

# The Development Phase

Implementation of the database management system for the Airbnb Use Case

Reported by: Muhannad Al-Ahmad



# Project Overview



## Purpose

To design and Implement relational database for managing Airbnb operations, incorporating tutor feedback from the conception phase to enhance functionality.



## Adjustments

- Linked Payment with User.
- Integrated contact info into the User table.
- Connected Location with User.
- Added HouseRules and Neighborhood tables.
- Added AccommodationHouseRules table to manage the many-to-many relationship between accommodations and House rules.



- The next slides will showcase the entities along with the SQL statements used to create them and populate them with data.
- Three test cases established to examine the performance and validate the relationships within the database

The background features a minimalist, abstract illustration of a data center. It consists of two sets of server racks, each represented by a stack of rectangular blocks in shades of purple, teal, and light blue. In the upper left, a smartphone is shown from a side-on perspective, its screen displaying a small icon of three dots. Lines connect this icon to the top of the leftmost server rack. The entire scene is set against a light gray gradient.

Showcase  
The Entities



```

CREATE TABLE IF NOT EXISTS `airbnb_db`.`user` (
  `UserID` INT NOT NULL AUTO_INCREMENT,          -- Unique identifier for each user, auto-incremented
  `Name` VARCHAR(100) NOT NULL,                  -- Full name of the user
  `Email` VARCHAR(100) NOT NULL,                  -- Unique email for user contact and login
  `PhoneNumber` VARCHAR(15) NOT NULL,             -- Unique phone number for user contact
  `ReferredByUserID` INT(11) NULL DEFAULT NULL,  -- References another user who referred this user (nullable)
  `LocationID` INT NOT NULL,                     -- Foreign key linking to the Location table
  PRIMARY KEY (`UserID`),
  UNIQUE INDEX `Email` (`Email` ASC) VISIBLE,    -- Ensures each email is unique
  UNIQUE INDEX `PhoneNumber` (`PhoneNumber` ASC) VISIBLE, -- Ensures each phone number is unique
  INDEX `ReferredByUserID` (`ReferredByUserID` ASC) VISIBLE, -- Index for foreign key reference
  INDEX `LocationID` (`LocationID` ASC) VISIBLE, -- Index for foreign key reference
  CONSTRAINT `user_ibfk_1`                      -- Defines the foreign key constraint for ReferredByUserID
    FOREIGN KEY (`ReferredByUserID`)
    REFERENCES `airbnb_db`.`user` (`UserID`)
  ON DELETE SET NULL                            -- If referred user is deleted, set ReferredByUserID to NULL
  ON UPDATE CASCADE,                           -- Updates ReferredByUserID if UserID changes
  CONSTRAINT `user_ibfk_2`                      -- Defines the foreign key constraint for LocationID
    FOREIGN KEY (`LocationID`)
    REFERENCES `airbnb_db`.`location` (`LocationID`)
  ON DELETE CASCADE,                           -- Deletes user if the related location is deleted
  ON UPDATE CASCADE,                           -- Updates LocationID in the User table if it changes in Location
) ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;                  -- Supports multi-language characters and emojis
                                                -- Case-insensitive collation for multi-language support

```

```

-- Sample Data for User Table
INSERT INTO `airbnb_db`.`user` (Name, Email, PhoneNumber, ReferredByUserID, LocationID) VALUES
('John Doe', 'johndoe@example.com', '1234567890', NULL, 1),
('Jane Smith', 'janeshmith@example.com', '0987654321', NULL, 2),
('Alice Johnson', 'alicej@example.com', '2345678901', 1, 3),
('Bob Brown', 'bobbrown@example.com', '3456789012', 2, 4),
('Chris Green', 'chrisgreen@example.com', '4567890123', 1, 5),
('Diana White', 'dianawhite@example.com', '5678901234', NULL, 6),
('Evan Black', 'evanblack@example.com', '6789012345', 3, 7),
('Fiona Gray', 'fionagray@example.com', '7890123456', NULL, 8),
('George Blue', 'georgeblue@example.com', '8901234567', 4, 9),
('Holly Yellow', 'hollyyellow@example.com', '9012345678', NULL, 10),
('Ivy Red', 'ivyred@example.com', '1230987654', 5, 11),
('Jake Violet', 'jakeviolet@example.com', '2345678091', NULL, 12),
('Kelly Orange', 'kellyorange@example.com', '3456780129', 3, 13),
('Liam Pink', 'liampink@example.com', '4567890231', NULL, 14),
('Mona Purple', 'monapurple@example.com', '5678902314', 2, 15),
('Nina Brown', 'ninabrown@example.com', '6789012346', NULL, 16),
('Oscar Gray', 'oscargray@example.com', '7890123459', 1, 17),
('Paul Silver', 'paulsilver@example.com', '8901234561', NULL, 18),
('Quinn Gold', 'quinngold@example.com', '9012345672', 10, 19),
('Rachel Bronze', 'rachelbronze@example.com', '1123456789', 12, 20);

```

UserID	Name	Email	PhoneNumber	ReferredByUserID	LocationID
1	John Doe	johndoe@example.com	1234567890	<null>	1
2	Jane Smith	janeshmith@example.com	0987654321	<null>	2
3	Alice Johnson	alicej@example.com	2345678901	1	3
4	Bob Brown	bobbrown@example.com	3456789012	2	4
5	Chris Green	chrisgreen@example.com	4567890123	1	5
6	Diana White	dianawhite@example.com	5678901234	<null>	6
7	Evan Black	evanblack@example.com	6789012345	3	7
8	Fiona Gray	fionagray@example.com	7890123456	<null>	8
9	George Blue	georgeblue@example.com	8901234567	4	9
10	Holly Yellow	hollyyellow@example.com	9012345678	<null>	10
11	Ivy Red	ivyred@example.com	1230987654	5	11
12	Jake Violet	jakeviolet@example.com	2345678091	<null>	12
13	Kelly Orange	kellyorange@example.com	3456780129	3	13
14	Liam Pink	liampink@example.com	4567890231	<null>	14
15	Mona Purple	monapurple@example.com	5678902314	2	15
16	Nina Brown	ninabrown@example.com	6789012346	<null>	16
17	Oscar Gray	oscargray@example.com	7890123459	1	17
18	Paul Silver	paulsilver@example.com	8901234561	<null>	18
19	Quinn Gold	quinngold@example.com	9012345672	10	19
20	Rachel Bronze	rachelbronze@example.com	1123456789	12	20



```
CREATE TABLE IF NOT EXISTS `airbnb_db`.`hostprofile` (
    `HostID` INT NOT NULL, -- Unique identifier for each host, linked to a user
    `UserID` INT NOT NULL, -- Foreign key linking the host to the User table
    `Rating` DECIMAL(3,2) NULL DEFAULT NULL, -- Host rating, stored as a decimal with 2 decimal places
    `VerificationStatus` ENUM('Verified', 'Pending', 'Unverified') NULL DEFAULT 'Pending', -- Verification status for the host
    PRIMARY KEY (`HostID`), -- Sets HostID as the primary key
    UNIQUE INDEX `UserID` (`UserID` ASC) VISIBLE, -- Ensures each UserID is unique in HostProfile
    CONSTRAINT `hostprofile_ibfk_1` -- Defines the foreign key constraint for UserID
        FOREIGN KEY (`UserID`)
        REFERENCES `airbnb_db`.`user` (`UserID`)
        ON DELETE CASCADE -- If user is deleted, delete associated host profile
        ON UPDATE CASCADE -- If user ID is updated, update the reference in host profile
) ENGINE = InnoDB
```

```
-- Sample Data for HostProfile Table
INSERT INTO `airbnb_db`.`hostprofile` (HostID, UserID, Rating, VerificationStatus) VALUES
( HostID 1, UserID 1, Rating 4.5, VerificationStatus 'Verified'),
( HostID 2, UserID 2, Rating 4.2, VerificationStatus 'Pending'),
( HostID 3, UserID 3, Rating 4.8, VerificationStatus 'Verified'),
( HostID 4, UserID 4, Rating 3.9, VerificationStatus 'Unverified'),
( HostID 5, UserID 5, Rating 4.1, VerificationStatus 'Verified'),
( HostID 6, UserID 6, Rating 4.3, VerificationStatus 'Verified'),
( HostID 7, UserID 7, Rating 4.0, VerificationStatus 'Pending'),
( HostID 8, UserID 8, Rating 4.7, VerificationStatus 'Verified'),
( HostID 9, UserID 9, Rating 3.5, VerificationStatus 'Unverified'),
( HostID 10, UserID 10, Rating 4.6, VerificationStatus 'Verified'),
( HostID 11, UserID 11, Rating 4.2, VerificationStatus 'Pending'),
( HostID 12, UserID 12, Rating 4.8, VerificationStatus 'Verified'),
( HostID 13, UserID 13, Rating 3.9, VerificationStatus 'Verified'),
( HostID 14, UserID 14, Rating 4.3, VerificationStatus 'Pending'),
( HostID 15, UserID 15, Rating 4.5, VerificationStatus 'Unverified'),
( HostID 16, UserID 16, Rating 4.1, VerificationStatus 'Verified'),
( HostID 17, UserID 17, Rating 4.4, VerificationStatus 'Verified'),
( HostID 18, UserID 18, Rating 4.2, VerificationStatus 'Pending'),
( HostID 19, UserID 19, Rating 4.9, VerificationStatus 'Verified'),
( HostID 20, UserID 20, Rating 3.8, VerificationStatus 'Unverified');
```

HostID	UserID	Rating	VerificationStatus
1	1	1	4.50 Verified
2	2	2	4.20 Pending
3	3	3	4.80 Verified
4	4	4	3.90 Unverified
5	5	5	4.10 Verified
6	6	6	4.30 Verified
7	7	7	4.00 Pending
8	8	8	4.70 Verified
9	9	9	3.50 Unverified
10	10	10	4.60 Verified
11	11	11	4.20 Pending
12	12	12	4.80 Verified
13	13	13	3.90 Unverified
14	14	14	4.30 Pending
15	15	15	4.50 Unverified
16	16	16	4.10 Verified
17	17	17	4.40 Verified
18	18	18	4.20 Pending
19	19	19	4.90 Verified
20	20	20	3.80 Unverified



# Location

This table contains geographical location details, including city and country.

```
CREATE TABLE IF NOT EXISTS `airbnb_db`.`location` (
    `LocationID` INT NOT NULL AUTO_INCREMENT,
    `City` VARCHAR(100) NOT NULL,
    `Country` VARCHAR(100) NOT NULL,
    `PostalCode` VARCHAR(10) NULL DEFAULT NULL,
    PRIMARY KEY (`LocationID`)
) ENGINE = InnoDB

-- Unique identifier for each location, auto-incremented
-- Name of the city
-- Name of the country
-- Postal code of the location, nullable
-- Sets LocationID as the primary key
```

```
-- Sample Data for Location Table
INSERT INTO `airbnb_db`.`location` (City, Country, PostalCode) VALUES
('New York', 'USA', '10001'),
('Los Angeles', 'USA', '90001'),
('Chicago', 'USA', '60601'),
('Houston', 'USA', '77001'),
('Phoenix', 'USA', '85001'),
('Philadelphia', 'USA', '19101'),
('San Antonio', 'USA', '78201'),
('San Diego', 'USA', '92101'),
('Dallas', 'USA', '75201'),
('San Jose', 'USA', '95101'),
('Austin', 'USA', '73301'),
('Jacksonville', 'USA', '32099'),
('Fort Worth', 'USA', '76101'),
('Columbus', 'USA', '43085'),
('Charlotte', 'USA', '28201'),
('San Francisco', 'USA', '94101'),
('Indianapolis', 'USA', '46201'),
('Seattle', 'USA', '98101'),
('Denver', 'USA', '80201'),
('Washington', 'USA', '20001');
```

	LocationID	City	Country	PostalCode
1	1	New York	USA	10001
2	2	Los Angeles	USA	90001
3	3	Chicago	USA	60601
4	4	Houston	USA	77001
5	5	Phoenix	USA	85001
6	6	Philadelphia	USA	19101
7	7	San Antonio	USA	78201
8	8	San Diego	USA	92101
9	9	Dallas	USA	75201
10	10	San Jose	USA	95101
11	11	Austin	USA	73301
12	12	Jacksonville	USA	32099
13	13	Fort Worth	USA	76101
14	14	Columbus	USA	43085
15	15	Charlotte	USA	28201
16	16	San Francisco	USA	94101
17	17	Indianapolis	USA	46201
18	18	Seattle	USA	98101
19	19	Denver	USA	80201
20	20	Washington	USA	20001



# Accommodation

This table stores accommodation details, including descriptions, prices, hosts, locations, and hierarchical relationships.

```
CREATE TABLE IF NOT EXISTS `airbnb_db`.`accommodation` (
  `AccommodationID` INT NOT NULL AUTO_INCREMENT,
  `HostID` INT NULL,
  `LocationID` INT NULL,
  `Description` TEXT NOT NULL,
  `PricePerNight` DECIMAL(10,2) NOT NULL,
  `ParentAccommodationID` INT(11) NULL DEFAULT NULL,
  PRIMARY KEY (`AccommodationID`),
  INDEX `ParentAccommodationID` (`ParentAccommodationID` ASC) VISIBLE, -- Index for self-referencing parent accommodation
  INDEX `accommodation_ibfk_1` (`HostID` ASC) VISIBLE, -- Index for HostID foreign key
  INDEX `accommodation_ibfk_2` (`LocationID` ASC) VISIBLE, -- Index for LocationID foreign key
  CONSTRAINT `accommodation_ibfk_1`
    FOREIGN KEY (`HostID`)
      REFERENCES `airbnb_db`.`hostprofile` (`HostID`)
      ON DELETE RESTRICT
      ON UPDATE CASCADE,
  CONSTRAINT `accommodation_ibfk_2`
    FOREIGN KEY (`LocationID`)
      REFERENCES `airbnb_db`.`location` (`LocationID`)
      ON DELETE SET NULL
      ON UPDATE CASCADE,
  CONSTRAINT `accommodation_ibfk_3`
    FOREIGN KEY (`ParentAccommodationID`)
      REFERENCES `airbnb_db`.`accommodation` (`AccommodationID`)
      ON DELETE SET NULL
      ON UPDATE CASCADE
) ENGINE = InnoDB
```

```
-- Sample Data for Accommodation Table
INSERT INTO `airbnb_db`.`accommodation` (HostID, LocationID, Description, PricePerNight, ParentAccommodationID) VALUES
( HostID 1, LocationID 1, Description 'Cozy studio in the heart of New York', PricePerNight 120.00, ParentAccommodationID NULL),
( HostID 2, LocationID 2, Description 'Modern apartment with a sea view in Los Angeles', PricePerNight 150.00, ParentAccommodationID NULL),
( HostID 3, LocationID 3, Description 'Spacious loft in downtown Chicago', PricePerNight 130.00, ParentAccommodationID NULL),
( HostID 4, LocationID 4, Description 'Luxury villa in Houston', PricePerNight 200.00, ParentAccommodationID NULL),
( HostID 5, LocationID 5, Description 'Charming home in Phoenix with a private pool', PricePerNight 140.00, ParentAccommodationID NULL),
( HostID 6, LocationID 6, Description 'Historic townhouse in Philadelphia', PricePerNight 110.00, ParentAccommodationID NULL),
( HostID 7, LocationID 7, Description 'Riverside apartment in San Antonio', PricePerNight 115.00, ParentAccommodationID NULL),
( HostID 8, LocationID 8, Description 'Beachfront condo in San Diego', PricePerNight 175.00, ParentAccommodationID NULL),
( HostID 9, LocationID 9, Description 'High-rise apartment in Dallas with city views', PricePerNight 160.00, ParentAccommodationID NULL),
( HostID 10, LocationID 10, Description 'Elegant guesthouse in San Jose', PricePerNight 135.00, ParentAccommodationID NULL),
( HostID 11, LocationID 11, Description 'Stylish flat in Austin', PricePerNight 125.00, ParentAccommodationID NULL),
( HostID 12, LocationID 12, Description 'Quaint bungalow in Jacksonville', PricePerNight 105.00, ParentAccommodationID NULL),
( HostID 13, LocationID 13, Description 'Suburban home in Fort Worth', PricePerNight 95.00, ParentAccommodationID NULL),
( HostID 14, LocationID 14, Description 'Modern duplex in Columbus', PricePerNight 100.00, ParentAccommodationID NULL),
( HostID 15, LocationID 15, Description 'Family-friendly house in Charlotte', PricePerNight 110.00, ParentAccommodationID NULL),
( HostID 16, LocationID 16, Description 'Penthouse in San Francisco', PricePerNight 250.00, ParentAccommodationID NULL),
( HostID 17, LocationID 17, Description 'Downtown apartment in Indianapolis', PricePerNight 120.00, ParentAccommodationID NULL),
( HostID 18, LocationID 18, Description 'Studio in Seattle with mountain views', PricePerNight 130.00, ParentAccommodationID NULL),
( HostID 19, LocationID 19, Description 'Eco-friendly cabin in Denver', PricePerNight 145.00, ParentAccommodationID NULL),
( HostID 20, LocationID 20, Description 'Classic row house in Washington, D.C.', PricePerNight 155.00, ParentAccommodationID NULL);
```

AccommodationID	HostID	LocationID	Description	PricePerNight	ParentAccommodationID
1	1	1	Cozy studio in the heart of New York	120.00	<null>
2	2	2	Modern apartment with a sea view in Los Angeles	150.00	<null>
3	3	3	Spacious loft in downtown Chicago	130.00	<null>
4	4	4	Luxury villa in Houston	200.00	<null>
5	5	5	Charming home in Phoenix with a private pool	140.00	<null>
6	6	6	Historic townhouse in Philadelphia	110.00	<null>
7	7	7	Riverside apartment in San Antonio	115.00	<null>
8	8	8	Beachfront condo in San Diego	175.00	<null>
9	9	9	High-rise apartment in Dallas with city views	160.00	<null>
10	10	10	Elegant guesthouse in San Jose	135.00	<null>
11	11	11	Stylish flat in Austin	125.00	<null>
12	12	12	Quaint bungalow in Jacksonville	105.00	<null>
13	13	13	Suburban home in Fort Worth	95.00	<null>
14	14	14	Modern duplex in Columbus	100.00	<null>
15	15	15	Family-friendly house in Charlotte	110.00	<null>
16	16	16	Penthouse in San Francisco	250.00	<null>
17	17	17	Downtown apartment in Indianapolis	120.00	<null>
18	18	18	Studio in Seattle with mountain views	130.00	<null>
19	19	19	Eco-friendly cabin in Denver	145.00	<null>
20	20	20	Classic row house in Washington, D.C.	155.00	<null>



# Amenities

Stores available amenities for accommodations (e.g., WiFi, parking).

```
CREATE TABLE IF NOT EXISTS `airbnb_db`.`amenities` (
    `AmenityID` INT NOT NULL AUTO_INCREMENT,
    `AmenityName` VARCHAR(50) NOT NULL,
    PRIMARY KEY (`AmenityID`),
    UNIQUE INDEX `AmenityName` (`AmenityName` ASC) VISIBLE -- Ensures each amenity name is unique
) ENGINE = InnoDB
```

```
-- Sample Data for Amenities Table
INSERT INTO `airbnb_db`.`amenities` (AmenityName) VALUES
(AmenityName 'WiFi'),
(AmenityName 'Air Conditioning'),
(AmenityName 'Heating'),
(AmenityName 'TV'),
(AmenityName 'Kitchen'),
(AmenityName 'Washer'),
(AmenityName 'Dryer'),
(AmenityName 'Free Parking'),
(AmenityName 'Gym'),
(AmenityName 'Pool'),
(AmenityName 'Hot Tub'),
(AmenityName 'Breakfast Included'),
(AmenityName 'Pets Allowed'),
(AmenityName 'Smoking Allowed'),
(AmenityName 'Essentials'),
(AmenityName 'Fireplace'),
(AmenityName 'Laptop-friendly Workspace'),
(AmenityName 'Private Entrance'),
(AmenityName 'Beach Access'),
(AmenityName 'Hair Dryer');
```

AmenityID	AmenityName
1	2 WiFi
2	19 Beach Access
3	12 Breakfast Included
4	7 Heating
5	15 Essentials
6	16 Fireplace
7	8 Free Parking
8	9 Gym
9	20 Hair Dryer
10	3 WiFi
11	11 Hot Tub
12	5 Kitchen
13	17 Laptop-friendly Workspace
14	13 Pets Allowed
15	10 Pool
16	18 Private Entrance
17	14 Smoking Allowed
18	4 TV
19	6 Washer
20	1 Air Conditioning



# AccommodationAmenities

This table links accommodations with their available amenities.

```
CREATE TABLE IF NOT EXISTS `airbnb_db`.`accommodationamenities` (
    `AccommodationID` INT NOT NULL, -- Foreign key linking to Accommodation, representing the accommodation
    `AmenityID` INT NOT NULL, -- Foreign key linking to Amenities, representing the amenity
    PRIMARY KEY (`AccommodationID`, `AmenityID`), -- Composite primary key combining AccommodationID and AmenityID
    INDEX `accommodationamenities_ibfk_2`(`AmenityID` ASC) VISIBLE,
    CONSTRAINT `accommodationamenities_ibfk_1` -- Defines the foreign key constraint for AccommodationID
        FOREIGN KEY (`AccommodationID`)
        REFERENCES `airbnb_db`.`accommodation`(`AccommodationID`)
        ON DELETE CASCADE -- Deletes entries if the related accommodation is deleted
        ON UPDATE CASCADE, -- Updates entries if the related accommodation ID changes
    CONSTRAINT `accommodationamenities_ibfk_2` -- Defines the foreign key constraint for AmenityID
        FOREIGN KEY (`AmenityID`)
        REFERENCES `airbnb_db`.`amenities`(`AmenityID`)
        ON DELETE CASCADE -- Deletes entries if the related amenity is deleted
        ON UPDATE CASCADE -- Updates entries if the related amenity ID changes
) ENGINE = InnoDB
```

```
-- Sample Data for AccommodationAmenities Table
INSERT INTO `airbnb_db`.`accommodationamenities` (AccommodationID, AmenityID) VALUES
( AccommodationID 1, AmenityID 1), -- Accommodation 1 has WiFi
( AccommodationID 1, AmenityID 2), -- Accommodation 1 has Air Conditioning
( AccommodationID 1, AmenityID 5), -- Accommodation 1 has Kitchen
( AccommodationID 2, AmenityID 1), -- Accommodation 2 has WiFi
( AccommodationID 2, AmenityID 3), -- Accommodation 2 has Heating
( AccommodationID 2, AmenityID 6), -- Accommodation 2 has Washer
( AccommodationID 3, AmenityID 4), -- Accommodation 3 has TV
( AccommodationID 3, AmenityID 5), -- Accommodation 3 has Kitchen
( AccommodationID 3, AmenityID 7), -- Accommodation 3 has Dryer
( AccommodationID 4, AmenityID 1), -- Accommodation 4 has WiFi
( AccommodationID 4, AmenityID 9), -- Accommodation 4 has Gym
( AccommodationID 5, AmenityID 1), -- Accommodation 5 has WiFi
( AccommodationID 5, AmenityID 10), -- Accommodation 5 has Pool
( AccommodationID 6, AmenityID 11), -- Accommodation 6 has Hot Tub
( AccommodationID 7, AmenityID 1), -- Accommodation 7 has WiFi
( AccommodationID 7, AmenityID 13), -- Accommodation 7 allows Pets
( AccommodationID 8, AmenityID 14), -- Accommodation 8 allows Smoking
( AccommodationID 9, AmenityID 1), -- Accommodation 9 has WiFi
( AccommodationID 9, AmenityID 12), -- Accommodation 9 includes Breakfast
( AccommodationID 10, AmenityID 1); -- Accommodation 10 has WiFi
```

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



This table stores administrative roles and related information for users with admin privileges.

```
CREATE TABLE IF NOT EXISTS `airbnb_db`.`admin` (
    `AdminID` INT NOT NULL,                                -- Unique identifier for each admin
    `UserID` INT NOT NULL,                                 -- Foreign key linking the admin to the User table
    `RoleDescription` VARCHAR(255) NULL DEFAULT NULL,      -- Description of the admin's role
    PRIMARY KEY (`AdminID`),                               -- Sets AdminID as the primary key
    UNIQUE INDEX `UserID` (`UserID` ASC) VISIBLE,          -- Ensures each UserID is unique in Admin
    CONSTRAINT `admin_ibfk_1`                            -- Defines the foreign key constraint for UserID
        FOREIGN KEY (`UserID`)
            REFERENCES `airbnb_db`.`user` (`UserID`)
            ON DELETE CASCADE
            ON UPDATE CASCADE
) ENGINE = InnoDB
```

AdminID	UserID	RoleDescription
1	1	System Administrator
2	2	Moderator
3	3	Content Manager
4	4	User Support Specialist
5	5	Security Administrator
6	6	Data Analyst
7	7	Operations Manager
8	8	Quality Assurance
9	9	Marketing Specialist
10	10	Financial Controller
11	11	Product Manager
12	12	Compliance Officer
13	13	Account Manager
14	14	Community Manager
15	15	Technical Support Lead
16	16	Customer Success Manager
17	17	Database Administrator
18	18	Training Coordinator
19	19	User Experience Designer
20	20	Research and Development

```
-- Sample Data for Admin Table
INSERT INTO `airbnb_db`.`admin` (AdminID, UserID, RoleDescription) VALUES
( AdminID 1, UserID 1, RoleDescription 'System Administrator'),
( AdminID 2, UserID 2, RoleDescription 'Moderator'),
( AdminID 3, UserID 3, RoleDescription 'Content Manager'),
( AdminID 4, UserID 4, RoleDescription 'User Support Specialist'),
( AdminID 5, UserID 5, RoleDescription 'Security Administrator'),
( AdminID 6, UserID 6, RoleDescription 'Data Analyst'),
( AdminID 7, UserID 7, RoleDescription 'Operations Manager'),
( AdminID 8, UserID 8, RoleDescription 'Quality Assurance'),
( AdminID 9, UserID 9, RoleDescription 'Marketing Specialist'),
( AdminID 10, UserID 10, RoleDescription 'Financial Controller'),
( AdminID 11, UserID 11, RoleDescription 'Product Manager'),
( AdminID 12, UserID 12, RoleDescription 'Compliance Officer'),
( AdminID 13, UserID 13, RoleDescription 'Account Manager'),
( AdminID 14, UserID 14, RoleDescription 'Community Manager'),
( AdminID 15, UserID 15, RoleDescription 'Technical Support Lead'),
( AdminID 16, UserID 16, RoleDescription 'Customer Success Manager'),
( AdminID 17, UserID 17, RoleDescription 'Database Administrator'),
( AdminID 18, UserID 18, RoleDescription 'Training Coordinator'),
( AdminID 19, UserID 19, RoleDescription 'User Experience Designer'),
( AdminID 20, UserID 20, RoleDescription 'Research and Development');
```



# Availability

This table tracks availability dates for each accommodation.

```
CREATE TABLE IF NOT EXISTS `airbnb_db`.`availability` (
    `AvailabilityID` INT NOT NULL AUTO_INCREMENT,          -- Unique identifier for each availability record, auto-incremented
    `AccommodationID` INT NOT NULL,                         -- Foreign key linking to the Accommodation table
    `AvailableDate` DATE NOT NULL,                          -- Date on which the accommodation is available
    `IsAvailable` TINYINT(1) NOT NULL,                      -- Availability status (1 for available, 0 for not available)
    PRIMARY KEY (`AvailabilityID`),                        -- Sets AvailabilityID as the primary key
    INDEX `availability_ibfk_1` (`AccommodationID` ASC) VISIBLE, -- Index for AccommodationID foreign key
    CONSTRAINT `availability_ibfk_1`                       -- Defines the foreign key constraint for AccommodationID
        FOREIGN KEY (`AccommodationID`)
            REFERENCES `airbnb_db`.`accommodation` (`AccommodationID`)
            ON DELETE CASCADE,                                -- Deletes availability records if the related accommodation is deleted
            ON UPDATE CASCADE                                -- Updates availability records if the related accommodation ID changes
) ENGINE = InnoDB
```

```
-- Sample Data for Availability Table
INSERT INTO `airbnb_db`.`availability` (AccommodationID, AvailableDate, IsAvailable) VALUES
( AccommodationID 1, AvailableDate '2024-12-01', IsAvailable 1),
( AccommodationID 1, AvailableDate '2024-12-02', IsAvailable 0),
( AccommodationID 2, AvailableDate '2024-12-01', IsAvailable 1),
( AccommodationID 2, AvailableDate '2024-12-03', IsAvailable 1),
( AccommodationID 3, AvailableDate '2024-12-01', IsAvailable 1),
( AccommodationID 3, AvailableDate '2024-12-04', IsAvailable 0),
( AccommodationID 4, AvailableDate '2024-12-01', IsAvailable 1),
( AccommodationID 4, AvailableDate '2024-12-05', IsAvailable 1),
( AccommodationID 5, AvailableDate '2024-12-01', IsAvailable 1),
( AccommodationID 5, AvailableDate '2024-12-06', IsAvailable 0),
( AccommodationID 6, AvailableDate '2024-12-01', IsAvailable 1),
( AccommodationID 6, AvailableDate '2024-12-07', IsAvailable 1),
( AccommodationID 7, AvailableDate '2024-12-01', IsAvailable 1),
( AccommodationID 7, AvailableDate '2024-12-08', IsAvailable 1),
( AccommodationID 8, AvailableDate '2024-12-01', IsAvailable 0),
( AccommodationID 8, AvailableDate '2024-12-09', IsAvailable 1),
( AccommodationID 9, AvailableDate '2024-12-01', IsAvailable 1),
( AccommodationID 9, AvailableDate '2024-12-10', IsAvailable 0),
( AccommodationID 10, AvailableDate '2024-12-01', IsAvailable 1),
( AccommodationID 10, AvailableDate '2024-12-11', IsAvailable 1),
( AccommodationID 11, AvailableDate '2024-12-01', IsAvailable 1),
( AccommodationID 11, AvailableDate '2024-12-12', IsAvailable 1),
( AccommodationID 12, AvailableDate '2024-12-01', IsAvailable 0),
( AccommodationID 12, AvailableDate '2024-12-13', IsAvailable 1);
```

AvailabilityID	AccommodationID	AvailableDate	IsAvailable
1	1	1 2024-12-01	1
2	2	1 2024-12-02	0
3	3	2 2024-12-01	1
4	4	2 2024-12-03	1
5	5	3 2024-12-01	1
6	6	3 2024-12-04	0
7	7	4 2024-12-01	1
8	8	4 2024-12-05	1
9	9	5 2024-12-01	1
10	10	5 2024-12-06	0
11	11	6 2024-12-01	1
12	12	6 2024-12-07	1
13	13	7 2024-12-01	1
14	14	7 2024-12-08	1
15	15	8 2024-12-01	0
16	16	8 2024-12-09	1
17	17	9 2024-12-01	1
18	18	9 2024-12-10	0
19	19	10 2024-12-01	1
20	20	10 2024-12-11	1
21	21	11 2024-12-01	1
22	22	11 2024-12-12	1
23	23	12 2024-12-01	0
24	24	12 2024-12-13	1



```
CREATE TABLE IF NOT EXISTS `airbnb_db`.`guestprofile` (
    `GuestID` INT NOT NULL, -- Unique identifier for each guest profile
    `UserID` INT NOT NULL, -- Foreign key linking the guest profile to the User table
    `Preferences` TEXT NULL DEFAULT NULL, -- Stores guest preferences (e.g., room type, amenities)
    PRIMARY KEY (`GuestID`), -- Sets GuestID as the primary key
    UNIQUE INDEX `UserID` (`UserID` ASC) VISIBLE, -- Ensures each UserID is unique in GuestProfile
    CONSTRAINT `guestprofile_ibfk_1` -- Defines the foreign key constraint for UserID
        FOREIGN KEY (`UserID`)
        REFERENCES `airbnb_db`.`user` (`UserID`)
        ON DELETE CASCADE -- If user is deleted, delete associated guest profile
        ON UPDATE CASCADE -- If user ID is updated, update the reference in guest profile
) ENGINE = InnoDB
```

```
-- Sample Data for GuestProfile Table
INSERT INTO `airbnb_db`.`guestprofile` (GuestID, UserID, Preferences) VALUES
( GuestID 1, UserID 21, Preferences 'Prefers non-smoking accommodations'),
( GuestID 2, UserID 22, Preferences 'Pet-friendly places only'),
( GuestID 3, UserID 23, Preferences 'Requires kitchen access'),
( GuestID 4, UserID 24, Preferences 'Near public transportation'),
( GuestID 5, UserID 25, Preferences 'Private bathroom required'),
( GuestID 6, UserID 26, Preferences 'High-speed internet essential'),
( GuestID 7, UserID 27, Preferences 'Close to city center'),
( GuestID 8, UserID 28, Preferences 'Near beach'),
( GuestID 9, UserID 29, Preferences 'No stairs'),
( GuestID 10, UserID 30, Preferences 'Wants parking available'),
( GuestID 11, UserID 31, Preferences 'Looking for quiet area'),
( GuestID 12, UserID 32, Preferences 'Nearby hiking trails preferred'),
( GuestID 13, UserID 33, Preferences 'Long-term stay friendly'),
( GuestID 14, UserID 34, Preferences 'Single bed'),
( GuestID 15, UserID 35, Preferences 'Requires workspace'),
( GuestID 16, UserID 36, Preferences 'Pet-free accommodations'),
( GuestID 17, UserID 37, Preferences 'Accessible accommodations only'),
( GuestID 18, UserID 38, Preferences 'Looking for a family-friendly space'),
( GuestID 19, UserID 39, Preferences 'Walking distance to restaurants'),
( GuestID 20, UserID 40, Preferences 'Prefers a balcony or terrace');
```

	GuestID	UserID	Preferences
1		1	21 Prefers non-smoking accommodations
2		2	22 Pet-friendly places only
3		3	23 Requires kitchen access
4		4	24 Near public transportation
5		5	25 Private bathroom required
6		6	26 High-speed internet essential
7		7	27 Close to city center
8		8	28 Near beach
9		9	29 No stairs
10		10	30 Wants parking available
11		11	31 Looking for quiet area
12		12	32 Nearby hiking trails preferred
13		13	33 Long-term stay friendly
14		14	34 Single bed
15		15	35 Requires workspace
16		16	36 Pet-free accommodations
17		17	37 Accessible accommodations only
18		18	38 Looking for a family-friendly space
19		19	39 Walking distance to restaurants
20		20	40 Prefers a balcony or terrace



```

CREATE TABLE IF NOT EXISTS `airbnb_db`.`booking` (
  `BookingID` INT NOT NULL AUTO_INCREMENT,
  `GuestID` INT NULL,
  `AccommodationID` INT NULL,
  `BookedByUserID` INT NULL,
  `BookingDate` DATE NOT NULL,
  `CheckInDate` DATE NOT NULL,
  `CheckOutDate` DATE NOT NULL,
  `TotalCost` DECIMAL(10,2) NOT NULL,
  PRIMARY KEY (`BookingID`),
  INDEX `AccommodationID` (`AccommodationID` ASC) VISIBLE,
  INDEX `GuestID` (`GuestID` ASC) VISIBLE,
  INDEX `BookedByUserID` (`BookedByUserID` ASC) VISIBLE,
  CONSTRAINT `booking_ibfk_1` FOREIGN KEY (`GuestID`)
    REFERENCES `airbnb_db`.`guestprofile`(`GuestID`)
    ON DELETE SET NULL
    ON UPDATE CASCADE,
  CONSTRAINT `booking_ibfk_2` FOREIGN KEY (`AccommodationID`)
    REFERENCES `airbnb_db`.`accommodation`(`AccommodationID`)
    ON DELETE SET NULL
    ON UPDATE CASCADE,
  CONSTRAINT `booking_ibfk_3` FOREIGN KEY (`BookedByUserID`)
    REFERENCES `airbnb_db`.`user`(`UserID`)
    ON DELETE SET NULL
    ON UPDATE CASCADE
) ENGINE = InnoDB
  
```

```

-- Sample Data for Booking Table
INSERT INTO `airbnb_db`.`booking` (BookingID, GuestID, AccommodationID, BookedByUserID, BookingDate, CheckInDate, CheckOutDate, TotalCost) VALUES
(BookingID 1, GuestID 1, AccommodationID 1, BookedByUserID 1, BookingDate '2024-11-01', CheckInDate '2024-12-01', CheckOutDate '2024-12-05', TotalCost 600.00),
(BookingID 2, GuestID 2, AccommodationID 2, BookedByUserID 2, BookingDate '2024-11-02', CheckInDate '2024-12-03', CheckOutDate '2024-12-07', TotalCost 750.00),
(BookingID 3, GuestID 3, AccommodationID 3, BookedByUserID 3, BookingDate '2024-11-03', CheckInDate '2024-12-08', CheckOutDate '2024-12-12', TotalCost 520.00),
(BookingID 4, GuestID 4, AccommodationID 4, BookedByUserID 4, BookingDate '2024-11-04', CheckInDate '2024-12-10', CheckOutDate '2024-12-15', TotalCost 1000.00),
(BookingID 5, GuestID 5, AccommodationID 5, BookedByUserID 5, BookingDate '2024-11-05', CheckInDate '2024-12-12', CheckOutDate '2024-12-14', TotalCost 280.00),
(BookingID 6, GuestID 6, AccommodationID 6, BookedByUserID 6, BookingDate '2024-11-06', CheckInDate '2024-12-14', CheckOutDate '2024-12-18', TotalCost 460.00),
(BookingID 7, GuestID 7, AccommodationID 7, BookedByUserID 7, BookingDate '2024-11-07', CheckInDate '2024-12-20', CheckOutDate '2024-12-23', TotalCost 345.00),
(BookingID 8, GuestID 8, AccommodationID 8, BookedByUserID 8, BookingDate '2024-11-08', CheckInDate '2024-12-25', CheckOutDate '2024-12-28', TotalCost 525.00),
(BookingID 9, GuestID 9, AccommodationID 9, BookedByUserID 9, BookingDate '2024-11-09', CheckInDate '2024-12-28', CheckOutDate '2024-12-30', TotalCost 320.00),
(BookingID 10, GuestID 10, AccommodationID 10, BookedByUserID 10, BookingDate '2024-11-10', CheckInDate '2024-12-01', CheckOutDate '2024-12-03', TotalCost 270.00),
(BookingID 11, GuestID 11, AccommodationID 11, BookedByUserID 11, BookingDate '2024-11-11', CheckInDate '2024-12-05', CheckOutDate '2024-12-07', TotalCost 250.00),
(BookingID 12, GuestID 12, AccommodationID 12, BookedByUserID 12, BookingDate '2024-11-12', CheckInDate '2024-12-08', CheckOutDate '2024-12-11', TotalCost 315.00),
(BookingID 13, GuestID 13, AccommodationID 13, BookedByUserID 13, BookingDate '2024-11-13', CheckInDate '2024-12-13', CheckOutDate '2024-12-15', TotalCost 200.00),
(BookingID 14, GuestID 14, AccommodationID 14, BookedByUserID 14, BookingDate '2024-11-14', CheckInDate '2024-12-18', CheckOutDate '2024-12-22', TotalCost 500.00),
(BookingID 15, GuestID 15, AccommodationID 15, BookedByUserID 15, BookingDate '2024-11-15', CheckInDate '2024-12-23', CheckOutDate '2024-12-26', TotalCost 330.00),
(BookingID 16, GuestID 16, AccommodationID 16, BookedByUserID 16, BookingDate '2024-11-16', CheckInDate '2024-12-27', CheckOutDate '2024-12-30', TotalCost 225.00),
(BookingID 17, GuestID 17, AccommodationID 17, BookedByUserID 17, BookingDate '2024-11-17', CheckInDate '2024-12-29', CheckOutDate '2024-12-31', TotalCost 390.00),
(BookingID 18, GuestID 18, AccommodationID 18, BookedByUserID 18, BookingDate '2024-11-18', CheckInDate '2025-01-02', CheckOutDate '2025-01-06', TotalCost 520.00),
(BookingID 19, GuestID 19, AccommodationID 19, BookedByUserID 19, BookingDate '2024-11-19', CheckInDate '2025-01-07', CheckOutDate '2025-01-10', TotalCost 460.00),
(BookingID 20, GuestID 20, AccommodationID 20, BookedByUserID 20, BookingDate '2024-11-20', CheckInDate '2025-01-12', CheckOutDate '2025-01-15', TotalCost 465.00);
  
```

BookingID	GuestID	AccommodationID	BookedByUserID	BookingDate	CheckInDate	CheckOutDate	TotalCost
1	1	1	1	2024-11-01	2024-12-01	2024-12-05	600.00
2	2	2	2	2024-11-02	2024-12-03	2024-12-07	750.00
3	3	3	3	2024-11-03	2024-12-08	2024-12-12	520.00
4	4	4	4	2024-11-04	2024-12-10	2024-12-15	1000.00
5	5	5	5	2024-11-05	2024-12-12	2024-12-14	280.00
6	6	6	6	2024-11-06	2024-12-14	2024-12-18	460.00
7	7	7	7	2024-11-07	2024-12-20	2024-12-23	345.00
8	8	8	8	2024-11-08	2024-12-25	2024-12-28	525.00
9	9	9	9	2024-11-09	2024-12-28	2024-12-30	320.00
10	10	10	10	2024-11-10	2024-12-01	2024-12-03	270.00
11	11	11	11	2024-11-11	2024-12-05	2024-12-07	250.00
12	12	12	12	2024-11-12	2024-12-08	2024-12-11	315.00
13	13	13	13	2024-11-13	2024-12-13	2024-12-15	200.00
14	14	14	14	2024-11-14	2024-12-18	2024-12-22	500.00
15	15	15	15	2024-11-15	2024-12-23	2024-12-26	330.00
16	16	16	16	2024-11-16	2024-12-27	2024-12-30	225.00
17	17	17	17	2024-11-17	2024-12-29	2024-12-31	390.00
18	18	18	18	2024-11-18	2025-01-02	2025-01-06	520.00
19	19	19	19	2024-11-19	2025-01-07	2025-01-10	460.00
20	20	20	20	2024-11-20	2025-01-12	2025-01-15	465.00



```
CREATE TABLE IF NOT EXISTS `airbnb_db`.`calendar` (
    `CalendarID` INT NOT NULL AUTO_INCREMENT, -- Unique identifier for each calendar entry, auto-incremented
    `AccommodationID` INT NOT NULL, -- Foreign key linking to the Accommodation table
    `Date` DATE NOT NULL, -- Date for the calendar entry
    `BookingStatus` ENUM('Available', 'Booked', 'Unavailable') NOT NULL, -- Status of the accommodation on the specified date
    PRIMARY KEY (`CalendarID`), -- Sets CalendarID as the primary key
    INDEX `calendar_ibfk_1` (`AccommodationID` ASC) VISIBLE, -- Index for AccommodationID foreign key
    CONSTRAINT `calendar_ibfk_1` -- Defines the foreign key constraint for AccommodationID
        FOREIGN KEY (`AccommodationID`)
        REFERENCES `airbnb_db`.`accommodation` (`AccommodationID`)
        ON DELETE CASCADE -- Deletes calendar entry if the related accommodation is deleted
        ON UPDATE CASCADE -- Updates calendar entry if the related accommodation ID changes
) ENGINE = InnoDB
```

CalendarID	AccommodationID	Date	BookingStatus
1	1	2024-12-01	Available
2	2	2024-12-02	Booked
3	3	2024-12-01	Available
4	4	2024-12-03	Available
5	5	2024-12-01	Available
6	6	2024-12-04	Unavailable
7	7	2024-12-01	Available
8	8	2024-12-05	Booked
9	9	2024-12-01	Available
10	10	2024-12-06	Unavailable
11	11	2024-12-01	Available
12	12	2024-12-07	Booked
13	13	2024-12-01	Available
14	14	2024-12-08	Available
15	15	2024-12-01	Unavailable
16	16	2024-12-09	Booked
17	17	2024-12-01	Available
18	18	2024-12-10	Unavailable
19	19	2024-12-01	Available
20	20	2024-12-11	Booked

```
-- Sample Data for Calendar Table
INSERT INTO `airbnb_db`.`calendar` (CalendarID, AccommodationID, Date, BookingStatus) VALUES
( CalendarID 1, AccommodationID 1, Date '2024-12-01', BookingStatus 'Available'),
( CalendarID 2, AccommodationID 1, Date '2024-12-02', BookingStatus 'Booked'),
( CalendarID 3, AccommodationID 2, Date '2024-12-01', BookingStatus 'Available'),
( CalendarID 4, AccommodationID 2, Date '2024-12-03', BookingStatus 'Available'),
( CalendarID 5, AccommodationID 3, Date '2024-12-01', BookingStatus 'Available'),
( CalendarID 6, AccommodationID 3, Date '2024-12-04', BookingStatus 'Unavailable'),
( CalendarID 7, AccommodationID 4, Date '2024-12-01', BookingStatus 'Available'),
( CalendarID 8, AccommodationID 4, Date '2024-12-05', BookingStatus 'Booked'),
( CalendarID 9, AccommodationID 5, Date '2024-12-01', BookingStatus 'Available'),
( CalendarID 10, AccommodationID 5, Date '2024-12-06', BookingStatus 'Unavailable'),
( CalendarID 11, AccommodationID 6, Date '2024-12-01', BookingStatus 'Available'),
( CalendarID 12, AccommodationID 6, Date '2024-12-07', BookingStatus 'Booked'),
( CalendarID 13, AccommodationID 7, Date '2024-12-01', BookingStatus 'Available'),
( CalendarID 14, AccommodationID 7, Date '2024-12-08', BookingStatus 'Available'),
( CalendarID 15, AccommodationID 8, Date '2024-12-01', BookingStatus 'Unavailable'),
( CalendarID 16, AccommodationID 8, Date '2024-12-09', BookingStatus 'Booked'),
( CalendarID 17, AccommodationID 9, Date '2024-12-01', BookingStatus 'Available'),
( CalendarID 18, AccommodationID 9, Date '2024-12-10', BookingStatus 'Unavailable'),
( CalendarID 19, AccommodationID 10, Date '2024-12-01', BookingStatus 'Available'),
( CalendarID 20, AccommodationID 10, Date '2024-12-11', BookingStatus 'Booked');
```



# Commission

This table tracks commission fees associated with bookings.

```
CREATE TABLE IF NOT EXISTS `airbnb_db`.`commission` (
    `CommissionID` INT NOT NULL AUTO_INCREMENT,          -- Unique identifier for each commission record, auto-incremented
    `BookingID` INT NOT NULL,                            -- Foreign key linking to the Booking table
    `PlatformFee` DECIMAL(10,2) NOT NULL,                -- Platform fee charged for the booking
    `HostFee` DECIMAL(10,2) NOT NULL,                    -- Fee paid to the host
    PRIMARY KEY (`CommissionID`),                        -- Sets CommissionID as the primary key
    INDEX `BookingID` (`BookingID` ASC) VISIBLE,        -- Index for BookingID foreign key
    CONSTRAINT `commission_ibfk_1`                      -- Defines the foreign key constraint for BookingID
        FOREIGN KEY (`BookingID`)
        REFERENCES `airbnb_db`.`booking` (`BookingID`)
        ON DELETE CASCADE                                -- Deletes commission record if the related booking is deleted
        ON UPDATE CASCADE                                -- Updates commission record if the related booking ID changes
) ENGINE = InnoDB
```

```
-- Sample Data for Commission Table
INSERT INTO `airbnb_db`.`commission` (CommissionID, BookingID, PlatformFee, HostFee) VALUES
(CommissionID 1, BookingID 1, PlatformFee 60.00, HostFee 540.00),
(CommissionID 2, BookingID 2, PlatformFee 75.00, HostFee 675.00),
(CommissionID 3, BookingID 3, PlatformFee 52.00, HostFee 468.00),
(CommissionID 4, BookingID 4, PlatformFee 100.00, HostFee 900.00),
(CommissionID 5, BookingID 5, PlatformFee 28.00, HostFee 252.00),
(CommissionID 6, BookingID 6, PlatformFee 46.00, HostFee 414.00),
(CommissionID 7, BookingID 7, PlatformFee 34.50, HostFee 310.50),
(CommissionID 8, BookingID 8, PlatformFee 52.50, HostFee 472.50),
(CommissionID 9, BookingID 9, PlatformFee 32.00, HostFee 288.00),
(CommissionID 10, BookingID 10, PlatformFee 27.00, HostFee 243.00),
(CommissionID 11, BookingID 11, PlatformFee 25.00, HostFee 225.00),
(CommissionID 12, BookingID 12, PlatformFee 31.50, HostFee 283.50),
(CommissionID 13, BookingID 13, PlatformFee 20.00, HostFee 180.00),
(CommissionID 14, BookingID 14, PlatformFee 50.00, HostFee 450.00),
(CommissionID 15, BookingID 15, PlatformFee 33.00, HostFee 297.00),
(CommissionID 16, BookingID 16, PlatformFee 22.50, HostFee 202.50),
(CommissionID 17, BookingID 17, PlatformFee 39.00, HostFee 351.00),
(CommissionID 18, BookingID 18, PlatformFee 52.00, HostFee 468.00),
(CommissionID 19, BookingID 19, PlatformFee 46.00, HostFee 414.00),
(CommissionID 20, BookingID 20, PlatformFee 46.50, HostFee 418.50);
```

CommissionID	BookingID	PlatformFee	HostFee
1	1	60.00	540.00
2	2	75.00	675.00
3	3	52.00	468.00
4	4	100.00	900.00
5	5	28.00	252.00
6	6	46.00	414.00
7	7	34.50	310.50
8	8	52.50	472.50
9	9	32.00	288.00
10	10	27.00	243.00
11	11	25.00	225.00
12	12	31.50	283.50
13	13	20.00	180.00
14	14	50.00	450.00
15	15	33.00	297.00
16	16	22.50	202.50
17	17	39.00	351.00
18	18	52.00	468.00
19	19	46.00	414.00
20	20	46.50	418.50



```
CREATE TABLE IF NOT EXISTS `airbnb_db`.`discount` (
    `DiscountID` INT NOT NULL AUTO_INCREMENT, -- Unique identifier for each discount record, auto-incremented
    `AccommodationID` INT NULL DEFAULT NULL, -- Foreign key linking to the Accommodation table
    `DiscountPercentage` DECIMAL(5,2) NOT NULL, -- Percentage discount offered on the accommodation
    `StartDate` DATE NOT NULL, -- Start date of the discount period
    `EndDate` DATE NOT NULL, -- End date of the discount period
    PRIMARY KEY (`DiscountID`), -- Sets DiscountID as the primary key
    INDEX `AccommodationID` (`AccommodationID` ASC) VISIBLE, -- Index for AccommodationID foreign key
    CONSTRAINT `discount_ibfk_1` -- Defines the foreign key constraint for AccommodationID
        FOREIGN KEY (`AccommodationID`)
        REFERENCES `airbnb_db`.`accommodation` (`AccommodationID`)
        ON DELETE SET NULL -- Sets AccommodationID to NULL if the related accommodation is deleted
        ON UPDATE CASCADE -- Updates AccommodationID if the related accommodation ID changes
) ENGINE = InnoDB
```

```
-- Sample Data for Discount Table
INSERT INTO `airbnb_db`.`discount` (DiscountID, AccommodationID, DiscountPercentage, StartDate, EndDate) VALUES
(DiscountID 1, AccommodationID 1, DiscountPercentage 10.00, StartDate '2024-12-01', EndDate '2024-12-10'),
(DiscountID 2, AccommodationID 2, DiscountPercentage 15.00, StartDate '2024-12-05', EndDate '2024-12-15'),
(DiscountID 3, AccommodationID 3, DiscountPercentage 5.00, StartDate '2024-12-08', EndDate '2024-12-12'),
(DiscountID 4, AccommodationID 4, DiscountPercentage 20.00, StartDate '2024-12-10', EndDate '2024-12-20'),
(DiscountID 5, AccommodationID 5, DiscountPercentage 10.00, StartDate '2024-12-12', EndDate '2024-12-18'),
(DiscountID 6, AccommodationID 6, DiscountPercentage 25.00, StartDate '2024-12-14', EndDate '2024-12-22'),
(DiscountID 7, AccommodationID 7, DiscountPercentage 15.00, StartDate '2024-12-16', EndDate '2024-12-25'),
(DiscountID 8, AccommodationID 8, DiscountPercentage 5.00, StartDate '2024-12-18', EndDate '2024-12-23'),
(DiscountID 9, AccommodationID 9, DiscountPercentage 10.00, StartDate '2024-12-20', EndDate '2024-12-28'),
(DiscountID 10, AccommodationID 10, DiscountPercentage 12.00, StartDate '2024-12-22', EndDate '2024-12-31'),
(DiscountID 11, AccommodationID 11, DiscountPercentage 7.00, StartDate '2024-12-24', EndDate '2025-01-05'),
(DiscountID 12, AccommodationID 12, DiscountPercentage 15.00, StartDate '2024-12-26', EndDate '2025-01-10'),
(DiscountID 13, AccommodationID 13, DiscountPercentage 10.00, StartDate '2024-12-28', EndDate '2025-01-12'),
(DiscountID 14, AccommodationID 14, DiscountPercentage 20.00, StartDate '2024-12-30', EndDate '2025-01-15'),
(DiscountID 15, AccommodationID 15, DiscountPercentage 8.00, StartDate '2025-01-01', EndDate '2025-01-10'),
(DiscountID 16, AccommodationID 16, DiscountPercentage 18.00, StartDate '2025-01-05', EndDate '2025-01-15'),
(DiscountID 17, AccommodationID 17, DiscountPercentage 10.00, StartDate '2025-01-07', EndDate '2025-01-20'),
(DiscountID 18, AccommodationID 18, DiscountPercentage 12.00, StartDate '2025-01-10', EndDate '2025-01-25'),
(DiscountID 19, AccommodationID 19, DiscountPercentage 5.00, StartDate '2025-01-12', EndDate '2025-01-22'),
(DiscountID 20, AccommodationID 20, DiscountPercentage 15.00, StartDate '2025-01-15', EndDate '2025-01-30');
```

DiscountID	AccommodationID	DiscountPercentage	StartDate	EndDate
1	1	10.00	2024-12-01	2024-12-10
2	2	15.00	2024-12-05	2024-12-15
3	3	5.00	2024-12-08	2024-12-12
4	4	20.00	2024-12-10	2024-12-20
5	5	10.00	2024-12-12	2024-12-18
6	6	25.00	2024-12-14	2024-12-22
7	7	15.00	2024-12-16	2024-12-25
8	8	5.00	2024-12-18	2024-12-23
9	9	10.00	2024-12-20	2024-12-28
10	10	12.00	2024-12-22	2024-12-31
11	11	7.00	2024-12-24	2025-01-05
12	12	15.00	2024-12-26	2025-01-10
13	13	10.00	2024-12-28	2025-01-12
14	14	20.00	2024-12-30	2025-01-15
15	15	8.00	2025-01-01	2025-01-10
16	16	18.00	2025-01-05	2025-01-15
17	17	10.00	2025-01-07	2025-01-20
18	18	12.00	2025-01-10	2025-01-25
19	19	5.00	2025-01-12	2025-01-22
20	20	15.00	2025-01-15	2025-01-30



```
CREATE TABLE IF NOT EXISTS `airbnb_db`.`incomecalculator` (
    `CalculatorID` INT NOT NULL AUTO_INCREMENT,          -- Unique identifier for each income calculator record, auto-incremented
    `HostID` INT NULL,                                    -- Foreign key linking to the HostProfile table
    `EstimatedIncome` DECIMAL(10,2) NOT NULL,            -- Estimated income for the host
    PRIMARY KEY (`CalculatorID`),                         -- Sets CalculatorID as the primary key
    INDEX `incomecalculator_ibfk_1` (`HostID` ASC) VISIBLE, -- Index for HostID foreign key
    CONSTRAINT `incomecalculator_ibfk_1`                  -- Defines the foreign key constraint for HostID
        FOREIGN KEY (`HostID`)
        REFERENCES `airbnb_db`.`hostprofile` (`HostID`)
        ON DELETE CASCADE,                                -- Deletes income calculator record if the related host profile is deleted
        ON UPDATE CASCADE                                -- Updates income calculator record if the related host ID changes
) ENGINE = InnoDB
```

```
-- Sample Data for IncomeCalculator Table
INSERT INTO `airbnb_db`.`incomecalculator` (CalculatorID, HostID, EstimatedIncome) VALUES
( CalculatorID 1, HostID 1, EstimatedIncome 2000.00),
( CalculatorID 2, HostID 2, EstimatedIncome 2500.00),
( CalculatorID 3, HostID 3, EstimatedIncome 1500.00),
( CalculatorID 4, HostID 4, EstimatedIncome 3000.00),
( CalculatorID 5, HostID 5, EstimatedIncome 1800.00),
( CalculatorID 6, HostID 6, EstimatedIncome 2200.00),
( CalculatorID 7, HostID 7, EstimatedIncome 2700.00),
( CalculatorID 8, HostID 8, EstimatedIncome 2400.00),
( CalculatorID 9, HostID 9, EstimatedIncome 1600.00),
( CalculatorID 10, HostID 10, EstimatedIncome 2900.00),
( CalculatorID 11, HostID 11, EstimatedIncome 3100.00),
( CalculatorID 12, HostID 12, EstimatedIncome 2800.00),
( CalculatorID 13, HostID 13, EstimatedIncome 2300.00),
( CalculatorID 14, HostID 14, EstimatedIncome 2600.00),
( CalculatorID 15, HostID 15, EstimatedIncome 1900.00),
( CalculatorID 16, HostID 16, EstimatedIncome 3200.00),
( CalculatorID 17, HostID 17, EstimatedIncome 2100.00),
( CalculatorID 18, HostID 18, EstimatedIncome 2500.00),
( CalculatorID 19, HostID 19, EstimatedIncome 1700.00),
( CalculatorID 20, HostID 20, EstimatedIncome 3000.00);
```

CalculatorID	HostID	EstimatedIncome
1	1	2000.00
2	2	2500.00
3	3	1500.00
4	4	3000.00
5	5	1800.00
6	6	2200.00
7	7	2700.00
8	8	2400.00
9	9	1600.00
10	10	2900.00
11	11	3100.00
12	12	2800.00
13	13	2300.00
14	14	2600.00
15	15	1900.00
16	16	3200.00
17	17	2100.00
18	18	2500.00
19	19	1700.00
20	20	3000.00



```
CREATE TABLE IF NOT EXISTS `airbnb_db`.`insurancepolicy` (
    `PolicyID` INT NOT NULL AUTO_INCREMENT,          -- Unique identifier for each insurance policy, auto-incremented
    `AccommodationID` INT NULL DEFAULT NULL,        -- Foreign key linking to the Accommodation table
    `CoverageAmount` DECIMAL(10,2) NOT NULL,          -- Amount covered by the insurance policy
    `Premium` DECIMAL(10,2) NOT NULL,                -- Premium amount for the insurance policy
    `ProviderName` VARCHAR(100) NOT NULL,            -- Name of the insurance provider
    PRIMARY KEY (`PolicyID`),                         -- Sets PolicyID as the primary key
    INDEX `AccommodationID` (`AccommodationID` ASC) VISIBLE, -- Index for AccommodationID foreign key
    CONSTRAINT `insurancepolicy_ibfk_1`              -- Defines the foreign key constraint for AccommodationID
        FOREIGN KEY (`AccommodationID`)
        REFERENCES `airbnb_db`.`accommodation` (`AccommodationID`)
        ON DELETE SET NULL                           -- Sets AccommodationID to NULL if the related accommodation is deleted
        ON UPDATE CASCADE                          -- Updates AccommodationID if the related accommodation ID changes
) ENGINE = InnoDB
```

```
-- Sample Data for InsurancePolicy Table
INSERT INTO `airbnb_db`.`insurancepolicy` (PolicyID, AccommodationID, CoverageAmount, Premium, ProviderName) VALUES
( PolicyID 1, AccommodationID 1, CoverageAmount 50000.00, Premium 200.00, ProviderName 'InsureCo'),
( PolicyID 2, AccommodationID 2, CoverageAmount 75000.00, Premium 250.00, ProviderName 'SafeGuard'),
( PolicyID 3, AccommodationID 3, CoverageAmount 60000.00, Premium 180.00, ProviderName 'ProtectPlus'),
( PolicyID 4, AccommodationID 4, CoverageAmount 100000.00, Premium 300.00, ProviderName 'ShieldSure'),
( PolicyID 5, AccommodationID 5, CoverageAmount 85000.00, Premium 220.00, ProviderName 'InsureCo'),
( PolicyID 6, AccommodationID 6, CoverageAmount 70000.00, Premium 210.00, ProviderName 'SafeGuard'),
( PolicyID 7, AccommodationID 7, CoverageAmount 90000.00, Premium 275.00, ProviderName 'ProtectPlus'),
( PolicyID 8, AccommodationID 8, CoverageAmount 65000.00, Premium 190.00, ProviderName 'ShieldSure'),
( PolicyID 9, AccommodationID 9, CoverageAmount 80000.00, Premium 240.00, ProviderName 'InsureCo'),
( PolicyID 10, AccommodationID 10, CoverageAmount 55000.00, Premium 200.00, ProviderName 'SafeGuard'),
( PolicyID 11, AccommodationID 11, CoverageAmount 70000.00, Premium 210.00, ProviderName 'ProtectPlus'),
( PolicyID 12, AccommodationID 12, CoverageAmount 75000.00, Premium 250.00, ProviderName 'ShieldSure'),
( PolicyID 13, AccommodationID 13, CoverageAmount 50000.00, Premium 200.00, ProviderName 'InsureCo'),
( PolicyID 14, AccommodationID 14, CoverageAmount 85000.00, Premium 230.00, ProviderName 'SafeGuard'),
( PolicyID 15, AccommodationID 15, CoverageAmount 60000.00, Premium 180.00, ProviderName 'ProtectPlus'),
( PolicyID 16, AccommodationID 16, CoverageAmount 95000.00, Premium 280.00, ProviderName 'ShieldSure'),
( PolicyID 17, AccommodationID 17, CoverageAmount 72000.00, Premium 215.00, ProviderName 'InsureCo'),
( PolicyID 18, AccommodationID 18, CoverageAmount 77000.00, Premium 260.00, ProviderName 'SafeGuard'),
( PolicyID 19, AccommodationID 19, CoverageAmount 58000.00, Premium 190.00, ProviderName 'ProtectPlus'),
( PolicyID 20, AccommodationID 20, CoverageAmount 85000.00, Premium 230.00, ProviderName 'ShieldSure');
```

PolicyID	AccommodationID	CoverageAmount	Premium	ProviderName
1	1	50000.00	200.00	InsureCo
2	2	75000.00	250.00	SafeGuard
3	3	60000.00	180.00	ProtectPlus
4	4	100000.00	300.00	ShieldSure
5	5	85000.00	220.00	InsureCo
6	6	70000.00	210.00	SafeGuard
7	7	90000.00	275.00	ProtectPlus
8	8	65000.00	190.00	ShieldSure
9	9	80000.00	240.00	InsureCo
10	10	55000.00	200.00	SafeGuard
11	11	70000.00	210.00	ProtectPlus
12	12	75000.00	250.00	ShieldSure
13	13	50000.00	200.00	InsureCo
14	14	85000.00	230.00	SafeGuard
15	15	60000.00	180.00	ProtectPlus
16	16	95000.00	280.00	ShieldSure
17	17	72000.00	215.00	InsureCo
18	18	77000.00	260.00	SafeGuard
19	19	58000.00	190.00	ProtectPlus
20	20	85000.00	230.00	ShieldSure



```
CREATE TABLE IF NOT EXISTS `airbnb_db`.`listingimage` (
    `ImageID` INT NOT NULL AUTO_INCREMENT, -- Unique identifier for each listing image, auto-incremented
    `AccommodationID` INT NOT NULL, -- Foreign key linking to the Accommodation table
    `ImagePath` VARCHAR(255) NOT NULL, -- File path or URL to the image
    PRIMARY KEY (`ImageID`), -- Sets ImageID as the primary key
    INDEX `listingimage_ibfk_1` (`AccommodationID` ASC) VISIBLE, -- Index for AccommodationID foreign key
    CONSTRAINT `listingimage_ibfk_1` -- Defines the foreign key constraint for AccommodationID
        FOREIGN KEY (`AccommodationID`)
        REFERENCES `airbnb_db`.`accommodation` (`AccommodationID`)
        ON DELETE CASCADE -- Deletes image record if the related accommodation is deleted
        ON UPDATE CASCADE -- Updates image record if the related accommodation ID changes
) ENGINE = InnoDB
```

```
-- Sample Data for ListingImage Table
INSERT INTO `airbnb_db`.`listingimage` (ImageID, AccommodationID, ImagePath) VALUES
( ImageID 1, AccommodationID 1, ImagePath '/images/accommodation1_img1.jpg'),
( ImageID 2, AccommodationID 2, ImagePath '/images/accommodation2_img1.jpg'),
( ImageID 3, AccommodationID 3, ImagePath '/images/accommodation3_img1.jpg'),
( ImageID 4, AccommodationID 4, ImagePath '/images/accommodation4_img1.jpg'),
( ImageID 5, AccommodationID 5, ImagePath '/images/accommodation5_img1.jpg'),
( ImageID 6, AccommodationID 6, ImagePath '/images/accommodation6_img1.jpg'),
( ImageID 7, AccommodationID 7, ImagePath '/images/accommodation7_img1.jpg'),
( ImageID 8, AccommodationID 8, ImagePath '/images/accommodation8_img1.jpg'),
( ImageID 9, AccommodationID 9, ImagePath '/images/accommodation9_img1.jpg'),
( ImageID 10, AccommodationID 10, ImagePath '/images/accommodation10_img1.jpg'),
( ImageID 11, AccommodationID 11, ImagePath '/images/accommodation11_img1.jpg'),
( ImageID 12, AccommodationID 12, ImagePath '/images/accommodation12_img1.jpg'),
( ImageID 13, AccommodationID 13, ImagePath '/images/accommodation13_img1.jpg'),
( ImageID 14, AccommodationID 14, ImagePath '/images/accommodation14_img1.jpg'),
( ImageID 15, AccommodationID 15, ImagePath '/images/accommodation15_img1.jpg'),
( ImageID 16, AccommodationID 16, ImagePath '/images/accommodation16_img1.jpg'),
( ImageID 17, AccommodationID 17, ImagePath '/images/accommodation17_img1.jpg'),
( ImageID 18, AccommodationID 18, ImagePath '/images/accommodation18_img1.jpg'),
( ImageID 19, AccommodationID 19, ImagePath '/images/accommodation19_img1.jpg'),
( ImageID 20, AccommodationID 20, ImagePath '/images/accommodation20_img1.jpg');
```

ImageID	AccommodationID	ImagePath
1	1	1 /images/accommodation1_img1.jpg
2	2	2 /images/accommodation2_img1.jpg
3	3	3 /images/accommodation3_img1.jpg
4	4	4 /images/accommodation4_img1.jpg
5	5	5 /images/accommodation5_img1.jpg
6	6	6 /images/accommodation6_img1.jpg
7	7	7 /images/accommodation7_img1.jpg
8	8	8 /images/accommodation8_img1.jpg
9	9	9 /images/accommodation9_img1.jpg
10	10	10 /images/accommodation10_img1.jpg
11	11	11 /images/accommodation11_img1.jpg
12	12	12 /images/accommodation12_img1.jpg
13	13	13 /images/accommodation13_img1.jpg
14	14	14 /images/accommodation14_img1.jpg
15	15	15 /images/accommodation15_img1.jpg
16	16	16 /images/accommodation16_img1.jpg
17	17	17 /images/accommodation17_img1.jpg
18	18	18 /images/accommodation18_img1.jpg
19	19	19 /images/accommodation19_img1.jpg
20	20	20 /images/accommodation20_img1.jpg



# Message

This table contains messages exchanged between users.

```
CREATE TABLE IF NOT EXISTS `airbnb_db`.`message` (
  `MessageID` INT NOT NULL AUTO_INCREMENT,
  `SenderUserID` INT NOT NULL,
  `ReceiverUserID` INT NOT NULL,
  `MessageContent` TEXT NOT NULL,
  `Timestamp` DATETIME NOT NULL,
  PRIMARY KEY (`MessageID`),
  INDEX `message_ibfk_1` (`SenderUserID` ASC) VISIBLE, -- Index for SenderUserID foreign key
  INDEX `message_ibfk_2` (`ReceiverUserID` ASC) VISIBLE, -- Index for ReceiverUserID foreign key
  CONSTRAINT `message_ibfk_1` -- Defines the foreign key constraint for SenderUserID
    FOREIGN KEY (`SenderUserID`)
      REFERENCES `airbnb_db`.`user` (`UserID`)
      ON DELETE CASCADE
      ON UPDATE CASCADE,
  CONSTRAINT `message_ibfk_2` -- Defines the foreign key constraint for ReceiverUserID
    FOREIGN KEY (`ReceiverUserID`)
      REFERENCES `airbnb_db`.`user` (`UserID`)
      ON DELETE CASCADE
      ON UPDATE CASCADE
) ENGINE = InnoDB

-- Sample Data for Message Table
INSERT INTO `airbnb_db`.`message` (MessageID, SenderUserID, ReceiverUserID, MessageContent, Timestamp) VALUES
(1, 1, 2, 'Hello, I am interested in your listing.', '2024-11-01 10:15:00'),
(2, 2, 1, 'Thank you for your interest! Let me know if you have any questions.', '2024-11-01 10:30:00'),
(3, 3, 4, 'Is the property available on the selected dates?', '2024-11-02 09:00:00'),
(4, 4, 3, 'Yes, it is available.', '2024-11-02 09:15:00'),
(5, 5, 6, 'Can I bring my pet?', '2024-11-03 14:45:00'),
(6, 6, 5, 'Yes, pets are allowed in the property.', '2024-11-03 15:00:00'),
(7, 7, 8, 'Do you have a WiFi connection?', '2024-11-04 16:20:00'),
(8, 8, 7, 'Yes, we offer high-speed WiFi.', '2024-11-04 16:30:00'),
(9, 9, 10, 'Is there free parking on-site?', '2024-11-05 11:10:00'),
(10, 10, 9, 'Yes, free parking is available.', '2024-11-05 11:25:00'),
(11, 11, 12, 'Can I check in early?', '2024-11-06 13:50:00'),
(12, 12, 11, 'Early check-in is possible upon request.', '2024-11-06 14:00:00'),
(13, 13, 14, 'Is the neighborhood safe?', '2024-11-07 18:30:00'),
(14, 14, 13, 'Yes, it is a very safe neighborhood.', '2024-11-07 18:45:00'),
(15, 15, 16, 'Are there restaurants nearby?', '2024-11-08 12:20:00'),
(16, 16, 15, 'Yes, there are several within walking distance.', '2024-11-08 12:35:00'),
(17, 17, 18, 'Can I extend my stay?', '2024-11-09 19:10:00'),
(18, 18, 17, 'Certainly, I will update the booking.', '2024-11-09 19:20:00'),
(19, 19, 20, 'Is there a pool on the property?', '2024-11-10 08:40:00'),
(20, 20, 19, 'Yes, the pool is available for guests.', '2024-11-10 08:55:00');
```

MessageID	SenderUserID	ReceiverUserID	MessageContent	Timestamp
1	1	2	Hello, I am interested in your listing.	2024-11-01 10:15:00
2	2	1	Thank you for your interest! Let me know if you have..	2024-11-01 10:30:00
3	3	4	Is the property available on the selected dates?	2024-11-02 09:00:00
4	4	3	Yes, it is available.	2024-11-02 09:15:00
5	5	6	Can I bring my pet?	2024-11-03 14:45:00
6	6	5	Yes, pets are allowed in the property.	2024-11-03 15:00:00
7	7	8	Do you have a WiFi connection?	2024-11-04 16:20:00
8	8	7	Yes, we offer high-speed WiFi.	2024-11-04 16:30:00
9	9	10	Is there free parking on-site?	2024-11-05 11:10:00
10	10	9	Yes, free parking is available.	2024-11-05 11:25:00
11	11	11	Can I check in early?	2024-11-06 13:50:00
12	12	12	Early check-in is possible upon request.	2024-11-06 14:00:00
13	13	13	Is the neighborhood safe?	2024-11-07 18:30:00
14	14	14	Yes, it is a very safe neighborhood.	2024-11-07 18:45:00
15	15	15	Are there restaurants nearby?	2024-11-08 12:20:00
16	16	16	Yes, there are several within walking distance.	2024-11-08 12:35:00
17	17	17	Can I extend my stay?	2024-11-09 19:10:00
18	18	18	Certainly, I will update the booking.	2024-11-09 19:20:00
19	19	19	Is there a pool on the property?	2024-11-10 08:40:00
20	20	20	Yes, the pool is available for guests.	2024-11-10 08:55:00



# Payment

This table stores payment details related to bookings, including amount, status, payment date and payer.

```
CREATE TABLE IF NOT EXISTS `airbnb_db`.`payment` (
  `PaymentID` INT NOT NULL AUTO_INCREMENT,
  `BookingID` INT NOT NULL,
  `PaidByUserID` INT NOT NULL,
  `Amount` DECIMAL(10,2) NOT NULL,
  `Status` ENUM('Pending', 'Completed', 'Failed') NOT NULL,
  `PaymentDate` DATE NOT NULL,
  PRIMARY KEY (`PaymentID`),
  INDEX `BookingID` (`BookingID` ASC) VISIBLE,
  INDEX `PaidByUserID` (`PaidByUserID` ASC) VISIBLE,
  CONSTRAINT `payment_ibfk_1` FOREIGN KEY (`BookingID`)
    REFERENCES `airbnb_db`.`booking` (`BookingID`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `payment_ibfk_2` FOREIGN KEY (`PaidByUserID`)
    REFERENCES `airbnb_db`.`user` (`UserID`)
    ON DELETE CASCADE
    ON UPDATE CASCADE
) ENGINE = InnoDB
```

```
-- Sample Data for Payment Table
INSERT INTO `airbnb_db`.`payment` (BookingID, PaidByUserID, Amount, Status, PaymentDate) VALUES
(1, 1, 600.00, 'Completed', '2024-11-01'),
(2, 2, 750.00, 'Completed', '2024-11-02'),
(3, 3, 520.00, 'Pending', '2024-11-03'),
(4, 4, 1000.00, 'Completed', '2024-11-04'),
(5, 5, 280.00, 'Completed', '2024-11-05'),
(6, 6, 460.00, 'Completed', '2024-11-06'),
(7, 7, 345.00, 'Failed', '2024-11-07'),
(8, 8, 525.00, 'Completed', '2024-11-08'),
(9, 9, 320.00, 'Pending', '2024-11-09'),
(10, 10, 270.00, 'Completed', '2024-11-10'),
(11, 11, 250.00, 'Completed', '2024-11-11'),
(12, 12, 315.00, 'Completed', '2024-11-12'),
(13, 13, 200.00, 'Pending', '2024-11-13'),
(14, 14, 500.00, 'Completed', '2024-11-14'),
(15, 15, 330.00, 'Completed', '2024-11-15'),
(16, 16, 225.00, 'Completed', '2024-11-16'),
(17, 17, 390.00, 'Pending', '2024-11-17'),
(18, 18, 520.00, 'Completed', '2024-11-18'),
(19, 19, 460.00, 'Failed', '2024-11-19'),
(20, 20, 465.00, 'Completed', '2024-11-20');
```

PaymentID	BookingID	PaidByUserID	Amount	Status	PaymentDate
1	1	1	600.00	Completed	2024-11-01
2	2	2	750.00	Completed	2024-11-02
3	3	3	520.00	Pending	2024-11-03
4	4	4	1000.00	Completed	2024-11-04
5	5	5	280.00	Completed	2024-11-05
6	6	6	460.00	Completed	2024-11-06
7	7	7	345.00	Failed	2024-11-07
8	8	8	525.00	Completed	2024-11-08
9	9	9	320.00	Pending	2024-11-09
10	10	10	270.00	Completed	2024-11-10
11	11	11	250.00	Completed	2024-11-11
12	12	12	315.00	Completed	2024-11-12
13	13	13	200.00	Pending	2024-11-13
14	14	14	500.00	Completed	2024-11-14
15	15	15	330.00	Completed	2024-11-15
16	16	16	225.00	Completed	2024-11-16
17	17	17	390.00	Pending	2024-11-17
18	18	18	520.00	Completed	2024-11-18
19	19	19	460.00	Failed	2024-11-19
20	20	20	465.00	Completed	2024-11-20



# PropertyPreference

This table contains user-specific preferences for property features.

```
CREATE TABLE IF NOT EXISTS `airbnb_db`.`propertypreference` (
    `PreferenceID` INT NOT NULL AUTO_INCREMENT,          -- Unique identifier for each property preference, auto-incremented
    `UserID` INT NOT NULL,                               -- Foreign key linking to the User table
    `PreferenceType` VARCHAR(50) NOT NULL,                -- Type of preference (e.g., room type, amenities)
    `PreferenceValue` VARCHAR(50) NOT NULL,              -- Value for the specified preference type
    PRIMARY KEY (`PreferenceID`),                         -- Sets PreferenceID as the primary key
    INDEX `propertypreference_ibfk_1` (`UserID` ASC) VISIBLE, -- Index for UserID foreign key
    CONSTRAINT `propertypreference_ibfk_1`                -- Defines the foreign key constraint for UserID
        FOREIGN KEY (`UserID`)
        REFERENCES `airbnb_db`.`user` (`UserID`)
        ON DELETE CASCADE                                -- Deletes property preference if the related user is deleted
        ON UPDATE CASCADE                                -- Updates property preference if the related user ID changes
) ENGINE = InnoDB
```

```
-- Sample Data for PropertyPreference Table
INSERT INTO `airbnb_db`.`propertypreference` (PreferenceID, UserID, PreferenceType, PreferenceValue) VALUES
(PreferenceID 1, UserID 1, PreferenceType 'Smoking', PreferenceValue 'Not Allowed'),
(PreferenceID 2, UserID 2, PreferenceType 'Pet-Friendly', PreferenceValue 'Yes'),
(PreferenceID 3, UserID 3, PreferenceType 'WiFi', PreferenceValue 'Required'),
(PreferenceID 4, UserID 4, PreferenceType 'Parking', PreferenceValue 'Free'),
(PreferenceID 5, UserID 5, PreferenceType 'Breakfast', PreferenceValue 'Included'),
(PreferenceID 6, UserID 6, PreferenceType 'Kitchen', PreferenceValue 'Shared'),
(PreferenceID 7, UserID 7, PreferenceType 'Laundry', PreferenceValue 'Available'),
(PreferenceID 8, UserID 8, PreferenceType 'Heating', PreferenceValue 'Required'),
(PreferenceID 9, UserID 9, PreferenceType 'Air Conditioning', PreferenceValue 'Preferred'),
(PreferenceID 10, UserID 10, PreferenceType 'Wheelchair Accessible', PreferenceValue 'Yes'),
(PreferenceID 11, UserID 11, PreferenceType 'Pool', PreferenceValue 'Not Required'),
(PreferenceID 12, UserID 12, PreferenceType 'Fireplace', PreferenceValue 'Preferred'),
(PreferenceID 13, UserID 13, PreferenceType 'Workspace', PreferenceValue 'Necessary'),
(PreferenceID 14, UserID 14, PreferenceType 'Gym', PreferenceValue 'Not Important'),
(PreferenceID 15, UserID 15, PreferenceType 'Elevator', PreferenceValue 'Preferred'),
(PreferenceID 16, UserID 16, PreferenceType 'Outdoor Space', PreferenceValue 'Required'),
(PreferenceID 17, UserID 17, PreferenceType 'Long-term Stays', PreferenceValue 'Yes'),
(PreferenceID 18, UserID 18, PreferenceType 'Events Allowed', PreferenceValue 'No'),
(PreferenceID 19, UserID 19, PreferenceType 'Hot Tub', PreferenceValue 'Optional'),
(PreferenceID 20, UserID 20, PreferenceType 'Child-Friendly', PreferenceValue 'Yes');
```

PreferenceID	UserID	PreferenceType	PreferenceValue
1	1	1 Smoking	Not Allowed
2	2	2 Pet-Friendly	Yes
3	3	3 WiFi	Required
4	4	4 Parking	Free
5	5	5 Breakfast	Included
6	6	6 Kitchen	Shared
7	7	7 Laundry	Available
8	8	8 Heating	Required
9	9	9 Air Conditioning	Preferred
10	10	10 Wheelchair Accessible	Yes
11	11	11 Pool	Not Required
12	12	12 Fireplace	Preferred
13	13	13 Workspace	Necessary
14	14	14 Gym	Not Important
15	15	15 Elevator	Preferred
16	16	16 Outdoor Space	Required
17	17	17 Long-term Stays	Yes
18	18	18 Events Allowed	No
19	19	19 Hot Tub	Optional
20	20	20 Child-Friendly	Yes



```
CREATE TABLE IF NOT EXISTS `airbnb_db`.`rating` (
    `RatingID` INT NOT NULL AUTO_INCREMENT,
    `AccommodationID` INT NOT NULL,
    `Score` INT NOT NULL,
    PRIMARY KEY (`RatingID`),
    INDEX `AccommodationID` (`AccommodationID` ASC) VISIBLE,
    CONSTRAINT `rating_ibfk_1`
        FOREIGN KEY (`AccommodationID`)
        REFERENCES `airbnb_db`.`accommodation` (`AccommodationID`)
        ON DELETE CASCADE
        ON UPDATE CASCADE
) ENGINE = InnoDB
```

```
-- Sample Data for Rating Table
INSERT INTO `airbnb_db`.`rating` (RatingID, AccommodationID, Score) VALUES
( RatingID 1, AccommodationID 1, Score 4),
( RatingID 2, AccommodationID 2, Score 5),
( RatingID 3, AccommodationID 3, Score 3),
( RatingID 4, AccommodationID 4, Score 5),
( RatingID 5, AccommodationID 5, Score 4),
( RatingID 6, AccommodationID 6, Score 4),
( RatingID 7, AccommodationID 7, Score 3),
( RatingID 8, AccommodationID 8, Score 5),
( RatingID 9, AccommodationID 9, Score 2),
( RatingID 10, AccommodationID 10, Score 4),
( RatingID 11, AccommodationID 11, Score 5),
( RatingID 12, AccommodationID 12, Score 3),
( RatingID 13, AccommodationID 13, Score 4),
( RatingID 14, AccommodationID 14, Score 5),
( RatingID 15, AccommodationID 15, Score 3),
( RatingID 16, AccommodationID 16, Score 4),
( RatingID 17, AccommodationID 17, Score 5),
( RatingID 18, AccommodationID 18, Score 2),
( RatingID 19, AccommodationID 19, Score 4),
( RatingID 20, AccommodationID 20, Score 5);
```

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



```
CREATE TABLE IF NOT EXISTS `airbnb_db`.`review` (
    `ReviewID` INT NOT NULL AUTO_INCREMENT, -- Unique identifier for each review, auto-incremented
    `AccommodationID` INT NULL DEFAULT NULL, -- Foreign key linking to the Accommodation table
    `UserID` INT NOT NULL, -- Foreign key linking to the User table
    `Rating` INT NOT NULL, -- Rating score given in the review
    `Comment` TEXT NULL DEFAULT NULL, -- Text comment provided in the review
    `Date` DATE NOT NULL, -- Date the review was written
    PRIMARY KEY (`ReviewID`), -- Sets ReviewID as the primary key
    INDEX `AccommodationID` (`AccommodationID` ASC) VISIBLE, -- Index for AccommodationID foreign key
    INDEX `review_ibfk_2` (`UserID` ASC) VISIBLE, -- Index for UserID foreign key
    CONSTRAINT `review_ibfk_1` -- Defines the foreign key constraint for AccommodationID
        FOREIGN KEY (`AccommodationID`)
        REFERENCES `airbnb_db`.`accommodation` (`AccommodationID`)
        ON DELETE SET NULL -- Sets AccommodationID to NULL if the related accommodation is deleted
        ON UPDATE CASCADE, -- Updates AccommodationID in the review if it changes in the accommodation table
    CONSTRAINT `review_ibfk_2` -- Defines the foreign key constraint for UserID
        FOREIGN KEY (`UserID`)
        REFERENCES `airbnb_db`.`user` (`UserID`)
        ON DELETE CASCADE -- Deletes review if the related user is deleted
        ON UPDATE CASCADE -- Updates UserID in the review if it changes in the user table
) ENGINE = InnoDB
```

```
-- Sample Data for Review Table
INSERT INTO `airbnb_db`.`review` (ReviewID, AccommodationID, UserID, Rating, Comment, Date) VALUES
(ReviewID 1, AccommodationID 1, UserID 1, Rating 4, Comment 'Great place, very cozy!', Date '2024-11-01'),
(ReviewID 2, AccommodationID 2, UserID 2, Rating 5, Comment 'Amazing view and clean rooms.', Date '2024-11-02'),
(ReviewID 3, AccommodationID 3, UserID 3, Rating 3, Comment 'Good location but noisy area.', Date '2024-11-03'),
(ReviewID 4, AccommodationID 4, UserID 4, Rating 5, Comment 'Perfect stay, would recommend!', Date '2024-11-04'),
(ReviewID 5, AccommodationID 5, UserID 5, Rating 4, Comment 'Nice amenities, comfortable bed.', Date '2024-11-05'),
(ReviewID 6, AccommodationID 6, UserID 6, Rating 4, Comment 'Good value for money.', Date '2024-11-06'),
(ReviewID 7, AccommodationID 7, UserID 7, Rating 3, Comment 'Average stay, a bit outdated.', Date '2024-11-07'),
(ReviewID 8, AccommodationID 8, UserID 8, Rating 5, Comment 'Beautiful property by the beach!', Date '2024-11-08'),
(ReviewID 9, AccommodationID 9, UserID 9, Rating 2, Comment 'Rooms were small and cramped.', Date '2024-11-09'),
(ReviewID 10, AccommodationID 10, UserID 10, Rating 4, Comment 'Convenient location, friendly host.', Date '2024-11-10'),
(ReviewID 11, AccommodationID 11, UserID 11, Rating 5, Comment 'Exceeded expectations, very clean.', Date '2024-11-11'),
(ReviewID 12, AccommodationID 12, UserID 12, Rating 3, Comment 'Nice area, but a bit pricey.', Date '2024-11-12'),
(ReviewID 13, AccommodationID 13, UserID 13, Rating 4, Comment 'Comfortable, felt like home.', Date '2024-11-13'),
(ReviewID 14, AccommodationID 14, UserID 14, Rating 5, Comment 'Would stay again, amazing host!', Date '2024-11-14'),
(ReviewID 15, AccommodationID 15, UserID 15, Rating 3, Comment 'Basic amenities, nothing special.', Date '2024-11-15'),
(ReviewID 16, AccommodationID 16, UserID 16, Rating 4, Comment 'Spacious and well-kept.', Date '2024-11-16'),
(ReviewID 17, AccommodationID 17, UserID 17, Rating 5, Comment 'Fantastic experience overall!', Date '2024-11-17'),
(ReviewID 18, AccommodationID 18, UserID 18, Rating 2, Comment 'Disappointing, needs improvements.', Date '2024-11-18'),
(ReviewID 19, AccommodationID 19, UserID 19, Rating 4, Comment 'Lovely decor, enjoyed my stay.', Date '2024-11-19'),
(ReviewID 20, AccommodationID 20, UserID 20, Rating 5, Comment 'Perfect getaway spot!', Date '2024-11-20');
```

ReviewID	AccommodationID	UserID	Rating	Comment	Date
1	1	1	1	4 Great place, very cozy!	2024-11-01
2	2	2	2	5 Amazing view and clean rooms.	2024-11-02
3	3	3	3	3 Good location but noisy area.	2024-11-03
4	4	4	4	5 Perfect stay, would recommend!	2024-11-04
5	5	5	5	4 Nice amenities, comfortable bed.	2024-11-05
6	6	6	6	4 Good value for money.	2024-11-06
7	7	7	7	3 Average stay, a bit outdated.	2024-11-07
8	8	8	8	5 Beautiful property by the beach!	2024-11-08
9	9	9	9	2 Rooms were small and cramped.	2024-11-09
10	10	10	10	4 Convenient location, friendly host.	2024-11-10
11	11	11	11	5 Exceeded expectations, very clean.	2024-11-11
12	12	12	12	3 Nice area, but a bit pricey.	2024-11-12
13	13	13	13	4 Comfortable, felt like home.	2024-11-13
14	14	14	14	5 Would stay again, amazing host!	2024-11-14
15	15	15	15	3 Basic amenities, nothing special.	2024-11-15
16	16	16	16	4 Spacious and well-kept.	2024-11-16
17	17	17	17	5 Fantastic experience overall.	2024-11-17
18	18	18	18	2 Disappointing, needs improvements.	2024-11-18
19	19	19	19	4 Lovely decor, enjoyed my stay.	2024-11-19
20	20	20	20	5 Perfect getaway spot!	2024-11-20



```
CREATE TABLE IF NOT EXISTS `airbnb_db`.`socialmedialink` (
    `LinkID` INT NOT NULL AUTO_INCREMENT, -- Unique identifier for each social media link, auto-incremented
    `UserID` INT NOT NULL, -- Foreign key linking to the User table
    `PlatformName` VARCHAR(50) NOT NULL, -- Name of the social media platform (e.g., Facebook, Twitter)
    `ProfileLink` VARCHAR(255) NOT NULL, -- URL to the user's profile on the social media platform
    PRIMARY KEY (`LinkID`), -- Sets LinkID as the primary key
    INDEX `socialmedialink_ibfk_1` (`UserID` ASC) VISIBLE, -- Index for UserID foreign key
    CONSTRAINT `socialmedialink_ibfk_1` -- Defines the foreign key constraint for UserID
        FOREIGN KEY (`UserID`)
        REFERENCES `airbnb_db`.`user` (`UserID`)
        ON DELETE CASCADE -- Deletes social media link if the related user is deleted
        ON UPDATE CASCADE -- Updates social media link if the related user ID changes
) ENGINE = InnoDB
```

```
-- Sample Data for SocialMediaLink Table
INSERT INTO `airbnb_db`.`socialmedialink` (LinkID, UserID, PlatformName, ProfileLink) VALUES
( LinkID 1, UserID 1, PlatformName 'Facebook', ProfileLink 'https://facebook.com/johndoe'),
( LinkID 2, UserID 2, PlatformName 'Instagram', ProfileLink 'https://instagram.com/janesmith'),
( LinkID 3, UserID 3, PlatformName 'Twitter', ProfileLink 'https://twitter.com/alicejohnson'),
( LinkID 4, UserID 4, PlatformName 'LinkedIn', ProfileLink 'https://linkedin.com/in/bobbrown'),
( LinkID 5, UserID 5, PlatformName 'Facebook', ProfileLink 'https://facebook.com/chrisgreen'),
( LinkID 6, UserID 6, PlatformName 'Instagram', ProfileLink 'https://instagram.com/dianawhite'),
( LinkID 7, UserID 7, PlatformName 'Twitter', ProfileLink 'https://twitter.com/evanblack'),
( LinkID 8, UserID 8, PlatformName 'LinkedIn', ProfileLink 'https://linkedin.com/in/fionagray'),
( LinkID 9, UserID 9, PlatformName 'Facebook', ProfileLink 'https://facebook.com/georgeblue'),
( LinkID 10, UserID 10, PlatformName 'Instagram', ProfileLink 'https://instagram.com/hollyyellow'),
( LinkID 11, UserID 11, PlatformName 'Twitter', ProfileLink 'https://twitter.com/ivyred'),
( LinkID 12, UserID 12, PlatformName 'LinkedIn', ProfileLink 'https://linkedin.com/in/jakeviolet'),
( LinkID 13, UserID 13, PlatformName 'Facebook', ProfileLink 'https://facebook.com/kellyorange'),
( LinkID 14, UserID 14, PlatformName 'Instagram', ProfileLink 'https://instagram.com/liampink'),
( LinkID 15, UserID 15, PlatformName 'Twitter', ProfileLink 'https://twitter.com/monapurple'),
( LinkID 16, UserID 16, PlatformName 'LinkedIn', ProfileLink 'https://linkedin.com/in/ninabrown'),
( LinkID 17, UserID 17, PlatformName 'Facebook', ProfileLink 'https://facebook.com/oscargray'),
( LinkID 18, UserID 18, PlatformName 'Instagram', ProfileLink 'https://instagram.com/paulsilver'),
( LinkID 19, UserID 19, PlatformName 'Twitter', ProfileLink 'https://twitter.com/quinngold'),
( LinkID 20, UserID 20, PlatformName 'LinkedIn', ProfileLink 'https://linkedin.com/in/rachelbronze');
```

LinkID	UserID	PlatformName	ProfileLink
1	1	Facebook	https://facebook.com/johndoe
2	2	Instagram	https://instagram.com/janesmith
3	3	Twitter	https://twitter.com/alicejohnson
4	4	LinkedIn	https://linkedin.com/in/bobbrown
5	5	Facebook	https://facebook.com/chrisgreen
6	6	Instagram	https://instagram.com/dianawhite
7	7	Twitter	https://twitter.com/evanblack
8	8	LinkedIn	https://linkedin.com/in/fionagray
9	9	Facebook	https://facebook.com/georgeblue
10	10	Instagram	https://instagram.com/hollyyellow
11	11	Twitter	https://twitter.com/ivyred
12	12	LinkedIn	https://linkedin.com/in/jakeviolet
13	13	Facebook	https://facebook.com/kellyorange
14	14	Instagram	https://instagram.com/liampink
15	15	Twitter	https://twitter.com/monapurple
16	16	LinkedIn	https://linkedin.com/in/ninabrown
17	17	Facebook	https://facebook.com/oscargray
18	18	Instagram	https://instagram.com/paulsilver
19	19	Twitter	https://twitter.com/quinngold
20	20	LinkedIn	https://linkedin.com/in/rachelbronze



# Transaction

This table tracks financial transactions associated with payments.

```
CREATE TABLE IF NOT EXISTS `airbnb_db`.`transaction` (
    `TransactionID` INT NOT NULL AUTO_INCREMENT,
    `PaymentID` INT NULL DEFAULT NULL,
    `TransactionDate` DATE NOT NULL,
    `TransactionAmount` DECIMAL(10,2) NOT NULL,
    PRIMARY KEY (`TransactionID`),
    INDEX `PaymentID` (`PaymentID` ASC) VISIBLE,
    CONSTRAINT `transaction_ibfk_1`
        FOREIGN KEY (`PaymentID`)
        REFERENCES `airbnb_db`.`payment` (`PaymentID`)
        ON DELETE SET NULL
        ON UPDATE CASCADE
) ENGINE = InnoDB
```

```
-- Sample Data for Transaction Table
INSERT INTO `airbnb_db`.`transaction` (TransactionID, PaymentID, TransactionDate, TransactionAmount) VALUES
(1, 1, '2024-11-01', 600.00),
(2, 2, '2024-11-02', 750.00),
(3, 3, '2024-11-03', 520.00),
(4, 4, '2024-11-04', 1000.00),
(5, 5, '2024-11-05', 280.00),
(6, 6, '2024-11-06', 460.00),
(7, 7, '2024-11-07', 345.00),
(8, 8, '2024-11-08', 525.00),
(9, 9, '2024-11-09', 320.00),
(10, 10, '2024-11-10', 270.00),
(11, 11, '2024-11-11', 250.00),
(12, 12, '2024-11-12', 315.00),
(13, 13, '2024-11-13', 200.00),
(14, 14, '2024-11-14', 500.00),
(15, 15, '2024-11-15', 330.00),
(16, 16, '2024-11-16', 225.00),
(17, 17, '2024-11-17', 390.00),
(18, 18, '2024-11-18', 520.00),
(19, 19, '2024-11-19', 460.00),
(20, 20, '2024-11-20', 465.00);
```

TransactionID	PaymentID	TransactionDate	TransactionAmount
1	1	2024-11-01	600.00
2	2	2024-11-02	750.00
3	3	2024-11-03	520.00
4	4	2024-11-04	1000.00
5	5	2024-11-05	280.00
6	6	2024-11-06	460.00
7	7	2024-11-07	345.00
8	8	2024-11-08	525.00
9	9	2024-11-09	320.00
10	10	2024-11-10	270.00
11	11	2024-11-11	250.00
12	12	2024-11-12	315.00
13	13	2024-11-13	200.00
14	14	2024-11-14	500.00
15	15	2024-11-15	330.00
16	16	2024-11-16	225.00
17	17	2024-11-17	390.00
18	18	2024-11-18	520.00
19	19	2024-11-19	460.00
20	20	2024-11-20	465.00



```
CREATE TABLE IF NOT EXISTS `airbnb_db`.`houserules` (
  `RuleID` INT NOT NULL AUTO_INCREMENT, -- Unique identifier for each rule
  `AccommodationID` INT NOT NULL, -- Foreign key linking to the Accommodation table
  `RuleDescription` TEXT NOT NULL, -- Description of the house rule
  PRIMARY KEY (`RuleID`), -- Sets RuleID as the primary key
  INDEX `AccommodationID` (`AccommodationID` ASC) VISIBLE, -- Index for foreign key
  CONSTRAINT `houserules_ibfk_1` -- Defines the foreign key constraint for AccommodationID
    FOREIGN KEY (`AccommodationID`)
    REFERENCES `airbnb_db`.`accommodation` (`AccommodationID`)
    ON DELETE CASCADE -- Deletes rules if the accommodation is deleted
    ON UPDATE CASCADE -- Updates rules if the accommodation ID changes
) ENGINE = InnoDB
```

```
-- Sample Data for HouseRules Table
INSERT INTO `airbnb_db`.`houserules` (RuleDescription) VALUES
('No smoking inside the house.'),
('No pets allowed.'),
('Quiet hours are from 10 PM to 8 AM.'),
('Guests must clean up after themselves.'),
('No parties or events allowed.'),
('Turn off lights and appliances when not in use.'),
('Do not disturb neighbors.'),
('Maximum occupancy is 4 people.'),
('No shoes inside the house.'),
('Pets are allowed with prior approval.'),
('Guests must take out the trash daily.'),
('No loud music after 9 PM.'),
('Do not use appliances for purposes other than intended.'),
('Guests must provide valid ID at check-in.'),
('Do not rearrange furniture.'),
('Swimming pool hours are from 9 AM to 7 PM.'),
('Guests must follow parking guidelines.'),
('Report any damages immediately.'),
('Do not leave windows open when leaving the house.'),
('Do not flush non-biodegradable items down the toilet.'));
```

RuleID	AccommodationID	RuleDescription
1	1	1 No smoking inside the house.
2	2	2 No pets allowed.
3	3	3 Quiet hours are from 10 PM to 8 AM.
4	4	4 Guests must clean up after themselves.
5	5	5 No parties or events allowed.
6	6	6 Turn off lights and appliances when not in use.
7	7	7 Do not disturb neighbors.
8	8	8 Maximum occupancy is 4 people.
9	9	9 No shoes inside the house.
10	10	10 Pets are allowed with prior approval.
11	11	11 Guests must take out the trash daily.
12	12	12 No loud music after 9 PM.
13	13	13 Do not use appliances for purposes other than intended.
14	14	14 Guests must provide valid ID at check-in.
15	15	15 Do not rearrange furniture.
16	16	16 Swimming pool hours are from 9 AM to 7 PM.
17	17	17 Guests must follow parking guidelines.
18	18	18 Report any damages immediately.
19	19	19 Do not leave windows open when leaving the house.
20	20	20 Do not flush non-biodegradable items down the toilet.



# Neighborhood

This table stores details about neighborhoods connected to locations.

```
CREATE TABLE IF NOT EXISTS `airbnb_db`.`neighborhood` (
  `NeighborhoodID` INT NOT NULL AUTO_INCREMENT, -- Unique identifier for the neighborhood
  `LocationID` INT NOT NULL, -- Foreign key linking to the Location table
  `Description` TEXT NOT NULL, -- Description of the neighborhood
  PRIMARY KEY (`NeighborhoodID`), -- Sets NeighborhoodID as the primary key
  INDEX `LocationID` ( `LocationID` ASC) VISIBLE, -- Index for foreign key
  CONSTRAINT `neighborhood_ibfk_1` -- Defines the foreign key constraint for LocationID
    FOREIGN KEY (`LocationID`)
    REFERENCES `airbnb_db`.`location` (`LocationID`)
    ON DELETE CASCADE -- Deletes neighborhood info if the location is deleted
    ON UPDATE CASCADE -- Updates neighborhood info if the location ID changes
) ENGINE = InnoDB
```

```
-- Sample Data for Neighborhood Table
INSERT INTO `airbnb_db`.`neighborhood` (LocationID, Description) VALUES
(1, 'Vibrant area with shopping centers and cafes nearby.'),
(2, 'Quiet suburban area with family-friendly parks.'),
(3, 'Bustling downtown area close to major attractions.'),
(4, 'Residential neighborhood with a mix of modern and historic homes.'),
(5, 'Eco-friendly community with bike paths and green spaces.'),
(6, 'Close to the beach with stunning sea views.'),
(7, 'Mountain-side neighborhood with hiking trails nearby.'),
(8, 'Artistic community with galleries and live music events.'),
(9, 'Industrial area transitioning to a trendy hotspot.'),
(10, 'Luxurious area with high-end shops and restaurants.'),
(11, 'Countryside setting with open fields and fresh air.'),
(12, 'Historic district with cobblestone streets and landmarks.'),
(13, 'Lively student area with pubs and affordable eateries.'),
(14, 'Tech-focused area with modern buildings and coworking spaces.'),
(15, 'Close to the airport with excellent transport connections.'),
(16, 'Peaceful retirement community with quiet streets.'),
(17, 'Close-knit neighborhood with frequent community events.'),
(18, 'Riverfront area with scenic walking paths.'),
(19, 'Tourist hub with easy access to museums and theaters.'),
(20, 'Lakeside neighborhood with water sports and fishing opportunities.');
```

NeighborhoodID	LocationID	Description
1	1	Vibrant area with shopping centers and cafes nearby.
2	2	Quiet suburban area with family-friendly parks.
3	3	Bustling downtown area close to major attractions.
4	4	Residential neighborhood with a mix of modern and historic homes.
5	5	Eco-friendly community with bike paths and green spaces.
6	6	Close to the beach with stunning sea views.
7	7	Mountain-side neighborhood with hiking trails nearby.
8	8	Artistic community with galleries and live music events.
9	9	Industrial area transitioning to a trendy hotspot.
10	10	Luxurious area with high-end shops and restaurants.
11	11	Countryside setting with open fields and fresh air.
12	12	Historic district with cobblestone streets and landmarks.
13	13	Lively student area with pubs and affordable eateries.
14	14	Tech-focused area with modern buildings and coworking spaces.
15	15	Close to the airport with excellent transport connections.
16	16	Peaceful retirement community with quiet streets.
17	17	Close-knit neighborhood with frequent community events.
18	18	Riverfront area with scenic walking paths.
19	19	Tourist hub with easy access to museums and theaters.
20	20	Lakeside neighborhood with water sports and fishing opportunities.



# AccommodationHouseRules

This is a junction table linking accommodations with their house rules

```
CREATE TABLE IF NOT EXISTS `airbnb_db`.`AccommodationHouseRules` (
  `AccommodationID` INT NOT NULL,          -- Foreign key linking to the Accommodation table
  `HouseRuleID` INT NOT NULL,              -- Foreign key linking to the HouseRules table
  PRIMARY KEY (`AccommodationID`, `HouseRuleID`), -- Composite primary key for unique pairs of AccommodationID and HouseRuleID
  CONSTRAINT `accommodation_houserules_ibfk_1`    -- Foreign key constraint for AccommodationID
    FOREIGN KEY (`AccommodationID`)
    REFERENCES `airbnb_db`.`accommodation` (`AccommodationID`)
    ON DELETE CASCADE                      -- Deletes entries if the corresponding accommodation is deleted
    ON UPDATE CASCADE,                     -- Updates AccommodationID if it changes in the Accommodation table
  CONSTRAINT `accommodation_houserules_ibfk_2`    -- Foreign key constraint for HouseRuleID
    FOREIGN KEY (`HouseRuleID`)
    REFERENCES `airbnb_db`.`houserules` (`RuleID`)
    ON DELETE CASCADE                      -- Deletes entries if the corresponding house rule is deleted
    ON UPDATE CASCADE                     -- Updates HouseRuleID if it changes in the HouseRules table
) ENGINE = InnoDB
```

```
-- Sample Data for AccommodationHouseRules Table
INSERT INTO `airbnb_db`.`accommodation_houserules` (AccommodationID, RuleID) VALUES
( AccommodationID 1, RuleID 1),    -- Cozy studio in New York: No smoking inside the house
( AccommodationID 2, RuleID 2),    -- Modern apartment in LA: No pets allowed
( AccommodationID 3, RuleID 3),    -- Spacious loft in Chicago: Quiet hours are from 10 PM to 8 AM
( AccommodationID 4, RuleID 4),    -- Luxury villa in Houston: Guests must clean up after themselves
( AccommodationID 5, RuleID 5),    -- Charming home in Phoenix: No parties or events allowed
( AccommodationID 6, RuleID 6),    -- Historic townhouse in Philadelphia: Turn off lights and appliances when not in use
( AccommodationID 7, RuleID 7),    -- Riverside apartment in San Antonio: Do not disturb neighbors
( AccommodationID 8, RuleID 8),    -- Beachfront condo in San Diego: Maximum occupancy is 4 people
( AccommodationID 9, RuleID 9),    -- High-rise apartment in Dallas: No shoes inside the house
( AccommodationID 10, RuleID 10),   -- Elegant guesthouse in San Jose: Pets are allowed with prior approval
( AccommodationID 11, RuleID 11),   -- Stylish flat in Austin: Guests must take out the trash daily
( AccommodationID 12, RuleID 12),   -- Quaint bungalow in Jacksonville: No loud music after 9 PM
( AccommodationID 13, RuleID 13),   -- Suburban home in Fort Worth: Do not use appliances for unintended purposes
( AccommodationID 14, RuleID 14),   -- Modern duplex in Columbus: Guests must provide valid ID at check-in
( AccommodationID 15, RuleID 15),   -- Family-friendly house in Charlotte: Do not rearrange furniture
( AccommodationID 16, RuleID 16),   -- Penthouse in San Francisco: Swimming pool hours are from 9 AM to 7 PM
( AccommodationID 17, RuleID 17),   -- Downtown apartment in Indianapolis: Guests must follow parking guidelines
( AccommodationID 18, RuleID 18),   -- Studio in Seattle: Report any damages immediately
( AccommodationID 19, RuleID 19),   -- Eco-friendly cabin in Denver: Do not leave windows open when leaving the house
( AccommodationID 20, RuleID 20);  -- Classic row house in Washington, D.C.: Do not flush non-biodegradable items down the toilet
```

AccommodationID	HouseRuleID
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



**Test Cases**



# Test Case 1 (User)

```
SELECT
    u.Name AS GuestName,
    a.Description AS Accommodation,
    b.TotalCost AS BookingCost
FROM
    user u
JOIN
    booking b ON u.UserID = b.BookedByUserID
JOIN
    accommodation a ON b.AccommodationID = a.AccommodationID
ORDER BY
    b.TotalCost DESC;
```

	GuestName	Accommodation	BookingCost
1	Bob Brown	Luxury villa in Houston	1000.00
2	Jane Smith	Modern apartment with a sea view in Los Angeles	750.00
3	John Doe	Cozy studio in the heart of New York	600.00
4	Fiona Gray	Beachfront condo in San Diego	525.00
5	Alice Johnson	Spacious loft in downtown Chicago	520.00
6	Paul Silver	Studio in Seattle with mountain views	520.00
7	Liam Pink	Modern duplex in Columbus	500.00
8	Rachel Bronze	Classic row house in Washington, D.C.	465.00
9	Diana White	Historic townhouse in Philadelphia	460.00
10	Quinn Gold	Eco-friendly cabin in Denver	460.00
11	Oscar Gray	Downtown apartment in Indianapolis	390.00
12	Evan Black	Riverside apartment in San Antonio	345.00
13	Mona Purple	Family-friendly house in Charlotte	330.00
14	George Blue	High-rise apartment in Dallas with city views	320.00
15	Jake Violet	Quaint bungalow in Jacksonville	315.00
16	Chris Green	Charming home in Phoenix with a private pool	280.00
17	Holly Yellow	Elegant guesthouse in San Jose	270.00
18	Ivy Red	Stylish flat in Austin	250.00
19	Nina Brown	Penthouse in San Francisco	225.00
20	Kelly Orange	Suburban home in Fort Worth	200.00

**Scenario:** Retrieve a list of users along with the accommodations they booked and the booking cost.



**Purpose:**

- Tests the relationship between user, booking, and accommodation.
- Validates foreign keys: BookedByUserID in booking and AccommodationID in accommodation.



## Test Case 2 (Accommodation)

```
SELECT
    a.Description AS Accommodation,
    h.HostID AS Host,
    COUNT(b.BookingID) AS TotalBookings
FROM
    accommodation a
LEFT JOIN
    hostprofile h ON a.HostID = h.HostID
LEFT JOIN
    booking b ON a.AccommodationID = b.AccommodationID
GROUP BY
    a.AccommodationID
ORDER BY
    TotalBookings DESC;
```

	Accommodation	Host	TotalBookings
1	Cozy studio in the heart of New York		1
2	Modern apartment with a sea view in Los Angeles		2
3	Spacious loft in downtown Chicago		1
4	Luxury villa in Houston		1
5	Charming home in Phoenix with a private pool		1
6	Historic townhouse in Philadelphia		1
7	Riverside apartment in San Antonio		1
8	Beachfront condo in San Diego		1
9	High-rise apartment in Dallas with city views		1
10	Elegant guesthouse in San Jose		1
11	Stylish flat in Austin		1
12	Quaint bungalow in Jacksonville		1
13	Suburban home in Fort Worth		1
14	Modern duplex in Columbus		1
15	Family-friendly house in Charlotte		1
16	Penthouse in San Francisco		1
17	Downtown apartment in Indianapolis		1
18	Studio in Seattle with mountain views		1
19	Eco-friendly cabin in Denver		1
20	Classic row house in Washington, D.C.		1



**Scenario:** Retrieve details of accommodations, their hosts, and the total number of bookings per accommodation.

**Purpose:**

- Tests the connection between accommodation, hostprofile, and booking.
- Validates the usage of HostID as a foreign key in accommodation and AccommodationID in booking.



## Test Case 3 (Booking)

```
SELECT
    g.GuestID,
    u.Name AS GuestName,
    b.CheckInDate,
    b.TotalCost
FROM
    booking b
JOIN
    guestprofile g ON b.GuestID = g.GuestID
JOIN
    user u ON b.BookedByUserID = u.UserID
WHERE
    b.TotalCost > 500
ORDER BY
    b.CheckInDate ASC;
```

	□ GuestID ▼	□ GuestName ▼	□ CheckInDate ▼	□ TotalCost ▼
1	1	John Doe	2024-12-01	600.00
2	2	Jane Smith	2024-12-03	750.00
3	3	Alice Johnson	2024-12-08	520.00
4	4	Bob Brown	2024-12-10	1000.00
5	8	Fiona Gray	2024-12-25	525.00
6	18	Paul Silver	2025-01-02	520.00



**Scenario:** Retrieve bookings made by guests, including guest names, check-in dates, and total costs, where the total cost exceeds 500.

**Purpose:**

- Tests the relationship between booking, guestprofile, and user.
- Validates foreign keys: GuestID in booking and BookedByUserID in user.



## Summary

In Phase 2, the Airbnb database was refined and expanded based on the tutor's feedback from Phase 1.

The enhancements included significant structural improvements to ensure functionality, scalability, and compliance with the requirements.

Key adjustments were made, such as linking the Payment table with User/Guest to track payment details, integrating contact information directly into the User table for simplicity, and connecting Location with User to store user-specific location data. Additionally, new tables, including HouseRules and Neighborhood, were introduced to enrich the system by adding listing-specific rules and neighborhood information.

The SQL schema was updated accordingly, and 20+ entries were populated for each table to meet data requirements.

Three meaningful test cases were designed and executed to validate relationships, data integrity, and the performance of the database. These test cases primarily focused on workflows like guest bookings, host accommodations, and payment tracking, utilizing joins to extract insights across multiple tables.

The results confirmed the effectiveness of the design in maintaining data consistency and supporting complex queries. This phase successfully lays the foundation for further development and testing in Phase 3.



## List of entities

<b>User</b> .....	Slide 04	<b>IncomeCalculator</b> .....	Slide 17
<b>HostProfile</b> .....	Slide 05	<b>InsurancePolicy</b> .....	Slide 18
<b>Location</b> .....	Slide 06	<b>ListingImage</b> .....	Slide 19
<b>Accommodation</b> .....	Slide 07	<b>Message</b> .....	Slide 20
<b>Amenities</b> .....	Slide 08	<b>Payment</b> .....	Slide 21
<b>AccommodationAmenities</b> ....	Slide 09	<b>PropertyPreference</b> .....	Slide 22
<b>Admin</b> .....	Slide 10	<b>Rating</b> .....	Slide 23
<b>Availability</b> .....	Slide 11	<b>Review</b> .....	Slide 24
<b>GuestProfile</b> .....	Slide 12	<b>SocialMediaLink</b> .....	Slide 25
<b>Booking</b> .....	Slide 13	<b>Transaction</b> .....	Slide 26
<b>Calendar</b> .....	Slide 14	<b>HouseRules</b> .....	Slide 27
<b>Commission</b> .....	Slide 15	<b>Neighborhood</b> .....	Slide 28
<b>Discount</b> .....	Slide 16	<b>AccommodationHouseRules</b> .....	Slide 29