	15	Listing approved.	2024-01-15 09:00:00
16	16	Booking reminder.	2024-01-16 08:00:00
17	17	Profile picture updated.	2024-01-17 07:00:00
18	18	Payment pending.	2024-01-18 06:00:00
19	19	Booking request received.	2024-01-19 05:00:00
20	20	Your booking is confirmed.	2024-01-20 04:00:00

Host

Captures details about hosts, such as the languages they speak. This table is used to store host-related information, including their language skills and their connection to user accounts.

Output:

HostID	UserID
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19
20	20

Create the table:

```
1 • USE Airbnb;
2
3 • CREATE TABLE Host (
4     HostID INT PRIMARY KEY,
5     UserID INT,
6     FOREIGN KEY (UserID) REFERENCES User(UserID)
7 );
```

Insert elements of the table:

```
INSERT INTO Host (HostID, UserID) VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 4),
(5, 5),
(6, 6),
(7, 7),
(8, 8),
(9, 9),
(10, 10),
(11, 11),
(12, 12),
(13, 13),
(14, 14),
(15, 15),
(16, 16),
(17, 17),
(18, 18),
(19, 19),
(20, 20);
```

Accommodation

Create the table:

```
USE Airbnb;
```

```
CREATE TABLE Accommodation (
    AccommodationID INT PRIMARY KEY,
    HostID INT,
    Description VARCHAR(255),
    Photos BLOB,
    Pricing FLOAT,
    Availability DATETIME,
    FOREIGN KEY (HostID) REFERENCES Host(HostID)
);
```

Insert elements of the table:

```
INSERT INTO Accommodation (AccommodationID, HostID, Description, Photos, Pricing, Availability) VALUES
(1, 1, 'Cozy apartment in city center.', NULL, 100.0, '2024-01-01 10:00:00'),
(2, 2, 'Modern house with garden.', NULL, 150.0, '2024-01-02 11:00:00'),
(3, 3, 'Beachfront villa.', NULL, 200.0, '2024-01-03 12:00:00'),
(4, 4, 'Mountain cabin.', NULL, 120.0, '2024-01-04 13:00:00'),
(5, 5, 'Luxury penthouse.', NULL, 300.0, '2024-01-05 14:00:00'),
(6, 6, 'Rustic cottage.', NULL, 80.0, '2024-01-06 15:00:00'),
(7, 7, 'City apartment with view.', NULL, 110.0, '2024-01-07 16:00:00'),
(8, 8, 'Country house.', NULL, 90.0, '2024-01-08 17:00:00'),
(9, 9, 'Charming bungalow.', NULL, 130.0, '2024-01-09 18:00:00'),
(10, 10, 'Spacious loft.', NULL, 140.0, '2024-01-10 19:00:00'),
(11, 11, 'Historic home.', NULL, 160.0, '2024-01-11 20:00:00'),
(12, 12, 'Modern condo.', NULL, 110.0, '2024-01-12 21:00:00'),
(13, 13, 'Lake house.', NULL, 180.0, '2024-01-13 22:00:00'),
(14, 14, 'Downtown studio.', NULL, 90.0, '2024-01-14 23:00:00'),
(15, 15, 'Country estate.', NULL, 250.0, '2024-01-15 09:00:00'),
(16, 16, 'Urban loft.', NULL, 130.0, '2024-01-16 08:00:00'),
(17, 17, 'Luxury apartment.', NULL, 220.0, '2024-01-17 07:00:00'),
(18, 18, 'Secluded cabin.', NULL, 95.0, '2024-01-18 06:00:00'),
(19, 19, 'Elegant townhouse.', NULL, 170.0, '2024-01-19 05:00:00'),
(20, 20, 'Cozy guesthouse.', NULL, 85.0, '2024-01-20 04:00:00');
```

Output:

AccommodationID	HostID	Description	Photos	Pricing	Availability
1	1	Cozy apartment in city center.	NULL	100	2024-01-01 10:00:00
2	2	Modern house with garden.	NULL	150	2024-01-02 11:00:00
3	3	Beachfront villa.	NULL	200	2024-01-03 12:00:00
4	4	Mountain cabin.	NULL	120	2024-01-04 13:00:00
5	5	Luxury penthouse.	NULL	300	2024-01-05 14:00:00
6	6	Rustic cottage.	NULL	80	2024-01-06 15:00:00
7	7	City apartment with view.	NULL	110	2024-01-07 16:00:00
8	8	Country house.	NULL	90	2024-01-08 17:00:00
9	9	Charming bungalow.	NULL	130	2024-01-09 18:00:00
10	10	Spacious loft.	NULL	140	2024-01-10 19:00:00
11	11	Historic home.	NULL	160	2024-01-11 20:00:00
12	12	Modern condo.	NULL	110	2024-01-12 21:00:00
13	13	Lake house.	NULL	180	2024-01-13 22:00:00
14	14	Downtown studio.	NULL	90	2024-01-14 23:00:00
15	15	Country estate.	NULL	250	2024-01-15 09:00:00
16	16	Urban loft.	NULL	130	2024-01-16 08:00:00
17	17	Luxury apartment.	NULL	220	2024-01-17 07:00:00
18	18	Secluded cabin.	NULL	95	2024-01-18 06:00:00
19	19	Elegant townhouse.	NULL	170	2024-01-19 05:00:00
20	20	Cozy guesthouse.	NULL	85	2024-01-20 04:00:00

Guest

Record guest-related details within the Airbnb platform.

Create the table:

```
USE Airbnb;

CREATE TABLE Guest (
    GuestID INT PRIMARY KEY,
    UserID INT,
    FOREIGN KEY (UserID) REFERENCES User(UserID)
);
```

Output:

GuestID	UserID
1	1
11	2
2	3
12	4
3	5
13	6
4	7
14	8
5	9
15	10
6	11
16	12
7	13
17	14
8	15
18	16
9	17
19	18
10	19
20	20

Insert elements of the table:

```
INSERT INTO Guest (GuestID, UserID) VALUES
(1, 1),
(2, 3),
(3, 5),
(4, 7),
(5, 9),
(6, 11),
(7, 13),
(8, 15),
(9, 17),
(10, 19),
(11, 2),
(12, 4),
(13, 6),
(14, 8),
(15, 10),
(16, 12),
(17, 14),
(18, 16),
(19, 18),
(20, 20);
```

Employee

Create the table:

```
USE Airbnb;

CREATE TABLE Employee (
    EmployeeID INT PRIMARY KEY,
    BirthDate DATE,
    HireDate DATE,
    Title VARCHAR(50),
    AnnualLeave INT,
    ReportsTo INT,
    FOREIGN KEY (ReportsTo) REFERENCES Employee(EmployeeID)
);
```

Insert elements of the table:

```
INSERT INTO Employee (EmployeeID, BirthDate, HireDate, Title, AnnualLeave, ReportsTo) VALUES
(1, '1980-01-01', '2010-01-01', 'Manager', 30, NULL),
(2, '1985-02-02', '2011-02-02', 'Assistant', 25, 1),
(3, '1990-03-03', '2012-03-03', 'Executive', 20, 1),
(4, '1982-04-04', '2013-04-04', 'Clerk', 15, 2),
(5, '1983-05-05', '2014-05-05', 'HR', 25, 3),
(6, '1984-06-06', '2015-06-06', 'Developer', 20, 3),
(7, '1986-07-07', '2016-07-07', 'Designer', 20, 1),
(8, '1987-08-08', '2017-08-08', 'Support', 20, 2),
(9, '1988-09-09', '2018-09-09', 'Analyst', 20, 4),
(10, '1989-10-10', '2019-10-10', 'Tester', 20, 5),
(11, '1991-11-11', '2020-11-11', 'Manager', 30, 6),
(12, '1992-12-12', '2021-12-12', 'Assistant', 25, 7),
(13, '1993-01-13', '2022-01-13', 'Executive', 20, 8),
(14, '1994-02-14', '2023-02-14', 'Clerk', 15, 9),
(15, '1995-03-15', '2024-03-15', 'HR', 25, 10),
(16, '1996-04-16', '2025-04-16', 'Developer', 20, 11),
(17, '1997-05-17', '2026-05-17', 'Designer', 20, 12),
(18, '1998-06-18', '2027-06-18', 'Support', 20, 13),
(19, '1999-07-19', '2028-07-19', 'Analyst', 20, 14),
(20, '2000-08-20', '2029-08-20', 'Tester', 20, 15);
```

Output:

EmployeeID	BirthDate	HireDate	Title	AnnualLeave	ReportsTo
1	1980-01-01	2010-01-01	Manager	30	NULL
2	1985-02-02	2011-02-02	Assistant	25	1
3	1990-03-03	2012-03-03	Executive	20	1
4	1982-04-04	2013-04-04	Clerk	15	2
5	1983-05-05	2014-05-05	HR	25	3
6	1984-06-06	2015-06-06	Developer	20	3
7	1986-07-07	2016-07-07	Designer	20	1
8	1987-08-08	2017-08-08	Support	20	2
9	1988-09-09	2018-09-09	Analyst	20	4
10	1989-10-10	2019-10-10	Tester	20	5
11	1991-11-11	2020-11-11	Manager	30	6
12	1992-12-12	2021-12-12	Assistant	25	7
13	1993-01-13	2022-01-13	Executive	20	8
14	1994-02-14	2023-02-14	Clerk	15	9
15	1995-03-15	2024-03-15	HR	25	10
16	1996-04-16	2025-04-16	Developer	20	11
17	1997-05-17	2026-05-17	Designer	20	12
18	1998-06-18	2027-06-18	Support	20	13
19	1999-07-19	2028-07-19	Analyst	20	14
20	2000-08-20	2029-08-20	Tester	20	15

Location

Create the table:

```
USE Airbnb;
```

```
CREATE TABLE Location (
    LocationID INT PRIMARY KEY,
    Country VARCHAR(50),
    Region VARCHAR(50),
    City VARCHAR(50),
    Street VARCHAR(50),
    PostCode VARCHAR(20)
);
```

Insert elements of the table:

```
INSERT INTO Location (LocationID, Country, Region, City, Street, PostCode) VALUES
(1, 'USA', 'California', 'Los Angeles', 'Main St', '90001'),
(2, 'USA', 'New York', 'New York', 'Broadway', '10001'),
(3, 'Canada', 'Ontario', 'Toronto', 'Queen St', 'M5H 2N2'),
(4, 'UK', 'England', 'London', 'High St', 'SW1A 1AA'),
(5, 'France', 'Île-de-France', 'Paris', 'Champs-Élysées', '75008'),
(6, 'Germany', 'Bavaria', 'Munich', 'Marienplatz', '80331'),
(7, 'Spain', 'Catalonia', 'Barcelona', 'La Rambla', '08002'),
(8, 'Italy', 'Lazio', 'Rome', 'Via del Corso', '00187'),
(9, 'Australia', 'New South Wales', 'Sydney', 'George St', '2000'),
(10, 'Japan', 'Tokyo', 'Shibuya', 'Center-gai', '150-0042'),
(11, 'China', 'Beijing', 'Chaoyang', 'Wangfujing', '100006'),
(12, 'India', 'Maharashtra', 'Mumbai', 'Marine Drive', '400020'),
(13, 'Brazil', 'São Paulo', 'São Paulo', 'Paulista Ave', '01310-100'),
(14, 'Mexico', 'Mexico City', 'Mexico City', 'Reforma', '06500'),
(15, 'Russia', 'Moscow', 'Moscow', 'Arbat St', '119019'),
(16, 'South Africa', 'Gauteng', 'Johannesburg', 'Soweto', '1868'),
(17, 'UAE', 'Dubai', 'Dubai', 'Sheikh Zayed Road', '00000'),
(18, 'Singapore', 'Singapore', 'Singapore', 'Orchard Road', '238841'),
(19, 'South Korea', 'Seoul', 'Seoul', 'Gangnam', '06027'),
(20, 'Sweden', 'Stockholm', 'Stockholm', 'Drottninggatan', '111 51');
```

Output:

LocationID	Country	Region	City	Street	PostCode
1	USA	California	Los Angeles	Main St	90001
2	USA	New York	New York	Broadway	10001
3	Canada	Ontario	Toronto	Queen St	M5H 2N2
4	UK	England	London	High St	SW1A 1AA
5	France	Île-de-France	Paris	Champs-Élysées	75008
6	Germany	Bavaria	Munich	Marienplatz	80331
7	Spain	Catalonia	Barcelona	La Rambla	08002
8	Italy	Lazio	Rome	Via del Corso	00187
9	Australia	New South Wales	Sydney	George St	2000
10	Japan	Tokyo	Shibuya	Center-gai	150-0042
11	China	Beijing	Chaoyang	Wangfujing	100006
12	India	Maharashtra	Mumbai	Marine Drive	400020
13	Brazil	São Paulo	São Paulo	Paulista Ave	01310-100
14	Mexico	Mexico City	Mexico City	Reforma	06500
15	Russia	Moscow	Moscow	Arbat St	119019
16	South Africa	Gauteng	Johannesburg	Soweto	1868
17	UAE	Dubai	Dubai	Sheikh Zayed Road	00000
18	Singapore	Singapore	Singapore	Orchard Road	238841
19	South Korea	Seoul	Seoul	Gangnam	06027
20	Sweden	Stockholm	Stockholm	Drottninggatan	111 51

Place

Create the table:

```
USE Airbnb;

CREATE TABLE Place (
    PlaceID INT PRIMARY KEY,
    PlaceLocationID INT,
    NumberOfRooms INT,
    PlaceSize FLOAT,
    MaxGuests INT,
    FOREIGN KEY (PlaceLocationID) REFERENCES Location(LocationID)
);
```

Insert elements of the table:

```
INSERT INTO Place (PlaceID, PlaceLocationID, NumberOfRooms, PlaceSize, MaxGuests) VALUES
(1, 1, 3, 100.0, 4),
(2, 2, 2, 80.0, 3),
(3, 3, 4, 120.0, 5),
(4, 4, 5, 150.0, 6),
(5, 5, 3, 90.0, 4),
(6, 6, 2, 70.0, 3),
(7, 7, 1, 50.0, 2),
(8, 8, 3, 100.0, 4),
(9, 9, 4, 120.0, 5),
(10, 10, 5, 150.0, 6),
(11, 11, 3, 90.0, 4),
(12, 12, 2, 70.0, 3),
(13, 13, 1, 50.0, 2),
(14, 14, 3, 100.0, 4),
(15, 15, 4, 120.0, 5),
(16, 16, 5, 150.0, 6),
(17, 17, 3, 90.0, 4),
(18, 18, 2, 70.0, 3),
(19, 19, 1, 50.0, 2),
(20, 20, 3, 100.0, 4);
```

Output:

PlaceID	PlaceLocationID	NumberOfRooms	PlaceSize	MaxGuests
1	1	3	100	4
2	2	2	80	3
3	3	4	120	5
4	4	5	150	6
5	5	3	90	4
6	6	2	70	3
7	7	1	50	2
8	8	3	100	4
9	9	4	120	5
10	10	5	150	6
11	11	3	90	4
12	12	2	70	3
13	13	1	50	2
14	14	3	100	4
15	15	4	120	5
16	16	5	150	6
17	17	3	90	4
18	18	2	70	3
19	19	1	50	2
20	20	3	100	4

Listing

Create the table:

```
USE Airbnb;

CREATE TABLE Listing (
    ListingID INT PRIMARY KEY,
    ListedPlaceID INT,
    CoverPhotoID INT,
    Description VARCHAR(255),
    FOREIGN KEY (ListedPlaceID) REFERENCES Place(PlaceID),
    FOREIGN KEY (CoverPhotoID) REFERENCES Media(MediaID)
);
```

Insert elements of the table:

```
INSERT INTO Listing (ListingID, ListedPlaceID, CoverPhotoID, Description) VALUES
(1, 1, 1, 'Beautiful city center apartment'),
(2, 2, 2, 'Modern house with garden'),
(3, 3, 3, 'Beachfront villa'),
(4, 4, 4, 'Mountain cabin'),
(5, 5, 5, 'Luxury penthouse'),
(6, 6, 6, 'Rustic cottage'),
(7, 7, 7, 'City apartment with view'),
(8, 8, 8, 'Country house'),
(9, 9, 9, 'Charming bungalow'),
(10, 10, 10, 'Spacious loft'),
(11, 11, 11, 'Historic home'),
(12, 12, 12, 'Modern condo'),
(13, 13, 13, 'Lake house'),
(14, 14, 14, 'Downtown studio'),
(15, 15, 15, 'Country estate'),
(16, 16, 16, 'Urban loft'),
(17, 17, 17, 'Luxury apartment'),
(18, 18, 18, 'Secluded cabin'),
(19, 19, 19, 'Elegant townhouse'),
(20, 20, 20, 'Cozy guesthouse');
```

Output:

ListingID	ListedPlaceID	CoverPhotoID	Description
1	1	1	Beautiful city center apartment
2	2	2	Modern house with garden
3	3	3	Beachfront villa
4	4	4	Mountain cabin
5	5	5	Luxury penthouse
6	6	6	Rustic cottage
7	7	7	City apartment with view
8	8	8	Country house
9	9	9	Charming bungalow
10	10	10	Spacious loft
11	11	11	Historic home
12	12	12	Modern condo
13	13	13	Lake house
14	14	14	Downtown studio
15	15	15	Country estate
16	16	16	Urban loft
17	17	17	Luxury apartment
18	18	18	Secluded cabin
19	19	19	Elegant townhouse
20	20	20	Cozy guesthouse

Amenities

Holds data on different amenities that can be linked to rental properties or listings within the system.

Create the table:

```
USE Airbnb;

CREATE TABLE Amenities (
    AmenityID INT PRIMARY KEY,
    Name VARCHAR(50),
    Description VARCHAR(255)
);
```

Output:

AmenityID	Name	Description
1	WiFi	Wireless Internet
2	Parking	Free Parking
3	Pool	Swimming Pool
4	Gym	Fitness Center
5	Air Conditioning	Cooling System
6	Heating	Heating System
7	Kitchen	Full Kitchen
8	Washer	Washing Machine
9	Dryer	Dryer Machine
10	TV	Television
11	Hot Tub	Hot Tub
12	Breakfast	Free Breakfast
13	Pet Friendly	Pets Allowed
14	Fireplace	Indoor Fireplace
15	Elevator	Elevator in Buildi...
16	Wheelchair Ac...	Wheelchair Access
17	Smoke Detector	Smoke Detector
18	Carbon Mono...	CO Detector
19	First Aid Kit	First Aid Kit
20	Hair Dryer	Hair Dryer

Insert elements of the table:

```
INSERT INTO Amenities (AmenityID, Name, Description) VALUES
(1, 'WiFi', 'Wireless Internet'),
(2, 'Parking', 'Free Parking'),
(3, 'Pool', 'Swimming Pool'),
(4, 'Gym', 'Fitness Center'),
(5, 'Air Conditioning', 'Cooling System'),
(6, 'Heating', 'Heating System'),
(7, 'Kitchen', 'Full Kitchen'),
(8, 'Washer', 'Washing Machine'),
(9, 'Dryer', 'Dryer Machine'),
(10, 'TV', 'Television'),
(11, 'Hot Tub', 'Hot Tub'),
(12, 'Breakfast', 'Free Breakfast'),
(13, 'Pet Friendly', 'Pets Allowed'),
(14, 'Fireplace', 'Indoor Fireplace'),
(15, 'Elevator', 'Elevator in Building'),
(16, 'Wheelchair Accessible', 'Wheelchair Access'),
(17, 'Smoke Detector', 'Smoke Detector'),
(18, 'Carbon Monoxide Detector', 'CO Detector'),
(19, 'First Aid Kit', 'First Aid Kit'),
(20, 'Hair Dryer', 'Hair Dryer');
```

AmentityListing

Create the table:

```
USE Airbnb;

CREATE TABLE AmentiyListing (
    AmentiyListingID INT PRIMARY KEY,
    PlaceID INT,
    AmentiyID INT,
    FOREIGN KEY (PlaceID) REFERENCES Place(PlaceID),
    FOREIGN KEY (AmentiyID) REFERENCES Amenities(AmentiyID)
);
```

Insert elements of the table:

```
INSERT INTO AmentiyListing (AmentiyListingID, PlaceID, AmentiyID) VALUES
(1, 1, 1),
(2, 2, 2),
(3, 3, 3),
(4, 4, 4),
(5, 5, 5),
(6, 6, 6),
(7, 7, 7),
(8, 8, 8),
(9, 9, 9),
(10, 10, 10),
(11, 11, 11),
(12, 12, 12),
(13, 13, 13),
(14, 14, 14),
(15, 15, 15),
(16, 16, 16),
(17, 17, 17),
(18, 18, 18),
(19, 19, 19),
(20, 20, 20);
```

Output:

AmentiyListingID	PlaceID	AmentiyID
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10	10	10
11	11	11
12	12	12
13	13	13
14	14	14
15	15	15
16	16	16
17	17	17
18	18	18
19	19	19
20	20	20

Income

Create the table:

```
USE Airbnb;

CREATE TABLE Income (
    IncomeID INT PRIMARY KEY,
    Year VARCHAR(4),
    Amount FLOAT,
    PlaceID INT,
    FOREIGN KEY (PlaceID) REFERENCES Place(PlaceID)
);
```

Insert elements of the table:

```
INSERT INTO Income (IncomeID, Year, Amount, PlaceID) VALUES
(1, '2024', 10000.0, 1),
(2, '2024', 15000.0, 2),
(3, '2024', 20000.0, 3),
(4, '2024', 12000.0, 4),
(5, '2024', 30000.0, 5),
(6, '2024', 8000.0, 6),
(7, '2024', 11000.0, 7),
(8, '2024', 9000.0, 8),
(9, '2024', 13000.0, 9),
(10, '2024', 14000.0, 10),
(11, '2024', 16000.0, 11),
(12, '2024', 11000.0, 12),
(13, '2024', 18000.0, 13),
(14, '2024', 9000.0, 14),
(15, '2024', 25000.0, 15),
(16, '2024', 13000.0, 16),
(17, '2024', 22000.0, 17),
(18, '2024', 9500.0, 18),
(19, '2024', 17000.0, 19),
(20, '2024', 8500.0, 20);
```

Output:

IncomeID	Year	Amount	PlaceID
1	2024	10000	1
2	2024	15000	2
3	2024	20000	3
4	2024	12000	4
5	2024	30000	5
6	2024	8000	6
7	2024	11000	7
8	2024	9000	8
9	2024	13000	9
10	2024	14000	10
11	2024	16000	11
12	2024	11000	12
13	2024	18000	13
14	2024	9000	14
15	2024	25000	15
16	2024	13000	16
17	2024	22000	17
18	2024	9500	18
19	2024	17000	19
20	2024	8500	20

Booking

Create the table:

```
USE Airbnb;
```

```
CREATE TABLE Booking (
    BookingID INT PRIMARY KEY,
    AccommodationID INT,
    GuestID INT,
    BookingDates DATETIME,
    PaymentStatus VARCHAR(50),
    TransactionDetails VARCHAR(255),
    FOREIGN KEY (AccommodationID) REFERENCES Accommodation(AccommodationID),
    FOREIGN KEY (GuestID) REFERENCES Guest(GuestID)
);
```

Insert elements of the table:

```
INSERT INTO Booking (BookingID, AccommodationID, GuestID, BookingDates, PaymentStatus, TransactionDetails) VALUES
(1, 1, 1, '2024-01-01 10:00:00', 'Completed', 'Transaction 1'),
(2, 2, 2, '2024-01-02 11:00:00', 'Pending', 'Transaction 2'),
(3, 3, 3, '2024-01-03 12:00:00', 'Completed', 'Transaction 3'),
(4, 4, 4, '2024-01-04 13:00:00', 'Canceled', 'Transaction 4'),
(5, 5, 5, '2024-01-05 14:00:00', 'Completed', 'Transaction 5'),
(6, 6, 6, '2024-01-06 15:00:00', 'Pending', 'Transaction 6'),
(7, 7, 7, '2024-01-07 16:00:00', 'Completed', 'Transaction 7'),
(8, 8, 8, '2024-01-08 17:00:00', 'Pending', 'Transaction 8'),
(9, 9, 9, '2024-01-09 18:00:00', 'Completed', 'Transaction 9'),
(10, 10, 10, '2024-01-10 19:00:00', 'Completed', 'Transaction 10'),
(11, 11, 11, '2024-01-11 20:00:00', 'Pending', 'Transaction 11'),
(12, 12, 12, '2024-01-12 21:00:00', 'Completed', 'Transaction 12'),
(13, 13, 13, '2024-01-13 22:00:00', 'Canceled', 'Transaction 13'),
(14, 14, 14, '2024-01-14 23:00:00', 'Completed', 'Transaction 14'),
(15, 15, 15, '2024-01-15 09:00:00', 'Pending', 'Transaction 15'),
(16, 16, 16, '2024-01-16 08:00:00', 'Completed', 'Transaction 16'),
(17, 17, 17, '2024-01-17 07:00:00', 'Pending', 'Transaction 17'),
(18, 18, 18, '2024-01-18 06:00:00', 'Completed', 'Transaction 18'),
(19, 19, 19, '2024-01-19 05:00:00', 'Completed', 'Transaction 19'),
(20, 20, 20, '2024-01-20 04:00:00', 'Pending', 'Transaction 20');
```

Output:

BookingID	AccommodationID	GuestID	BookingDates	PaymentStatus	TransactionDetails
1	1	1	2024-01-01 10:00:00	Completed	Transaction 1
2	2	2	2024-01-02 11:00:00	Pending	Transaction 2
3	3	3	2024-01-03 12:00:00	Completed	Transaction 3
4	4	4	2024-01-04 13:00:00	Canceled	Transaction 4
5	5	5	2024-01-05 14:00:00	Completed	Transaction 5
6	6	6	2024-01-06 15:00:00	Pending	Transaction 6
7	7	7	2024-01-07 16:00:00	Completed	Transaction 7
8	8	8	2024-01-08 17:00:00	Pending	Transaction 8
9	9	9	2024-01-09 18:00:00	Completed	Transaction 9
10	10	10	2024-01-10 19:00:00	Completed	Transaction 10
11	11	11	2024-01-11 20:00:00	Pending	Transaction 11
12	12	12	2024-01-12 21:00:00	Completed	Transaction 12
13	13	13	2024-01-13 22:00:00	Canceled	Transaction 13
14	14	14	2024-01-14 23:00:00	Completed	Transaction 14
15	15	15	2024-01-15 09:00:00	Pending	Transaction 15
16	16	16	2024-01-16 08:00:00	Completed	Transaction 16
17	17	17	2024-01-17 07:00:00	Pending	Transaction 17
18	18	18	2024-01-18 06:00:00	Completed	Transaction 18
19	19	19	2024-01-19 05:00:00	Completed	Transaction 19
20	20	20	2024-01-20 04:00:00	Pending	Transaction 20

Payment

Captures details of payments associated with reservations.

Create the table:

```
USE Airbnb;

CREATE TABLE Payment (
    PaymentID INT PRIMARY KEY,
    BookingID INT,
    PaymentMethod VARCHAR(50),
    PaymentStatus VARCHAR(50),
    FOREIGN KEY (BookingID) REFERENCES Booking(BookingID)
);
```

Insert elements of the table:

```
INSERT INTO Payment (PaymentID, BookingID, PaymentMethod, PaymentStatus) VALUES
(1, 1, 'Credit Card', 'Completed'),
(2, 2, 'PayPal', 'Pending'),
(3, 3, 'Bank Transfer', 'Completed'),
(4, 4, 'Credit Card', 'Canceled'),
(5, 5, 'PayPal', 'Completed'),
(6, 6, 'Bank Transfer', 'Pending'),
(7, 7, 'Credit Card', 'Completed'),
(8, 8, 'PayPal', 'Pending'),
(9, 9, 'Bank Transfer', 'Completed'),
(10, 10, 'Credit Card', 'Completed'),
(11, 11, 'PayPal', 'Pending'),
(12, 12, 'Bank Transfer', 'Completed'),
(13, 13, 'Credit Card', 'Canceled'),
(14, 14, 'PayPal', 'Completed'),
(15, 15, 'Bank Transfer', 'Pending'),
(16, 16, 'Credit Card', 'Completed'),
(17, 17, 'PayPal', 'Pending'),
(18, 18, 'Bank Transfer', 'Completed'),
(19, 19, 'Credit Card', 'Completed'),
(20, 20, 'PayPal', 'Pending');
```

Output:

PaymentID	BookingID	PaymentMethod	PaymentStatus
1	1	Credit Card	Completed
2	2	PayPal	Pending
3	3	Bank Transfer	Completed
4	4	Credit Card	Canceled
5	5	PayPal	Completed
6	6	Bank Transfer	Pending
7	7	Credit Card	Completed
8	8	PayPal	Pending
9	9	Bank Transfer	Completed
10	10	Credit Card	Completed
11	11	PayPal	Pending
12	12	Bank Transfer	Completed
13	13	Credit Card	Canceled
14	14	PayPal	Completed
15	15	Bank Transfer	Pending
16	16	Credit Card	Completed
17	17	PayPal	Pending
18	18	Bank Transfer	Completed
19	19	Credit Card	Completed
20	20	PayPal	Pending

Reservation

Records reservations made by guests, detailing check-in and check-out dates for specific listings.

Create the table:

```
USE Airbnb;
```

```
CREATE TABLE Reservation (
    ReservationID INT PRIMARY KEY,
    BookingDate DATETIME,
    Checkin DATETIME,
    Checkout DATETIME,
    PaymentID INT,
    FOREIGN KEY (PaymentID) REFERENCES Payment(PaymentID)
);
```

Insert elements of the table:

```
INSERT INTO Reservation (ReservationID, BookingDate, Checkin, Checkout, PaymentID) VALUES
(1, '2024-01-01 10:00:00', '2024-01-10 15:00:00', '2024-01-15 11:00:00', 1),
(2, '2024-01-02 11:00:00', '2024-01-11 16:00:00', '2024-01-16 12:00:00', 2),
(3, '2024-01-03 12:00:00', '2024-01-12 17:00:00', '2024-01-17 13:00:00', 3),
(4, '2024-01-04 13:00:00', '2024-01-13 18:00:00', '2024-01-18 14:00:00', 4),
(5, '2024-01-05 14:00:00', '2024-01-14 19:00:00', '2024-01-19 15:00:00', 5),
(6, '2024-01-06 15:00:00', '2024-01-15 20:00:00', '2024-01-20 16:00:00', 6),
(7, '2024-01-07 16:00:00', '2024-01-16 21:00:00', '2024-01-21 17:00:00', 7),
(8, '2024-01-08 17:00:00', '2024-01-17 22:00:00', '2024-01-22 18:00:00', 8),
(9, '2024-01-09 18:00:00', '2024-01-18 23:00:00', '2024-01-23 19:00:00', 9),
(10, '2024-01-10 19:00:00', '2024-01-19 00:00:00', '2024-01-24 20:00:00', 10),
(11, '2024-01-11 20:00:00', '2024-01-20 01:00:00', '2024-01-25 21:00:00', 11),
(12, '2024-01-12 21:00:00', '2024-01-21 02:00:00', '2024-01-26 22:00:00', 12),
(13, '2024-01-13 22:00:00', '2024-01-22 03:00:00', '2024-01-27 23:00:00', 13),
(14, '2024-01-14 23:00:00', '2024-01-23 04:00:00', '2024-01-28 00:00:00', 14),
(15, '2024-01-15 00:00:00', '2024-01-24 05:00:00', '2024-01-29 01:00:00', 15),
(16, '2024-01-16 01:00:00', '2024-01-25 06:00:00', '2024-01-30 02:00:00', 16),
(17, '2024-01-17 02:00:00', '2024-01-26 07:00:00', '2024-01-31 03:00:00', 17),
(18, '2024-01-18 03:00:00', '2024-01-27 08:00:00', '2024-02-01 04:00:00', 18),
(19, '2024-01-19 04:00:00', '2024-01-28 09:00:00', '2024-02-02 05:00:00', 19),
(20, '2024-01-20 05:00:00', '2024-01-29 10:00:00', '2024-02-03 06:00:00', 20);
```

Output:

ReservationID	BookingDate	Checkin	Checkout	PaymentID
1	2024-01-01 10:00:00	2024-01-10 15:00:00	2024-01-15 11:00:00	1
2	2024-01-02 11:00:00	2024-01-11 16:00:00	2024-01-16 12:00:00	2
3	2024-01-03 12:00:00	2024-01-12 17:00:00	2024-01-17 13:00:00	3
4	2024-01-04 13:00:00	2024-01-13 18:00:00	2024-01-18 14:00:00	4
5	2024-01-05 14:00:00	2024-01-14 19:00:00	2024-01-19 15:00:00	5
6	2024-01-06 15:00:00	2024-01-15 20:00:00	2024-01-20 16:00:00	6
7	2024-01-07 16:00:00	2024-01-16 21:00:00	2024-01-21 17:00:00	7
8	2024-01-08 17:00:00	2024-01-17 22:00:00	2024-01-22 18:00:00	8
9	2024-01-09 18:00:00	2024-01-18 23:00:00	2024-01-23 19:00:00	9
10	2024-01-10 19:00:00	2024-01-19 00:00:00	2024-01-24 20:00:00	10
11	2024-01-11 20:00:00	2024-01-20 01:00:00	2024-01-25 21:00:00	11
12	2024-01-12 21:00:00	2024-01-21 02:00:00	2024-01-26 22:00:00	12
13	2024-01-13 22:00:00	2024-01-22 03:00:00	2024-01-27 23:00:00	13
14	2024-01-14 23:00:00	2024-01-23 04:00:00	2024-01-28 00:00:00	14
15	2024-01-15 00:00:00	2024-01-24 05:00:00	2024-01-29 01:00:00	15
16	2024-01-16 01:00:00	2024-01-25 06:00:00	2024-01-30 02:00:00	16
17	2024-01-17 02:00:00	2024-01-26 07:00:00	2024-01-31 03:00:00	17
18	2024-01-18 03:00:00	2024-01-27 08:00:00	2024-02-01 04:00:00	18
19	2024-01-19 04:00:00	2024-01-28 09:00:00	2024-02-02 05:00:00	19
20	2024-01-20 05:00:00	2024-01-29 10:00:00	2024-02-03 06:00:00	20

Complaint

Create the table:

```
USE Airbnb;
```

```
CREATE TABLE Complaint (
    ComplaintID INT PRIMARY KEY,
    Explanation VARCHAR(255),
    Status VARCHAR(50),
    ResolutionDate DATETIME,
    RelatedGuestID INT,
    RelatedHostID INT,
    FOREIGN KEY (RelatedGuestID) REFERENCES Guest(GuestID),
    FOREIGN KEY (RelatedHostID) REFERENCES Host(HostID)
);
```

Insert elements of the table:

```
INSERT INTO Complaint (ComplaintID, Explanation, Status, ResolutionDate, RelatedGuestID, RelatedHostID) VALUES
(1, 'Noisy neighborhood', 'Resolved', '2024-01-01', 1, 2),
(2, 'Unclean room', 'Pending', NULL, 3, 4),
(3, 'Rude host', 'Resolved', '2024-01-03', 5, 6),
(4, 'Late check-in', 'Pending', NULL, 7, 8),
(5, 'No Wi-Fi', 'Resolved', '2024-01-05', 9, 10),
(6, 'Broken appliances', 'Pending', NULL, 11, 12),
(7, 'Poor service', 'Resolved', '2024-01-07', 13, 14),
(8, 'Misleading description', 'Pending', NULL, 15, 16),
(9, 'Uncomfortable bed', 'Resolved', '2024-01-09', 17, 18),
(10, 'Bad location', 'Pending', NULL, 19, 20),
(11, 'Noisy neighborhood', 'Resolved', '2024-01-01', 1, 2),
(12, 'Unclean room', 'Pending', NULL, 3, 4),
(13, 'Rude host', 'Resolved', '2024-01-03', 5, 6),
(14, 'Late check-in', 'Pending', NULL, 7, 8),
(15, 'No Wi-Fi', 'Resolved', '2024-01-05', 9, 10),
(16, 'Broken appliances', 'Pending', NULL, 11, 12),
(17, 'Poor service', 'Resolved', '2024-01-07', 13, 14),
(18, 'Misleading description', 'Pending', NULL, 15, 16),
(19, 'Uncomfortable bed', 'Resolved', '2024-01-09', 17, 18),
(20, 'Bad location', 'Pending', NULL, 19, 20);
```

Output:

ComplaintID	Explanation	Status	ResolutionDate	RelatedGuestID	RelatedHostID
1	Noisy neighborhood	Resolved	2024-01-01 00:00:00	1	2
2	Unclean room	Pending	NULL	3	4
3	Rude host	Resolved	2024-01-03 00:00:00	5	6
4	Late check-in	Pending	NULL	7	8
5	No Wi-Fi	Resolved	2024-01-05 00:00:00	9	10
6	Broken appliances	Pending	NULL	11	12
7	Poor service	Resolved	2024-01-07 00:00:00	13	14
8	Misleading description	Pending	NULL	15	16
9	Uncomfortable bed	Resolved	2024-01-09 00:00:00	17	18
10	Bad location	Pending	NULL	19	20
11	Noisy neighborhood	Resolved	2024-01-01 00:00:00	1	2
12	Unclean room	Pending	NULL	3	4
13	Rude host	Resolved	2024-01-03 00:00:00	5	6
14	Late check-in	Pending	NULL	7	8
15	No Wi-Fi	Resolved	2024-01-05 00:00:00	9	10
16	Broken appliances	Pending	NULL	11	12
17	Poor service	Resolved	2024-01-07 00:00:00	13	14
18	Misleading description	Pending	NULL	15	16
19	Uncomfortable bed	Resolved	2024-01-09 00:00:00	17	18
20	Bad location	Pending	NULL	19	20

Media

Manages different types of media content utilized within the platform.

Create the table:

```
USE Airbnb;

CREATE TABLE Media (
    MediaID INT PRIMARY KEY,
    Type VARCHAR(50),
    Media BLOB
);
```

Insert elements of the table:

```
INSERT INTO Media (MediaID, Type, Media) VALUES
(1, 'Image', 'Placeholder for image data 1'),
(2, 'Image', 'Placeholder for image data 2'),
(3, 'Image', 'Placeholder for image data 3'),
(4, 'Image', 'Placeholder for image data 4'),
(5, 'Image', 'Placeholder for image data 5'),
(6, 'Video', 'Placeholder for video data 1'),
(7, 'Video', 'Placeholder for video data 2'),
(8, 'Video', 'Placeholder for video data 3'),
(9, 'Video', 'Placeholder for video data 4'),
(10, 'Video', 'Placeholder for video data 5'),
(11, 'Audio', 'Placeholder for audio data 1'),
(12, 'Audio', 'Placeholder for audio data 2'),
(13, 'Audio', 'Placeholder for audio data 3'),
(14, 'Audio', 'Placeholder for audio data 4'),
(15, 'Audio', 'Placeholder for audio data 5'),
(16, 'Document', 'Placeholder for document data 1'),
(17, 'Document', 'Placeholder for document data 2'),
(18, 'Document', 'Placeholder for document data 3'),
(19, 'Document', 'Placeholder for document data 4'),
(20, 'Document', 'Placeholder for document data 5');
```

Output:

MediaID	Type	Media
1	Image	BLOB
2	Image	BLOB
3	Image	BLOB
4	Image	BLOB
5	Image	BLOB
6	Video	BLOB
7	Video	BLOB
8	Video	BLOB
9	Video	BLOB
10	Video	BLOB
11	Audio	BLOB
12	Audio	BLOB
13	Audio	BLOB
14	Audio	BLOB
15	Audio	BLOB
16	Document	BLOB
17	Document	BLOB
18	Document	BLOB
19	Document	BLOB
20	Document	BLOB

House Rules

Create the table:

```
USE Airbnb;

CREATE TABLE HouseRules (
    HouseRulesID INT PRIMARY KEY,
    Text TEXT
);
```

Insert elements of the table:

```
INSERT INTO HouseRules (HouseRulesID, Text) VALUES
(1, 'No smoking inside the house.'),
(2, 'No pets allowed.'),
(3, 'Quiet hours are from 10 PM to 7 AM.'),
(4, 'No parties or events without prior approval.'),
(5, 'Please take off your shoes when entering the house.'),
(6, 'Dispose of trash in the designated bins.'),
(7, 'Do not use the towels to remove makeup or clean spills.'),
(8, 'Turn off lights and appliances when not in use.'),
(9, 'No unregistered guests are allowed.'),
(10, 'Parking is only allowed in the designated area.'),
(11, 'Please respect the neighbors and keep noise to a minimum.'),
(12, 'Do not rearrange the furniture.'),
(13, 'Report any damages immediately.'),
(14, 'Clean up after yourself in the common areas.'),
(15, 'Do not leave personal belongings in the hallway.'),
(16, 'No eating or drinking in the bedrooms.'),
(17, 'Use the provided coasters to avoid damaging the furniture.'),
(18, 'Check out is by 11 AM unless otherwise arranged.'),
(19, 'Please lock all doors and windows when leaving the house.'),
(20, 'Follow all posted pool and spa rules.');
```

Output:

HouseRulesID	Text
1	No smoking inside the house.
2	No pets allowed.
3	Quiet hours are from 10 PM to 7 AM.
4	No parties or events without prior approval.
5	Please take off your shoes when entering the house.
6	Dispose of trash in the designated bins.
7	Do not use the towels to remove makeup or clean spills.
8	Turn off lights and appliances when not in use.
9	No unregistered guests are allowed.
10	Parking is only allowed in the designated area.
11	Please respect the neighbors and keep noise to a minimum.
12	Do not rearrange the furniture.
13	Report any damages immediately.
14	Clean up after yourself in the common areas.
15	Do not leave personal belongings in the hallway.
16	No eating or drinking in the bedrooms.
17	Use the provided coasters to avoid damaging the furniture.
18	Check out is by 11 AM unless otherwise arranged.
19	Please lock all doors and windows when leaving the house.
20	Follow all posted pool and spa rules.

Test Case 1: Retrieving Booking Details with Linked Accommodation, Host, and Guest Info

Goal: The aim is to pull together information about bookings, including details about the accommodations, hosts, and guests involved.

Query:

```
SELECT
    Booking.BookingID,
    Booking.BookingDates,
    Booking.PaymentStatus,
    Booking.TransactionDetails,
    Accommodation.Description AS AccommodationDescription,
    Accommodation.Pricing AS AccommodationPricing,
    Host.HostID,
    Guest.GuestID,
    User.Name AS GuestName
FROM
    Booking
JOIN
    Accommodation ON Booking.AccommodationID = Accommodation.AccommodationID
JOIN
    Host ON Accommodation.HostID = Host.HostID
JOIN
    Guest ON Booking.GuestID = Guest.GuestID
JOIN
    User ON Guest.UserID = User.UserID
WHERE
    Booking.PaymentStatus = 'Completed';
```

Output:

BookingID	BookingDates	PaymentStatus	TransactionDetails	AccommodationDescription	AccommodationPricing	HostID	GuestID	GuestName
1	2024-01-01 10:00:00	Completed	Transaction 1	Cozy apartment in city center.	100	1	1	John Doe
3	2024-01-03 12:00:00	Completed	Transaction 3	Beachfront villa.	200	3	3	Michael Wilson
5	2024-01-05 14:00:00	Completed	Transaction 5	Luxury penthouse.	300	5	5	James Martinez
7	2024-01-07 16:00:00	Completed	Transaction 7	City apartment with view.	110	7	7	Charles Gonzalez
9	2024-01-09 18:00:00	Completed	Transaction 9	Charming bungalow.	130	9	9	Christopher Lee
10	2024-01-10 19:00:00	Completed	Transaction 10	Spacious loft.	140	10	10	Daniel Moore
12	2024-01-12 21:00:00	Completed	Transaction 12	Modern condo.	110	12	12	Emily Davis
14	2024-01-14 23:00:00	Completed	Transaction 14	Downtown studio.	90	14	14	Laura Garcia
16	2024-01-16 08:00:00	Completed	Transaction 16	Urban loft.	130	16	16	Patricia Lopez
18	2024-01-18 06:00:00	Completed	Transaction 18	Secluded cabin.	95	18	18	Jessica Martinez
19	2024-01-19 05:00:00	Completed	Transaction 19	Elegant townhouse.	170	19	19	Karen Taylor

Expected Result: The query should list all bookings that have a "Completed" payment status. It will include booking specifics, descriptions of the accommodations, pricing details, and the IDs of the associated host and guest. Additionally, the query will fetch the guest's name from the User table.

Test Case 2: Gathering Details on Amenities, Listings, and Income

Goal: The purpose here is to collect information about property listings, the amenities they offer, and the income generated by each listing.

Query:

```
SELECT
    Listing.ListingID,
    Listing.Description AS ListingDescription,
    Amenities.Name AS AmenityName,
    Amenities.Description AS AmenityDescription,
    Income.Amount AS IncomeAmount,
    Income.Year AS IncomeYear
FROM
    Listing
JOIN
    AmenityListing ON Listing.ListedPlaceID = AmenityListing.PlaceID
JOIN
    Amenities ON AmenityListing.AmenityID = Amenities.AmenityID
JOIN
    Income ON Listing.ListedPlaceID = Income.PlaceID
WHERE
    Income.Year = '2024';
```

Output:

ListingID	ListingDescription	AmenityName	AmenityDescription	IncomeAmount	IncomeYear
1	Beautiful city center apartment	WiFi	Wireless Internet	10000	2024
2	Modern house with garden	Parking	Free Parking	15000	2024
3	Beachfront villa	Pool	Swimming Pool	20000	2024
4	Mountain cabin	Gym	Fitness Center	12000	2024
5	Luxury penthouse	Air Conditioning	Cooling System	30000	2024
6	Rustic cottage	Heating	Heating System	8000	2024
7	City apartment with view	Kitchen	Full Kitchen	11000	2024
8	Country house	Washer	Washing Machine	9000	2024
9	Charming bungalow	Dryer	Dryer Machine	13000	2024
10	Spacious loft	TV	Television	14000	2024
11	Historic home	Hot Tub	Hot Tub	16000	2024
12	Modern condo	Breakfast	Free Breakfast	11000	2024
13	Lake house	Pet Friendly	Pets Allowed	18000	2024
14	Downtown studio	Fireplace	Indoor Fireplace	9000	2024
15	Country estate	Elevator	Elevator in Building	25000	2024
16	Urban loft	Wheelchair Accessible	Wheelchair Access	13000	2024
17	Luxury apartment	Smoke Detector	Smoke Detector	22000	2024
18	Secluded cabin	Carbon Monoxide Detector	CO Detector	9500	2024
19	Elegant townhouse	First Aid Kit	First Aid Kit	17000	2024
20	Cozy guesthouse	Hair Dryer	Hair Dryer	8500	2024

Expected Result: The query should provide details for each listing, including a description of the property, the amenities available, and the income earned in 2024. The output will show the ListingID, the property description, the name and description of each amenity, and the income details (amount and year).

Thank You!

