Conception Phase

1. What roles (person/user groups) are there?
   Host, guest, admin
2. What actions do these roles perform?
   Host: account management (create/update/delete), property management (add/update/delete property), booking management, monitor cancellations and disputes
   Guest: account management (create/update/delete), search and browse (by property/location/type/price/amenities/discounts), booking a property, adding/removing a property to/from the wishlist, update/delete/add payment details, submit reviews, raise/track disputes
   Admin: user management, monitor user verification statuses, transaction management, property management, solve disputes
3. Which data and functions are required?

Data dictionary

Attribute	**Data Type**	**Description**
Host_id	INT (Primary Key)	Unique identifier for the host
Guest_id	INT (Primary Key)	Unique identifier for the guest
full_name	VARCHAR (255)	Full name of the host/guest/admin
email	VARCHAR (255)	Email address for communication
phone	VARCHAR (15)	Contact phone number
address	TEXT	Physical address
rating	FLOAT	Rating score for a host or guest
profile_picture	VARCHAR (255)	URL of the profile picture
Verification_id	INT (Primary Key)	Unique identifier for a verification entry
email_verified	BOOLEAN	Indicates if the email is verified (true/false)
phone_verified	BOOLEAN	Indicates if the phone is verified (true/false)
id_verified	BOOLEAN	Indicates if the ID is verified (true/false)
verification_date	TIMESTAMP	Date and time of verification
Payment_details_id	INT (Primary Key)	Unique identifier for payment details
payment_method	VARCHAR (255)	Method of payment (e.g., credit card, PayPal)
account_number	VARCHAR (255)	Account or card number
expiry_date	DATE	Expiry date of the payment method
Admin_id	INT (Primary Key)	Unique identifier for the admin
admin_role	VARCHAR (255)	Role of the admin (e.g., manager, support)
Location_id	INT (Primary Key)	Unique identifier for the location

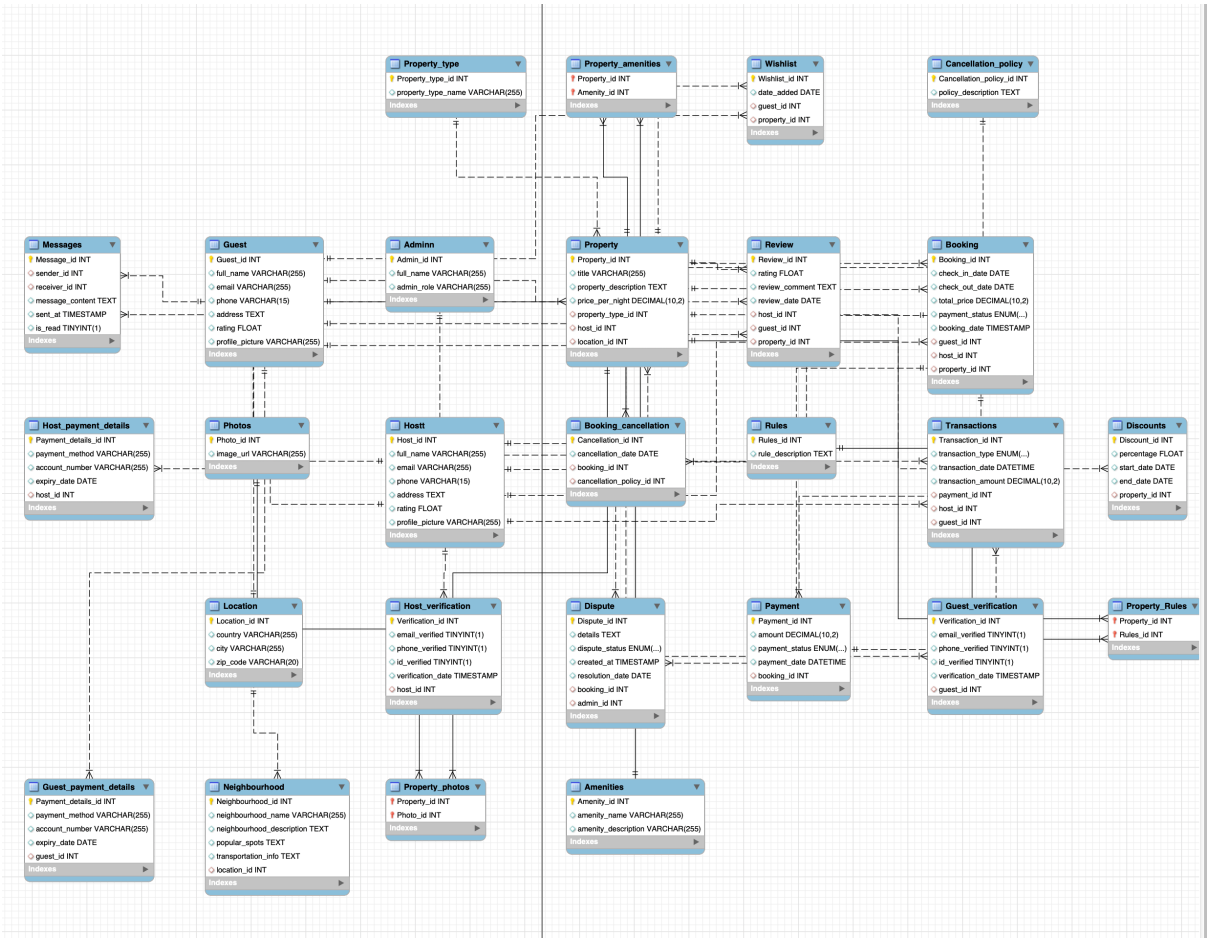
Conception Phase

country	VARCHAR (255)	Country name
city	VARCHAR (255)	City name
zip_code	VARCHAR (20)	ZIP or postal code
Property_type_id	INT (Primary Key)	Unique identifier for the property type
property_type_name	VARCHAR (255)	Name of the property type (e.g., apartment, house)
Property_id	INT (Primary Key)	Unique identifier for the property
title	VARCHAR (255)	Title of the property listing
property_description	TEXT	Detailed description of the property
price_per_night	DECIMAL (10, 2)	Cost of staying per night
Amenity_id	INT (Primary Key)	Unique identifier for an amenity
amenity_name	VARCHAR (255)	Name of the amenity (e.g., Wi-Fi, parking)
amenity_description	VARCHAR (255)	Description of the amenity
Photo_id	INT (Primary Key)	Unique identifier for a photo
image_url	VARCHAR (255)	URL of the property photo
Booking_id	INT (Primary Key)	Unique identifier for a booking
check_in_date	DATE	Check-in date for the booking
check_out_date	DATE	Check-out date for the booking
total_price	DECIMAL (10, 2)	Total price for the booking
payment_status	ENUM	Status of the payment (e.g., Paid, Pending)
booking_date	TIMESTAMP	Date and time the booking was made
Cancellation_policy_id	INT (Primary Key)	Unique identifier for a cancellation policy
policy_description	TEXT	Description of the cancellation policy
Cancellation_id	INT (Primary Key)	Unique identifier for a booking cancellation
Cancellation_date	DATE	Cancellation date of a booking
Payment_id	INT (Primary Key)	Unique identifier for a payment
amount	DECIMAL (10, 2)	Amount paid
payment_date	DATE	Date of payment
Transaction_id	INT (Primary Key)	Unique identifier for a transaction
transaction_type	ENUM	Type of transaction (Credit/Debit/PayPal)
transaction_date	DATE	Date of the transaction
transaction_amount	DECIMAL (10, 2)	Amount involved in the transaction
Review_id	INT (Primary Key)	Unique identifier for a review
review_comment	TEXT	Comments left in a review
Review_date	DATE	Date when the review was written
Wishlist_id	INT (Primary Key)	Unique identifier for a Wishlist entry

Date_added	DATE	Date when the property was added to the wishlist
Discount_id	INT (Primary Key)	Unique identifier for a discount entry
percentage	FLOAT	Discount percentage
Start_date	DATE	Date the discount starts
End_date	DATE	Date the discount ends
Rules_id	INT (Primary Key)	Unique identifier for a rule
rule_description	TEXT	Description of a rule
Neighbourhood_id	INT (Primary Key)	Unique identifier for a neighbourhood
neighbourhood_name	VARCHAR (255)	Name of the neighbourhood
Descriptionn	TEXT	A detailed description of about the neighbourhood's characteristics
Popular_spots	TEXT	List of popular attractions (e.g., landmarks, parks, restaurants or other points of interest)
Transportation_info	TEXT	Information about the available transportation options in the neighbourhood (e.g., bus stops, metro stations)
Dispute_id	INT (Primary Key)	Unique identifier for a dispute
details	TEXT	Details of the dispute
dispute_status	ENUM	Current status of a dispute (e.g., Open, Resolved)
Created_at	TIMESTAMP	Date and time the dispute was created
Message_id	INT (Primary Key)	Unique identifier for a message
Resolution_date	DATE	Date the dispute was solved
Sent_at	TIMESTAMP	Date and time the message was sent
Sender_id	INT	Identifier for a sender (guest)
Receiver_idc	INT	Identifier for a receiver (host)
message_content	TEXT	Content of the message sent between users
is_read	BOOLEAN	Indicates if a message has been read (true/false)
Wishlist_id	INT (Primary Key)	Unique identifier for a Wishlist entry

Conception Phase

Entity Relationship Model (ERM):



The project focuses on creating a well-designed database for a property rental platform like Airbnb, that connects guests and hosts. The platform allows users to list, book and review properties while including features such as property management, booking systems, secure payment processing, dispute handling and user verification.

The main challenge was to design an efficient and scalable database that can support all the requirements of the platform. This involved tackling issues like maintaining data integrity, avoiding redundance and managing complex relationships between entities.

To solve this, the database was structured using relational design principles and normalization. Key tables, such as hostt, guest, property, booking and payment form the foundation, while additional tables handle reviews, transactions, amenities and other features. Careful attention was given to enforcing constraints and enabling cascading updates or deletions to maintain data accuracy. The system also supports features like user verification, wishlist functionality, property rules and discounts.

DEVELOPMENT & REFLECTION PHASE

Anel Naukan, 102207227

January 15, 2025

Hostt table

SQL code

```sql
-- Hostt table stores host information
CREATE TABLE Hostt (
    Host_id INT AUTO_INCREMENT Primary KEY, -- Unique identifier for the host
    full_name VARCHAR(255), -- Full name of the host
    email VARCHAR(255), -- Email address for communication
    phone VARCHAR(15), -- Contact phone number
    address TEXT, -- Physical address
    rating FLOAT, -- Rating score for a host
    profile_picture VARCHAR(255) -- URL of the profile picture
);
```

20 Entries

```sql
INSERT INTO Hostt (full_name, email, phone, address, rating, profile_picture)
VALUES
('John Doe', 'johndoe@example.com', '1234567890', '123 Elm Street, Springfield', 3.5, 'profile1.jpg'),
('Jane Smith', 'janesmith@example.com', '9876543210', '456 Oak Avenue, Metropolis', 4.6, 'profile2.jpg'),
('Emily Johnson', 'emilyj@example.com', '5678901234', '789 Pine Road, Gotham', 4.9, 'profile3.jpg'),
('Daniel Smith', 'daniels@example.com', '6789012345', '456 Maple Street, Metropolis', 3.7, 'profile4.jpg'),
('Sophia Davis', 'sophiad@example.com', '7890123456', '123 Oak Lane, Star City', 4.2, 'profile5.jpg'),
('Liam Brown', 'liamb@example.com', '8901234567', '987 Elm Street, Springfield', 4.1, 'profile6.jpg'),
('Olivia Wilson', 'oliviaw@example.com', '9012345678', '654 Cedar Avenue, Smallville', 3.9, 'profile7.jpg'),
('James Martinez', 'jamesm@example.com', '1234567890', '321 Birch Road, Riverdale', 4.6, 'profile8.jpg'),
('Ava Taylor', 'avat@example.com', '2345678901', '258 Fir Boulevard, Hill Valley', 4.8, 'profile9.jpg'),
('Benjamin White', 'benjaminw@example.com', '3456789012', '369 Spruce Drive, Mystic Falls', 3.8, 'profile10.jpg'),
('Mia Anderson', 'miaa@example.com', '4567890123', '741 Redwood Lane, Raccoon City', 2.9, 'profile11.jpg'),
('Ethan Clark', 'ethanc@example.com', '5678901234', '852 Willow Road, Silent Hill', 2.5, 'profile12.jpg'),
('Charlotte Moore', 'charlottem@example.com', '6789012345', '963 Ash Crescent, Rosewood', 4.1, 'profile13.jpg'),
('Lucas Harris', 'lucash@example.com', '7890123456', '159 Cherry Court, Sunnydale', 4.3, 'profile14.jpg'),
('Grace Lee', 'gracel@example.com', '8901234567', '357 Beech Avenue, Pallet Town', 3.2, 'profile15.jpg'),
('Henry Adams', 'henrya@example.com', '9012345678', '951 Dogwood Street, Emerald City', 2.8, 'profile16.jpg'),
('Isabella Murphy', 'isabellam@example.com', '1234567890', '753 Magnolia Way, Stars Hollow', 4.5, 'profile17.jpg'),
('William Scott', 'williams@example.com', '2345678901', '147 Cypress Boulevard, Amityville', 3.1, 'profile18.jpg'),
('Amelia Robinson', 'ameliar@example.com', '3456789012', '258 Fir Road, Hawkins', 4.7, 'profile19.jpg'),
('Sara White', 'sara.white@example.com', '3456789012', '999 Maple Street, Star City', 3.5, 'profile20.jpg');
```

Host_id	full_name	email	phone	address	rating	profile_pict...
1	John Doe	johndoe@example.com	1234567890	123 Elm Street, Springfield	3.5	profile1.jpg
2	Jane Smith	janesmith@example.com	9876543210	456 Oak Avenue, Metropolis	4.6	profile2.jpg
3	Emily Johnson	emilyj@example.com	5678901234	789 Pine Road, Gotham	4.9	profile3.jpg
4	Daniel Smith	daniels@example.com	6789012345	456 Maple Street, Metropolis	3.7	profile4.jpg
5	Sophia Davis	sophiad@example.com	7890123456	123 Oak Lane, Star City	4.2	profile5.jpg
6	Liam Brown	liamb@example.com	8901234567	987 Elm Street, Springfield	4.1	profile6.jpg
7	Olivia Wilson	oliviaw@example.com	9012345678	654 Cedar Avenue, Smallville	3.9	profile7.jpg
8	James Martinez	jamesm@example.com	1234567890	321 Birch Road, Riverdale	4.6	profile8.jpg
9	Ava Taylor	avat@example.com	2345678901	258 Fir Boulevard, Hill Valley	4.8	profile9.jpg
10	Benjamin White	benjaminw@example.com	3456789012	369 Spruce Drive, Mystic Falls	3.8	profile10.jpg
11	Mia Anderson	miaa@example.com	4567890123	741 Redwood Lane, Raccoo...	2.9	profile11.jpg
12	Ethan Clark	ethanc@example.com	5678901234	852 Willow Road, Silent Hill	2.5	profile12.jpg
13	Charlotte Moore	charlottem@example.com	6789012345	963 Ash Crescent, Rosewood	4.1	profile13.jpg
14	Lucas Harris	lucash@example.com	7890123456	159 Cherry Court, Sunnydale	4.3	profile14.jpg
15	Grace Lee	gracel@example.com	8901234567	357 Beech Avenue, Pallet To...	3.2	profile15.jpg
16	Henry Adams	henrya@example.com	9012345678	951 Dogwood Street, Emeral...	2.8	profile16.jpg
17	Isabella Murphy	isabellam@example.com	1234567890	753 Magnolia Way, Stars Hol...	4.5	profile17.jpg
18	William Scott	williams@example.com	2345678901	147 Cypress Boulevard, Amit...	3.1	profile18.jpg
19	Amelia Robins...	ameliar@example.com	3456789012	258 Fir Road, Hawkins	4.7	profile19.jpg
20	Sara White	sara.white@example.com	3456789012	999 Maple Street, Star City	3.5	profile20.jpg
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Development & Reflection Phase

Host_verification table

SQL code

```sql
-- Host_verification table stores information about host profile verification
CREATE TABLE Host_verification (
    Verification_id INT AUTO_INCREMENT PRIMARY KEY, -- Unique identifier for a verification entry
    email_verified BOOLEAN, -- Indicates if the email is verified (true/false)
    phone_verified BOOLEAN, -- Indicates if the phone is verified (true/false)
    id_verified BOOLEAN, -- Indicates if the ID is verified (true/false)
    verification_date TIMESTAMP, -- Date and time of verification
    host_id INT, -- Unique identifier for the host
    FOREIGN KEY (host_id) REFERENCES Hostt(Host_id) -- Foreign key referencing Hostt table
);
```

20 Entries

```sql
INSERT INTO Host_verification (email_verified, phone_verified, id_verified, verification_date, host_id)
VALUES
(TRUE, TRUE, TRUE, '2023-01-15 12:00:00', 1),
(TRUE, FALSE, TRUE, '2023-02-20 15:30:00', 2),
(FALSE, TRUE, FALSE, '2023-03-05 09:00:00', 3),
(TRUE, FALSE, TRUE, '2023-09-16 12:47:06', 4),
(FALSE, TRUE, TRUE, '2023-04-13 02:47:06', 5),
(TRUE, FALSE, TRUE, '2024-01-17 14:45:36', 6),
(FALSE, FALSE, TRUE, '2021-11-15 10:47:06', 7),
(TRUE, TRUE, TRUE, '2019-01-15 12:00:56', 8),
(TRUE, FALSE, FALSE, '2022-11-23 13:27:09', 9),
(TRUE, TRUE, FALSE, '2023-12-01 17:47:06', 10),
(TRUE, FALSE, FALSE, '2015-11-15 10:19:04', 11),
(TRUE, TRUE, TRUE, '2023-01-15 09:47:06', 12),
(TRUE, FALSE, TRUE, '2021-04-23 06:17:33', 13),
(TRUE, TRUE, TRUE, '2023-11-14 10:43:36', 14),
(FALSE, TRUE, TRUE, '2024-06-03 08:32:18', 15),
(FALSE, TRUE, FALSE, '2025-01-03 13:55:31', 16),
(TRUE, FALSE, TRUE, '2022-08-14 18:23:53', 17),
(TRUE, TRUE, FALSE, '2014-12-03 23:12:10', 18),
(FALSE, TRUE, TRUE, '2022-09-24 22:52:48', 19),
(TRUE, TRUE, TRUE, '2023-12-01 18:45:00', 20);
```

Verification...	email_verifi...	phone_verifi...	id_verified	verification_date	host_id
1	1	1	1	2023-01-15 12:00:00	1
2	1	0	1	2023-02-20 15:30:00	2
3	0	1	0	2023-03-05 09:00:00	3
4	1	0	1	2023-09-16 12:47:06	4
5	0	1	1	2023-04-13 02:47:06	5
6	1	0	1	2024-01-17 14:45:36	6
7	0	0	1	2021-11-15 10:47:06	7
8	1	1	1	2019-01-15 12:00:56	8
9	1	0	0	2022-11-23 13:27:09	9
10	1	1	0	2023-12-01 17:47:06	10
11	1	0	0	2015-11-15 10:19:04	11
12	1	1	1	2023-01-15 09:47:06	12
13	1	0	1	2021-04-23 06:17:33	13
14	1	1	1	2023-11-14 10:43:36	14
15	0	1	1	2024-06-03 08:32:18	15
16	0	1	0	2025-01-03 13:55:31	16
17	1	0	1	2022-08-14 18:23:53	17
18	1	1	0	2014-12-03 23:12:10	18
19	0	1	1	2022-09-24 22:52:48	19
20	1	1	1	2023-12-01 18:45:00	20
NULL	NULL	NULL	NULL	NULL	NULL

Host_payment_details table

SQL code

```sql
-- Host_payment_details stores information about payment details of host
CREATE TABLE Host_payment_details (
    Payment_details_id INT AUTO_INCREMENT PRIMARY KEY, -- Unique identifier for payment details
    payment_method VARCHAR(255), -- Method of payment (e.g., credit card, PayPal)
    account_number VARCHAR(255), -- Account or card number
    expiry_date DATE, -- Expiry date of the payment method
    host_id INT, -- Unique identifier for the host
    FOREIGN KEY (host_id) REFERENCES Hostt(Host_id) -- Foreign key referencing Hostt table
);
```

20 Entries

```sql
INSERT INTO Host_payment_details (payment_method, account_number, expiry_date, host_id)
VALUES
('Credit Card', '4111111111111111', '2025-12-31', 1),
('Bank Transfer', 'IBAN1234567890', '2026-06-30', 2),
('PayPal', 'emily.johnson@paypal.com', '2025-03-31', 3),
('Credit Card', '1891111111411111', '2026-11-21', 4),
('PayPal', 'sophiad@paypal.com', '2025-10-17', 5),
('Credit Card', '1591116111416754', '2028-12-29', 6),
('Credit Card', '2341317111411111', '2026-01-21', 7),
('Bank Transfer', 'IBAN3456789012', '2027-07-28', 8),
('Bank Transfer', 'IBAN1254567891', '2025-01-30', 9),
('PayPal', 'benjaminw@paypal.com', '2026-03-25', 10),
('PayPal', 'miaa@paypal.com', '2025-12-13', 11),
('Credit Card', '6591115431416754', '2028-06-19', 12),
('PayPal', 'charlottem@paypal.com', '2029-03-23', 13),
('PayPal', 'lucash@paypal.com', '2026-12-07', 14),
('Bank Transfer', 'IBAN9700567890', '2025-06-30', 15),
('Bank Transfer', 'IBAN4533897135', '2026-08-24', 16),
('Credit Card', '6591465131416754', '2028-04-04', 17),
('PayPal', 'williams@paypal.com', '2027-10-07', 18),
('PayPal', 'ameliar@paypal.com', '2025-10-23', 19),
('Credit Card', '5555555555554444', '2027-05-15', 20);
```

Payment_details_...	payment_meth...	account_number	expiry_date	host_id
1	Credit Card	4111111111111111	2025-12-31	1
2	Bank Transfer	IBAN1234567890	2026-06-30	2
3	PayPal	emily.johnson@paypal.com	2025-03-31	3
4	Credit Card	1891111111411111	2026-11-21	4
5	PayPal	sophiad@paypal.com	2025-10-17	5
6	Credit Card	1591116111416754	2028-12-29	6
7	Credit Card	2341317111411111	2026-01-21	7
8	Bank Transfer	IBAN3456789012	2027-07-28	8
9	Bank Transfer	IBAN1254567891	2025-01-30	9
10	PayPal	benjaminw@paypal.com	2026-03-25	10
11	PayPal	miaa@paypal.com	2025-12-13	11
12	Credit Card	6591115431416754	2028-06-19	12
13	PayPal	charlottem@paypal.com	2029-03-23	13
14	PayPal	lucash@paypal.com	2026-12-07	14
15	Bank Transfer	IBAN9700567890	2025-06-30	15
16	Bank Transfer	IBAN4533897135	2026-08-24	16
17	Credit Card	6591465131416754	2028-04-04	17
18	PayPal	williams@paypal.com	2027-10-07	18
19	PayPal	ameliar@paypal.com	2025-10-23	19
20	Credit Card	5555555555554444	2027-05-15	20
NULL	NULL	NULL	NULL	NULL

Development & Reflection Phase

Guest table

SQL code

```sql
-- Guest table stores guest information
CREATE TABLE Guest (
    Guest_id INT AUTO_INCREMENT PRIMARY KEY, -- Unique identifier for the guest
    full_name VARCHAR(255), -- Full name of the guest
    email VARCHAR(255), -- Email address for communication
    phone VARCHAR(15), -- Contact phone number
    address TEXT, -- Physical address
    rating FLOAT, -- Rating sscore for a guest
    profile_picture VARCHAR(255) -- URL of the profile picture
);
```

20 Entries

```sql
INSERT INTO Guest (full_name, email, phone, address, rating, profile_picture)
VALUES
('Alice Brown', 'alice.brown@example.com', '1112223333', '123 Birch Road, Smallville', 3.9, 'guest1.jpg'),
('Bob Carter', 'bob.carter@example.com', '4445478666', '456 Cedar Avenue, Hill Valley', 2.7, 'guest2.jpg'),
('Cathy Davis', 'cathy.davis@example.com', '7778889999', '789 Walnut Street, Riverdale', 4.5, 'guest3.jpg'),
('Emily Carter', 'emily.carter@example.com', '11122267433', '123 Birch Road, Smallville', 4.1, 'guest4.jpg'),
('James Anderson', 'james.anderson@example.com', '2233734744', '456 Oak Street, Metropolis', 4.6, 'guest5.jpg'),
('Olivia Bennett', 'olivia.bennett@example.com', '2336445755', '789 Pine Avenue, Star City', 3.8, 'guest6.jpg'),
('Michael Harris', 'michael.harris@example.com', '9445556866', '321 Maple Lane, Riverdale', 4.6, 'guest7.jpg'),
('Sophia Taylor', 'sophia.taylor@example.com', '9956667777', '654 Cedar Drive, Springfield', 4.4, 'guest8.jpg'),
('William Rickson', 'william.rickson@example.com', '6267778888', '987 Elm Road, Gotham', 4.3, 'guest9.jpg'),
('Isabella Walker', 'isabella.walker@example.com', '3478889999', '147 Spruce Avenue, Hill Valley', 3.7, 'guest10.jpg'),
('Charlotte Murphy', 'charlotte.murphy@example.com', '9812223334', '852 Willow Lane, Pallet Town', 4.7, 'guest11.jpg'),
('Lucas Moore', 'lucas.moore@example.com', '2123334445', '963 Ash Way, Mystic Falls', 3.9, 'guest12.jpg'),
('Ava Clark', 'ava.clark@example.com', '1334445556', '159 Cherry Crescent, Rosewood', 4.6, 'guest13.jpg'),
('Henry Lewis', 'henry.lewis@example.com', '4874555667', '357 Beech Court, Sunnydale', 4.7, 'guest14.jpg'),
('Mia Young', 'mia.young@example.com', '9156667778', '951 Dogwood Avenue, Hawkins', 4.8, 'guest15.jpg'),
('Daniel Hall', 'daniel.hall@example.com', '7767778889', '753 Magnolia Street, Stars Hollow', 4.1, 'guest16.jpg'),
('Grace White', 'grace.white@example.com', '2178849993', '159 Cypress Road, Amityville', 4.4, 'guest17.jpg'),
('Benjamin Lee', 'benjamin.lee@example.com', '3685990000', '258 Fir Street, Emerald City', 4.5, 'guest18.jpg'),
('Amelia Thomas', 'amelia.thomas@example.com', '1390501711', '369 Walnut Boulevard, Raccoon City', 4.6, 'guest19.jpg'),
('Zoe Quinn', 'zoe.quinn@example.com', '1233211234', '123 Main Street, Wonderland', 4.8, 'guest20.jpg');
```

Guest_id	full_name	email	phone	address	rating	profile_pict…
1	Alice Brown	alice.brown@example.com	1112223333	123 Birch Road, Smallville	3.9	guest1.jpg
2	Bob Carter	bob.carter@example.com	4445478666	456 Cedar Avenue, Hill Valley	2.7	guest2.jpg
3	Cathy Davis	cathy.davis@example.com	7778889999	789 Walnut Street, Riverdale	4.5	guest3.jpg
4	Emily Carter	emily.carter@example.com	11122267433	123 Birch Road, Smallville	4.1	guest4.jpg
5	James Anderson	james.anderson@example.com	2233734744	456 Oak Street, Metropolis	4.6	guest5.jpg
6	Olivia Bennett	olivia.bennett@example.com	2336445755	789 Pine Avenue, Star City	3.8	guest6.jpg
7	Michael Harris	michael.harris@example.com	9445556866	321 Maple Lane, Riverdale	4.6	guest7.jpg
8	Sophia Taylor	sophia.taylor@example.com	9956667777	654 Cedar Drive, Springfield	4.4	guest8.jpg
9	William Rickson	william.rickson@example.com	6267778888	987 Elm Road, Gotham	4.3	guest9.jpg
10	Isabella Walker	isabella.walker@example.com	3478889999	147 Spruce Avenue, Hill Valley	3.7	guest10.jpg
11	Charlotte Murphy	charlotte.murphy@example.com	9812223334	852 Willow Lane, Pallet Town	4.7	guest11.jpg
12	Lucas Moore	lucas.moore@example.com	2123334445	963 Ash Way, Mystic Falls	3.9	guest12.jpg
13	Ava Clark	ava.clark@example.com	1334445556	159 Cherry Crescent, Rosew…	4.6	guest13.jpg
14	Henry Lewis	henry.lewis@example.com	4874555667	357 Beech Court, Sunnydale	4.7	guest14.jpg
15	Mia Young	mia.young@example.com	9156667778	951 Dogwood Avenue, Hawki…	4.8	guest15.jpg
16	Daniel Hall	daniel.hall@example.com	7767778889	753 Magnolia Street, Stars H…	4.1	guest16.jpg
17	Grace White	grace.white@example.com	2178849993	159 Cypress Road, Amityville	4.4	guest17.jpg
18	Benjamin Lee	benjamin.lee@example.com	3685990000	258 Fir Street, Emerald City	4.5	guest18.jpg
19	Amelia Thomas	amelia.thomas@example.com	1390501711	369 Walnut Boulevard, Racc…	4.6	guest19.jpg
20	Zoe Quinn	zoe.quinn@example.com	1233211234	123 Main Street, Wonderland	4.8	guest20.jpg
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Guest_verification table

SQL code

```sql
-- Guest_verification table stores information about guest profile verification
CREATE TABLE Guest_verification (
    Verification_id INT AUTO_INCREMENT PRIMARY KEY, -- Unique identifier for a verification entry
    email_verified BOOLEAN, -- Indicates if the email is verified (true/false)
    phone_verified BOOLEAN, -- Indicates if the phone is verified (true/false)
    id_verified BOOLEAN, -- Indicates if the ID is verified (true/false)
    verification_date TIMESTAMP, -- Date and time of verification
    guest_id INT, -- Unique identifier for the guest
    FOREIGN KEY (guest_id) REFERENCES Guest(Guest_id) -- Foreign key referencing Guest table
);
```

20 Entries

```sql
INSERT INTO Guest_verification (email_verified, phone_verified, id_verified, verification_date, guest_id)
VALUES
(TRUE, FALSE, TRUE, '2021-01-15 12:00:00', 1),
(TRUE, FALSE, FALSE, '2022-12-20 15:30:00', 2),
(FALSE, TRUE, TRUE, '2012-03-05 09:10:07', 3),
(TRUE, FALSE, FALSE, '2013-11-16 12:47:06', 4),
(TRUE, TRUE, TRUE, '2023-04-13 07:47:06', 5),
(TRUE, FALSE, TRUE, '2021-01-17 15:45:36', 6),
(FALSE, FALSE, TRUE, '2021-11-15 11:47:06', 7),
(TRUE, TRUE, TRUE, '2019-01-15 12:10:56', 8),
(TRUE, FALSE, FALSE, '2022-11-23 13:27:09', 9),
(FALSE, TRUE, TRUE, '2014-12-01 17:47:06', 10),
(TRUE, TRUE, FALSE, '2015-11-15 11:19:14', 11),
(TRUE, TRUE, TRUE, '2023-01-15 09:47:26', 12),
(TRUE, FALSE, TRUE, '2021-04-23 16:11:33', 13),
(TRUE, FALSE, TRUE, '2017-01-04 10:42:36', 14),
(FALSE, TRUE, TRUE, '2022-06-03 08:42:18', 15),
(FALSE, TRUE, FALSE, '2020-01-03 13:55:31', 16),
(TRUE, FALSE, TRUE, '2012-08-14 18:18:18', 17),
(FALSE, TRUE, FALSE, '2014-12-03 17:17:17', 18),
(FALSE, TRUE, TRUE, '2010-09-24 22:52:48', 19),
(TRUE, TRUE, TRUE, '2021-12-01 18:45:00', 20);
```

Verification…	email_verifi…	phone_verifi…	id_verified	verification_date	guest_id
1	1	0	1	2021-01-15 12:00:00	1
2	1	0	0	2022-12-20 15:30:00	2
3	0	1	1	2012-03-05 09:10:07	3
4	1	0	0	2013-11-16 12:47:06	4
5	1	1	1	2023-04-13 07:47:06	5
6	1	0	1	2021-01-17 15:45:36	6
7	0	0	1	2021-11-15 11:47:06	7
8	1	1	1	2019-01-15 12:10:56	8
9	1	0	0	2022-11-23 13:27:09	9
10	0	1	1	2014-12-01 17:47:06	10
11	1	1	0	2015-11-15 11:19:14	11
12	1	1	1	2023-01-15 09:47:26	12
13	1	0	1	2021-04-23 16:11:33	13
14	1	0	1	2017-01-04 10:42:36	14
15	0	1	1	2022-06-03 08:42:18	15
16	0	1	0	2020-01-03 13:55:31	16
17	1	0	1	2012-08-14 18:18:18	17
18	0	1	0	2014-12-03 17:17:17	18
19	0	1	1	2010-09-24 22:52:48	19
20	1	1	1	2021-12-01 18:45:00	20
NULL	NULL	NULL	NULL	NULL	NULL

Development & Reflection Phase

Guest_payment_details table

SQL code

```sql
-- Guest_payment_details stores information about payment details of guest
CREATE TABLE Guest_payment_details (
    Payment_details_id INT AUTO_INCREMENT PRIMARY KEY, -- Unique identifier for payment details
    payment_method VARCHAR(255), -- Method of payment (e.g., credit card, PayPal)
    account_number VARCHAR(255), -- Account or card number
    expiry_date DATE, -- Expiry date of the payment method
    guest_id INT, -- Unique identifier for the guest
    FOREIGN KEY (guest_id) REFERENCES Guest(Guest_id) -- Foreign key referencing Guest table
);
```

20 Entries

```sql
INSERT INTO Guest_payment_details (payment_method, account_number, expiry_date, guest_id)
VALUES
('Bank Transfer', 'IBAN1234567890', '2026-06-30', 1),
('Credit Card', '4111111568111111', '2025-11-29', 2),
('PayPal', 'cathy.davis@paypal.com', '2025-03-31', 3),
('Credit Card', '1341100111411111', '2026-11-21', 4),
('PayPal', 'james.anderson@paypal.com', '2026-11-17', 5),
('Credit Card', '4591116111416754', '2028-12-19', 6),
('Credit Card', '6541317611411111', '2026-04-21', 7),
('Bank Transfer', 'IBAN3876789012', '2027-03-28', 8),
('Bank Transfer', 'IBAN1255407891', '2025-04-30', 9),
('PayPal', 'isabella.walker@paypal.com', '2026-05-25', 10),
('PayPal', 'charlotte.murphy@paypal.com', '2025-12-24', 11),
('Credit Card', '6591340531416754', '2028-01-19', 12),
('PayPal', 'ava.clark@paypal.com', '2029-03-27', 13),
('PayPal', 'henry.lewis@paypal.com', '2026-12-17', 14),
('Bank Transfer', 'IBAN9700533890', '2025-08-29', 15),
('Bank Transfer', 'IBAN4533812135', '2026-08-24', 16),
('Credit Card', '4891005131457754', '2028-04-13', 17),
('PayPal', 'benjamin.lee@paypal.com', '2026-10-27', 18),
('PayPal', 'amelia.thomas@paypal.com', '2025-11-23', 19),
('Credit Card', '2311855005532414', '2027-05-15', 20);
```

Payment_details_...	payment_meth...	account_number	expiry_date	guest_id
1	Bank Transfer	IBAN1234567890	2026-06-30	1
2	Credit Card	4111111568111111	2025-11-29	2
3	PayPal	cathy.davis@paypal.com	2025-03-31	3
4	Credit Card	1341100111411111	2026-11-21	4
5	PayPal	james.anderson@paypal.com	2026-11-17	5
6	Credit Card	4591116111416754	2028-12-19	6
7	Credit Card	6541317611411111	2026-04-21	7
8	Bank Transfer	IBAN3876789012	2027-03-28	8
9	Bank Transfer	IBAN1255407891	2025-04-30	9
10	PayPal	isabella.walker@paypal.com	2026-05-25	10
11	PayPal	charlotte.murphy@paypal.com	2025-12-24	11
12	Credit Card	6591340531416754	2028-01-19	12
13	PayPal	ava.clark@paypal.com	2029-03-27	13
14	PayPal	henry.lewis@paypal.com	2026-12-17	14
15	Bank Transfer	IBAN9700533890	2025-08-29	15
16	Bank Transfer	IBAN4533812135	2026-08-24	16
17	Credit Card	4891005131457754	2028-04-13	17
18	PayPal	benjamin.lee@paypal.com	2026-10-27	18
19	PayPal	amelia.thomas@paypal.com	2025-11-23	19
20	Credit Card	2311855005532414	2027-05-15	20
NULL	NULL	NULL	NULL	NULL

Admin table

SQL code

```sql
-- Admin table stores admin information
CREATE TABLE Adminn (
    Admin_id INT AUTO_INCREMENT PRIMARY KEY, -- Unique identifier for the admin
    full_name VARCHAR(255), -- Full name of the admin
    admin_role VARCHAR(255) -- Role of the admin (e.g., manager, support)
);
```

20 Entries

```sql
INSERT INTO Adminn (full_name, admin_role)
VALUES
('Adam Murphy', 'System Administrator'),
('Alexia Lee', 'Security Analyst'),
('Michael Walker', 'Data Privacy Officer'),
('Catherine Antoir', 'User Support Specialist'),
('Isabell Gurdun', 'Operations Manager'),
('Marien Spezz', 'Policy Compliance Officer'),
('Thomas Martin', 'Quality Assurance Analyst'),
('Luke Luther', 'System Administrator'),
('Olaf Mitchell', 'Content Moderator'),
('Ben Ovich', 'Database Manager'),
('Amat Davidson', 'Data Privacy Officer'),
('Odysseum Clark', 'Technical Support Lead'),
('Emilie Cruize', 'User Support Specialist'),
('Maya Vitan', 'Risk Manager'),
('Leo Vinci', 'Legal Advisor'),
('Zeodu Uglu', 'User Support Specialist'),
('Wiktorin Rutter', 'Data Privacy Officer'),
('Mark Lewis', 'Security Analyst'),
('Kevin Carter', 'Fraud Detection Specialist'),
('Bob Brown', 'User Support Specialist');
```

Admin_id	full_name	admin_role
1	Adam Murphy	System Administrator
2	Alexia Lee	Security Analyst
3	Michael Walker	Data Privacy Officer
4	Catherine Antoir	User Support Specialist
5	Isabell Gurdun	Operations Manager
6	Marien Spezz	Policy Compliance Officer
7	Thomas Martin	Quality Assurance Analyst
8	Luke Luther	System Administrator
9	Olaf Mitchell	Content Moderator
10	Ben Ovich	Database Manager
11	Amat Davidson	Data Privacy Officer
12	Odysseum Clark	Technical Support Lead
13	Emilie Cruize	User Support Specialist
14	Maya Vitan	Risk Manager
15	Leo Vinci	Legal Advisor
16	Zeodu Uglu	User Support Specialist
17	Wiktorin Rutter	Data Privacy Officer
18	Mark Lewis	Security Analyst
19	Kevin Carter	Fraud Detection Specialist
20	Bob Brown	User Support Specialist
NULL	NULL	NULL

Development & Reflection Phase

Location table

SQL code

```sql
-- Location table stores information about location of properties
CREATE TABLE Location (
    Location_id INT AUTO_INCREMENT PRIMARY KEY, -- Unique identifier for the location
    country VARCHAR(255), -- Country name
    city VARCHAR(255), -- City name
    zip_code VARCHAR(20) -- ZIP or postal code
);
```

20 Entries

```sql
INSERT INTO Location (country, city, zip_code)
VALUES
('Germany', 'Cologne', '50667'),
('Germany', 'Bonn', '53179'),
('Italy', 'Milan', '20021'),
('Spain', 'Madrid', '28002'),
('Italy', 'Rome', '00128'),
('Germany', 'Berlin', '10115'),
('France', 'Paris', '70123'),
('France', 'Paris', '75000'),
('Germany', 'Bonn', '53115'),
('USA', 'Boston', '02118'),
('Greece', 'Athens', '104 38'),
('Norway', 'Oslo', '0154'),
('Denmark', 'Copenhagen', '1064'),
('Sweden', 'Stockholm', '111 15'),
('USA', 'Baltimore', '21208'),
('Argentina', 'Buenos Aires', 'B1094'),
('Turkey', 'Istanbul', '34017'),
('USA', 'Atanta', '30301'),
('Netherlands', 'The Hague', '1187 LZ'),
('Finland', 'Helsinki', '00290');
```

Location_id	country	city	zip_code
1	Germany	Cologne	50667
2	Germany	Bonn	53179
3	Italy	Milan	20021
4	Spain	Madrid	28002
5	Italy	Rome	00128
6	Germany	Berlin	10115
7	France	Paris	70123
8	France	Paris	75000
9	Germany	Bonn	53115
10	USA	Boston	02118
11	Greece	Athens	104 38
12	Norway	Oslo	0154
13	Denmark	Copen…	1064
14	Sweden	Stockh…	111 15
15	USA	Baltim…	21208
16	Argentina	Bueno…	B1094
17	Turkey	Istanbul	34017
18	USA	Atanta	30301
19	Netherl…	The H…	1187 LZ
20	Finland	Helsinki	00290
NULL	NULL	NULL	NULL

Property_type table

SQL code

```sql
-- Property_type stores information about property types
CREATE TABLE Property_type (
    Property_type_id INT auto_increment PRIMARY KEY, -- Unique identifier for the property type
    property_type_name VARCHAR(255) -- Name of the property type (e.g., apartment, house)
);
```

20 Entries

```sql
INSERT INTO Property_type (property_type_name)
VALUES
('Apartment'),
('Cabin'),
('Apartment'),
('House'),
('Villa'),
('Castle'),
('Cabin'),
('Apartment'),
('House'),
('Mansion'),
('Apartment'),
('Castle'),
('House'),
('Castle'),
('Mansion'),
('Cabin'),
('Apartment'),
('Villa'),
('Tree House'),
('House');
```

Property_type…	property_type_na…
1	Apartment
2	Cabin
3	Apartment
4	House
5	Villa
6	Castle
7	Cabin
8	Apartment
9	House
10	Mansion
11	Apartment
12	Castle
13	House
14	Castle
15	Mansion
16	Cabin
17	Apartment
18	Villa
19	Tree House
20	House
NULL	NULL

Development & Reflection Phase

Property table

SQL code

```sql
-- Property table stores information about properties
CREATE TABLE Property (
    Property_id INT AUTO_INCREMENT PRIMARY KEY, -- Unique identifier for the property
    title VARCHAR(255), -- Title of the property listing
    property_description TEXT, -- Detailed description of the property
    price_per_night DECIMAL(10, 2), -- Cost of staying per night
    property_type_id INT, -- Unique identifier for the property type
    host_id INT, -- Unique identifier for the host
    location_id INT, -- Unique identifier for the location
    FOREIGN KEY (property_type_id) REFERENCES Property_type(Property_type_id), -- Foreign key referencing Property table
    FOREIGN KEY (host_id) REFERENCES Hostt(Host_id), -- Foreign key referencing Hostt table
    FOREIGN KEY (location_id) REFERENCES Location(Location_id) -- Foreign key referencing Location table
);
```

20 Entries

```sql
INSERT INTO Property (title, property_description, price_per_night, property_type_id, host_id, location_id)
VALUES
('Bright Apartment in Bonn', 'Bright apartment located in the centre of Bonn', 110.00, 14, 20, 1),
('Baumhaus', 'The cabin where you can spend some time alone', 132.00, 7, 1, 2),
('Beautiful Apartment in The Heart of Milan', '2-room apartment in the middle of Milan', 102.00, 3, 5, 3),
('Una Casa Grande en Madrid', '4-room house, ideal for families', 150.00, 9, 7, 4),
('Holidays Villa in Rome', 'Spend a wonderful time in Rome in a majestic villa', 1950.00, 11, 13, 5),
('Majestic Castle', 'Be ready to become part of the history in our historic castle', 1021.00, 17, 11, 6),
('Spacious Cabin For Two', 'Run away from the city noise for the weekend', 97.00, 2, 2, 7),
('Cozy Apartment', 'Beautiful apartment in Paris with the view oif Eiffel Tower', 134.00, 4, 19, 8),
('House With Sauna', 'Big house with sauna, ideal to relax', 167.00, 5, 3, 9),
('Beautiful Mansion With Pool', 'Spend your entire holiday in a heated-pool in Boston', 2455.00, 8, 4, 10),
('Luxurious Apartment in Natural Area', 'Experience privacy in comfort in our stylish apartment', 125.00, 12, 8, 11),
('Breathtaking Castle in Norway', 'Lovely place to spend time with family and friends', 1165.00, 10, 15, 12),
('Tiny House With Amazing Views', 'Private and quiet house in Copenhagen', 89.00, 1, 18, 13),
('Charming Swedish Castle', 'Castle in a lovely area of Stockholm', 2040.00, 19, 14, 14),
('9-Bedroom Villa With Huge Indoor Sports Court', 'Mansion for a big family stay or friends gathering', 3047.00, 20, 6, 15),
('Wooden House to Relax', 'Beautiful wooden cabin by native greenery and wonderful views', 135.00, 6, 9, 16),
('Big flat With a Great View of Istanbul', 'Loveve apartment in the centre of Istanbul', 107.00, 13, 10, 17),
('Beautiful 8-people Villa', 'Villa for 8-people stay, with 10 bedroom, 8 bathrooms and a lot more', 1899.00, 15, 12, 18),
('Treehouse For a Romantic Stay', 'In a treehouse your stay becomes very special', 303.00, 16, 17, 19),
('Designer House', 'Welcome to our Designer House to enjoy your stay', 146.00, 18, 16, 20);
```

Property_id	title	property_description	price_per_ni...	property_type...	host_id	location_id
1	Bright Apartment in Bonn	Bright apartment located in the centre of Bonn	110.00	14	20	1
2	Baumhaus	The cabin where you can spend some time alone	132.00	7	1	2
3	Beautiful Apartment in The Heart of Milan	2-room apartment in the middle of Milan	102.00	3	5	3
4	Una Casa Grande en Madrid	4-room house, ideal for families	150.00	9	7	4
5	Holidays Villa in Rome	Spend a wonderful time in Rome in a majestic villa	1950.00	11	13	5
6	Majestic Castle	Be ready to become part of the history in our his...	1021.00	17	11	6
7	Spacious Cabin For Two	Run away from the city noise for the weekend	97.00	2	2	7
8	Cozy Apartment	Beautiful apartment in Paris with the view oif Eiff...	134.00	4	19	8
9	House With Sauna	Big house with sauna, ideal to relax	167.00	5	3	9
10	Beautiful Mansion With Pool	Spend your entire holiday in a heated-pool in Bo...	2455.00	8	4	10
11	Luxurious Apartment in Natural Area	Experience privacy in comfort in our stylish apar...	125.00	12	8	11
12	Breathtaking Castle in Norway	Lovely place to spend time with family and friends	1165.00	10	15	12
13	Tiny House With Amazing Views	Private and quiet house in Copenhagen	89.00	1	18	13
14	Charming Swedish Castle	Castle in a lovely area of Stockholm	2040.00	19	14	14
15	9-Bedroom Villa With Huge Indoor Spor...	Mansion for a big family stay or friends gathering	3047.00	20	6	15
16	Wooden House to Relax	Beautiful wooden cabin by native greenery and...	135.00	6	9	16
17	Big flat With a Great View of Istanbul	Loveve apartment in the centre of Istanbul	107.00	13	10	17
18	Beautiful 8-people Villa	Villa for 8-people stay, with 10 bedroom, 8 bathr...	1899.00	15	12	18
19	Treehouse For a Romantic Stay	In a treehouse your stay becomes very special	303.00	16	17	19
20	Designer House	Welcome to our Designer House to enjoy your s...	146.00	18	16	20
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Amenities table

SQL code

```sql
-- Amenities table stores information about amenities available in the property
CREATE TABLE Amenities (
    Amenity_id INT AUTO_INCREMENT PRIMARY KEY, -- Unique identifier for an amenity
    amenity_name VARCHAR(255), -- Name of the amenity (e.g., Wi-Fi, parking)
    amenity_description VARCHAR(255) -- Description of the amenity
);
```

20 Entries

```sql
INSERT INTO Amenities (amenity_name, amenity_description)
VALUES
('Wifi','High-speed wireless internet'),
('Swimming pool','Private swimming pool on property'),
('Heating','Central heating system for colder seasons'),
('Air Conditioning','Cool air conditioning for hot weather'),
('Smoke Alarm','Safety smoke detector installed'),
('Kitchen','Fully-equipped kitchen'),
('TV','Flat-screen television with cable channels'),
('Gym','Access to gyma and fitness equipment'),
('Pet friendly','Accomodation allows pets'),
('Fireplace','Cozy indoor fireplace'),
('Hot Tub','Private hot-tub or jacuzzi'),
('Balcony','Private balcony with scenic view'),
('Patio','Private patio with outdoor furniture'),
('Long-term stay','Allow to stay for more than 28 days'),
('Private Entrance','Separate street or building entrance'),
('Washer','Washing machine for laundry'),
('Hair Dryer','Electric hair dryer available'),
('BBQ Grill','Outdoor barbecue grill available'),
('Iron','Iron and ironing board provided'),
('Parking','Private parking space available');
```

Amenity_id	amenity_name	amenity_description
1	Wifi	High-speed wireless internet
2	Swimming pool	Private swimming pool on property
3	Heating	Central heating system for colder seasons
4	Air Conditioning	Cool air conditioning for hot weather
5	Smoke Alarm	Safety smoke detector installed
6	Kitchen	Fully-equipped kitchen
7	TV	Flat-screen television with cable channels
8	Gym	Access to gyma and fitness equipment
9	Pet friendly	Accomodation allows pets
10	Fireplace	Cozy indoor fireplace
11	Hot Tub	Private hot-tub or jacuzzi
12	Balcony	Private balcony with scenic view
13	Patio	Private patio with outdoor furniture
14	Long-term stay	Allow to stay for more than 28 days
15	Private Entrance	Separate street or building entrance
16	Washer	Washing machine for laundry
17	Hair Dryer	Electric hair dryer available
18	BBQ Grill	Outdoor barbecue grill available
19	Iron	Iron and ironing board provided
20	Parking	Private parking space available
NULL	NULL	NULL

Development & Reflection Phase

Property_amenities table

SQL code

```sql
-- Property_amenities table establishes relatinoship between Properties and Amenities tables
CREATE TABLE Property_amenities (
    Property_id INT, -- Unique identifier for a property
    Amenity_id INT, -- Unique identifier for an amenity
    PRIMARY KEY (Property_id, Amenity_id), -- Composite primary key
    FOREIGN KEY (Property_id) REFERENCES Property(Property_id), -- Foreign key referencing Property table
    FOREIGN KEY (Amenity_id) REFERENCES Amenities(Amenity_id) -- Foreign key referencing Amenities table
);
```

20 Entries

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

Property_id	Amenity_id
1	1
1	3
1	5
1	6
2	3
2	5
2	6
2	17
3	1
3	3
3	5
3	7
4	1
4	6
4	9
4	20
5	1
5	11
5	18
5	20
6	3
6	10
6	13
6	18
7	1
7	4
7	5
7	17

Property_id	Amenity_id
8	5
8	12
8	16
8	17
9	2
9	5
9	9
9	10
10	8
10	10
10	13
10	20
11	1
11	3
11	7
11	19
12	9
12	11
12	15
12	20
13	3
13	12
13	14
13	17
14	8
14	11
14	13
14	20

Property_id	Amenity_id
14	13
14	20
15	9
15	10
15	11
15	15
16	4
16	6
16	7
16	19
17	1
17	9
17	12
17	19
18	1
18	2
18	18
18	20
19	3
19	4
19	7
19	17
20	1
20	8
20	9
20	11
NULL	NULL

Photos table

SQL code

```sql
-- Photos table stores photos of properties
CREATE TABLE Photos (
    Photo_id INT AUTO_INCREMENT PRIMARY KEY, -- Unique identifier for a photo
    image_url VARCHAR(255) -- URL of the property photo
);
```

20 Entries

```sql
INSERT INTO Photos (image_url)
VALUES
('image_url/1'),
('image_url/2'),
('image_url/3'),
('image_url/4'),
('image_url/5'),
('image_url/6'),
('image_url/7'),
('image_url/8'),
('image_url/9'),
('image_url/10'),
('image_url/11'),
('image_url/12'),
('image_url/13'),
('image_url/14'),
('image_url/15'),
('image_url/16'),
('image_url/17'),
('image_url/18'),
('image_url/19'),
('image_url/20');
```

Photo_id	image_url
1	image_url/1
2	image_url/2
3	image_url/3
4	image_url/4
5	image_url/5
6	image_url/6
7	image_url/7
8	image_url/8
9	image_url/9
10	image_url/10
11	image_url/11
12	image_url/12
13	image_url/13
14	image_url/14
15	image_url/15
16	image_url/16
17	image_url/17
18	image_url/18
19	image_url/19
20	image_url/20
NULL	NULL

Development & Reflection Phase

Property_photos table

SQL code

```sql
-- Property_photos table establishes relationship between Property and Photos tables
CREATE TABLE Property_photos (
    Property_id INT, -- Unique identifier for a property
    Photo_id INT, -- Unique identifier for a photo
    PRIMARY KEY (Property_id, Photo_id), --  Composite primary key
    FOREIGN KEY (Property_id) REFERENCES Property(Property_id), -- Foreign key referencing Property table
    FOREIGN KEY (Photo_id) REFERENCES Photos(Photo_id) -- Foreign key referencing Photos table
);
```

20 Entries

```sql
INSERT INTO Property_photos (Property_id, Photo_id)
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);
```

Property_id	Photo_id
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
NULL	NULL

Booking table

SQL code

```sql
-- Booking table stores information about bookings made by guests
CREATE TABLE Booking (
    Booking_id INT AUTO_INCREMENT PRIMARY KEY, -- Unique identifier for a booking
    check_in_date DATE, -- Check-in date for the booking
    check_out_date DATE, -- Check-out date for the booking
    total_price DECIMAL(10, 2), -- Total price for the booking
    payment_status ENUM('Paid', 'Pending'), -- Status of the payment (e.g., Paid, Pending)
    booking_date TIMESTAMP, -- Date and time the booking was made
    guest_id INT, -- Unique identifier for the guest
    host_id INT, -- Unique identifier for the host
    property_id INT, -- Unique identifier for the property
    FOREIGN KEY (guest_id) REFERENCES Guest(Guest_id), -- Foreign key referencing Guest table
    FOREIGN KEY (host_id) REFERENCES Hostt(Host_id), -- Foreign key referencing Hostt table
    FOREIGN KEY (property_id) REFERENCES Property(Property_id) -- Foreign key referencing Property table
);
```

20 Entries

```sql
INSERT INTO Booking (check_in_date, check_out_date, total_price, payment_status, booking_date, property_id, host_id, guest_id)
/ALUES
('2025-01-01', '2025-01-06', 550.00, 'Paid', '2024-12-30 12:30:00', 14, 20, 1),
('2025-01-02', '2025-01-06', 528.00, 'Pending', '2025-01-01 09:15:00', 7, 1, 2),
('2025-01-03', '2025-01-06', 306.00, 'Paid', '2025-01-02 14:45:00', 3, 5, 3),
('2025-01-04', '2025-01-08', 600.00, 'Paid', '2025-01-03 17:20:00', 9, 7, 4),
('2025-01-05', '2025-01-09', 7800.00, 'Paid', '2025-01-04 11:00:00', 11, 13, 5),
('2025-01-06', '2025-01-10', 4084.00, 'Pending', '2025-01-05 10:15:00', 17, 11, 6),
('2025-01-07', '2025-01-13', 582.00, 'Paid', '2025-01-06 13:50:00', 2, 2, 7),
('2025-01-08', '2025-01-13', 670.00, 'Paid', '2025-01-07 08:30:00', 4, 19, 8),
('2025-01-09', '2025-01-14', 835.00, 'Paid', '2025-01-08 16:45:00', 5, 3, 9),
('2025-01-10', '2025-01-14', 9820.00, 'Pending', '2025-01-09 14:00:00', 8, 4, 10),
('2025-01-11', '2025-01-16', 625.00, 'Paid', '2025-01-10 09:45:00', 12, 8, 11),
('2025-01-12', '2025-01-18', 6990.00, 'Paid', '2025-01-11 10:20:00', 10, 15, 12),
('2025-01-13', '2025-01-18', 445.00, 'Paid', '2025-01-12 15:35:00', 1, 18, 13),
('2025-01-14', '2025-01-19', 10200.00, 'Paid', '2025-01-13 13:10:00', 19, 14, 14),
('2025-01-15', '2025-01-20', 15235.00, 'Paid', '2025-01-14 11:55:00', 20, 6, 15),
('2025-01-16', '2025-01-21', 675.00, 'Pending', '2025-01-15 17:40:00', 6, 9, 16),
('2025-01-17', '2025-01-21', 428.00, 'Paid', '2025-03-16 18:30:00', 13, 10, 17),
('2025-01-18', '2025-01-23', 9495.00, 'Paid', '2025-01-17 08:20:00', 15, 12, 18),
('2025-01-19', '2025-01-24', 1515.00, 'Paid', '2025-01-18 14:10:00', 16, 17, 19),
('2025-01-20', '2025-01-26', 876.00, 'Pending', '2025-01-19 16:50:00', 18, 16, 20);
```

Booking_id	check_in_da...	check_out_da...	total_pri...	payment_stat...	booking_date	guest_id	host_id	property_id
1	2025-01-01	2025-01-06	550.00	Paid	2024-12-30 12:30:00	1	20	14
2	2025-01-02	2025-01-06	528.00	Pending	2025-01-01 09:15:00	2	1	7
3	2025-01-03	2025-01-06	306.00	Paid	2025-01-02 14:45:00	3	5	3
4	2025-01-04	2025-01-08	600.00	Paid	2025-01-03 17:20:00	4	7	9
5	2025-01-05	2025-01-09	7800.00	Paid	2025-01-04 11:00:00	5	13	11
6	2025-01-06	2025-01-10	4084.00	Pending	2025-01-05 10:15:00	6	11	17
7	2025-01-07	2025-01-13	582.00	Paid	2025-01-06 13:50:00	7	2	2
8	2025-01-08	2025-01-13	670.00	Paid	2025-01-07 08:30:00	8	19	4
9	2025-01-09	2025-01-14	835.00	Paid	2025-01-08 16:45:00	9	3	5
10	2025-01-10	2025-01-14	9820.00	Pending	2025-01-09 14:00:00	10	4	8
11	2025-01-11	2025-01-16	625.00	Paid	2025-01-10 09:45:00	11	8	12
12	2025-01-12	2025-01-18	6990.00	Paid	2025-01-11 10:20:00	12	15	10
13	2025-01-13	2025-01-18	445.00	Paid	2025-01-12 15:35:00	13	18	1
14	2025-01-14	2025-01-19	10200.00	Paid	2025-01-13 13:10:00	14	14	19
15	2025-01-15	2025-01-20	15235.00	Paid	2025-01-14 11:55:00	15	6	20
16	2025-01-16	2025-01-21	675.00	Pending	2025-01-15 17:40:00	16	9	6
17	2025-01-17	2025-01-21	428.00	Paid	2025-03-16 18:30:00	17	10	13
18	2025-01-18	2025-01-23	9495.00	Paid	2025-01-17 08:20:00	18	12	15
19	2025-01-19	2025-01-24	1515.00	Paid	2025-01-18 14:10:00	19	17	16
20	2025-01-20	2025-01-26	876.00	Pending	2025-01-19 16:50:00	20	16	18
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Development & Reflection Phase

Cancellation_policy table

SQL code

```sql
-- Cancellation_policy stores information about cancellation policies for properties
CREATE TABLE Cancellation_policy (
    Cancellation_policy_id INT AUTO_INCREMENT PRIMARY KEY, -- Unique identifier for a cancellation policy
    policy_description TEXT -- Description of the cancellation policy
);
```

20 Entries

```sql
INSERT INTO Cancellation_policy (policy_description)
VALUES
('Flexible: Full refund up to 1 day before check-in.'),
('Moderate: Full refund up to 5 days before check-in.'),
('Strict: 50% refund up to 7 days before check-in.'),
('Non-Refundable: No refund under any circumstances.'),
('Super Strict 30 Days: 50% refund up to 30 days before check-in.'),
('Super Strict 60 Days: 50% refund up to 60 days before check-in.'),
('Last-Minute Friendly: Full refund up to 12 hours before check-in.'),
('Seasonal: Refund depends on the season; up to 75% refund.'),
('Long Stay: 75% refund for stays longer than 14 days.'),
('Corporate: Full refund for corporate bookings up to 10 days before.'),
('Custom Policy 1: Custom terms; contact the host for details.'),
('Weekend Strict: No refund for weekend stays.'),
('Holiday Policy: No refund during holidays.'),
('Family Friendly: Full refund up to 3 days before check-in.'),
('Early Bird: Full refund if cancelled within 24 hours of booking.'),
('Adventure Trip: Partial refund for adventure properties.'),
('Luxury Stay: Refund terms vary; up to 80% refund.'),
('Pet Friendly: Full refund if pets are not allowed.'),
('No Questions Asked: Full refund anytime.'),
('Strict with Grace Period: Full refund within 48 hours of booking for stays more than 14 days away.');
```

Cancellation_policy...	policy_description
1	Flexible: Full refund up to 1 day before check-in.
2	Moderate: Full refund up to 5 days before check...
3	Strict: 50% refund up to 7 days before check-in.
4	Non-Refundable: No refund under any circumst...
5	Super Strict 30 Days: 50% refund up to 30 days...
6	Super Strict 60 Days: 50% refund up to 60 days...
7	Last-Minute Friendly: Full refund up to 12 hours...
8	Seasonal: Refund depends on the season; up t...
9	Long Stay: 75% refund for stays longer than 14...
10	Corporate: Full refund for corporate bookings up...
11	Custom Policy 1: Custom terms; contact the hos...
12	Weekend Strict: No refund for weekend stays.
13	Holiday Policy: No refund during holidays.
14	Family Friendly: Full refund up to 3 days before...
15	Early Bird: Full refund if cancelled within 24 hou...
16	Adventure Trip: Partial refund for adventure pro...
17	Luxury Stay: Refund terms vary; up to 80% refu...
18	Pet Friendly: Full refund if pets are not allowed.
19	No Questions Asked: Full refund anytime.
20	Strict with Grace Period: Full refund within 48 h...
NULL	NULL

Booking_cancellation table

SQL code

```sql
-- Booking_cancellation establishes relationship between Booking and Cancellation_policy tables
CREATE TABLE Booking_cancellation (
    Cancellation_id INT AUTO_INCREMENT PRIMARY KEY, -- Unique identifier for a booking cancellation
    cancellation_date DATE, -- Cancellation date of a booking
    booking_id INT, -- Unique identifier for a booking
    cancellation_policy_id INT, -- Unique identifier for a cancellation policy
    FOREIGN KEY (booking_id) REFERENCES Booking(Booking_id), -- Foreign key referencing Booking table
    FOREIGN KEY (cancellation_policy_id) REFERENCES Cancellation_policy(Cancellation_policy_id) -- Foreign key referencing Cancellation_policy table
);
```

20 Entries

```sql
INSERT INTO Booking_cancellation (cancellation_date, booking_id, cancellation_policy_id)
VALUES
('2025-01-01', 1, 1),
('2025-01-02', 2, 3),
('2025-01-03', 3, 2),
('2025-01-04', 4, 4),
('2025-01-05', 5, 5),
('2025-01-06', 6, 6),
('2025-01-07', 7, 7),
('2025-01-08', 8, 8),
('2025-01-09', 9, 9),
('2025-01-10', 10, 10),
('2025-01-11', 11, 11),
('2025-01-12', 12, 12),
('2025-01-13', 13, 13),
('2025-01-14', 14, 14),
('2025-01-15', 15, 15),
('2025-01-16', 16, 16),
('2025-01-17', 17, 17),
('2025-01-18', 18, 18),
('2025-01-19', 19, 19),
('2025-01-20', 20, 20);
```

Cancellation_id	cancellation_d...	booking_id	cancellation_policy...
1	2025-01-01	1	1
2	2025-01-02	2	3
3	2025-01-03	3	2
4	2025-01-04	4	4
5	2025-01-05	5	5
6	2025-01-06	6	6
7	2025-01-07	7	7
8	2025-01-08	8	8
9	2025-01-09	9	9
10	2025-01-10	10	10
11	2025-01-11	11	11
12	2025-01-12	12	12
13	2025-01-13	13	13
14	2025-01-14	14	14
15	2025-01-15	15	15
16	2025-01-16	16	16
17	2025-01-17	17	17
18	2025-01-18	18	18
19	2025-01-19	19	19
20	2025-01-20	20	20
NULL	NULL	NULL	NULL

Development & Reflection Phase

Payment table

SQL code

```sql
-- Payment table stores information about payments made for bookings
CREATE TABLE Payment (
    Payment_id INT AUTO_INCREMENT PRIMARY KEY, -- Unique identifier for a payment
    amount DECIMAL(10, 2), -- Amount paid
    payment_status ENUM('Completed', 'Failed', 'Pending'), -- Status of the payment (e.g., Completed, Failed, Pending)
    payment_date DATETIME, -- Date of payment
    booking_id INT, -- Unique identifier for a booking
    FOREIGN KEY (booking_id) REFERENCES Booking(Booking_id) -- Foreign key referencing Booking table
);
```

20 Entries

```sql
INSERT INTO Payment (amount, payment_status, payment_date, booking_id)
VALUES
(550.00, 'Completed', '2024-12-30 12:30:00', 1),
(528.00, 'Pending', '2025-01-01 09:15:00', 2),
(306.00, 'Completed', '2025-01-02 14:45:00', 3),
(600.00, 'Completed', '2025-01-03 17:20:00', 4),
(7800.00, 'Completed', '2025-01-04 11:00:00', 5),
(4084.00, 'Pending', '2025-01-05 10:15:00', 6),
(582.00, 'Completed', '2025-01-06 13:50:00', 7),
(670.00, 'Completed', '2025-01-07 08:30:00', 8),
(835.00, 'Completed', '2025-01-08 16:45:00', 9),
(9820.00, 'Pending', '2025-01-09 14:00:00', 10),
(625.00, 'Failed', '2025-01-10 09:45:00', 11),
(6990.00, 'Completed', '2025-01-11 10:20:00', 12),
(445.00, 'Failed', '2025-01-12 15:35:00', 13),
(10200.00, 'Completed', '2025-01-13 13:10:00', 14),
(15235.00, 'Completed', '2025-01-14 11:55:00', 15),
(675.00, 'Pending', '2025-01-15 17:40:00', 16),
(428.00, 'Completed', '2025-03-16 18:30:00', 17),
(9495.00, 'Completed', '2025-01-17 08:20:00', 18),
(1515.00, 'Completed', '2025-01-18 14:10:00', 19),
(876.00, 'Pending', '2025-01-19 16:50:00', 20);
```

Payment_id	amount	payment_stat...	payment_date	booking_id
1	550.00	Completed	2024-12-30 12:30:00	1
2	528.00	Pending	2025-01-01 09:15:00	2
3	306.00	Completed	2025-01-02 14:45:00	3
4	600.00	Completed	2025-01-03 17:20:00	4
5	7800.00	Completed	2025-01-04 11:00:00	5
6	4084.00	Pending	2025-01-05 10:15:00	6
7	582.00	Completed	2025-01-06 13:50:00	7
8	670.00	Completed	2025-01-07 08:30:00	8
9	835.00	Completed	2025-01-08 16:45:00	9
10	9820.00	Pending	2025-01-09 14:00:00	10
11	625.00	Failed	2025-01-10 09:45:00	11
12	6990.00	Completed	2025-01-11 10:20:00	12
13	445.00	Failed	2025-01-12 15:35:00	13
14	10200....	Completed	2025-01-13 13:10:00	14
15	15235....	Completed	2025-01-14 11:55:00	15
16	675.00	Pending	2025-01-15 17:40:00	16
17	428.00	Completed	2025-03-16 18:30:00	17
18	9495.00	Completed	2025-01-17 08:20:00	18
19	1515.00	Completed	2025-01-18 14:10:00	19
20	876.00	Pending	2025-01-19 16:50:00	20
NULL	NULL	NULL	NULL	NULL

Transactions table

SQL code

```sql
-- Transactions table stores information about transactions made
CREATE TABLE Transactions (
    Transaction_id INT AUTO_INCREMENT PRIMARY KEY, -- Unique identifier for a transaction
    transaction_type ENUM('Credit', 'Debit', 'Paypal'), -- Type of transaction (Credit/Debit/PayPal)
    transaction_date DATETIME, -- Date of the transaction
    transaction_amount DECIMAL(10, 2), -- Amount involved in the transaction
    payment_id INT, -- Unique identifier for a payment
    host_id INT, -- Unique identifier for a host
    guest_id INT, -- Unique identifier for a payment
    FOREIGN KEY (payment_id) REFERENCES Payment(Payment_id), -- Foreign key referencing Payment table
    FOREIGN KEY (host_id) REFERENCES Hostt(Host_id), -- Foreign key referencing Hostt table
    FOREIGN KEY (guest_id) REFERENCES Guest(Guest_id) -- Foreign key referencing Guest table
);
```

20 Entries

```sql
INSERT INTO Transactions (transaction_type, transaction_date, transaction_amount, payment_id, host_id, guest_id)
VALUES
('Debit', '2024-12-30 12:30:00', 550.00, 1, 20, 1),
('Credit', '2025-01-01 09:15:00', 528.00, 2, 1, 2),
('PayPal', '2025-01-02 14:45:00', 306.00, 3, 5, 3),
('Credit', '2025-01-03 17:20:00', 600.00, 4, 7, 4),
('PayPal', '2025-01-04 11:00:00', 7800.00, 5, 13, 5),
('Credit', '2025-01-05 10:15:00', 4084.00, 6, 11, 6),
('Credit', '2025-01-06 13:50:00', 582.00, 7, 2, 7),
('Debit', '2025-01-07 08:30:00', 670.00, 8, 19, 8),
('Debit', '2025-01-08 16:45:00', 835.00, 9, 3, 9),
('PayPal', '2025-01-09 14:00:00', 9820.00, 10, 4, 10),
('PayPal', '2025-01-10 09:45:00', 625.00, 11, 8, 11),
('Credit', '2025-01-11 10:20:00', 6990.00,  12, 15, 12),
('PayPal', '2025-01-12 15:35:00', 445.00, 13, 18, 13),
('PayPal', '2025-01-13 13:10:00', 10200.00, 14, 14, 14),
('Debit', '2025-01-14 11:55:00', 15235.00, 15, 6, 15),
('Debit', '2025-01-15 17:40:00', 675.00, 16, 9, 16),
('Credit', '2025-03-16 18:30:00', 428.00, 17, 10, 17),
('PayPal', '2025-01-17 08:20:00', 9495.00, 18, 12, 18),
('PayPal', '2025-01-18 14:10:00', 1515.00, 19, 17, 19),
('Credit', '2025-01-19 16:50:00', 876.00, 20, 16, 20);
```

Transaction_...	transaction_ty...	transaction_date	transaction_amo...	payment_id	host_id	guest_id
1	Debit	2024-12-30 12:30:00	550.00	1	20	1
2	Credit	2025-01-01 09:15:00	528.00	2	1	2
3	Paypal	2025-01-02 14:45:00	306.00	3	5	3
4	Credit	2025-01-03 17:20:00	600.00	4	7	4
5	Paypal	2025-01-04 11:00:00	7800.00	5	13	5
6	Credit	2025-01-05 10:15:00	4084.00	6	11	6
7	Credit	2025-01-06 13:50:00	582.00	7	2	7
8	Debit	2025-01-07 08:30:00	670.00	8	19	8
9	Debit	2025-01-08 16:45:00	835.00	9	3	9
10	Paypal	2025-01-09 14:00:00	9820.00	10	4	10
11	Paypal	2025-01-10 09:45:00	625.00	11	8	11
12	Credit	2025-01-11 10:20:00	6990.00	12	15	12
13	Paypal	2025-01-12 15:35:00	445.00	13	18	13
14	Paypal	2025-01-13 13:10:00	10200.00	14	14	14
15	Debit	2025-01-14 11:55:00	15235.00	15	6	15
16	Debit	2025-01-15 17:40:00	675.00	16	9	16
17	Credit	2025-03-16 18:30:00	428.00	17	10	17
18	Paypal	2025-01-17 08:20:00	9495.00	18	12	18
19	Paypal	2025-01-18 14:10:00	1515.00	19	17	19
20	Credit	2025-01-19 16:50:00	876.00	20	16	20
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Development & Reflection Phase

Review table

SQL code

```sql
-- Review table stores information about reviews given  by guests for properties
CREATE TABLE Review (
    Review_id INT AUTO_INCREMENT PRIMARY KEY, -- Unique identifier for a review
    rating FLOAT, -- Rating score for a property
    review_comment TEXT, -- Comments left in a review
    review_date DATE, -- Date when the review was written
    host_id INT, -- Unique identifier for a host
    guest_id INT, -- Unique identifier for a guest
    property_id INT, -- Unique identifier for a property
    FOREIGN KEY (host_id) REFERENCES Hostt(Host_id), -- Foreign key referencing Hostt table
    FOREIGN KEY (guest_id) REFERENCES Guest(Guest_id), -- Foreign key referencing Guest table
    FOREIGN KEY (property_id) REFERENCES Property(Property_id) -- Foreign key referencing Property table
);
```

20 Entries

```sql
INSERT INTO Review (rating, review_comment, review_date, guest_id, property_id, host_id)
VALUES
(4, 'Really enjoyed the stay in the apartment', '2025-02-01', 1, 14, 20),
(4, 'The cabin was cozy', '2025-02-02', 2, 7, 1),
(5, 'The apartment is located in the centre of Milan, which is convenient for tourists', '2025-02-03', 3, 3, 5),
(3, 'No kitchen', '2025-02-04', 4, 9, 7),
(5, 'Suitable for a vacation with a big family or bunch of friends', '2025-02-05', 5, 11, 13),
(3, 'Overpriced', '2025-02-06', 6, 17, 11),
(4, 'Nice cabin', '2025-02-07', 7, 2, 2),
(5, 'Cozy apartment', '2025-02-08', 8, 4, 19),
(5, 'Pet friendly house', '2025-02-09', 9, 5, 3),
(4, 'Wonderful mansion but there is no wifi', '2025-02-10', 10, 8, 4),
(3, 'It was a bit dirty', '2025-02-11', 11, 12, 8),
(3, 'Majestic castle but it was a bit far from the city', '2025-02-12', 12, 10, 15),
(4, 'Good house for the price', '2025-02-13', 13, 1, 18),
(4, 'Never stayed in the castle before, recommended', '2025-02-14', 14, 19, 14),
(5, 'Perfect place to throw parties, had everything we needed', '2025-02-15', 15, 20, 6),
(5, 'Ideal for people trying to get away from the city and the crowd', '2025-02-16', 16, 6, 9),
(5, 'Such a lovely place to stay', '2025-02-17', 17, 13, 10),
(4, 'Fantastically beautiful', '2025-02-18', 18, 15, 12),
(4, 'Our stay in the treehouse was a dream', '2025-02-19', 19, 16, 17),
(5, 'Beautiful location and it is dog-friendly', '2025-02-20', 20, 18, 16);
```

Review_id	rating	review_comment	review_date	host_id	guest_id	property_id
1	4	Really enjoyed the stay in the apartment	2025-02-01	20	1	14
2	4	The cabin was cozy	2025-02-02	1	2	7
3	5	The apartment is located in the centre of Milan,...	2025-02-03	5	3	3
4	3	No kitchen	2025-02-04	7	4	9
5	5	Suitable for a vacation with a big family or bunc...	2025-02-05	13	5	11
6	3	Overpriced	2025-02-06	11	6	17
7	4	Nice cabin	2025-02-07	2	7	2
8	5	Cozy apartment	2025-02-08	19	8	4
9	5	Pet friendly house	2025-02-09	3	9	5
10	4	Wonderful mansion but there is no wifi	2025-02-10	4	10	8
11	3	It was a bit dirty	2025-02-11	8	11	12
12	3	Majestic castle but it was a bit far from the city	2025-02-12	15	12	10
13	4	Good house for the price	2025-02-13	18	13	1
14	4	Never stayed in the castle before, recommended	2025-02-14	14	14	19
15	5	Perfect place to throw parties, had everything w...	2025-02-15	6	15	20
16	5	Ideal for people trying to get away from the city...	2025-02-16	9	16	6
17	5	Such a lovely place to stay	2025-02-17	10	17	13
18	4	Fantastically beautiful	2025-02-18	12	18	15
19	4	Our stay in the treehouse was a dream	2025-02-19	17	19	16
20	5	Beautiful location and it is dog-friendly	2025-02-20	16	20	18
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Wishlist table

SQL code

```sql
-- Wishlist table stores information about properties guests want to visit
CREATE TABLE Wishlist (
    Wishlist_id INT AUTO_INCREMENT PRIMARY KEY, -- Unique identifier for a Wishlist entry
    date_added DATE, -- Date when the property was added to the wishlist
    guest_id INT, -- Unique identifier for a guest
    property_id INT, -- Unique identifier for a property
    FOREIGN KEY (guest_id) REFERENCES Guest(Guest_id), -- Foreign key referencing Guest table
    FOREIGN KEY (property_id) REFERENCES Property(Property_id) -- Foreign key referencing Property table
);
```

20 Entries

```sql
INSERT INTO Wishlist (date_added, guest_id, property_id)
VALUES
('2025-03-01', 1, 1),
('2025-03-02', 2, 3),
('2025-03-03', 3, 2),
('2025-03-04', 4, 4),
('2025-03-05', 5, 5),
('2025-03-06', 6, 6),
('2025-03-07', 7, 7),
('2025-03-08', 8, 8),
('2025-03-09', 9, 9),
('2025-03-10', 10, 10),
('2025-03-11', 11, 11),
('2025-03-12', 12, 12),
('2025-03-13', 13, 13),
('2025-03-14', 14, 14),
('2025-03-15', 15, 15),
('2025-03-16', 16, 16),
('2025-03-17', 17, 17),
('2025-03-18', 18, 18),
('2025-03-19', 19, 19),
('2025-03-20', 20, 20);
```

Wishlist_id	date_added	guest_id	property_id
1	2025-03-01	1	1
2	2025-03-02	2	3
3	2025-03-03	3	2
4	2025-03-04	4	4
5	2025-03-05	5	5
6	2025-03-06	6	6
7	2025-03-07	7	7
8	2025-03-08	8	8
9	2025-03-09	9	9
10	2025-03-10	10	10
11	2025-03-11	11	11
12	2025-03-12	12	12
13	2025-03-13	13	13
14	2025-03-14	14	14
15	2025-03-15	15	15
16	2025-03-16	16	16
17	2025-03-17	17	17
18	2025-03-18	18	18
19	2025-03-19	19	19
20	2025-03-20	20	20
NULL	NULL	NULL	NULL

Development & Reflection Phase

Discounts table

SQL code

```sql
-- Discounts table stores information about discounts for properties
CREATE TABLE Discounts (
    Discount_id INT AUTO_INCREMENT PRIMARY KEY, -- Unique identifier for a discount entry
    percentage FLOAT, -- Discount percentage
    start_date DATE, -- Date the discount starts
    end_date DATE, -- Date the discount ends
    property_id INT, -- Unique identifier for a property
    FOREIGN KEY (property_id) REFERENCES Property(Property_id) -- Foreign key referencing Property table
);
```

20 Entries

```sql
INSERT INTO Discounts (percentage, start_date, end_date, property_id)
VALUES
(10, '2025-04-01', '2025-04-08', 1),
(10, '2025-04-02', '2025-04-09', 2),
(15, '2025-04-03', '2025-04-10', 3),
(20, '2025-04-04', '2025-04-11', 4),
(10, '2025-04-05', '2025-04-12', 5),
(30, '2025-04-06', '2025-04-13', 6),
(15, '2025-04-07', '2025-04-14', 7),
(25, '2025-04-08', '2025-04-15', 8),
(40, '2025-04-09', '2025-04-16', 9),
(10, '2025-04-10', '2025-04-17', 10),
(50, '2025-04-11', '2025-04-18', 11),
(20, '2025-04-12', '2025-04-19', 12),
(25, '2025-04-13', '2025-04-20', 13),
(35, '2025-04-14', '2025-04-21', 14),
(10, '2025-04-15', '2025-04-22', 15),
(40, '2025-04-16', '2025-04-23', 16),
(10, '2025-04-17', '2025-04-24', 17),
(45, '2025-04-18', '2025-04-25', 18),
(70, '2025-04-19', '2025-04-26', 19),
(35, '2025-04-20', '2025-04-27', 20);
```

Discount_id	percentage	start_date	end_date	property_id
1	10	2025-04-01	2025-04-08	1
2	10	2025-04-02	2025-04-09	2
3	15	2025-04-03	2025-04-10	3
4	20	2025-04-04	2025-04-11	4
5	10	2025-04-05	2025-04-12	5
6	30	2025-04-06	2025-04-13	6
7	15	2025-04-07	2025-04-14	7
8	25	2025-04-08	2025-04-15	8
9	40	2025-04-09	2025-04-16	9
10	10	2025-04-10	2025-04-17	10
11	50	2025-04-11	2025-04-18	11
12	20	2025-04-12	2025-04-19	12
13	25	2025-04-13	2025-04-20	13
14	35	2025-04-14	2025-04-21	14
15	10	2025-04-15	2025-04-22	15
16	40	2025-04-16	2025-04-23	16
17	10	2025-04-17	2025-04-24	17
18	45	2025-04-18	2025-04-25	18
19	70	2025-04-19	2025-04-26	19
20	35	2025-04-20	2025-04-27	20
NULL	NULL	NULL	NULL	NULL

Dispute table

SQL code

```sql
-- Dispute tables stores information about disputes
CREATE TABLE Dispute (
    Dispute_id INT AUTO_INCREMENT PRIMARY KEY, -- Unique identifier for a dispute
    details TEXT, -- Details of the dispute
    dispute_status ENUM('Open', 'In Progress', 'Resolved', 'Closed'), -- Current status of a dispute (e.g., Open, Resolved)
    created_at TIMESTAMP, -- Date and time the dispute was created
    resolution_date DATE, -- Date the dispute was solved
    booking_id INT, -- Unique identifier for a booking
    admin_id INT, -- Unique identifier for an admin
    FOREIGN KEY (booking_id) REFERENCES Booking(Booking_id), -- Foreign key referencing Booking table
    FOREIGN KEY (admin_id) REFERENCES Adminn(Admin_id) -- Foreign key referencing Admin table
);
```

20 Entries

```sql
INSERT INTO Dispute (details, dispute_status, created_at, resolution_date, booking_id, admin_id)
VALUES
('Guest reported unclean property.', 'Open', '2025-01-05 10:00:00', NULL, 1, 1),
('Host claimed damage to furniture.', 'In Progress', '2025-01-06 12:00:00', NULL, 2, 2),
('Payment dispute for extra charges.', 'Resolved', '2025-01-07 09:30:00', '2025-01-08', 3, 3),
('Guest complained about noisy neighbors.', 'Closed', '2025-01-08 14:00:00', '2025-01-09', 4, 1),
('Host requested compensation for cancellation.', 'In Progress', '2025-01-09 11:00:00', NULL, 5, 4),
('Double booking issue reported.', 'Resolved', '2025-01-10 16:45:00', '2025-01-11', 6, 2),
('Amenities not as described.', 'Open', '2025-01-11 08:00:00', NULL, 7, 5),
('Guest found expired food in the fridge.', 'Closed', '2025-01-12 18:00:00', '2025-01-13', 8, 3),
('Unresponsive host during check-in.', 'In Progress', '2025-01-13 09:15:00', NULL, 9, 6),
('Guest refused to vacate property on time.', 'Open', '2025-01-14 10:30:00', NULL, 10, 4),
('Overcharged for additional guests.', 'Resolved', '2025-01-15 07:00:00', '2025-01-16', 11, 1),
('Host accused guest of theft.', 'In Progress', '2025-01-16 11:30:00', NULL, 12, 7),
('Property was locked, guest couldn't access.', 'Closed', '2025-01-17 15:00:00', '2025-01-18', 13, 5),
('Guest reported mold in the bathroom.', 'Resolved', '2025-01-18 13:20:00', '2025-01-19', 14, 6),
('Unauthorized pet on the property.', 'In Progress', '2025-01-19 17:00:00', NULL, 15, 2),
('Host canceled last minute.', 'Open', '2025-01-20 08:00:00', NULL, 16, 3),
('Dispute over security deposit refund.', 'Closed', '2025-01-21 10:00:00', '2025-01-22', 17, 7),
('Guest claims false advertising of amenities.', 'In Progress', '2025-01-22 14:45:00', NULL, 18, 4),
('Payment issue for additional cleaning charges.', 'Resolved', '2025-01-23 09:00:00', '2025-01-24', 19, 6),
('Host accused of discrimination.', 'Open', '2025-01-24 12:30:00', NULL, 20, 5);
```

Dispute_id	details	dispute_stat...	created_at	resolution_da...	booking_id	admin_id
1	Guest reported unclean property.	Open	2025-01-05 10:00:00	NULL	1	1
2	Host claimed damage to furniture.	In Progress	2025-01-06 12:00:00	NULL	2	2
3	Payment dispute for extra charges.	Resolved	2025-01-07 09:30:00	2025-01-08	3	3
4	Guest complained about noisy neighbors.	Closed	2025-01-08 14:00:00	2025-01-09	4	1
5	Host requested compensation for cancellation.	In Progress	2025-01-09 11:00:00	NULL	5	4
6	Double booking issue reported.	Resolved	2025-01-10 16:45:00	2025-01-11	6	2
7	Amenities not as described.	Open	2025-01-11 08:00:00	NULL	7	5
8	Guest found expired food in the fridge.	Closed	2025-01-12 18:00:00	2025-01-13	8	3
9	Unresponsive host during check-in.	In Progress	2025-01-13 09:15:00	NULL	9	6
10	Guest refused to vacate property on time.	Open	2025-01-14 10:30:00	NULL	10	4
11	Overcharged for additional guests.	Resolved	2025-01-15 07:00:00	2025-01-16	11	1
12	Host accused guest of theft.	In Progress	2025-01-16 11:30:00	NULL	12	7
13	Property was locked, guest couldn't access.	Closed	2025-01-17 15:00:00	2025-01-18	13	5
14	Guest reported mold in the bathroom.	Resolved	2025-01-18 13:20:00	2025-01-19	14	6
15	Unauthorized pet on the property.	In Progress	2025-01-19 17:00:00	NULL	15	2
16	Host canceled last minute.	Open	2025-01-20 08:00:00	NULL	16	3
17	Dispute over security deposit refund.	Closed	2025-01-21 10:00:00	2025-01-22	17	7
18	Guest claims false advertising of amenities.	In Progress	2025-01-22 14:45:00	NULL	18	4
19	Payment issue for additional cleaning charges.	Resolved	2025-01-23 09:00:00	2025-01-24	19	6
20	Host accused of discrimination.	Open	2025-01-24 12:30:00	NULL	20	5
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Development & Reflection Phase

Rules table

SQL code

```sql
-- Rules table stores information about the rules of property
CREATE TABLE Rules (
    Rules_id INT AUTO_INCREMENT PRIMARY KEY, -- Unique identifier for a rule
    rule_description TEXT -- Description of a rule
);
```

20 Entries

```sql
INSERT INTO Rules (rule_description)
VALUES
('Check-in: 4:00 PM - 6:00 PM'),
('Check-in: 12:00 PM - 2:00 PM'),
('Check-in: 1:00 PM - 3:00 PM'),
('Check-in after 3:00 PM'),
('Checkout before 10:00 AM'),
('Checkout before 11:00 AM'),
('Checkout before 12:00 AM'),
('Checkout before 1:00 PM'),
('No pets'),
('No smoking'),
('No parties or events'),
('2 guests maximum'),
('3 guests maximum'),
('4 guests maximum'),
('5 guests maximum'),
('Self check-in with smart lock'),
('Self check-in with lockbox'),
('Commercial photography allowed'),
('Additional rules: Please take off your shoes, slippers are available'),
('Quiet hours: 11:00 PM - 7:00 AM');
```

Rules_id	rule_description
1	Check-in: 4:00 PM - 6:00 PM
2	Check-in: 12:00 PM - 2:00 PM
3	Check-in: 1:00 PM - 3:00 PM
4	Check-in after 3:00 PM
5	Checkout before 10:00 AM
6	Checkout before 11:00 AM
7	Checkout before 12:00 AM
8	Checkout before 1:00 PM
9	No pets
10	No smoking
11	No parties or events
12	2 guests maximum
13	3 guests maximum
14	4 guests maximum
15	5 guests maximum
16	Self check-in with smart lock
17	Self check-in with lockbox
18	Commercial photography all…
19	Additional rules: Please take…
20	Quiet hours: 11:00 PM - 7:00…
NULL	NULL

Property_rules table

SQL code

```sql
-- Property_rules table establishes the relationship between Property and Rules tables
CREATE TABLE Property_Rules (
    Property_id INT, -- Unique identifier for a property
    Rules_id INT, -- Unique identifier for a rule
    PRIMARY KEY (Property_id, Rules_id), -- Composite primary key
    FOREIGN KEY (Property_id) REFERENCES Property(Property_id), -- Foreign key referencing Property table
    FOREIGN KEY (Rules_id) REFERENCES Rules(Rules_id) -- Foreign key referencing Rules table
);
```

20 Entries

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

Property_id	Rules_id
1	3
1	7
1	13
1	18
2	3
2	9
2	12
2	17
3	1
3	10
3	19
3	20
4	2
4	8
4	10
4	19
5	3
5	8
5	16
5	18
6	3
6	7
6	18
6	20
7	1
7	7
7	12
7	16

Property_id	Rules_id
8	1
8	5
8	9
8	14
9	2
9	5
9	16
9	19
10	1
10	8
10	9
10	18
11	1
11	11
11	12
11	16
12	4
12	8
12	11
12	20
13	2
13	7
13	12
13	20
14	3
14	6
14	10
14	16

Property_id	Rules_id
14	6
14	10
14	16
15	3
15	8
15	16
15	18
16	1
16	5
16	10
16	13
17	3
17	6
17	11
17	12
18	2
18	8
18	16
18	18
19	1
19	7
19	10
19	12
20	2
20	8
20	19
20	20
NULL	NULL

Development & Reflection Phase

Neighbourhood table

SQL code

```sql
-- Neighbourhood table stores neighbourhood information
CREATE TABLE Neighbourhood (
    Neighbourhood_id INT AUTO_INCREMENT PRIMARY KEY, -- Unique identifier for a neighbourhood
    neighbourhood_name VARCHAR(255), -- Name of the neighbourhood
    neighbourhood_description TEXT, -- A detailed description of about the neighbourhood's characteristics
    popular_spots TEXT, -- List of popular attractions
    transportation_info TEXT, -- Information about the available transportation options in the neighbourhood
    location_id INT, -- Unique identifier for a location
    FOREIGN KEY (location_id) REFERENCES Location (Location_id) -- Foreign key referencing Location table
);
```

20 Entries

```sql
INSERT INTO Neighbourhood (neighbourhood_name, neighbourhood_description, popular_spots, transportation_info, location_id)
VALUES
('Altstadt', 'Historic old town with Gothic architecture and shopping streets.', 'Cologne Cathedral, Museum Ludwig', 'Trams 1, 5; Metro Line 12', 1),
('Poppelsdorf', 'A peaceful area with botanical gardens and cafes.', 'Poppelsdorf Palace, University of Bonn', 'Bus routes 610, 611', 2),
('Naviglì', 'Trendy district with canals, restaurants, and bars.', 'Naviglio Grande, Porta Ticinese', 'Tram lines 9, 15; Metro Line M2', 3),
('Salamanca', 'Upscale area known for shopping and dining.', 'Plaza de Toros, Calle Serrano', 'Metro Lines 4, 6; Bus 53', 4),
('Trastevere', 'Charming neighborhood with narrow streets and historic landmarks.', 'Santa Maria in Trastevere, Janiculum Hill', 'Tram Line 8, Bus H', 5),
('Kreuzberg', 'Trendy area with art galleries and diverse dining options.', 'East Side Gallery, Tempelhofer Feld', 'U-Bahn Lines U1, U8', 6),
('Montmartre', 'Hilly district known for its artistic history.', 'Sacré-Coeur, Place du Tertre', 'Metro Lines 12, 2', 7),
('Le Marais', 'Fashionable neighborhood with narrow streets and boutiques.', 'Place des Vosges, Musée Carnavalet', 'Metro Lines 1, 8', 8),
('Südstadt', 'Residential area with parks and historic churches.', 'Rheinau Park, Bonn Minster', 'Tram Line 62; Bus 550', 9),
('South End', 'Charming area with brick townhouses and diverse restaurants.', 'Boston Center for the Arts, Tremont Street', 'Subway Orange Line; Bus 43', 10),
('Plaka', 'Picturesque neighborhood with winding streets and neoclassical architecture.', 'Acropolis Museum, Anafiotika', 'Metro Line 3, Bus 230', 11),
('Grünerløkka', 'Hip neighborhood with vintage shops and cafes.', 'Mathallen Food Hall, Sofienberg Park', 'Tram Line 11, Bus 30', 12),
('Nyhavn', 'Colorful waterfront district with historic ships and restaurants.', 'Amalienborg Palace, Nyhavn Canal', 'Metro Line M3, Bus 66', 13),
('Gamla Stan', 'Old Town with cobbled streets and medieval architecture.', 'Royal Palace, Nobel Museum', 'Metro Line 13, Bus 53', 14),
('Inner Harbor', 'Waterfront area with museums and seafood restaurants.', 'National Aquarium, Federal Hill Park', 'Light Rail Line 1, Bus 64', 15),
('Palermo', 'Chic district with parks, boutiques, and nightlife.', 'Bosques de Palermo, Plaza Serrano', 'Subte Line D, Bus 152', 16),
('Sultanahmet', 'Historic area with landmarks from the Ottoman Empire.', 'Hagia Sophia, Blue Mosque', 'Metro Line M1A, Tram T1', 17),
('Midtown', 'Dynamic urban district with skyscrapers and cultural landmarks.', 'Fox Theatre, Atlanta Botanical Garden', 'MARTA Red and Gold Lines', 18),
('Scheveningen', 'Seaside resort with beaches and a pier.', 'Scheveningen Beach, Sea Life', 'Tram Line 1, Bus 22', 19),
('Kallio', 'Trendy area with vintage shops and a lively nightlife.', 'Hakaniemi Market, Kallio Church', 'Tram Lines 3, 9', 20);
```

Neighbourhood...	neighbourhood_na...	neighbourhood_description
1	Altstadt	Historic old town with Gothic architecture and sh...
2	Poppelsdorf	A peaceful area with botanical gardens and cafes
3	Navigli	Trendy district with canals, restaurants, and bars.
4	Salamanca	Upscale area known for shopping and dining.
5	Trastevere	Charming neighborhood with narrow streets and...
6	Kreuzberg	Trendy area with art galleries and diverse dining...
7	Montmartre	Hilly district known for its artistic history.
8	Le Marais	Fashionable neighborhood with narrow streets a...
9	Südstadt	Residential area with parks and historic churches.
10	South End	Charming area with brick townhouses and diver...
11	Plaka	Picturesque neighborhood with winding streets...
12	Grünerløkka	Hip neighborhood with vintage shops and cafes.
13	Nyhavn	Colorful waterfront district with historic ships an...
14	Gamla Stan	Old Town with cobbled streets and medieval arc...
15	Inner Harbor	Waterfront area with museums and seafood rest...
16	Palermo	Chic district with parks, boutiques, and nightlife.
17	Sultanahmet	Historic area with landmarks from the Ottoman...
18	Midtown	Dynamic urban district with skyscrapers and cult...
19	Scheveningen	Seaside resort with beaches and a pier.
20	Kallio	Trendy area with vintage shops and a lively nigh...
NULL	NULL	NULL

popular_spots	transportation_info	location_id
Cologne Cathedral, Museum Ludwig	Trams 1, 5; Metro Line 12	1
Poppelsdorf Palace, University of Bonn	Bus routes 610, 611	2
Naviglio Grande, Porta Ticinese	Tram lines 9, 15; Metro Line M2	3
Plaza de Toros, Calle Serrano	Metro Lines 4, 6; Bus 53	4
Santa Maria in Trastevere, Janiculum Hill	Tram Line 8, Bus H	5
East Side Gallery, Tempelhofer Feld	U-Bahn Lines U1, U8	6
Sacré-Coeur, Place du Tertre	Metro Lines 12, 2	7
Place des Vosges, Musée Carnavalet	Metro Lines 1, 8	8
Rheinau Park, Bonn Minster	Tram Line 62; Bus 550	9
Boston Center for the Arts, Tremont Street	Subway Orange Line; Bus 43	10
Acropolis Museum, Anafiotika	Metro Line 3, Bus 230	11
Mathallen Food Hall, Sofienberg Park	Tram Line 11, Bus 30	12
Amalienborg Palace, Nyhavn Canal	Metro Line M3, Bus 66	13
Royal Palace, Nobel Museum	Metro Line 13, Bus 53	14
National Aquarium, Federal Hill Park	Light Rail Line 1, Bus 64	15
Bosques de Palermo, Plaza Serrano	Subte Line D, Bus 152	16
Hagia Sophia, Blue Mosque	Metro Line M1A, Tram T1	17
Fox Theatre, Atlanta Botanical Garden	MARTA Red and Gold Lines	18
Scheveningen Beach, Sea Life	Tram Line 1, Bus 22	19
Hakaniemi Market, Kallio Church	Tram Lines 3, 9	20
NULL	NULL	NULL

Messages table

SQL code

```sql
-- Messages table stores infromation about messages between a host and a guest
CREATE TABLE Messages (
    Message_id INT AUTO_INCREMENT  PRIMARY KEY, -- Unique identifier for a message
    sender_id INT, -- Unique identifier for a sender (guest)
    receiver_id INT, -- Unique identifier for a receiver (host)
    message_content TEXT, -- Content of the message sent between users
    sent_at TIMESTAMP, -- Date and time the message was sent
    is_read BOOLEAN, -- Indicates if a message has been read (true/false)
    FOREIGN KEY (sender_id) REFERENCES Guest(Guest_id), -- Foreign key referencing Guest table
    FOREIGN KEY (receiver_id) REFERENCES Hostt(Host_id) -- Foreign key referencing Hostt table
);
```

20 Entries

```sql
INSERT INTO Messages (sender_id, receiver_id, message_content, sent_at, is_read)
VALUES
(1, 20, 'Hi, I would like to know more about the property availability for next month.', '2025-01-01 09:30:00', FALSE),
(2, 19, 'Can I get a discount for a long-term stay?', '2025-01-02 14:45:00', TRUE),
(3, 18, 'The property is available from the 15th to the 30th of next month.', '2025-01-01 10:00:00', TRUE),
(4, 17, 'Could you please provide additional photos of the property?', '2025-01-03 08:15:00', FALSE),
(5, 16, 'Can I check in earlier than 3 PM?', '2025-01-04 13:20:00', TRUE),
(6, 15, 'Do you allow pets on the property?', '2025-01-05 17:10:00', FALSE),
(7, 14, 'Thank you for confirming the reservation!', '2025-01-06 12:30:00', TRUE),
(8, 13, 'Can I cancel my booking? What are the penalties?', '2025-01-07 16:45:00', FALSE),
(9, 12, 'How far is the property from the city center?', '2025-01-08 10:50:00', TRUE),
(10, 11, 'I noticed an issue with the property during my stay. Please assist.', '2025-01-08 19:15:00', FALSE),
(11, 10, 'Thank you for your inquiry! I will get back to you shortly.', '2025-01-01 09:35:00', TRUE),
(12, 9, 'The property is currently not available for those dates.', '2025-01-02 15:00:00', TRUE),
(13, 8, 'Sure, I will send the additional photos shortly.', '2025-01-03 08:45:00', TRUE),
(14, 7, 'Unfortunately, early check-in is not possible.', '2025-01-04 13:50:00', TRUE),
(15, 6, 'Yes, we allow pets with an additional cleaning fee.', '2025-01-05 17:40:00', TRUE),
(16, 5, 'You are welcome! Enjoy your stay.', '2025-01-06 13:00:00', TRUE),
(17, 4, 'Cancellation penalties depend on the policy. Please check the details.', '2025-01-07 17:10:00', TRUE),
(18, 3, 'The property is about 10 minutes by car from the city center.', '2025-01-08 11:15:00', TRUE),
(19, 2, 'Thank you for reporting the issue. We will address it immediately.', '2025-01-08 19:40:00', TRUE),
(20, 1, 'I will be available to answer your inquiries tomorrow.', '2025-01-01 18:25:00', TRUE);
```

Message_id	sender_id	receiver_id	message_content	sent_at	is_read
1	1	20	Hi, I would like to know more about the property...	2025-01-01 09:30:00	0
2	2	19	Can I get a discount for a long-term stay?	2025-01-02 14:45:00	1
3	3	18	The property is available from the 15th to the 30...	2025-01-01 10:00:00	1
4	4	17	Could you please provide additional photos of th...	2025-01-03 08:15:00	0
5	5	16	Can I check in earlier than 3 PM?	2025-01-04 13:20:00	1
6	6	15	Do you allow pets on the property?	2025-01-05 17:10:00	0
7	7	14	Thank you for confirming the reservation!	2025-01-06 12:30:00	1
8	8	13	Can I cancel my booking? What are the penalties?	2025-01-07 16:45:00	0
9	9	12	How far is the property from the city center?	2025-01-08 10:50:00	1
10	10	11	I noticed an issue with the property during my st...	2025-01-08 19:15:00	0
11	11	10	Thank you for your inquiry! I will get back to you...	2025-01-01 09:35:00	1
12	12	9	The property is currently not available for those...	2025-01-02 15:00:00	1
13	13	8	Sure, I will send the additional photos shortly.	2025-01-03 08:45:00	1
14	14	7	Unfortunately, early check-in is not possible.	2025-01-04 13:50:00	1
15	15	6	Yes, we allow pets with an additional cleaning fee.	2025-01-05 17:40:00	1
16	16	5	You are welcome! Enjoy your stay.	2025-01-06 13:00:00	1
17	17	4	Cancellation penalties depend on the policy. Ple...	2025-01-07 17:10:00	1
18	18	3	The property is about 10 minutes by car from th...	2025-01-08 11:15:00	1
19	19	2	Thank you for reporting the issue. We will addre...	2025-01-08 19:40:00	1
20	20	1	I will be available to answer your inquiries tomor...	2025-01-01 18:25:00	1
NULL	NULL	NULL	NULL	NULL	NULL

Test cases

Case 1

- Guest Books a property

Expected scenario:
1. Guest searches for available properties in Cologne, Germany

2. The guest selects a property and books it for a specific date

3. The booking is recorded in the database, and the status is updated to “Paid.”

Case 2

- Booking with payment details

Expected scenario:

Retrieve booking information along with payment details for completed transactions.

Case 1

Step 1

```sql
-- Search Properties in Cologne
SELECT
    Property.title AS Property_Title,
    Property.property_description AS Property_Description,
    Property.price_per_night AS Price_Per_Night,
    Location.city AS City
FROM
    Property
JOIN Location ON Property.location_id = Location.Location_id
WHERE
    Location.city = 'Cologne';
```

Property_Title	Property_Description	Price_Per_Night	City
Bright Apartment in Bonn	Bright apartment located in the centre of Bonn	110.00	Cologne

Step 2

```sql
-- Insert Booking
INSERT INTO Booking (guest_id, property_id, check_in_date, check_out_date, total_price, payment_status)
VALUES (2, 1, '2025-01-20', '2025-01-25', 500.00, 'Pending');
```

Booking_id	check_in_da...	check_out_da...	total_pri...	payment_stat...	booking_date	guest_id	host_id	property_id
1	2025-01-01	2025-01-06	550.00	Pending	2024-12-30 12:30:00	1	20	14
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Step 3

```sql
-- Update payment Status
UPDATE Booking
SET payment_status = 'Paid'
WHERE Booking_id = 1;
```

Booking_id	check_in_da...	check_out_da...	total_pri...	payment_stat...	booking_date	guest_id	host_id	property_id
1	2025-01-01	2025-01-06	550.00	Paid	2024-12-30 12:30:00	1	20	14
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Case 2

```sql

-- Retrieve booking information along with payment details
SELECT
    Booking.Booking_id,
    Guest.full_name AS Guest_Name,
    Payment.amount AS Payment_Amount,
    Payment.payment_date AS Payment_Date
FROM
    Booking
JOIN Guest ON Booking.guest_id = Guest.Guest_id
JOIN Payment ON Booking.Booking_id = Payment.booking_id
WHERE
    Payment.payment_status = 'Completed';

```

Booking_id	Guest_Name	Payment_Amount	Payment_Date
1	Alice Brown	550.00	2024-12-30 12:30:00
3	Cathy Davis	306.00	2025-01-02 14:45:00
4	Emily Carter	600.00	2025-01-03 17:20:00
5	James Anderson	7800.00	2025-01-04 11:00:00
7	Michael Harris	582.00	2025-01-06 13:50:00
8	Sophia Taylor	670.00	2025-01-07 08:30:00
9	William Rickson	835.00	2025-01-08 16:45:00
12	Lucas Moore	6990.00	2025-01-11 10:20:00
14	Henry Lewis	10200.00	2025-01-13 13:10:00
15	Mia Young	15235.00	2025-01-14 11:55:00
17	Grace White	428.00	2025-03-16 18:30:00
18	Benjamin Lee	9495.00	2025-01-17 08:20:00
19	Amelia Thomas	1515.00	2025-01-18 14:10:00

The test cases simulate realistic user interactions with the Airbnb database. In the first case, a guest searches for properties, books one in Cologne, and updates the payment status from “Pending” to “Paid,” validating the booking workflow and status updates. The second case retrieves booking and payment details for completed transactions, ensuring accurate linkage between bookings and payments while validating data integrity and relationship consistency.

Finalization phase

Abstract

The Airbnb Database Management System is a robust solution designed to manage and streamline operations for an online rental platform. This database system facilitates seamless interactions between guests, hosts, administrators, and properties, ensuring data integrity and optimized performance. The system adheres to a normalized design, minimizing redundancy and maintaining efficient data retrieval processes. Implemented in SQL, it supports scalable and accurate handling of real-world scenarios related to property bookings, payments, cancellations, and user communications.

Functionality

1. Property listings
    a. Hosts can add properties with detailed attributes such as location, amenities, and neighbourhood information.
    b. Neighbourhoods include data on popular spots and transportation info, enriching user experience
2. User management
    a. Separate tables for guests, hosts and administrators, ensuring role-based access and data isolation
    b. Guest and host tables profiles include essential details for personalised interactions
3. Booking and payments
    a. Booking table tracks reservations with properties linked to guests and hosts
    b. Payment integration ensures smooth transactions, tracking amounts, payment dates and statutes
4. Cancellation and disputes
    a. Cancellation policies enforce platform rules, while disputes handle post-booking issues
5. Messaging system
    a. Enables communication between guests and hosts with real-time message tracking and read statuses

Key features

- Relational integrity across all tables with well-defined foreign key constraints
- Test cases validating booking workflows, payment linkage and cancellation scenarios

Finalization phase

- Comprehensive indexing for faster query execution

Metadata

Number of tables: 27 tables

Entries: each table contains 20 sample entries for testing and demonstration purposes

Database size: approximately 2MB

Summary

This database system offers a complete solution for managing Airbnb-like operations, ensuring scalability, reliability and ease of use. With its modular design, it supports future enhancements, such as advanced reporting and analytics. All SQL scripts, test cases and documentation have been prepared for installation and use.