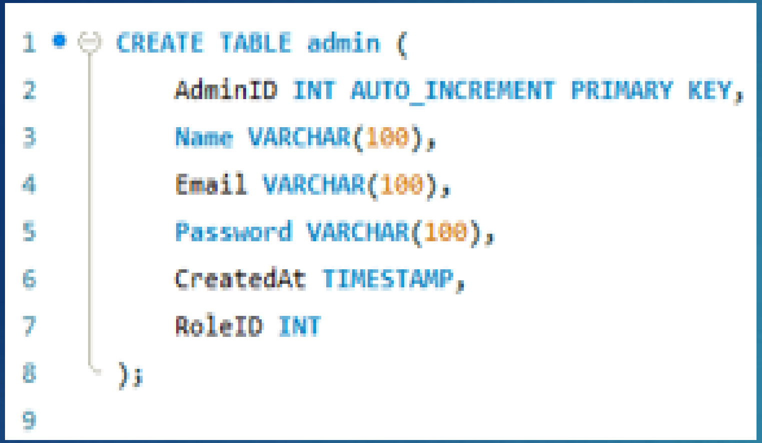
SQL DATA FILE

Creating the database schema named as ‘myairbnb’:

The Data base comprises of 24 interconnected tables, that has been crafted to show specific entities and relationships within the Airbnb database.

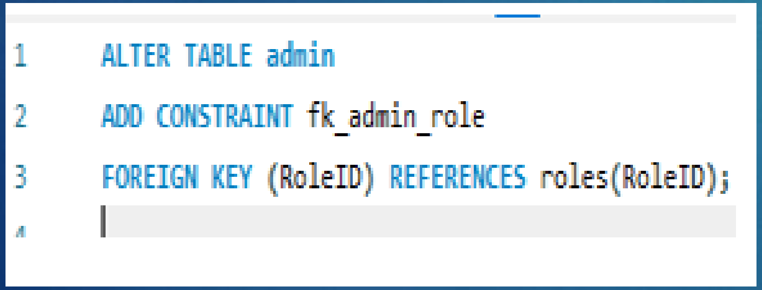
Admin

Is a special role within the user’s table. Use the SQL statement CREATE TABLE to define the structure of the admin table.



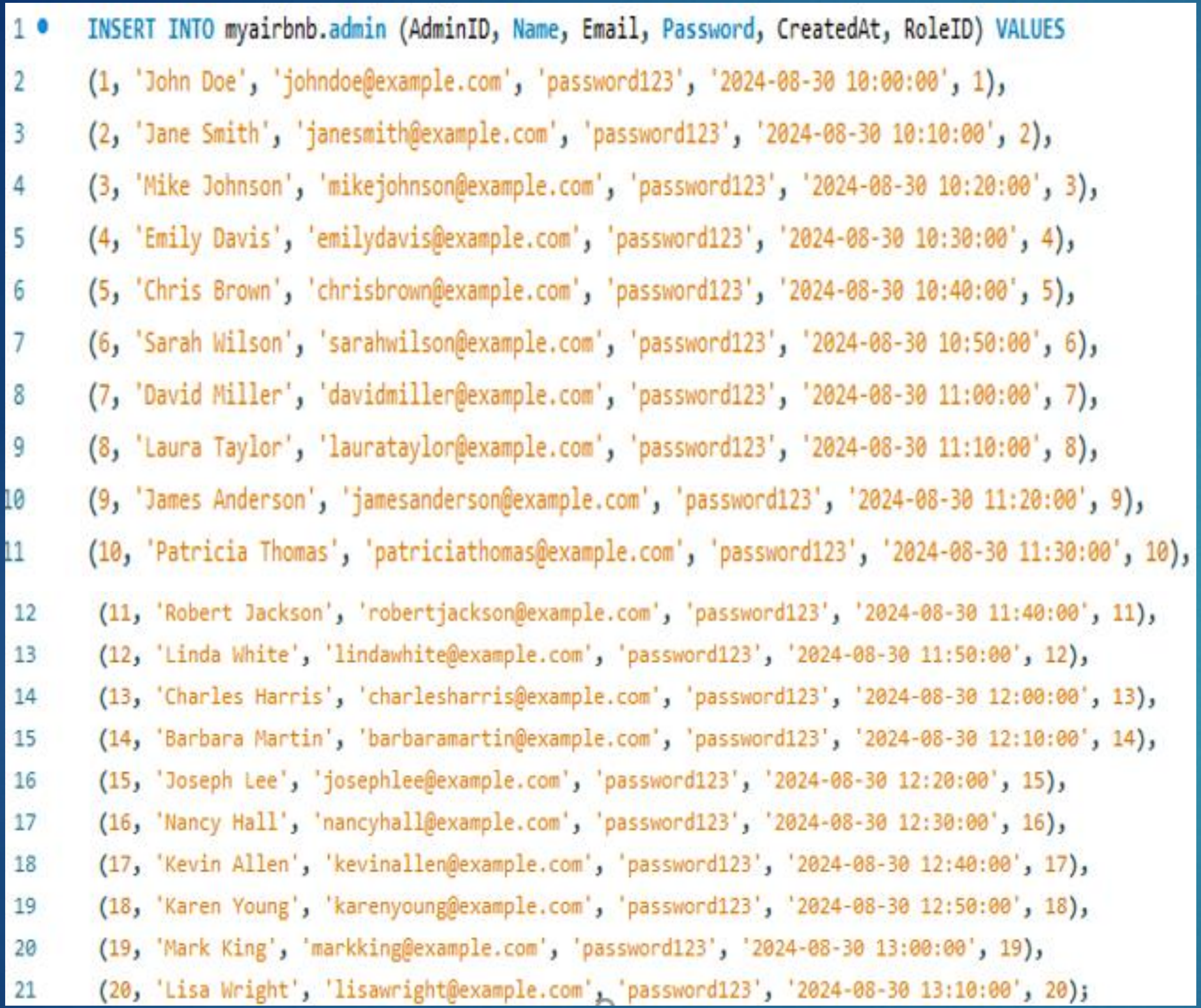
```
CREATE TABLE admin (
    AdminID INT AUTO_INCREMENT PRIMARY KEY,
    Name VARCHAR(100),
    Email VARCHAR(100),
    Password VARCHAR(100),
    CreatedAt TIMESTAMP,
    RoleID INT
);
```

ALTER TABLE : is used to modify the table in this case adding a foreign key.

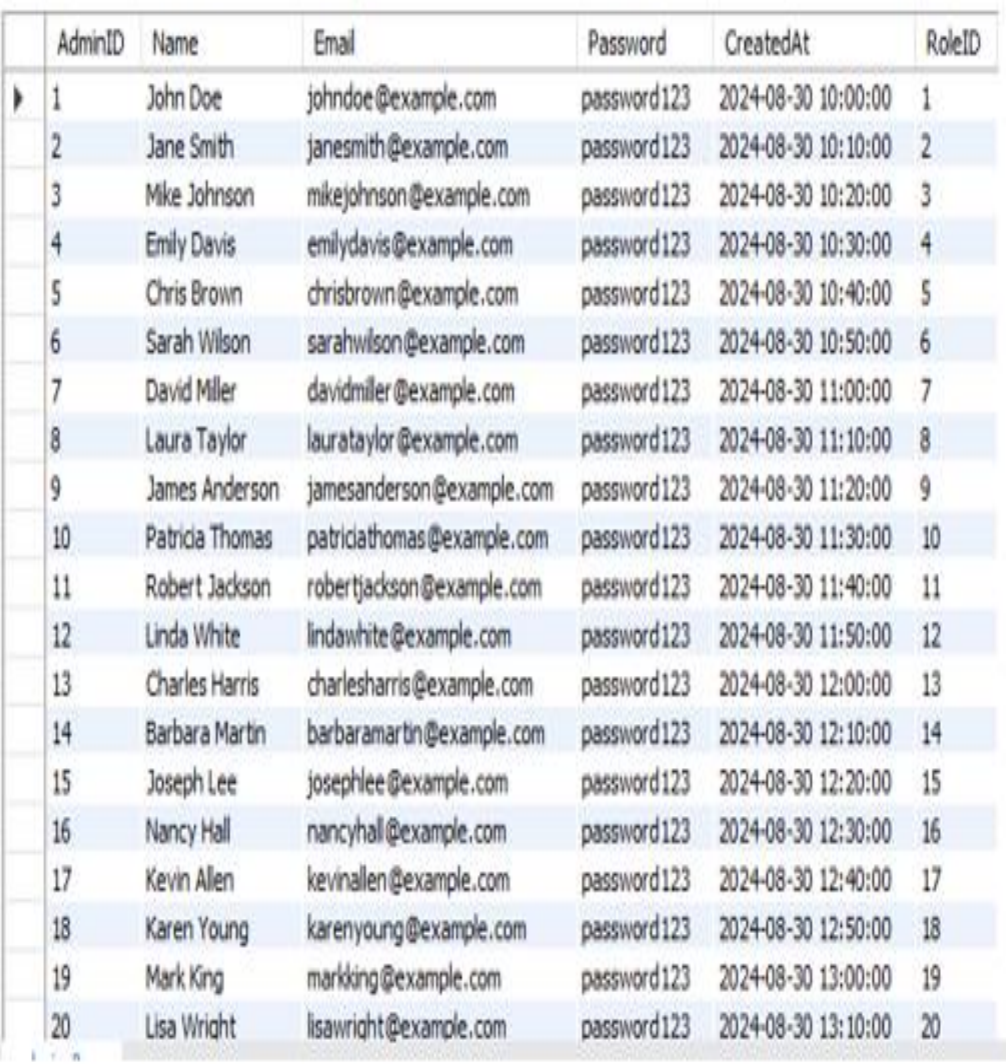


```
ALTER TABLE admin
ADD CONSTRAINT fk_admin_role
FOREIGN KEY (RoleID) REFERENCES roles(RoleID);
```

Admin (e.g ’legal admin’, ‘sales admin’,) personal information can be inserted into the table using the INSERT INTO statement.



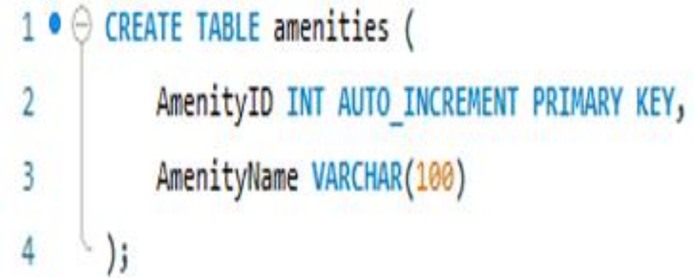
```sql
INSERT INTO myairbnb.admin (AdminID, Name, Email, Password, CreatedAt, RoleID) VALUES
(1, 'John Doe', 'johndoe@example.com', 'password123', '2024-08-30 10:00:00', 1),
(2, 'Jane Smith', 'janesmith@example.com', 'password123', '2024-08-30 10:10:00', 2),
(3, 'Mike Johnson', 'mikejohnson@example.com', 'password123', '2024-08-30 10:20:00', 3),
(4, 'Emily Davis', 'emilydavis@example.com', 'password123', '2024-08-30 10:30:00', 4),
(5, 'Chris Brown', 'chrisbrown@example.com', 'password123', '2024-08-30 10:40:00', 5),
(6, 'Sarah Wilson', 'sarahwilson@example.com', 'password123', '2024-08-30 10:50:00', 6),
(7, 'David Miller', 'davidmiller@example.com', 'password123', '2024-08-30 11:00:00', 7),
(8, 'Laura Taylor', 'laurataylor@example.com', 'password123', '2024-08-30 11:10:00', 8),
(9, 'James Anderson', 'jamesanderson@example.com', 'password123', '2024-08-30 11:20:00', 9),
(10, 'Patricia Thomas', 'patriciathomas@example.com', 'password123', '2024-08-30 11:30:00', 10),
 (11, 'Robert Jackson', 'robertjackson@example.com', 'password123', '2024-08-30 11:40:00', 11),
 (12, 'Linda White', 'lindawhite@example.com', 'password123', '2024-08-30 11:50:00', 12),
 (13, 'Charles Harris', 'charlesharris@example.com', 'password123', '2024-08-30 12:00:00', 13),
 (14, 'Barbara Martin', 'barbaramartin@example.com', 'password123', '2024-08-30 12:10:00', 14),
 (15, 'Joseph Lee', 'josephlee@example.com', 'password123', '2024-08-30 12:20:00', 15),
 (16, 'Nancy Hall', 'nancyhall@example.com', 'password123', '2024-08-30 12:30:00', 16),
 (17, 'Kevin Allen', 'kevinallen@example.com', 'password123', '2024-08-30 12:40:00', 17),
 (18, 'Karen Young', 'karenyoung@example.com', 'password123', '2024-08-30 12:50:00', 18),
 (19, 'Mark King', 'markking@example.com', 'password123', '2024-08-30 13:00:00', 19),
 (20, 'Lisa Wright', 'lisawright@example.com', 'password123', '2024-08-30 13:10:00', 20);
```



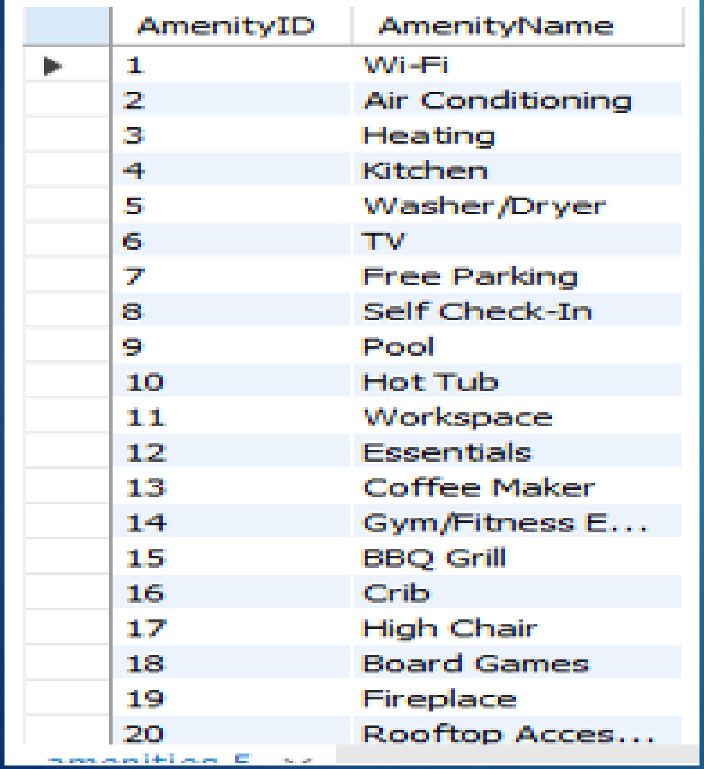
AdminID	Name	Email	Password	CreatedAt	RoleID
1	John Doe	johndoe@example.com	password123	2024-08-30 10:00:00	1
2	Jane Smith	janesmith@example.com	password123	2024-08-30 10:10:00	2
3	Mike Johnson	mikejohnson@example.com	password123	2024-08-30 10:20:00	3
4	Emily Davis	emilydavis@example.com	password123	2024-08-30 10:30:00	4
5	Chris Brown	chrisbrown@example.com	password123	2024-08-30 10:40:00	5
6	Sarah Wilson	sarahwilson@example.com	password123	2024-08-30 10:50:00	6
7	David Miller	davidmiller@example.com	password123	2024-08-30 11:00:00	7
8	Laura Taylor	laurataylor@example.com	password123	2024-08-30 11:10:00	8
9	James Anderson	jamesanderson@example.com	password123	2024-08-30 11:20:00	9
10	Patricia Thomas	patriciathomas@example.com	password123	2024-08-30 11:30:00	10
11	Robert Jackson	robertjackson@example.com	password123	2024-08-30 11:40:00	11
12	Linda White	lindawhite@example.com	password123	2024-08-30 11:50:00	12
13	Charles Harris	charlesharris@example.com	password123	2024-08-30 12:00:00	13
14	Barbara Martin	barbaramartin@example.com	password123	2024-08-30 12:10:00	14
15	Joseph Lee	josephlee@example.com	password123	2024-08-30 12:20:00	15
16	Nancy Hall	nancyhall@example.com	password123	2024-08-30 12:30:00	16
17	Kevin Allen	kevinallen@example.com	password123	2024-08-30 12:40:00	17
18	Karen Young	karenyoung@example.com	password123	2024-08-30 12:50:00	18
19	Mark King	markking@example.com	password123	2024-08-30 13:00:00	19
20	Lisa Wright	lisawright@example.com	password123	2024-08-30 13:10:00	20

Amenities

Stores list of amenities available. Uses the SQL statement CREATE TABLE to define the structure of the of amenities table.

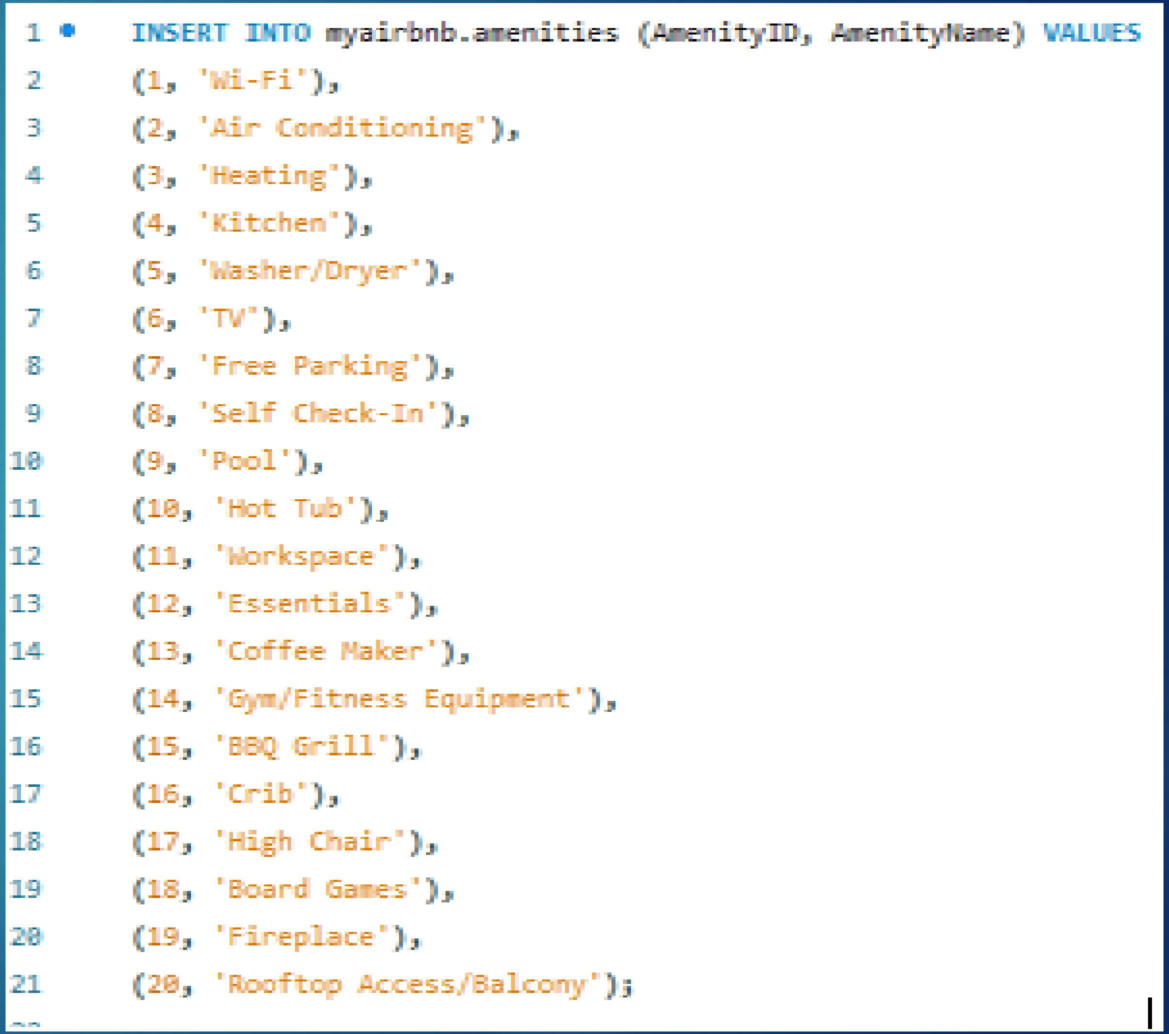


```sql
CREATE TABLE amenities (
    AmenityID INT AUTO_INCREMENT PRIMARY KEY,
    AmenityName VARCHAR(100)
);
```



	AmenityID	AmenityName
▶	1	Wi-Fi
	2	Air Conditioning
	3	Heating
	4	Kitchen
	5	Washer/Dryer
	6	TV
	7	Free Parking
	8	Self Check-In
	9	Pool
	10	Hot Tub
	11	Workspace
	12	Essentials
	13	Coffee Maker
	14	Gym/Fitness E...
	15	BBQ Grill
	16	Crib
	17	High Chair
	18	Board Games
	19	Fireplace
	20	Rooftop Acces...

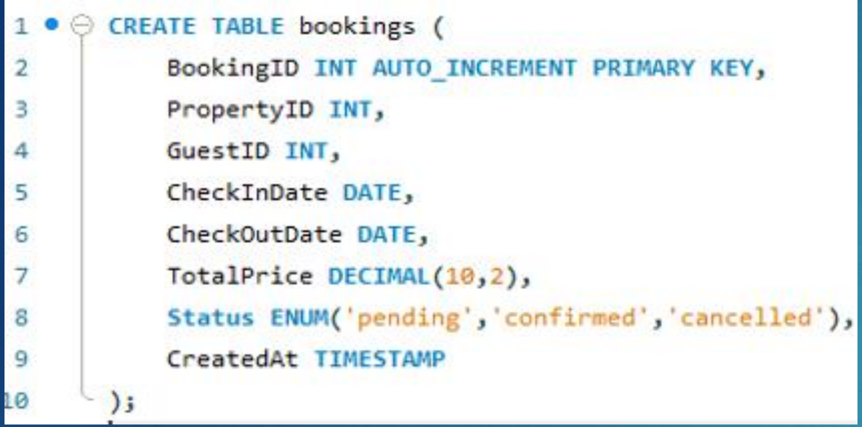
Amenities (e.g ‘WIFI’, ‘Pool’, ’Balcony’) and their ID can be inserted into the amenities table using the INSERT INTO statement.



```sql
INSERT INTO myairbnb.amenities (AmenityID, AmenityName) VALUES
(1, 'Wi-Fi'),
(2, 'Air Conditioning'),
(3, 'Heating'),
(4, 'Kitchen'),
(5, 'Washer/Dryer'),
(6, 'TV'),
(7, 'Free Parking'),
(8, 'Self Check-In'),
(9, 'Pool'),
(10, 'Hot Tub'),
(11, 'Workspace'),
(12, 'Essentials'),
(13, 'Coffee Maker'),
(14, 'Gym/Fitness Equipment'),
(15, 'BBQ Grill'),
(16, 'Crib'),
(17, 'High Chair'),
(18, 'Board Games'),
(19, 'Fireplace'),
(20, 'Rooftop Access/Balcony');
```

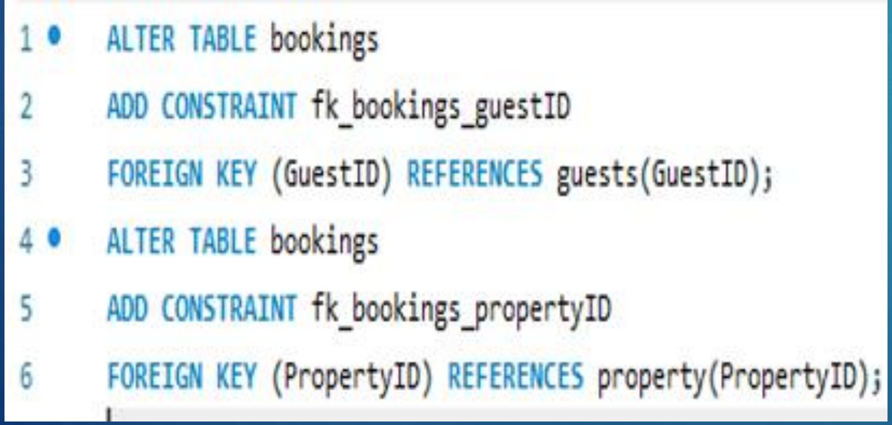
Bookings

Stores list of bookings. Uses the SQL statement CREATE TABLE to define the structure of the of bookings table.



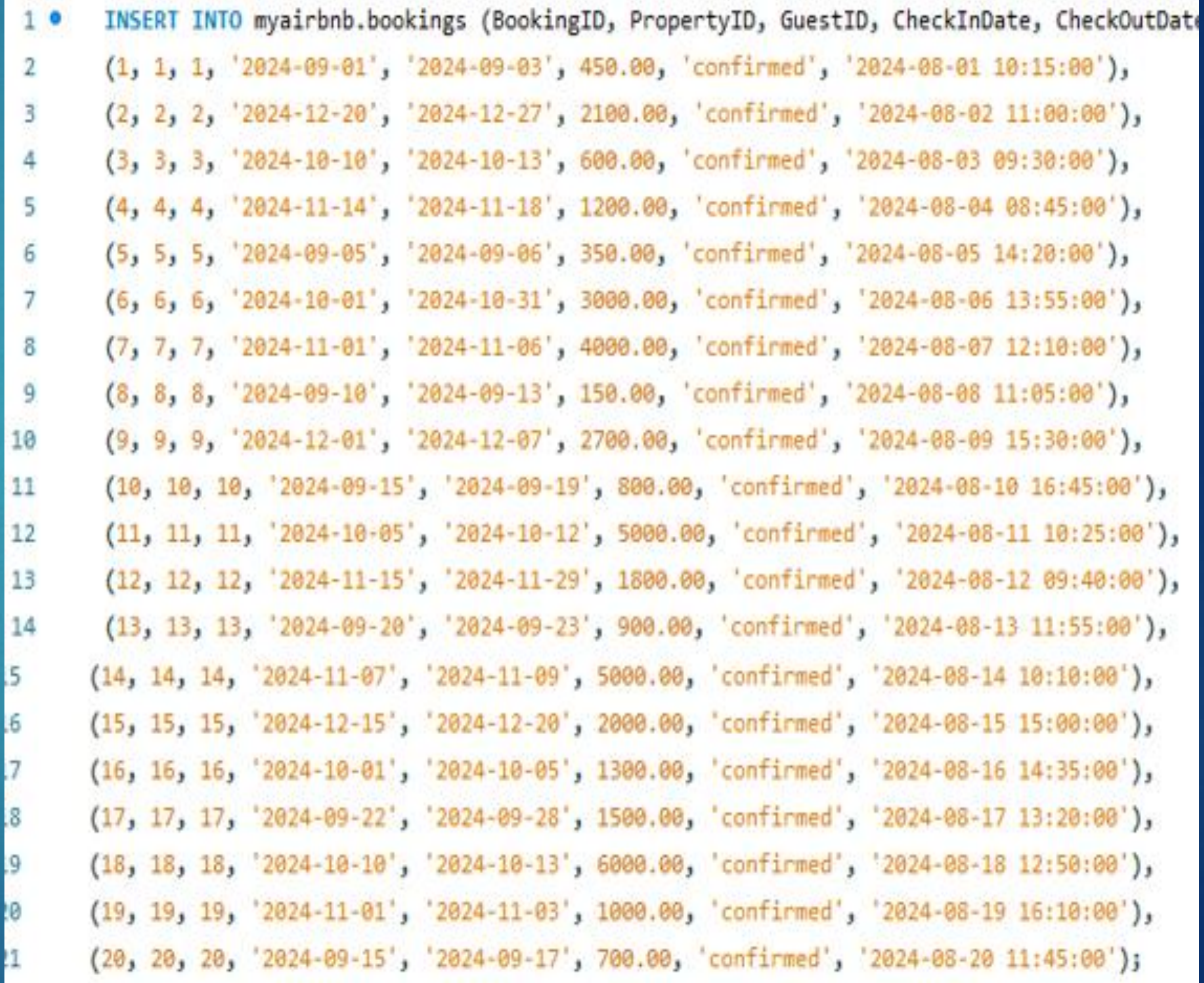
```
CREATE TABLE bookings (
    BookingID INT AUTO_INCREMENT PRIMARY KEY,
    PropertyID INT,
    GuestID INT,
    CheckInDate DATE,
    CheckOutDate DATE,
    TotalPrice DECIMAL(10,2),
    Status ENUM('pending','confirmed','cancelled'),
    CreatedAt TIMESTAMP
);
```

ALTER TABLE : is used to modify the table in this case adding a foreign key.

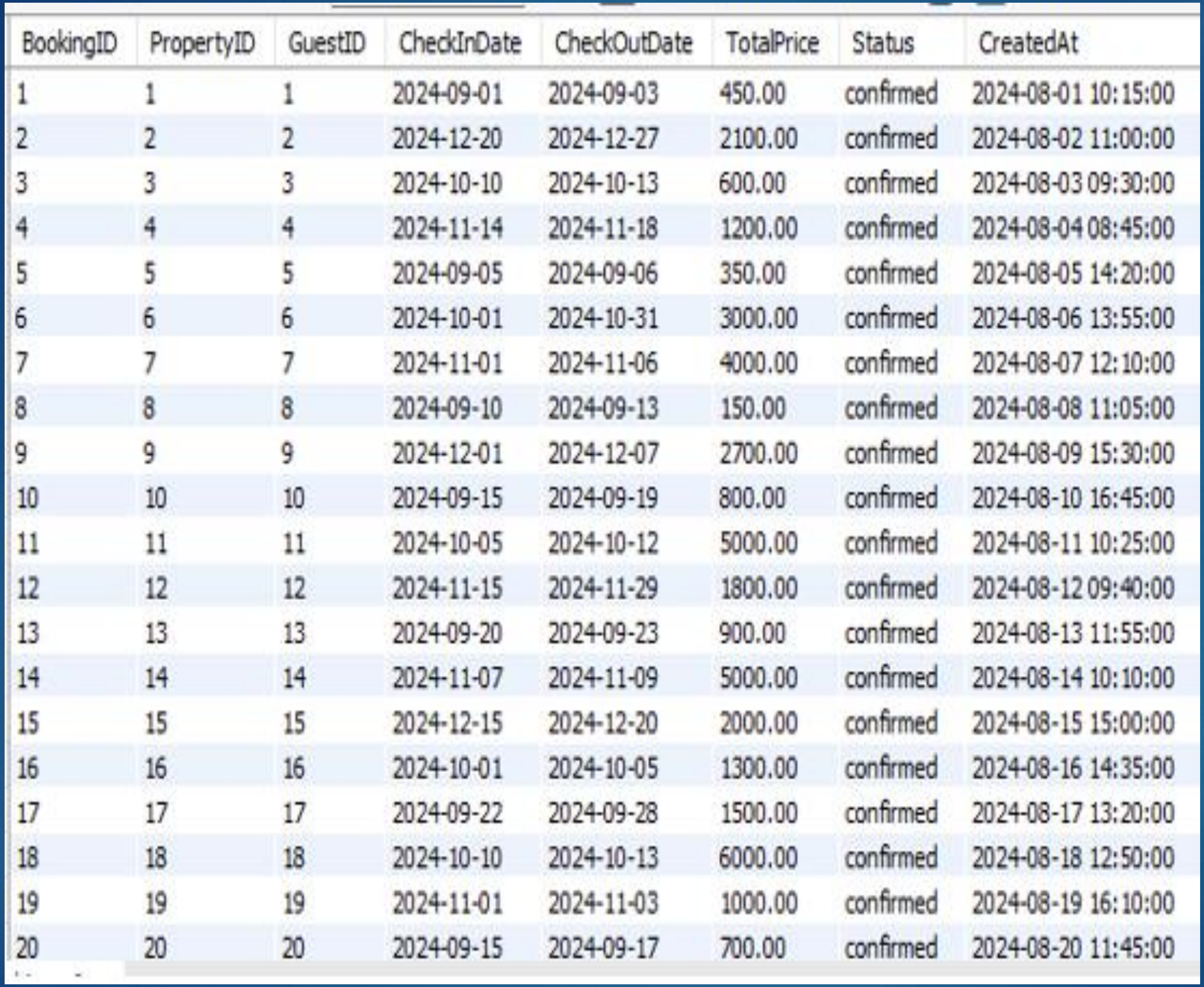


```
ALTER TABLE bookings
ADD CONSTRAINT fk_bookings_guestID
FOREIGN KEY (GuestID) REFERENCES guests(GuestID);
ALTER TABLE bookings
ADD CONSTRAINT fk_bookings_propertyID
FOREIGN KEY (PropertyID) REFERENCES property(PropertyID);
```

Bookings (e.g ‘Luxury Villa in Bali’, ‘Romantic Getaway in Paris’, ’Remote Work Stay in Lisbon’) and their details can be inserted into the table using the INSERT INTO statement.



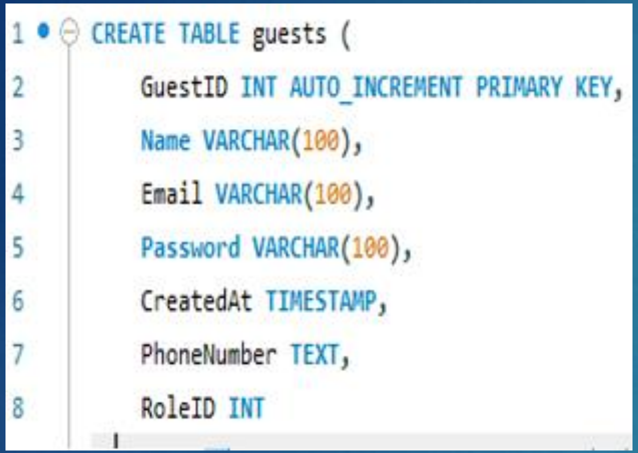
```
INSERT INTO myairbnb.bookings (BookingID, PropertyID, GuestID, CheckInDate, CheckOutDate
 (1, 1, 1, '2024-09-01', '2024-09-03', 450.00, 'confirmed', '2024-08-01 10:15:00'),
 (2, 2, 2, '2024-12-20', '2024-12-27', 2100.00, 'confirmed', '2024-08-02 11:00:00'),
 (3, 3, 3, '2024-10-10', '2024-10-13', 600.00, 'confirmed', '2024-08-03 09:30:00'),
 (4, 4, 4, '2024-11-14', '2024-11-18', 1200.00, 'confirmed', '2024-08-04 08:45:00'),
 (5, 5, 5, '2024-09-05', '2024-09-06', 350.00, 'confirmed', '2024-08-05 14:20:00'),
 (6, 6, 6, '2024-10-01', '2024-10-31', 3000.00, 'confirmed', '2024-08-06 13:55:00'),
 (7, 7, 7, '2024-11-01', '2024-11-06', 4000.00, 'confirmed', '2024-08-07 12:10:00'),
 (8, 8, 8, '2024-09-10', '2024-09-13', 150.00, 'confirmed', '2024-08-08 11:05:00'),
 (9, 9, 9, '2024-12-01', '2024-12-07', 2700.00, 'confirmed', '2024-08-09 15:30:00'),
 (10, 10, 10, '2024-09-15', '2024-09-19', 800.00, 'confirmed', '2024-08-10 16:45:00'),
 (11, 11, 11, '2024-10-05', '2024-10-12', 5000.00, 'confirmed', '2024-08-11 10:25:00'),
 (12, 12, 12, '2024-11-15', '2024-11-29', 1800.00, 'confirmed', '2024-08-12 09:40:00'),
 (13, 13, 13, '2024-09-20', '2024-09-23', 900.00, 'confirmed', '2024-08-13 11:55:00'),
(14, 14, 14, '2024-11-07', '2024-11-09', 5000.00, 'confirmed', '2024-08-14 10:10:00'),
(15, 15, 15, '2024-12-15', '2024-12-20', 2000.00, 'confirmed', '2024-08-15 15:00:00'),
(16, 16, 16, '2024-10-01', '2024-10-05', 1300.00, 'confirmed', '2024-08-16 14:35:00'),
(17, 17, 17, '2024-09-22', '2024-09-28', 1500.00, 'confirmed', '2024-08-17 13:20:00'),
(18, 18, 18, '2024-10-10', '2024-10-13', 6000.00, 'confirmed', '2024-08-18 12:50:00'),
(19, 19, 19, '2024-11-01', '2024-11-03', 1000.00, 'confirmed', '2024-08-19 16:10:00'),
(20, 20, 20, '2024-09-15', '2024-09-17', 700.00, 'confirmed', '2024-08-20 11:45:00');
```



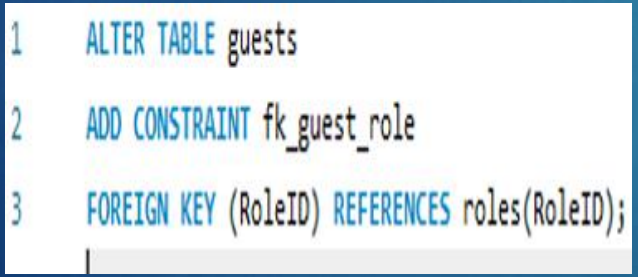
BookingID	PropertyID	GuestID	CheckInDate	CheckOutDate	TotalPrice	Status	CreatedAt
1	1	1	2024-09-01	2024-09-03	450.00	confirmed	2024-08-01 10:15:00
2	2	2	2024-12-20	2024-12-27	2100.00	confirmed	2024-08-02 11:00:00
3	3	3	2024-10-10	2024-10-13	600.00	confirmed	2024-08-03 09:30:00
4	4	4	2024-11-14	2024-11-18	1200.00	confirmed	2024-08-04 08:45:00
5	5	5	2024-09-05	2024-09-06	350.00	confirmed	2024-08-05 14:20:00
6	6	6	2024-10-01	2024-10-31	3000.00	confirmed	2024-08-06 13:55:00
7	7	7	2024-11-01	2024-11-06	4000.00	confirmed	2024-08-07 12:10:00
8	8	8	2024-09-10	2024-09-13	150.00	confirmed	2024-08-08 11:05:00
9	9	9	2024-12-01	2024-12-07	2700.00	confirmed	2024-08-09 15:30:00
10	10	10	2024-09-15	2024-09-19	800.00	confirmed	2024-08-10 16:45:00
11	11	11	2024-10-05	2024-10-12	5000.00	confirmed	2024-08-11 10:25:00
12	12	12	2024-11-15	2024-11-29	1800.00	confirmed	2024-08-12 09:40:00
13	13	13	2024-09-20	2024-09-23	900.00	confirmed	2024-08-13 11:55:00
14	14	14	2024-11-07	2024-11-09	5000.00	confirmed	2024-08-14 10:10:00
15	15	15	2024-12-15	2024-12-20	2000.00	confirmed	2024-08-15 15:00:00
16	16	16	2024-10-01	2024-10-05	1300.00	confirmed	2024-08-16 14:35:00
17	17	17	2024-09-22	2024-09-28	1500.00	confirmed	2024-08-17 13:20:00
18	18	18	2024-10-10	2024-10-13	6000.00	confirmed	2024-08-18 12:50:00
19	19	19	2024-11-01	2024-11-03	1000.00	confirmed	2024-08-19 16:10:00
20	20	20	2024-09-15	2024-09-17	700.00	confirmed	2024-08-20 11:45:00

Guests

Stores information about users who book properties. Uses the SQL statement CREATE TABLE to define the structure of the of guests table. While ALTER TABLE is used to modify the table, in this case adding a foreign key.

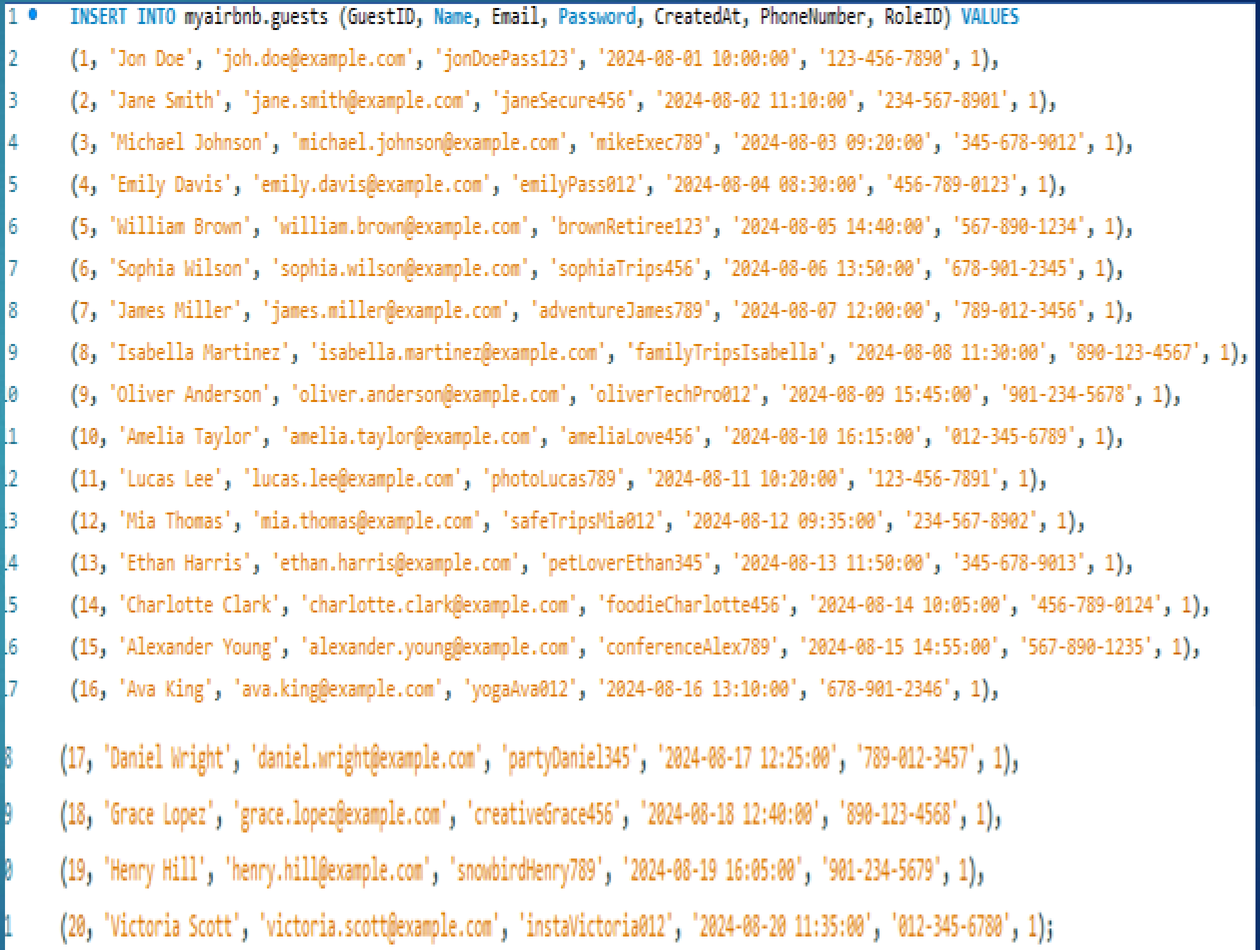


```sql
CREATE TABLE guests (
    GuestID INT AUTO_INCREMENT PRIMARY KEY,
    Name VARCHAR(100),
    Email VARCHAR(100),
    Password VARCHAR(100),
    CreatedAt TIMESTAMP,
    PhoneNumber TEXT,
    RoleID INT
```

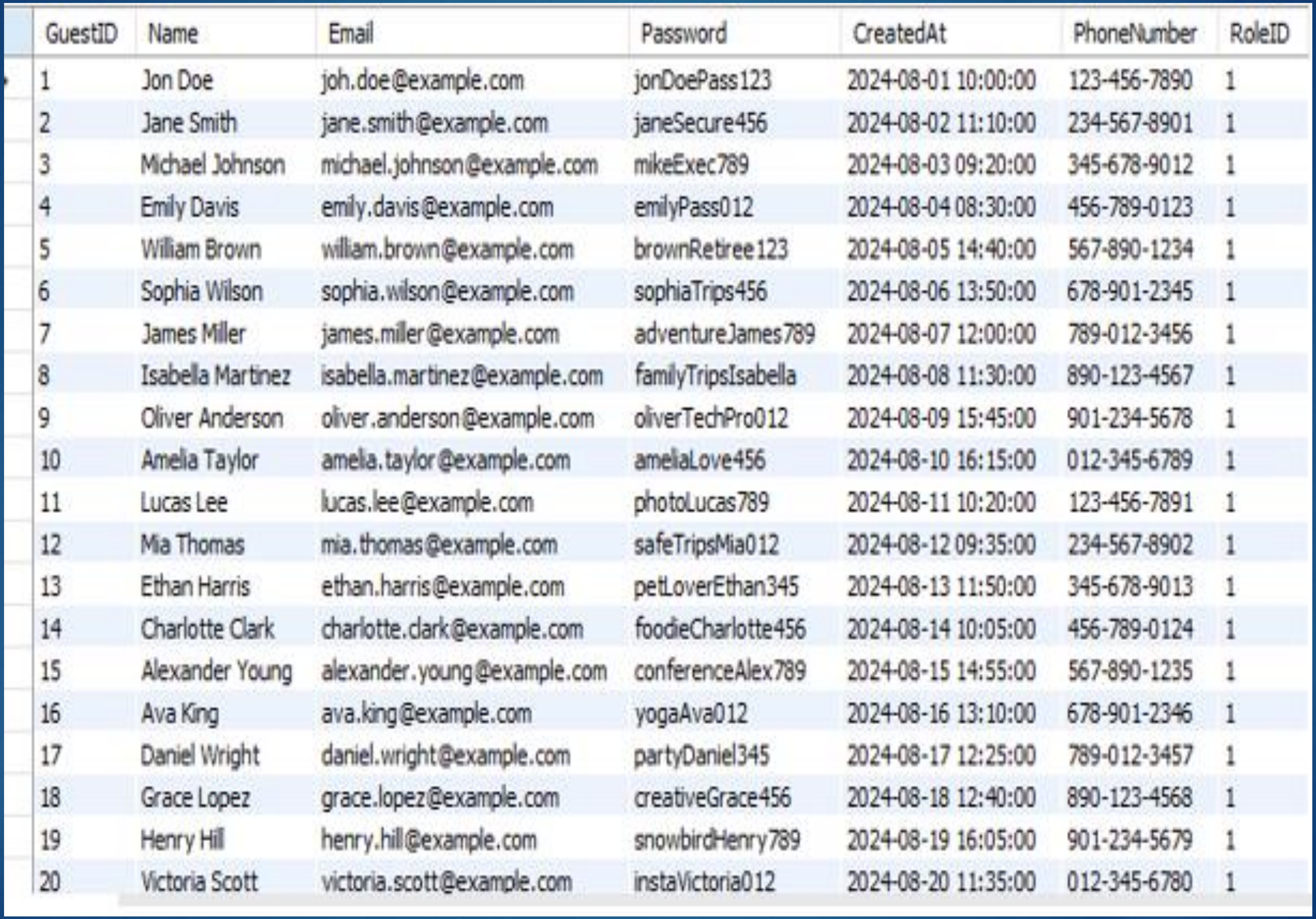


```sql
ALTER TABLE guests
ADD CONSTRAINT fk_guest_role
FOREIGN KEY (RoleID) REFERENCES roles(RoleID);
```

Guests (e.g Business traveller, students, Couples) and their personal details can be inserted into the table using the INSERT INTO statement.



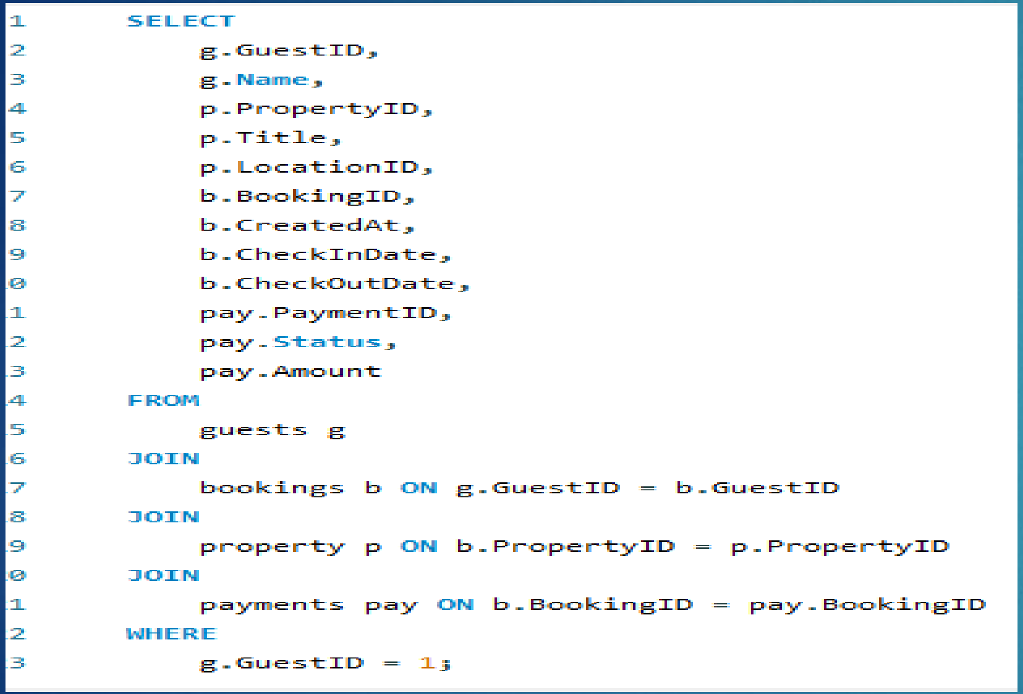
```sql
INSERT INTO myairbnb.guests (GuestID, Name, Email, Password, CreatedAt, PhoneNumber, RoleID) VALUES
(1, 'Jon Doe', 'joh.doe@example.com', 'jonDoePass123', '2024-08-01 10:00:00', '123-456-7890', 1),
(2, 'Jane Smith', 'jane.smith@example.com', 'janeSecure456', '2024-08-02 11:10:00', '234-567-8901', 1),
(3, 'Michael Johnson', 'michael.johnson@example.com', 'mikeExec789', '2024-08-03 09:20:00', '345-678-9012', 1),
(4, 'Emily Davis', 'emily.davis@example.com', 'emilyPass012', '2024-08-04 08:30:00', '456-789-0123', 1),
(5, 'William Brown', 'william.brown@example.com', 'brownRetiree123', '2024-08-05 14:40:00', '567-890-1234', 1),
(6, 'Sophia Wilson', 'sophia.wilson@example.com', 'sophiaTrips456', '2024-08-06 13:50:00', '678-901-2345', 1),
(7, 'James Miller', 'james.miller@example.com', 'adventureJames789', '2024-08-07 12:00:00', '789-012-3456', 1),
(8, 'Isabella Martinez', 'isabella.martinez@example.com', 'familyTripsIsabella', '2024-08-08 11:30:00', '890-123-4567', 1),
(9, 'Oliver Anderson', 'oliver.anderson@example.com', 'oliverTechPro012', '2024-08-09 15:45:00', '901-234-5678', 1),
(10, 'Amelia Taylor', 'amelia.taylor@example.com', 'ameliaLove456', '2024-08-10 16:15:00', '012-345-6789', 1),
(11, 'Lucas Lee', 'lucas.lee@example.com', 'photoLucas789', '2024-08-11 10:20:00', '123-456-7891', 1),
(12, 'Mia Thomas', 'mia.thomas@example.com', 'safeTripsMia012', '2024-08-12 09:35:00', '234-567-8902', 1),
(13, 'Ethan Harris', 'ethan.harris@example.com', 'petLoverEthan345', '2024-08-13 11:50:00', '345-678-9013', 1),
(14, 'Charlotte Clark', 'charlotte.clark@example.com', 'foodieCharlotte456', '2024-08-14 10:05:00', '456-789-0124', 1),
(15, 'Alexander Young', 'alexander.young@example.com', 'conferenceAlex789', '2024-08-15 14:55:00', '567-890-1235', 1),
(16, 'Ava King', 'ava.king@example.com', 'yogaAva012', '2024-08-16 13:10:00', '678-901-2346', 1),

(17, 'Daniel Wright', 'daniel.wright@example.com', 'partyDaniel345', '2024-08-17 12:25:00', '789-012-3457', 1),
(18, 'Grace Lopez', 'grace.lopez@example.com', 'creativeGrace456', '2024-08-18 12:40:00', '890-123-4568', 1),
(19, 'Henry Hill', 'henry.hill@example.com', 'snowbirdHenry789', '2024-08-19 16:05:00', '901-234-5679', 1),
(20, 'Victoria Scott', 'victoria.scott@example.com', 'instaVictoria012', '2024-08-20 11:35:00', '012-345-6780', 1);
```



GuestID	Name	Email	Password	CreatedAt	PhoneNumber	RoleID
1	Jon Doe	joh.doe@example.com	jonDoePass123	2024-08-01 10:00:00	123-456-7890	1
2	Jane Smith	jane.smith@example.com	janeSecure456	2024-08-02 11:10:00	234-567-8901	1
3	Michael Johnson	michael.johnson@example.com	mikeExec789	2024-08-03 09:20:00	345-678-9012	1
4	Emily Davis	emily.davis@example.com	emilyPass012	2024-08-04 08:30:00	456-789-0123	1
5	William Brown	william.brown@example.com	brownRetiree123	2024-08-05 14:40:00	567-890-1234	1
6	Sophia Wilson	sophia.wilson@example.com	sophiaTrips456	2024-08-06 13:50:00	678-901-2345	1
7	James Miller	james.miller@example.com	adventureJames789	2024-08-07 12:00:00	789-012-3456	1
8	Isabella Martinez	isabella.martinez@example.com	familyTripsIsabella	2024-08-08 11:30:00	890-123-4567	1
9	Oliver Anderson	oliver.anderson@example.com	oliverTechPro012	2024-08-09 15:45:00	901-234-5678	1
10	Amelia Taylor	amelia.taylor@example.com	ameliaLove456	2024-08-10 16:15:00	012-345-6789	1
11	Lucas Lee	lucas.lee@example.com	photoLucas789	2024-08-11 10:20:00	123-456-7891	1
12	Mia Thomas	mia.thomas@example.com	safeTripsMia012	2024-08-12 09:35:00	234-567-8902	1
13	Ethan Harris	ethan.harris@example.com	petLoverEthan345	2024-08-13 11:50:00	345-678-9013	1
14	Charlotte Clark	charlotte.clark@example.com	foodieCharlotte456	2024-08-14 10:05:00	456-789-0124	1
15	Alexander Young	alexander.young@example.com	conferenceAlex789	2024-08-15 14:55:00	567-890-1235	1
16	Ava King	ava.king@example.com	yogaAva012	2024-08-16 13:10:00	678-901-2346	1
17	Daniel Wright	daniel.wright@example.com	partyDaniel345	2024-08-17 12:25:00	789-012-3457	1
18	Grace Lopez	grace.lopez@example.com	creativeGrace456	2024-08-18 12:40:00	890-123-4568	1
19	Henry Hill	henry.hill@example.com	snowbirdHenry789	2024-08-19 16:05:00	901-234-5679	1
20	Victoria Scott	victoria.scott@example.com	instaVictoria012	2024-08-20 11:35:00	012-345-6780	1

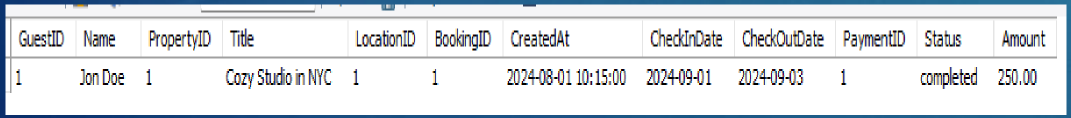
Guests: Test case: Combine data from three related tables

This query retrieves information for a specific guest (with GuestID = 1) from four related tables (guests, bookings, payments and property).



```
SELECT
    g.GuestID,
    g.Name,
    p.PropertyID,
    p.Title,
    p.LocationID,
    b.BookingID,
    b.CreatedAt,
    b.CheckInDate,
    b.CheckOutDate,
    pay.PaymentID,
    pay.Status,
    pay.Amount
FROM
    guests g
JOIN
    bookings b ON g.GuestID = b.GuestID
JOIN
    property p ON b.PropertyID = p.PropertyID
JOIN
    payments pay ON b.BookingID = pay.BookingID
WHERE
    g.GuestID = 1;
```

Result



GuestID	Name	PropertyID	Title	LocationID	BookingID	CreatedAt	CheckInDate	CheckOutDate	PaymentID	Status	Amount
1	Jon Doe	1	Cozy Studio in NYC	1	1	2024-08-01 10:15:00	2024-09-01	2024-09-03	1	completed	250.00

SELECT Clause: g.GuestID retrieves the unique ID of the guest from the guests table. g.Name retrieves the Name of the guest from the guests table. . p.PropertyID retrieves the unique property ID from the property table. p.Title retrieves the title (name) of the property from the property table. p.LocationID retrieves the location ID of the property from the property table. b.BookingID retrieves the unique booking ID from the bookings table. b.CreatedAt, b.CheckinDate and b.CheckoutDate retrieves the dates of creation, check in and check out of the booking from the bookings table. Pay.PaymentID, pay.Status, pay.Amount are columns from the payment table.

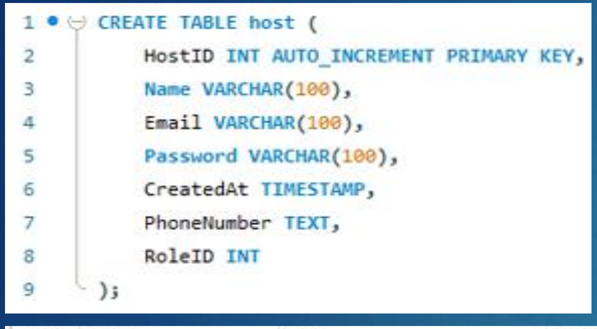
FROM Clause: The query starts from the guests table, which is the primary table in this context. This table contains information about guests.

JOIN Clauses: The first Join clause joins the guests table with the bookings table on the condition that the GuestID in the guests table matches the GuestID in the bookings table. The result will include only those rows where there is a matching GuestID in both tables, effectively linking each guest to their respective bookings. The second Join clause joins the bookings table with the property table on the condition that the PropertyID in the bookings table matches the PropertyID in the property table. The result will include only those rows where there is a matching PropertyID in both tables, effectively linking each booking to the details of the property being booked. The third join the booking table with the payment table.

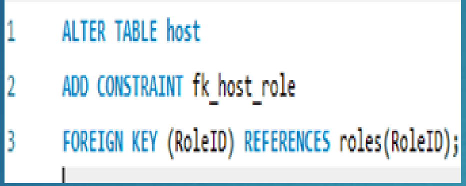
WHERE Clause: Filters the result based on the condition specified that the query should only return results where the GuestID in the guests table is equal to 1.

Host

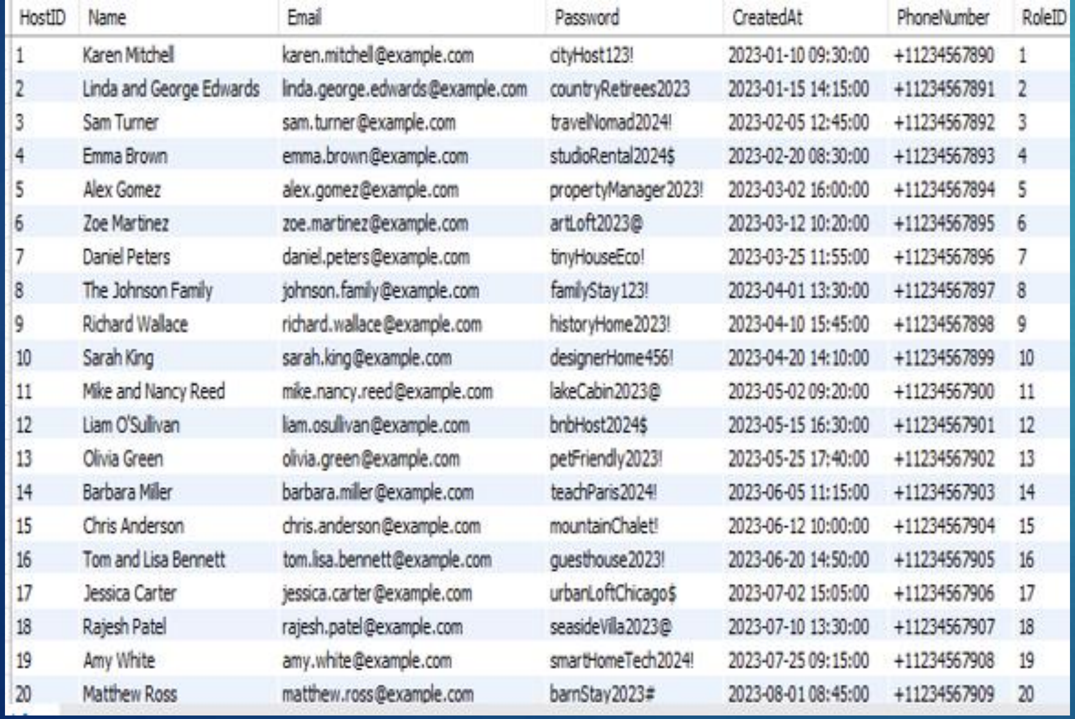
Stores information about users who list properties. Uses the SQL statement CREATE TABLE to define the structure of the of host table. While ALTER TABLE is used to modify the table, in this case adding a foreign key.



```sql
CREATE TABLE host (
    HostID INT AUTO_INCREMENT PRIMARY KEY,
    Name VARCHAR(100),
    Email VARCHAR(100),
    Password VARCHAR(100),
    CreatedAt TIMESTAMP,
    PhoneNumber TEXT,
    RoleID INT
);
```

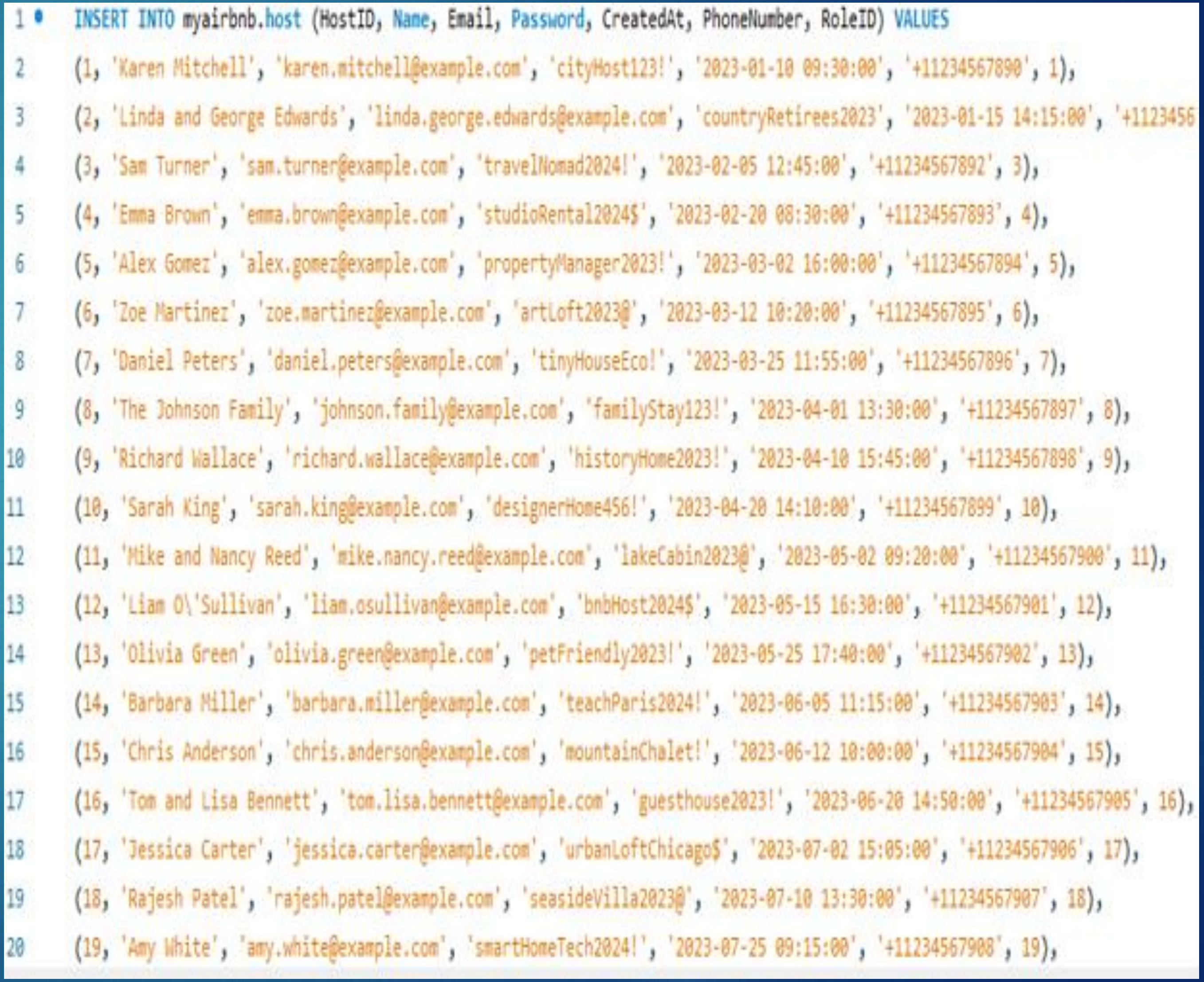


```sql
ALTER TABLE host
ADD CONSTRAINT fk_host_role
FOREIGN KEY (RoleID) REFERENCES roles(RoleID);
```



HostID	Name	Email	Password	CreatedAt	PhoneNumber	RoleID
1	Karen Mitchell	karen.mitchell@example.com	cityHost123!	2023-01-10 09:30:00	+11234567890	1
2	Linda and George Edwards	linda.george.edwards@example.com	countryRetirees2023	2023-01-15 14:15:00	+11234567891	2
3	Sam Turner	sam.turner@example.com	travelNomad2024!	2023-02-05 12:45:00	+11234567892	3
4	Emma Brown	emma.brown@example.com	studioRental2024$	2023-02-20 08:30:00	+11234567893	4
5	Alex Gomez	alex.gomez@example.com	propertyManager2023!	2023-03-02 16:00:00	+11234567894	5
6	Zoe Martinez	zoe.martinez@example.com	artLoft2023@	2023-03-12 10:20:00	+11234567895	6
7	Daniel Peters	daniel.peters@example.com	tinyHouseEco!	2023-03-25 11:55:00	+11234567896	7
8	The Johnson Family	johnson.family@example.com	familyStay123!	2023-04-01 13:30:00	+11234567897	8
9	Richard Wallace	richard.wallace@example.com	historyHome2023!	2023-04-10 15:45:00	+11234567898	9
10	Sarah King	sarah.king@example.com	designerHome456!	2023-04-20 14:10:00	+11234567899	10
11	Mike and Nancy Reed	mike.nancy.reed@example.com	lakeCabin2023@	2023-05-02 09:20:00	+11234567900	11
12	Liam O'Sullivan	liam.osullivan@example.com	bnbHost2024$	2023-05-15 16:30:00	+11234567901	12
13	Olivia Green	olivia.green@example.com	petFriendly2023!	2023-05-25 17:40:00	+11234567902	13
14	Barbara Miller	barbara.miller@example.com	teachParis2024!	2023-06-05 11:15:00	+11234567903	14
15	Chris Anderson	chris.anderson@example.com	mountainChalet!	2023-06-12 10:00:00	+11234567904	15
16	Tom and Lisa Bennett	tom.lisa.bennett@example.com	guesthouse2023!	2023-06-20 14:50:00	+11234567905	16
17	Jessica Carter	jessica.carter@example.com	urbanLoftChicago$	2023-07-02 15:05:00	+11234567906	17
18	Rajesh Patel	rajesh.patel@example.com	seasideVilla2023@	2023-07-10 13:30:00	+11234567907	18
19	Amy White	amy.white@example.com	smartHomeTech2024!	2023-07-25 09:15:00	+11234567908	19
20	Matthew Ross	matthew.ross@example.com	barnStay2023#	2023-08-01 08:45:00	+11234567909	20

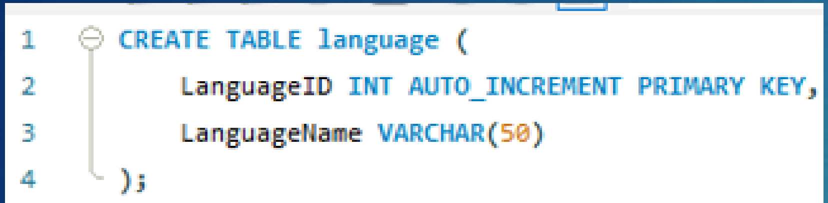
Host (e.g Family renting out a room) and their personal details can be inserted into the table using the INSERT INTO statement.



```sql
INSERT INTO myairbnb.host (HostID, Name, Email, Password, CreatedAt, PhoneNumber, RoleID) VALUES
(1, 'Karen Mitchell', 'karen.mitchell@example.com', 'cityHost123!', '2023-01-10 09:30:00', '+11234567890', 1),
(2, 'Linda and George Edwards', 'linda.george.edwards@example.com', 'countryRetirees2023', '2023-01-15 14:15:00', '+1123456
(3, 'Sam Turner', 'sam.turner@example.com', 'travelNomad2024!', '2023-02-05 12:45:00', '+11234567892', 3),
(4, 'Emma Brown', 'emma.brown@example.com', 'studioRental2024$', '2023-02-20 08:30:00', '+11234567893', 4),
(5, 'Alex Gomez', 'alex.gomez@example.com', 'propertyManager2023!', '2023-03-02 16:00:00', '+11234567894', 5),
(6, 'Zoe Martinez', 'zoe.martinez@example.com', 'artLoft2023@', '2023-03-12 10:20:00', '+11234567895', 6),
(7, 'Daniel Peters', 'daniel.peters@example.com', 'tinyHouseEco!', '2023-03-25 11:55:00', '+11234567896', 7),
(8, 'The Johnson Family', 'johnson.family@example.com', 'familyStay123!', '2023-04-01 13:30:00', '+11234567897', 8),
(9, 'Richard Wallace', 'richard.wallace@example.com', 'historyHome2023!', '2023-04-10 15:45:00', '+11234567898', 9),
(10, 'Sarah King', 'sarah.king@example.com', 'designerHome456!', '2023-04-20 14:10:00', '+11234567899', 10),
(11, 'Mike and Nancy Reed', 'mike.nancy.reed@example.com', 'lakeCabin2023@', '2023-05-02 09:20:00', '+11234567900', 11),
(12, 'Liam O\'Sullivan', 'liam.osullivan@example.com', 'bnbHost2024$', '2023-05-15 16:30:00', '+11234567901', 12),
(13, 'Olivia Green', 'olivia.green@example.com', 'petFriendly2023!', '2023-05-25 17:40:00', '+11234567902', 13),
(14, 'Barbara Miller', 'barbara.miller@example.com', 'teachParis2024!', '2023-06-05 11:15:00', '+11234567903', 14),
(15, 'Chris Anderson', 'chris.anderson@example.com', 'mountainChalet!', '2023-06-12 10:00:00', '+11234567904', 15),
(16, 'Tom and Lisa Bennett', 'tom.lisa.bennett@example.com', 'guesthouse2023!', '2023-06-20 14:50:00', '+11234567905', 16),
(17, 'Jessica Carter', 'jessica.carter@example.com', 'urbanLoftChicago$', '2023-07-02 15:05:00', '+11234567906', 17),
(18, 'Rajesh Patel', 'rajesh.patel@example.com', 'seasideVilla2023@', '2023-07-10 13:30:00', '+11234567907', 18),
(19, 'Amy White', 'amy.white@example.com', 'smartHomeTech2024!', '2023-07-25 09:15:00', '+11234567908', 19),
```

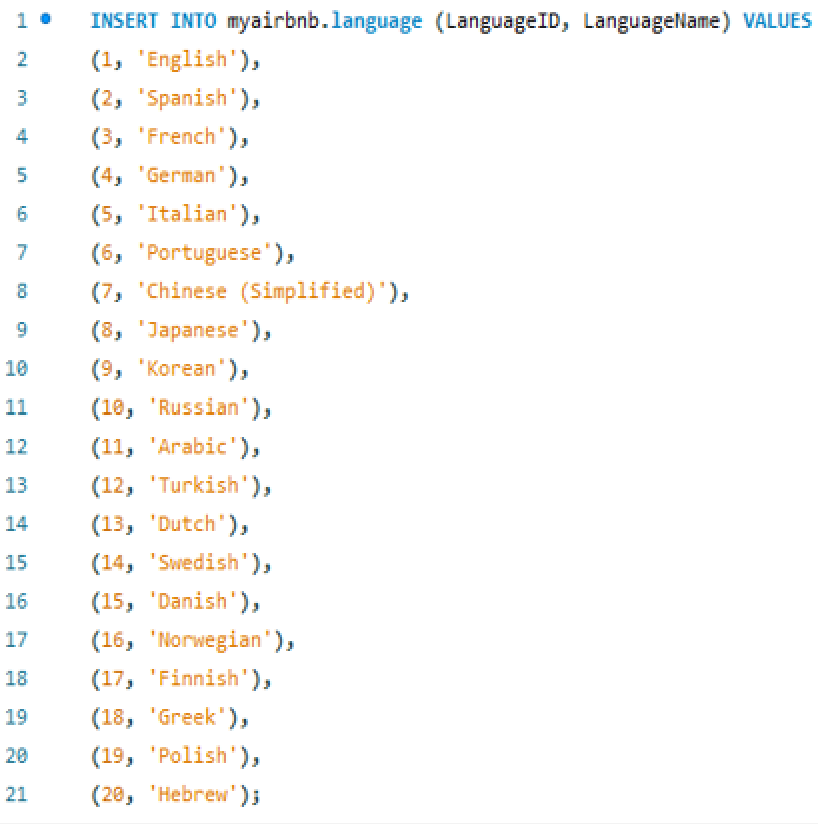
Language

Stores the different languages that can be supported or used on the platform. Uses the SQL statement CREATE TABLE to define the structure of the of language table.

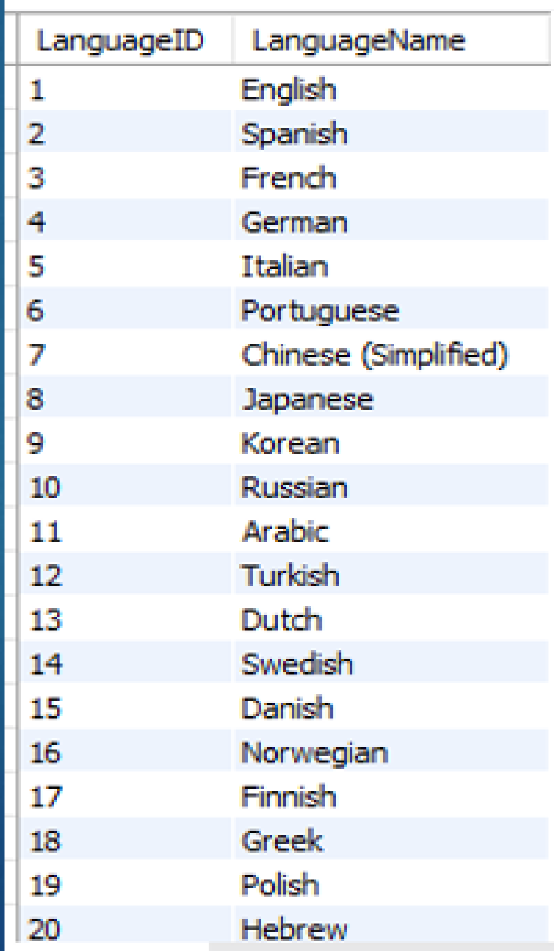


```sql
CREATE TABLE language (
    LanguageID INT AUTO_INCREMENT PRIMARY KEY,
    LanguageName VARCHAR(50)
);
```

Language (e.g English) and their ID can be inserted into the table using the INSERT INTO statement.



```sql
INSERT INTO myairbnb.language (LanguageID, LanguageName) VALUES
(1, 'English'),
(2, 'Spanish'),
(3, 'French'),
(4, 'German'),
(5, 'Italian'),
(6, 'Portuguese'),
(7, 'Chinese (Simplified)'),
(8, 'Japanese'),
(9, 'Korean'),
(10, 'Russian'),
(11, 'Arabic'),
(12, 'Turkish'),
(13, 'Dutch'),
(14, 'Swedish'),
(15, 'Danish'),
(16, 'Norwegian'),
(17, 'Finnish'),
(18, 'Greek'),
(19, 'Polish'),
(20, 'Hebrew');
```

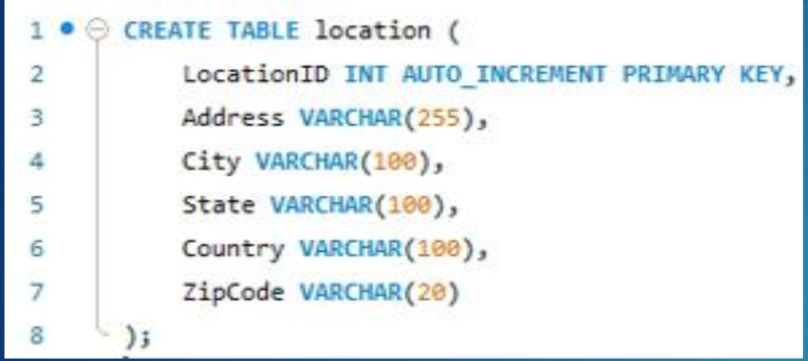


LanguageID	LanguageName
1	English
2	Spanish
3	French
4	German
5	Italian
6	Portuguese
7	Chinese (Simplified)
8	Japanese
9	Korean
10	Russian
11	Arabic
12	Turkish
13	Dutch
14	Swedish
15	Danish
16	Norwegian
17	Finnish
18	Greek
19	Polish
20	Hebrew

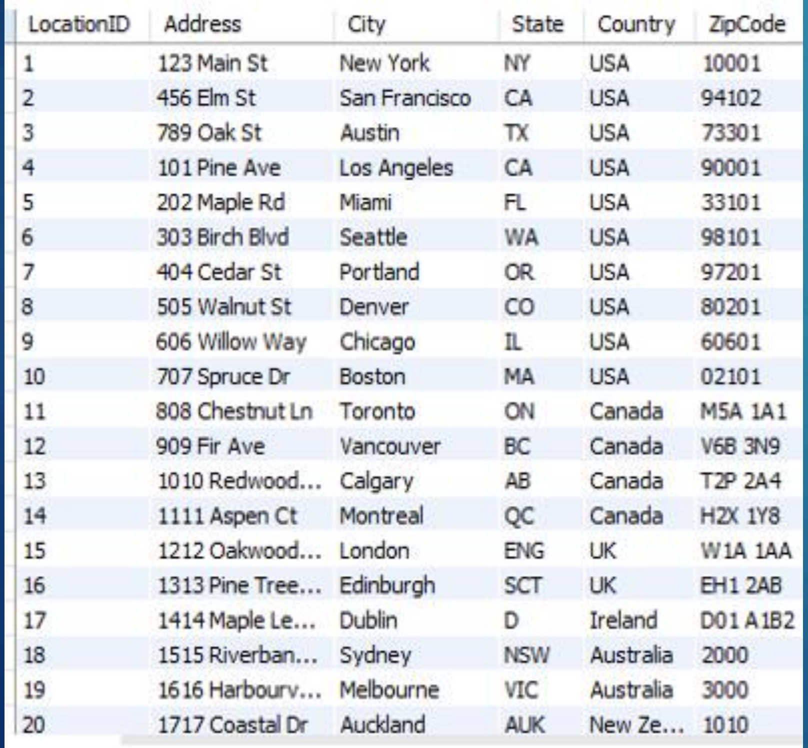
Location

Stores location details for properties. Uses the SQL statement CREATE TABLE to define the structure of the of location table.

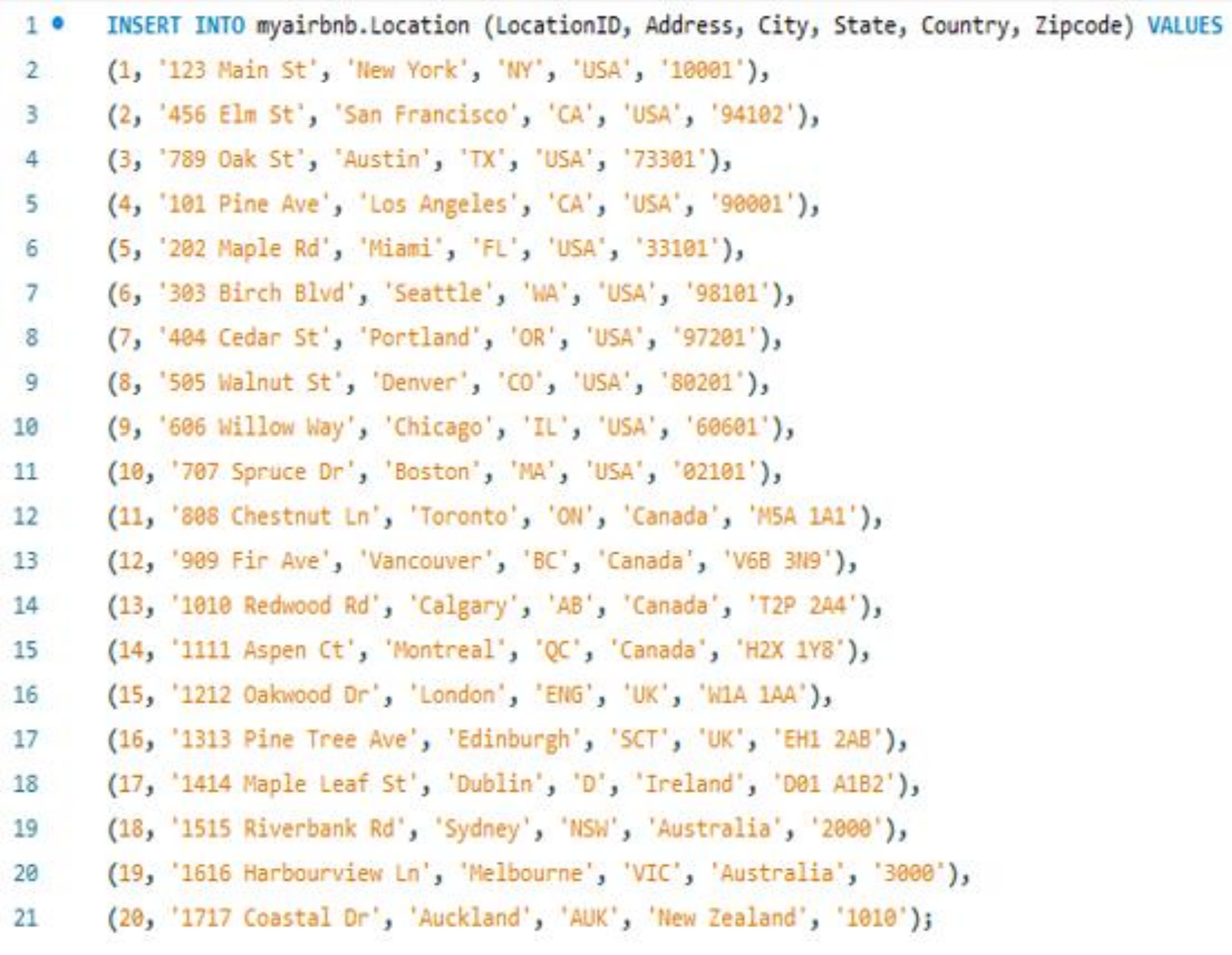
Location (e.g Edinburgh, Scotland) and their location details can be inserted into the table using the INSERT INTO statement.



```sql
CREATE TABLE location (
    LocationID INT AUTO_INCREMENT PRIMARY KEY,
    Address VARCHAR(255),
    City VARCHAR(100),
    State VARCHAR(100),
    Country VARCHAR(100),
    ZipCode VARCHAR(20)
);
```



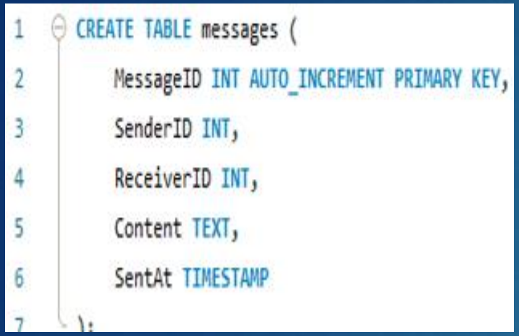
LocationID	Address	City	State	Country	ZipCode
1	123 Main St	New York	NY	USA	10001
2	456 Elm St	San Francisco	CA	USA	94102
3	789 Oak St	Austin	TX	USA	73301
4	101 Pine Ave	Los Angeles	CA	USA	90001
5	202 Maple Rd	Miami	FL	USA	33101
6	303 Birch Blvd	Seattle	WA	USA	98101
7	404 Cedar St	Portland	OR	USA	97201
8	505 Walnut St	Denver	CO	USA	80201
9	606 Willow Way	Chicago	IL	USA	60601
10	707 Spruce Dr	Boston	MA	USA	02101
11	808 Chestnut Ln	Toronto	ON	Canada	M5A 1A1
12	909 Fir Ave	Vancouver	BC	Canada	V6B 3N9
13	1010 Redwood...	Calgary	AB	Canada	T2P 2A4
14	1111 Aspen Ct	Montreal	QC	Canada	H2X 1Y8
15	1212 Oakwood...	London	ENG	UK	W1A 1AA
16	1313 Pine Tree...	Edinburgh	SCT	UK	EH1 2AB
17	1414 Maple Le...	Dublin	D	Ireland	D01 A1B2
18	1515 Riverban...	Sydney	NSW	Australia	2000
19	1616 Harbourv...	Melbourne	VIC	Australia	3000
20	1717 Coastal Dr	Auckland	AUK	New Ze...	1010



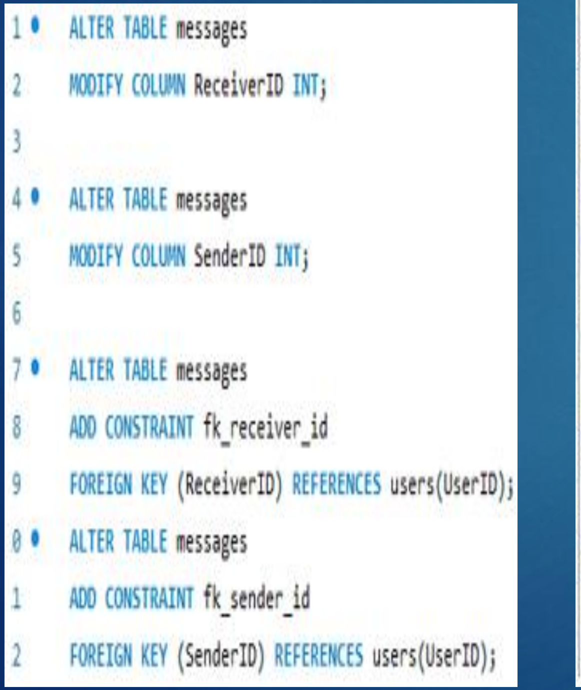
```sql
INSERT INTO myairbnb.Location (LocationID, Address, City, State, Country, Zipcode) VALUES
(1, '123 Main St', 'New York', 'NY', 'USA', '10001'),
(2, '456 Elm St', 'San Francisco', 'CA', 'USA', '94102'),
(3, '789 Oak St', 'Austin', 'TX', 'USA', '73301'),
(4, '101 Pine Ave', 'Los Angeles', 'CA', 'USA', '90001'),
(5, '202 Maple Rd', 'Miami', 'FL', 'USA', '33101'),
(6, '303 Birch Blvd', 'Seattle', 'WA', 'USA', '98101'),
(7, '404 Cedar St', 'Portland', 'OR', 'USA', '97201'),
(8, '505 Walnut St', 'Denver', 'CO', 'USA', '80201'),
(9, '606 Willow Way', 'Chicago', 'IL', 'USA', '60601'),
(10, '707 Spruce Dr', 'Boston', 'MA', 'USA', '02101'),
(11, '808 Chestnut Ln', 'Toronto', 'ON', 'Canada', 'M5A 1A1'),
(12, '909 Fir Ave', 'Vancouver', 'BC', 'Canada', 'V6B 3N9'),
(13, '1010 Redwood Rd', 'Calgary', 'AB', 'Canada', 'T2P 2A4'),
(14, '1111 Aspen Ct', 'Montreal', 'QC', 'Canada', 'H2X 1Y8'),
(15, '1212 Oakwood Dr', 'London', 'ENG', 'UK', 'W1A 1AA'),
(16, '1313 Pine Tree Ave', 'Edinburgh', 'SCT', 'UK', 'EH1 2AB'),
(17, '1414 Maple Leaf St', 'Dublin', 'D', 'Ireland', 'D01 A1B2'),
(18, '1515 Riverbank Rd', 'Sydney', 'NSW', 'Australia', '2000'),
(19, '1616 Harbourview Ln', 'Melbourne', 'VIC', 'Australia', '3000'),
(20, '1717 Coastal Dr', 'Auckland', 'AUK', 'New Zealand', '1010');
```

Messages

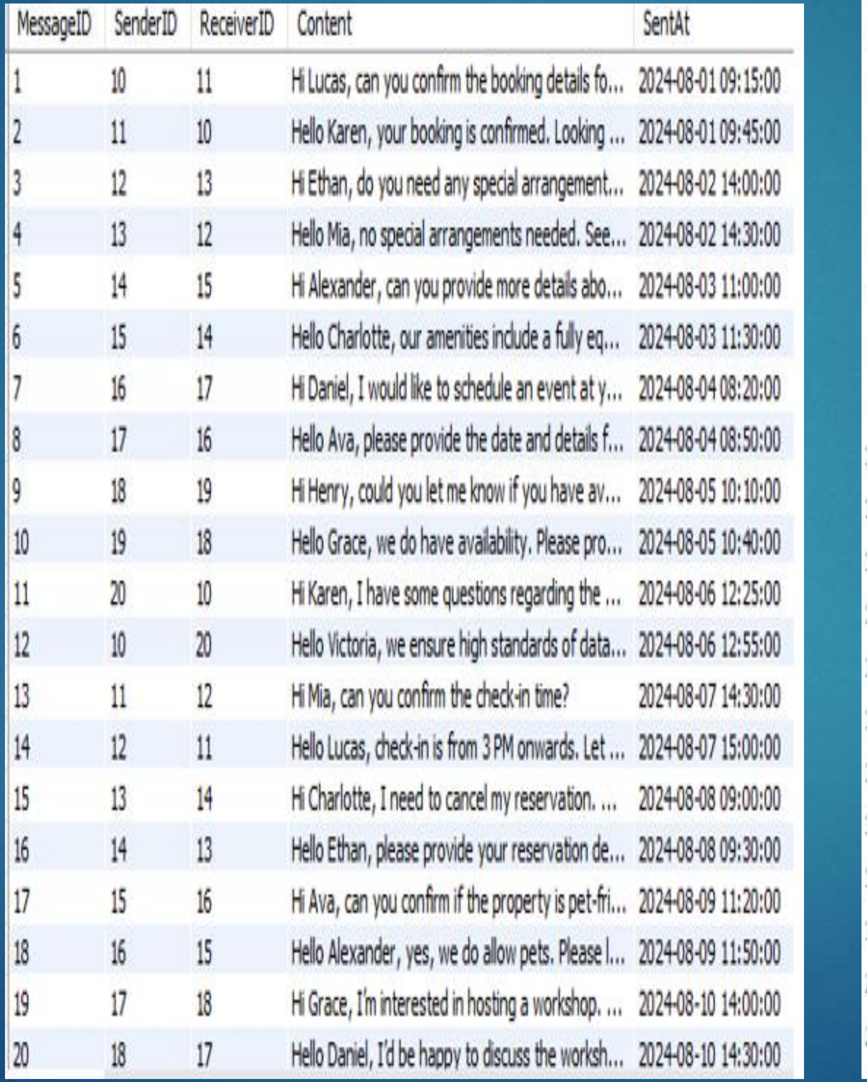
Stores communication messages between users (e.g., guest to host). Uses the SQL statement CREATE TABLE to define the structure of the of messages table. While ALTER TABLE is used to modify the table, in this case adding a foreign key.



```
CREATE TABLE messages (
    MessageID INT AUTO_INCREMENT PRIMARY KEY,
    SenderID INT,
    ReceiverID INT,
    Content TEXT,
    SentAt TIMESTAMP
);
```

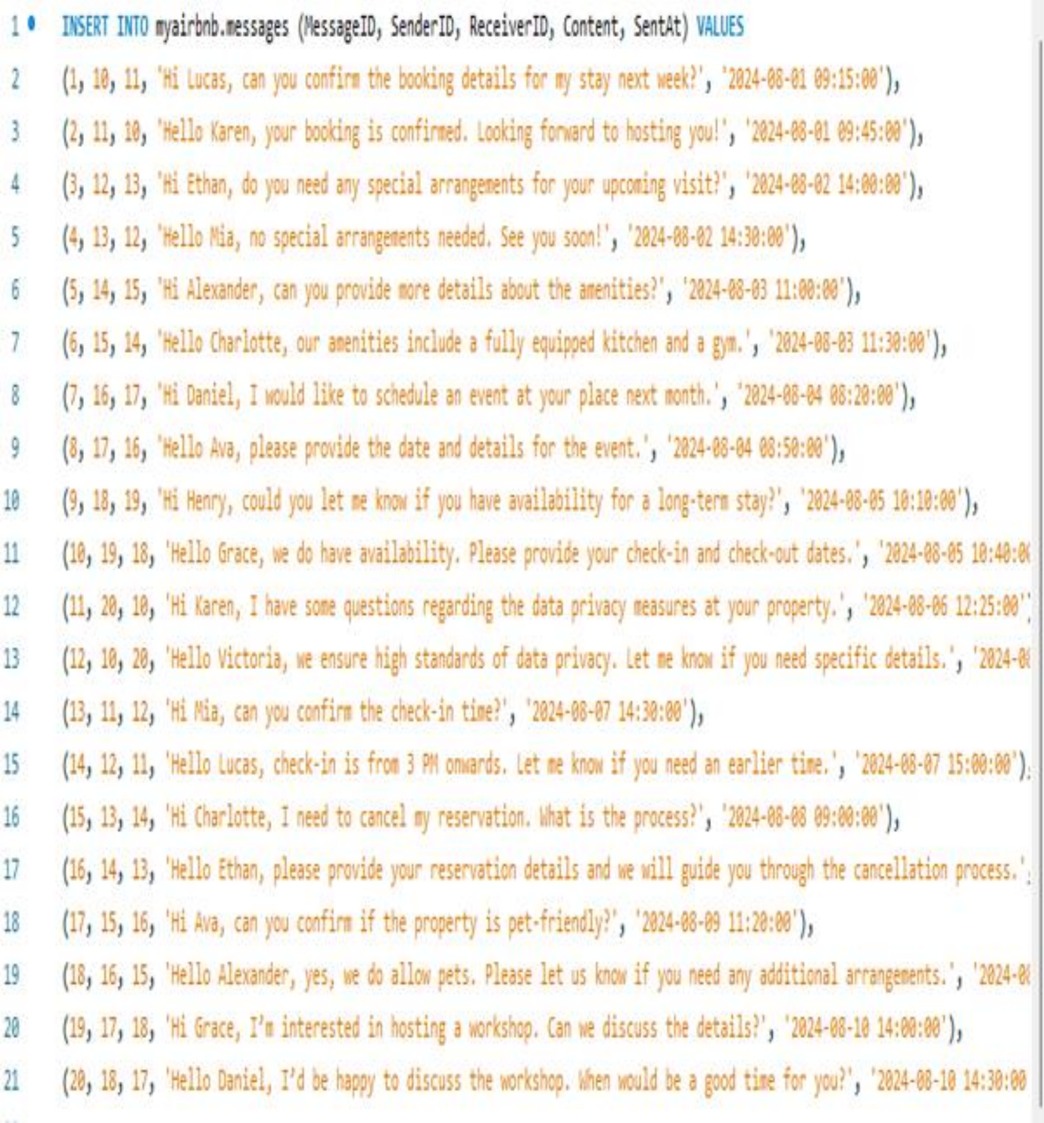


```
ALTER TABLE messages
MODIFY COLUMN ReceiverID INT;

ALTER TABLE messages
MODIFY COLUMN SenderID INT;

ALTER TABLE messages
ADD CONSTRAINT fk_receiver_id
FOREIGN KEY (ReceiverID) REFERENCES users(UserID);
ALTER TABLE messages
ADD CONSTRAINT fk_sender_id
FOREIGN KEY (SenderID) REFERENCES users(UserID);
```



MessageID	SenderID	ReceiverID	Content	SentAt
1	10	11	Hi Lucas, can you confirm the booking details fo...	2024-08-01 09:15:00
2	11	10	Hello Karen, your booking is confirmed. Looking ...	2024-08-01 09:45:00
3	12	13	Hi Ethan, do you need any special arrangement...	2024-08-02 14:00:00
4	13	12	Hello Mia, no special arrangements needed. See...	2024-08-02 14:30:00
5	14	15	Hi Alexander, can you provide more details abo...	2024-08-03 11:00:00
6	15	14	Hello Charlotte, our amenities include a fully eq...	2024-08-03 11:30:00
7	16	17	Hi Daniel, I would like to schedule an event at y...	2024-08-04 08:20:00
8	17	16	Hello Ava, please provide the date and details f...	2024-08-04 08:50:00
9	18	19	Hi Henry, could you let me know if you have av...	2024-08-05 10:10:00
10	19	18	Hello Grace, we do have availability. Please pro...	2024-08-05 10:40:00
11	20	10	Hi Karen, I have some questions regarding the ...	2024-08-06 12:25:00
12	10	20	Hello Victoria, we ensure high standards of data...	2024-08-06 12:55:00
13	11	12	Hi Mia, can you confirm the check-in time?	2024-08-07 14:30:00
14	12	11	Hello Lucas, check-in is from 3 PM onwards. Let ...	2024-08-07 15:00:00
15	13	14	Hi Charlotte, I need to cancel my reservation. ...	2024-08-08 09:00:00
16	14	13	Hello Ethan, please provide your reservation de...	2024-08-08 09:30:00
17	15	16	Hi Ava, can you confirm if the property is pet-fri...	2024-08-09 11:20:00
18	16	15	Hello Alexander, yes, we do allow pets. Please l...	2024-08-09 11:50:00
19	17	18	Hi Grace, I'm interested in hosting a workshop. ...	2024-08-10 14:00:00
20	18	17	Hello Daniel, I'd be happy to discuss the worksh...	2024-08-10 14:30:00

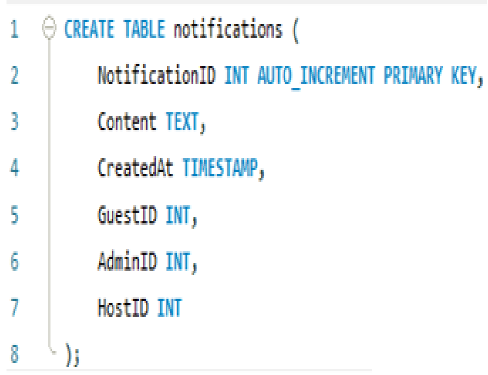
Details about messages can be inserted into the table using the INSERT INTO statement.



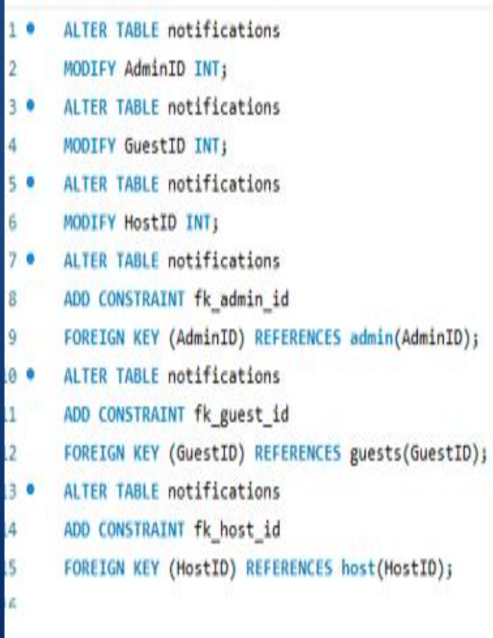
```
INSERT INTO myairbnb.messages (MessageID, SenderID, ReceiverID, Content, SentAt) VALUES
(1, 10, 11, 'Hi Lucas, can you confirm the booking details for my stay next week?', '2024-08-01 09:15:00'),
(2, 11, 10, 'Hello Karen, your booking is confirmed. Looking forward to hosting you!', '2024-08-01 09:45:00'),
(3, 12, 13, 'Hi Ethan, do you need any special arrangements for your upcoming visit?', '2024-08-02 14:00:00'),
(4, 13, 12, 'Hello Mia, no special arrangements needed. See you soon!', '2024-08-02 14:30:00'),
(5, 14, 15, 'Hi Alexander, can you provide more details about the amenities?', '2024-08-03 11:00:00'),
(6, 15, 14, 'Hello Charlotte, our amenities include a fully equipped kitchen and a gym.', '2024-08-03 11:30:00'),
(7, 16, 17, 'Hi Daniel, I would like to schedule an event at your place next month.', '2024-08-04 08:20:00'),
(8, 17, 16, 'Hello Ava, please provide the date and details for the event.', '2024-08-04 08:50:00'),
(9, 18, 19, 'Hi Henry, could you let me know if you have availability for a long-term stay?', '2024-08-05 10:10:00'),
(10, 19, 18, 'Hello Grace, we do have availability. Please provide your check-in and check-out dates.', '2024-08-05 10:40:0
(11, 20, 10, 'Hi Karen, I have some questions regarding the data privacy measures at your property.', '2024-08-06 12:25:00'
(12, 10, 20, 'Hello Victoria, we ensure high standards of data privacy. Let me know if you need specific details.', '2024-0
(13, 11, 12, 'Hi Mia, can you confirm the check-in time?', '2024-08-07 14:30:00'),
(14, 12, 11, 'Hello Lucas, check-in is from 3 PM onwards. Let me know if you need an earlier time.', '2024-08-07 15:00:00')
(15, 13, 14, 'Hi Charlotte, I need to cancel my reservation. What is the process?', '2024-08-08 09:00:00'),
(16, 14, 13, 'Hello Ethan, please provide your reservation details and we will guide you through the cancellation process.'
(17, 15, 16, 'Hi Ava, can you confirm if the property is pet-friendly?', '2024-08-09 11:20:00'),
(18, 16, 15, 'Hello Alexander, yes, we do allow pets. Please let us know if you need any additional arrangements.', '2024-0
(19, 17, 18, 'Hi Grace, I’m interested in hosting a workshop. Can we discuss the details?', '2024-08-10 14:00:00'),
(20, 18, 17, 'Hello Daniel, I’d be happy to discuss the workshop. When would be a good time for you?', '2024-08-10 14:30:00
```

Notifications

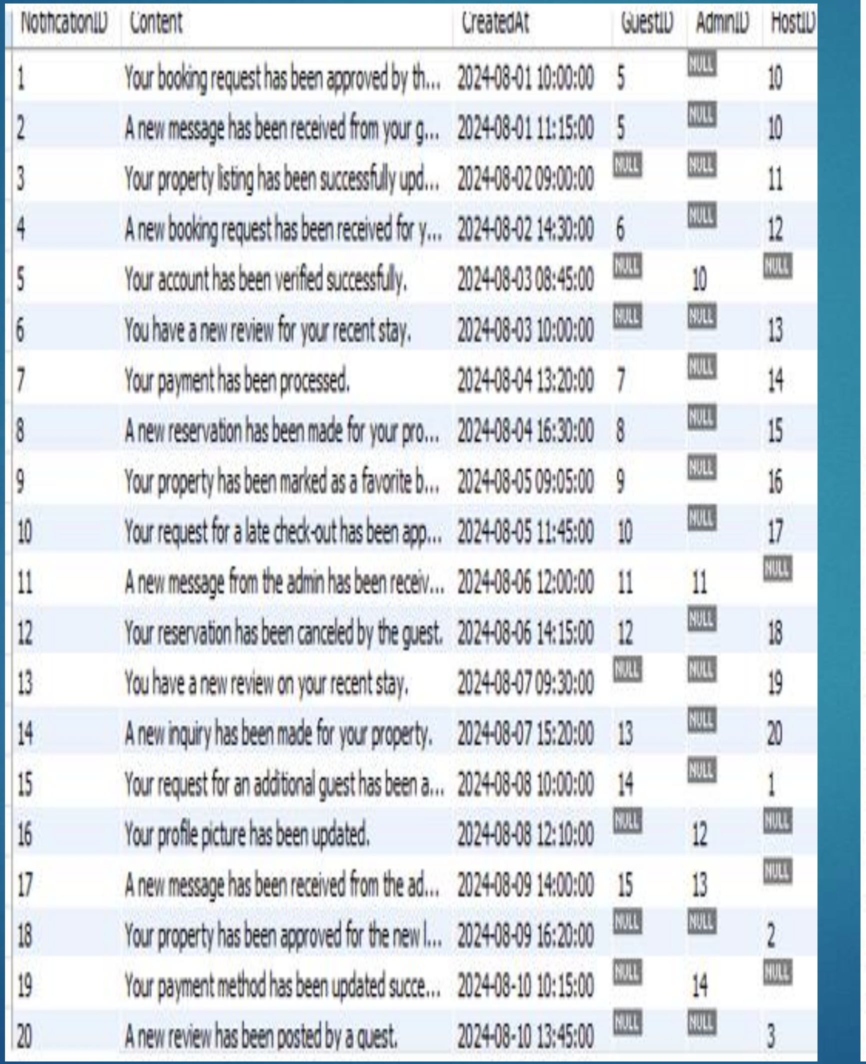
Stores notifications sent to users (guests, hosts, admins). The SQL statement CREATE TABLE is used to define the structure of the of notifications table. While ALTER TABLE is used to modify the table, in this case adding a foreign key.



```sql
CREATE TABLE notifications (
    NotificationID INT AUTO_INCREMENT PRIMARY KEY,
    Content TEXT,
    CreatedAt TIMESTAMP,
    GuestID INT,
    AdminID INT,
    HostID INT
);
```

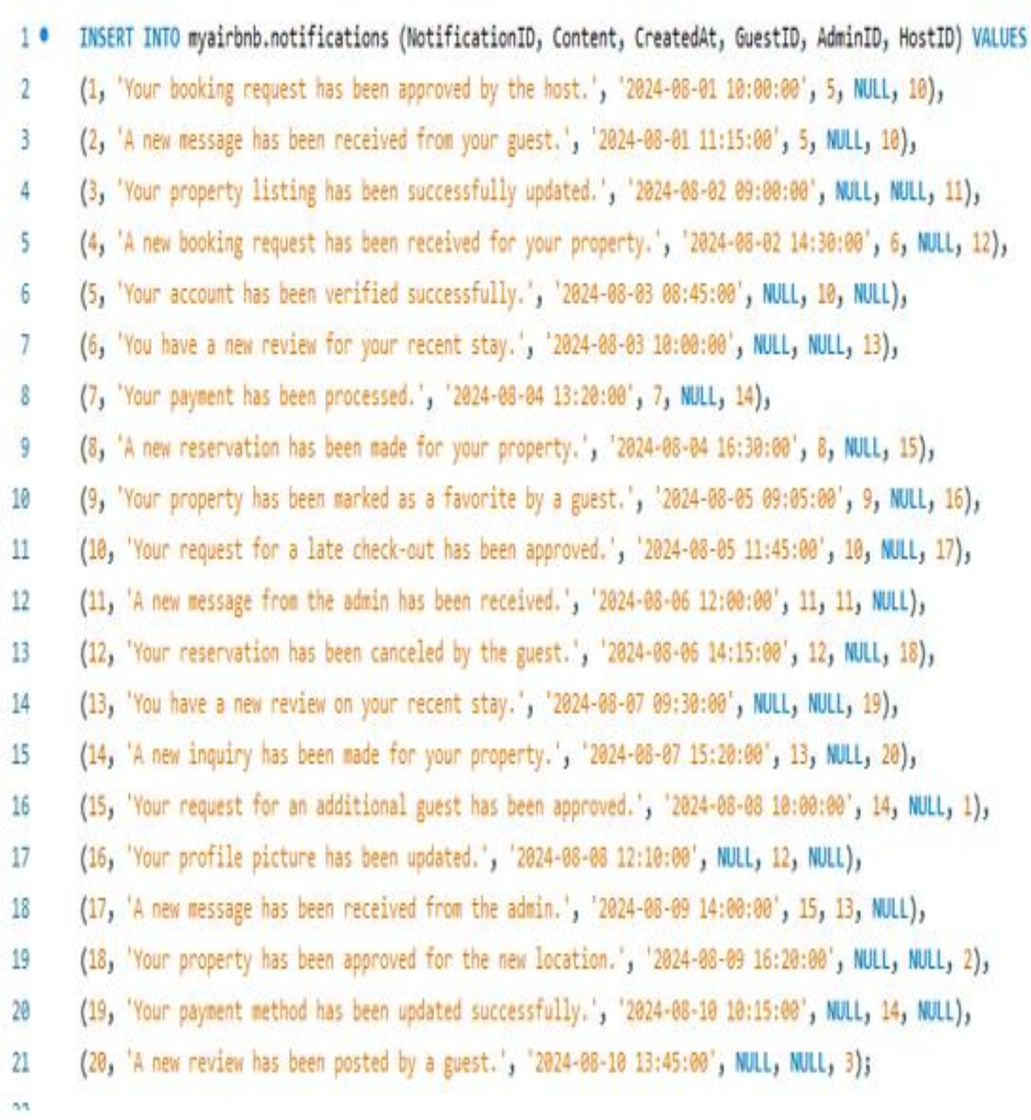


```sql
ALTER TABLE notifications
MODIFY AdminID INT;
ALTER TABLE notifications
MODIFY GuestID INT;
ALTER TABLE notifications
MODIFY HostID INT;
ALTER TABLE notifications
ADD CONSTRAINT fk_admin_id
FOREIGN KEY (AdminID) REFERENCES admin(AdminID);
ALTER TABLE notifications
ADD CONSTRAINT fk_guest_id
FOREIGN KEY (GuestID) REFERENCES guests(GuestID);
ALTER TABLE notifications
ADD CONSTRAINT fk_host_id
FOREIGN KEY (HostID) REFERENCES host(HostID);
```



NotificationID	Content	CreatedAt	GuestID	AdminID	HostID
1	Your booking request has been approved by th...	2024-08-01 10:00:00	5	NULL	10
2	A new message has been received from your g...	2024-08-01 11:15:00	5	NULL	10
3	Your property listing has been successfully upd...	2024-08-02 09:00:00	NULL	NULL	11
4	A new booking request has been received for y...	2024-08-02 14:30:00	6	NULL	12
5	Your account has been verified successfully.	2024-08-03 08:45:00	NULL	10	NULL
6	You have a new review for your recent stay.	2024-08-03 10:00:00	NULL	NULL	13
7	Your payment has been processed.	2024-08-04 13:20:00	7	NULL	14
8	A new reservation has been made for your pro...	2024-08-04 16:30:00	8	NULL	15
9	Your property has been marked as a favorite b...	2024-08-05 09:05:00	9	NULL	16
10	Your request for a late check-out has been app...	2024-08-05 11:45:00	10	NULL	17
11	A new message from the admin has been receiv...	2024-08-06 12:00:00	11	11	NULL
12	Your reservation has been canceled by the guest.	2024-08-06 14:15:00	12	NULL	18
13	You have a new review on your recent stay.	2024-08-07 09:30:00	NULL	NULL	19
14	A new inquiry has been made for your property.	2024-08-07 15:20:00	13	NULL	20
15	Your request for an additional guest has been a...	2024-08-08 10:00:00	14	NULL	1
16	Your profile picture has been updated.	2024-08-08 12:10:00	NULL	12	NULL
17	A new message has been received from the ad...	2024-08-09 14:00:00	15	13	NULL
18	Your property has been approved for the new l...	2024-08-09 16:20:00	NULL	NULL	2
19	Your payment method has been updated succe...	2024-08-10 10:15:00	NULL	14	NULL
20	A new review has been posted by a guest.	2024-08-10 13:45:00	NULL	NULL	3

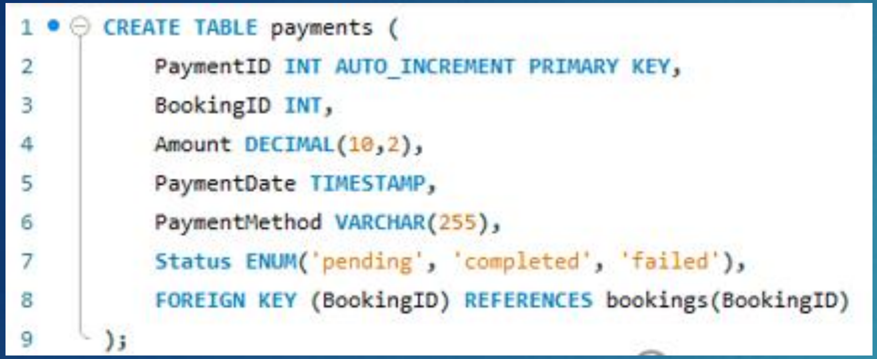
Notifications can be inserted into the table using the INSERT INTO statement.



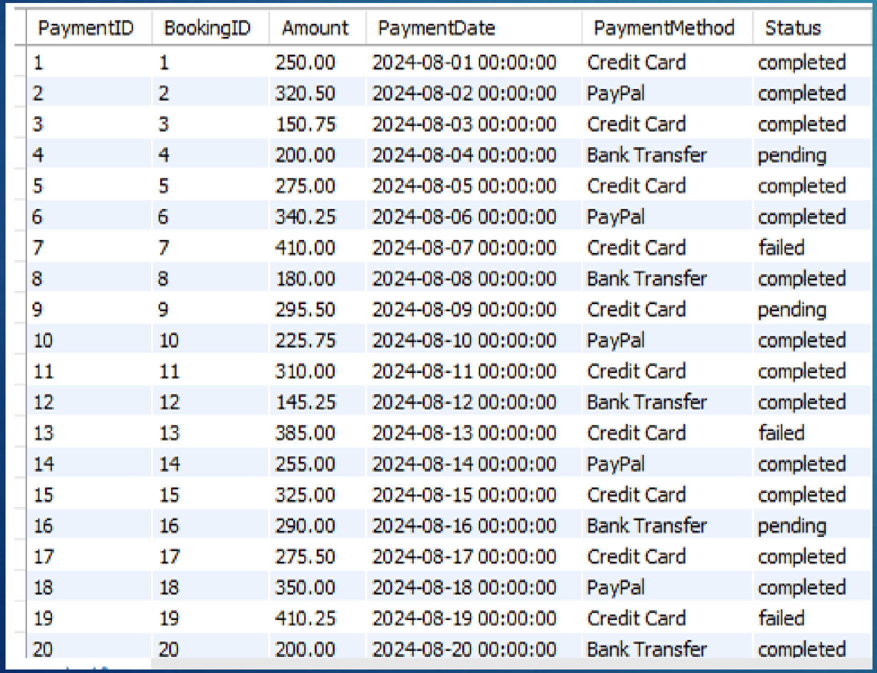
```sql
INSERT INTO myairbnb.notifications (NotificationID, Content, CreatedAt, GuestID, AdminID, HostID) VALUES
(1, 'Your booking request has been approved by the host.', '2024-08-01 10:00:00', 5, NULL, 10),
(2, 'A new message has been received from your guest.', '2024-08-01 11:15:00', 5, NULL, 10),
(3, 'Your property listing has been successfully updated.', '2024-08-02 09:00:00', NULL, NULL, 11),
(4, 'A new booking request has been received for your property.', '2024-08-02 14:30:00', 6, NULL, 12),
(5, 'Your account has been verified successfully.', '2024-08-03 08:45:00', NULL, 10, NULL),
(6, 'You have a new review for your recent stay.', '2024-08-03 10:00:00', NULL, NULL, 13),
(7, 'Your payment has been processed.', '2024-08-04 13:20:00', 7, NULL, 14),
(8, 'A new reservation has been made for your property.', '2024-08-04 16:30:00', 8, NULL, 15),
(9, 'Your property has been marked as a favorite by a guest.', '2024-08-05 09:05:00', 9, NULL, 16),
(10, 'Your request for a late check-out has been approved.', '2024-08-05 11:45:00', 10, NULL, 17),
(11, 'A new message from the admin has been received.', '2024-08-06 12:00:00', 11, 11, NULL),
(12, 'Your reservation has been canceled by the guest.', '2024-08-06 14:15:00', 12, NULL, 18),
(13, 'You have a new review on your recent stay.', '2024-08-07 09:30:00', NULL, NULL, 19),
(14, 'A new inquiry has been made for your property.', '2024-08-07 15:20:00', 13, NULL, 20),
(15, 'Your request for an additional guest has been approved.', '2024-08-08 10:00:00', 14, NULL, 1),
(16, 'Your profile picture has been updated.', '2024-08-08 12:10:00', NULL, 12, NULL),
(17, 'A new message has been received from the admin.', '2024-08-09 14:00:00', 15, 13, NULL),
(18, 'Your property has been approved for the new location.', '2024-08-09 16:20:00', NULL, NULL, 2),
(19, 'Your payment method has been updated successfully.', '2024-08-10 10:15:00', NULL, 14, NULL),
(20, 'A new review has been posted by a guest.', '2024-08-10 13:45:00', NULL, NULL, 3);
```

Payments

Stores information about payments made for bookings. The SQL statement CREATE TABLE is used to define the structure of the of payments table.

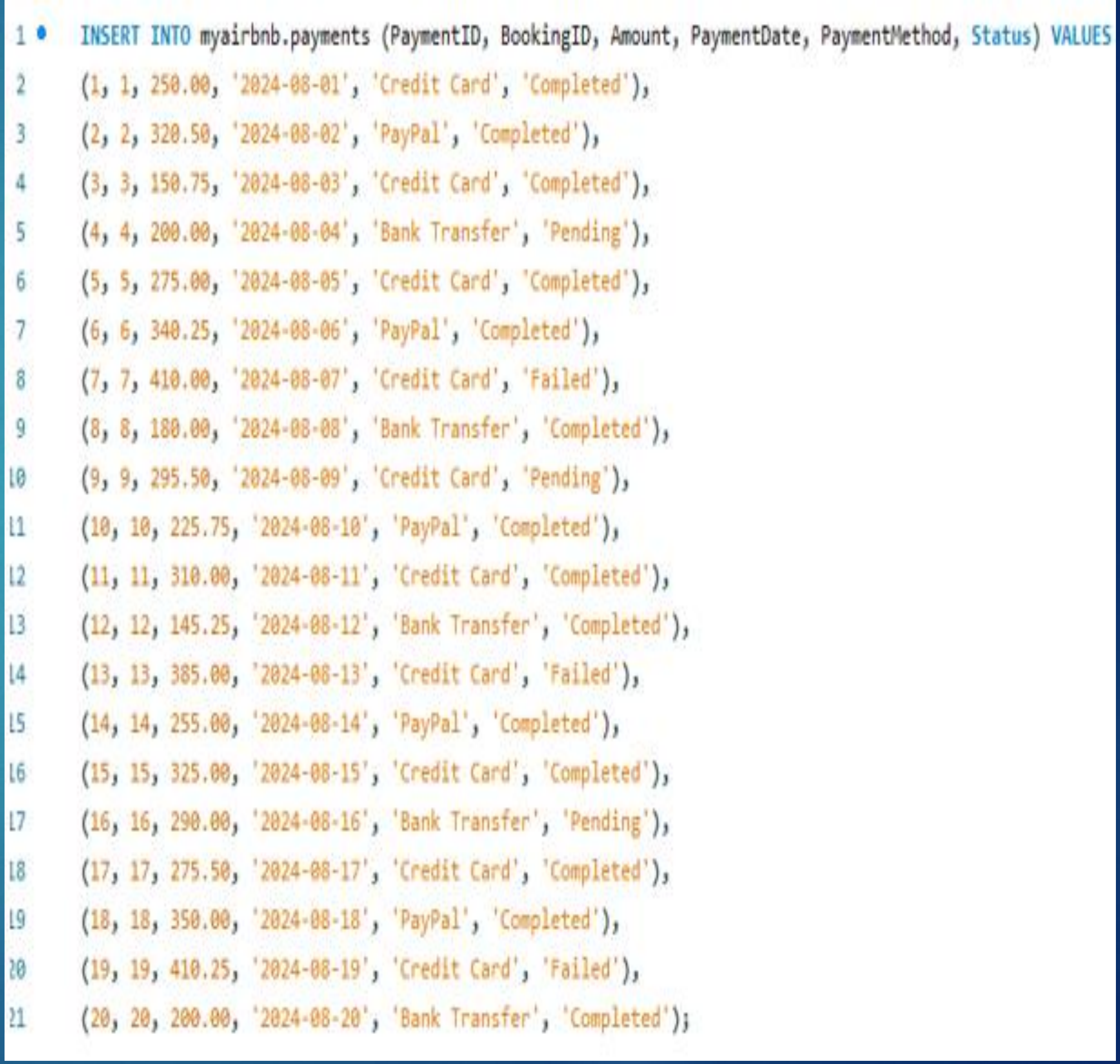


```sql
CREATE TABLE payments (
        PaymentID INT AUTO_INCREMENT PRIMARY KEY,
        BookingID INT,
        Amount DECIMAL(10,2),
        PaymentDate TIMESTAMP,
        PaymentMethod VARCHAR(255),
        Status ENUM('pending', 'completed', 'failed'),
        FOREIGN KEY (BookingID) REFERENCES bookings(BookingID)
);
```



PaymentID	BookingID	Amount	PaymentDate	PaymentMethod	Status
1	1	250.00	2024-08-01 00:00:00	Credit Card	completed
2	2	320.50	2024-08-02 00:00:00	PayPal	completed
3	3	150.75	2024-08-03 00:00:00	Credit Card	completed
4	4	200.00	2024-08-04 00:00:00	Bank Transfer	pending
5	5	275.00	2024-08-05 00:00:00	Credit Card	completed
6	6	340.25	2024-08-06 00:00:00	PayPal	completed
7	7	410.00	2024-08-07 00:00:00	Credit Card	failed
8	8	180.00	2024-08-08 00:00:00	Bank Transfer	completed
9	9	295.50	2024-08-09 00:00:00	Credit Card	pending
10	10	225.75	2024-08-10 00:00:00	PayPal	completed
11	11	310.00	2024-08-11 00:00:00	Credit Card	completed
12	12	145.25	2024-08-12 00:00:00	Bank Transfer	completed
13	13	385.00	2024-08-13 00:00:00	Credit Card	failed
14	14	255.00	2024-08-14 00:00:00	PayPal	completed
15	15	325.00	2024-08-15 00:00:00	Credit Card	completed
16	16	290.00	2024-08-16 00:00:00	Bank Transfer	pending
17	17	275.50	2024-08-17 00:00:00	Credit Card	completed
18	18	350.00	2024-08-18 00:00:00	PayPal	completed
19	19	410.25	2024-08-19 00:00:00	Credit Card	failed
20	20	200.00	2024-08-20 00:00:00	Bank Transfer	completed

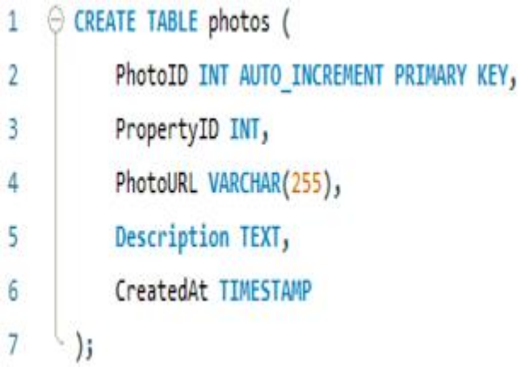
Payment details can be inserted into the table using the INSERT INTO statement.



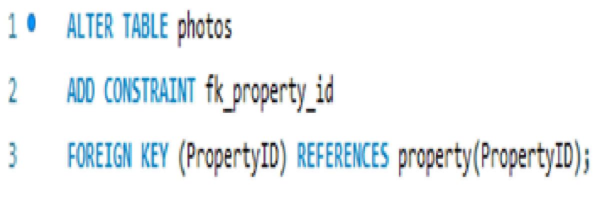
```sql
INSERT INTO myairbnb.payments (PaymentID, BookingID, Amount, PaymentDate, PaymentMethod, Status) VALUES
(1, 1, 250.00, '2024-08-01', 'Credit Card', 'Completed'),
(2, 2, 320.50, '2024-08-02', 'PayPal', 'Completed'),
(3, 3, 150.75, '2024-08-03', 'Credit Card', 'Completed'),
(4, 4, 200.00, '2024-08-04', 'Bank Transfer', 'Pending'),
(5, 5, 275.00, '2024-08-05', 'Credit Card', 'Completed'),
(6, 6, 340.25, '2024-08-06', 'PayPal', 'Completed'),
(7, 7, 410.00, '2024-08-07', 'Credit Card', 'Failed'),
(8, 8, 180.00, '2024-08-08', 'Bank Transfer', 'Completed'),
(9, 9, 295.50, '2024-08-09', 'Credit Card', 'Pending'),
(10, 10, 225.75, '2024-08-10', 'PayPal', 'Completed'),
(11, 11, 310.00, '2024-08-11', 'Credit Card', 'Completed'),
(12, 12, 145.25, '2024-08-12', 'Bank Transfer', 'Completed'),
(13, 13, 385.00, '2024-08-13', 'Credit Card', 'Failed'),
(14, 14, 255.00, '2024-08-14', 'PayPal', 'Completed'),
(15, 15, 325.00, '2024-08-15', 'Credit Card', 'Completed'),
(16, 16, 290.00, '2024-08-16', 'Bank Transfer', 'Pending'),
(17, 17, 275.50, '2024-08-17', 'Credit Card', 'Completed'),
(18, 18, 350.00, '2024-08-18', 'PayPal', 'Completed'),
(19, 19, 410.25, '2024-08-19', 'Credit Card', 'Failed'),
(20, 20, 200.00, '2024-08-20', 'Bank Transfer', 'Completed');
```

Photos

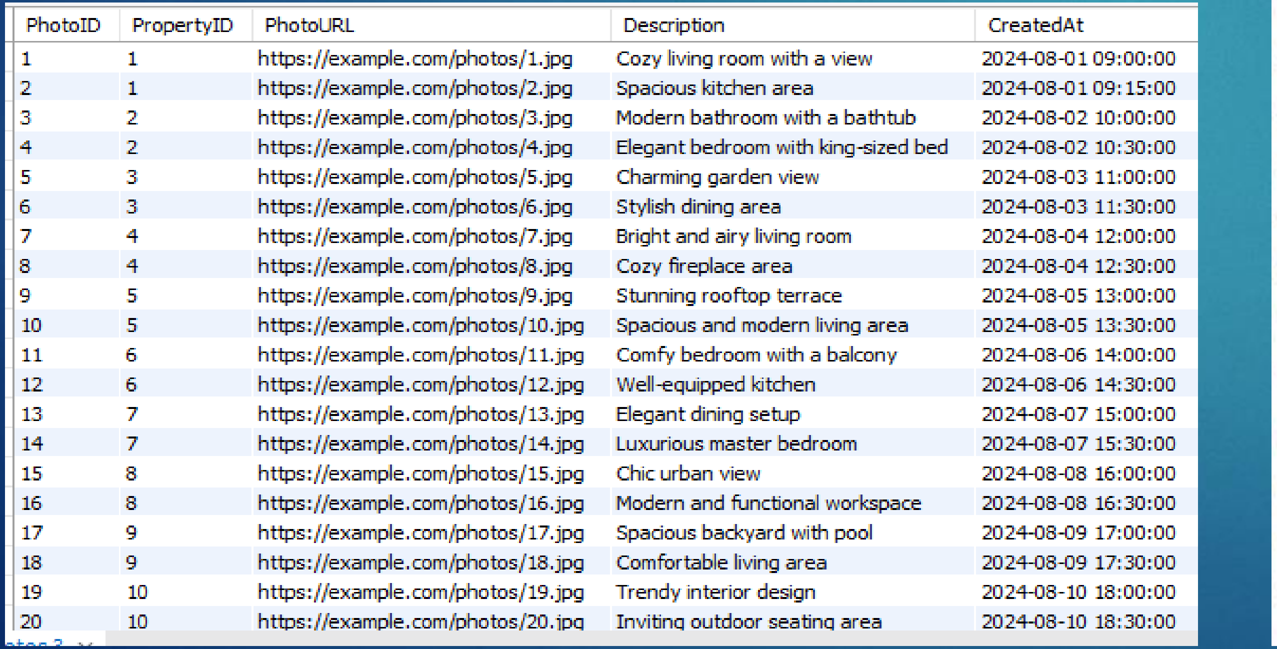
Stores photos associated with properties. The SQL statement CREATE TABLE is used to define the structure of the of photos table. While ALTER TABLE is used to modify the table, in this case adding a foreign key.



```sql
CREATE TABLE photos (
    PhotoID INT AUTO_INCREMENT PRIMARY KEY,
    PropertyID INT,
    PhotoURL VARCHAR(255),
    Description TEXT,
    CreatedAt TIMESTAMP
);
```

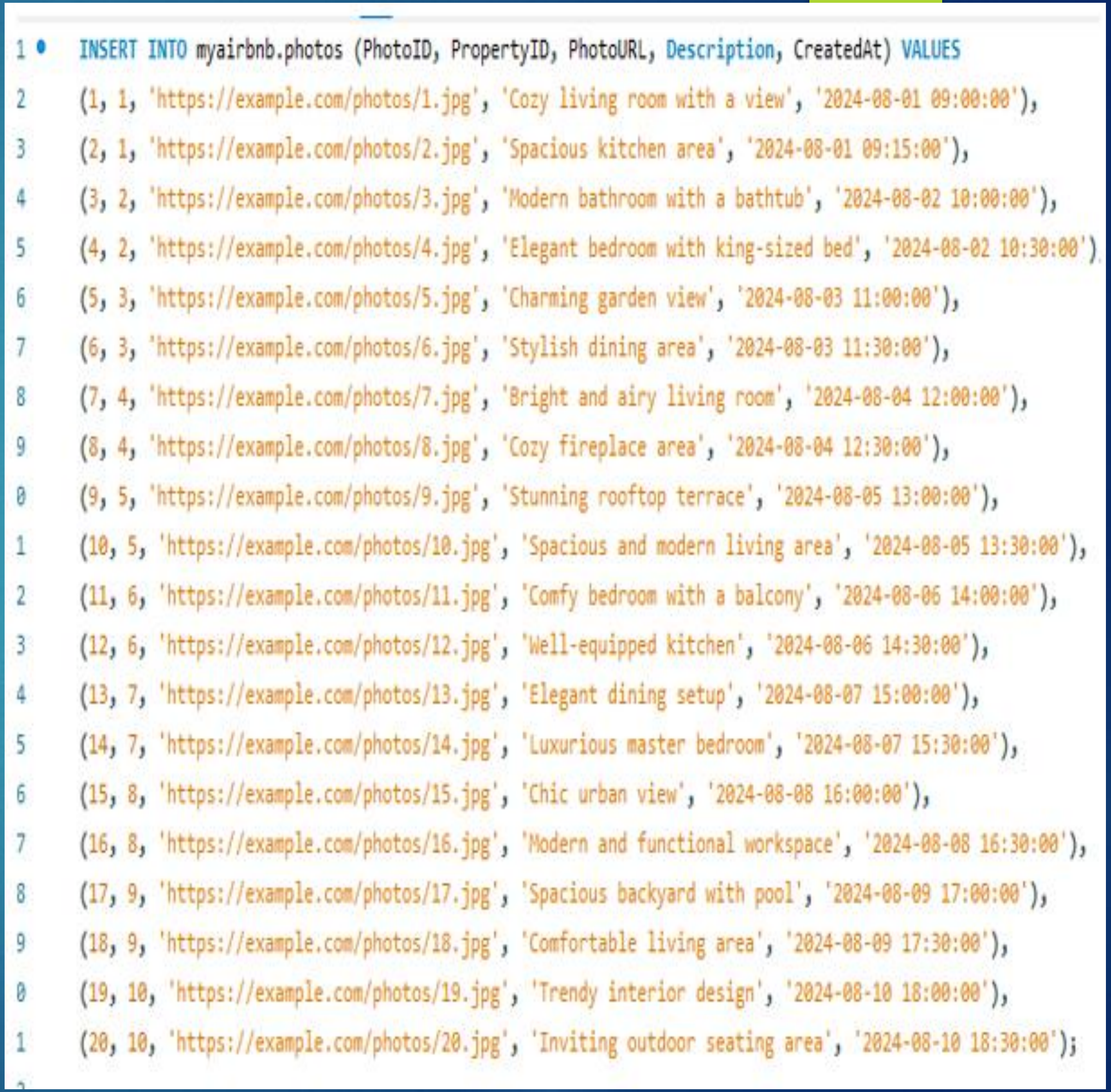


```sql
ALTER TABLE photos
ADD CONSTRAINT fk_property_id
FOREIGN KEY (PropertyID) REFERENCES property(PropertyID);
```



PhotoID	PropertyID	PhotoURL	Description	CreatedAt
1	1	https://example.com/photos/1.jpg	Cozy living room with a view	2024-08-01 09:00:00
2	1	https://example.com/photos/2.jpg	Spacious kitchen area	2024-08-01 09:15:00
3	2	https://example.com/photos/3.jpg	Modern bathroom with a bathtub	2024-08-02 10:00:00
4	2	https://example.com/photos/4.jpg	Elegant bedroom with king-sized bed	2024-08-02 10:30:00
5	3	https://example.com/photos/5.jpg	Charming garden view	2024-08-03 11:00:00
6	3	https://example.com/photos/6.jpg	Stylish dining area	2024-08-03 11:30:00
7	4	https://example.com/photos/7.jpg	Bright and airy living room	2024-08-04 12:00:00
8	4	https://example.com/photos/8.jpg	Cozy fireplace area	2024-08-04 12:30:00
9	5	https://example.com/photos/9.jpg	Stunning rooftop terrace	2024-08-05 13:00:00
10	5	https://example.com/photos/10.jpg	Spacious and modern living area	2024-08-05 13:30:00
11	6	https://example.com/photos/11.jpg	Comfy bedroom with a balcony	2024-08-06 14:00:00
12	6	https://example.com/photos/12.jpg	Well-equipped kitchen	2024-08-06 14:30:00
13	7	https://example.com/photos/13.jpg	Elegant dining setup	2024-08-07 15:00:00
14	7	https://example.com/photos/14.jpg	Luxurious master bedroom	2024-08-07 15:30:00
15	8	https://example.com/photos/15.jpg	Chic urban view	2024-08-08 16:00:00
16	8	https://example.com/photos/16.jpg	Modern and functional workspace	2024-08-08 16:30:00
17	9	https://example.com/photos/17.jpg	Spacious backyard with pool	2024-08-09 17:00:00
18	9	https://example.com/photos/18.jpg	Comfortable living area	2024-08-09 17:30:00
19	10	https://example.com/photos/19.jpg	Trendy interior design	2024-08-10 18:00:00
20	10	https://example.com/photos/20.jpg	Inviting outdoor seating area	2024-08-10 18:30:00

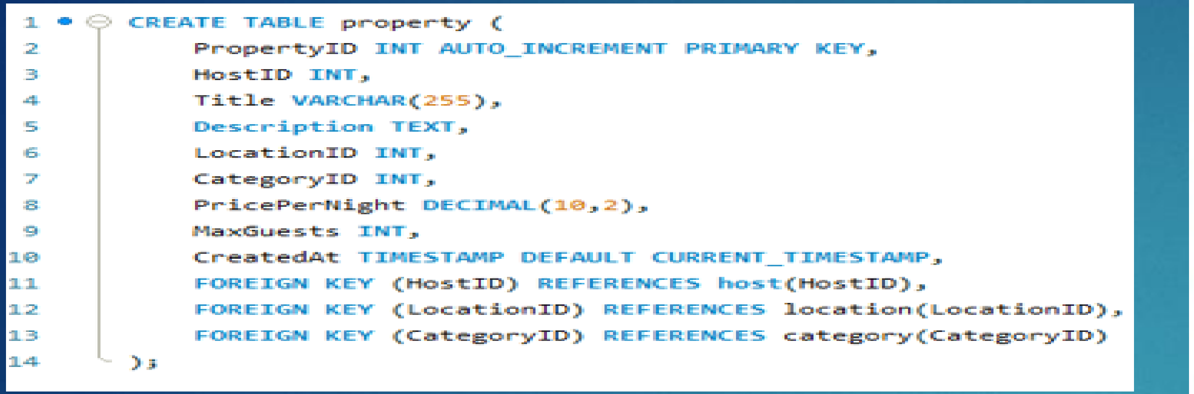
Photo details can be inserted into the table using the INSERT INTO statement.



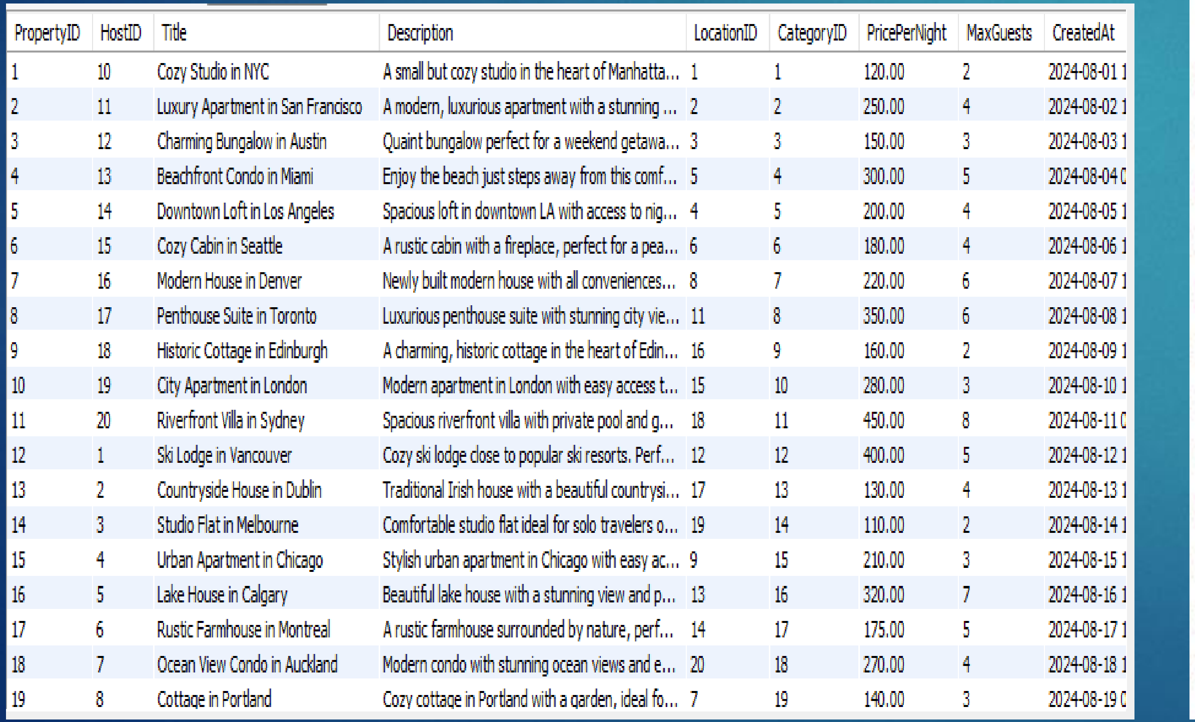
```sql
INSERT INTO myairbnb.photos (PhotoID, PropertyID, PhotoURL, Description, CreatedAt) VALUES
(1, 1, 'https://example.com/photos/1.jpg', 'Cozy living room with a view', '2024-08-01 09:00:00'),
(2, 1, 'https://example.com/photos/2.jpg', 'Spacious kitchen area', '2024-08-01 09:15:00'),
(3, 2, 'https://example.com/photos/3.jpg', 'Modern bathroom with a bathtub', '2024-08-02 10:00:00'),
(4, 2, 'https://example.com/photos/4.jpg', 'Elegant bedroom with king-sized bed', '2024-08-02 10:30:00')
(5, 3, 'https://example.com/photos/5.jpg', 'Charming garden view', '2024-08-03 11:00:00'),
(6, 3, 'https://example.com/photos/6.jpg', 'Stylish dining area', '2024-08-03 11:30:00'),
(7, 4, 'https://example.com/photos/7.jpg', 'Bright and airy living room', '2024-08-04 12:00:00'),
(8, 4, 'https://example.com/photos/8.jpg', 'Cozy fireplace area', '2024-08-04 12:30:00'),
(9, 5, 'https://example.com/photos/9.jpg', 'Stunning rooftop terrace', '2024-08-05 13:00:00'),
(10, 5, 'https://example.com/photos/10.jpg', 'Spacious and modern living area', '2024-08-05 13:30:00'),
(11, 6, 'https://example.com/photos/11.jpg', 'Comfy bedroom with a balcony', '2024-08-06 14:00:00'),
(12, 6, 'https://example.com/photos/12.jpg', 'Well-equipped kitchen', '2024-08-06 14:30:00'),
(13, 7, 'https://example.com/photos/13.jpg', 'Elegant dining setup', '2024-08-07 15:00:00'),
(14, 7, 'https://example.com/photos/14.jpg', 'Luxurious master bedroom', '2024-08-07 15:30:00'),
(15, 8, 'https://example.com/photos/15.jpg', 'Chic urban view', '2024-08-08 16:00:00'),
(16, 8, 'https://example.com/photos/16.jpg', 'Modern and functional workspace', '2024-08-08 16:30:00'),
(17, 9, 'https://example.com/photos/17.jpg', 'Spacious backyard with pool', '2024-08-09 17:00:00'),
(18, 9, 'https://example.com/photos/18.jpg', 'Comfortable living area', '2024-08-09 17:30:00'),
(19, 10, 'https://example.com/photos/19.jpg', 'Trendy interior design', '2024-08-10 18:00:00'),
(20, 10, 'https://example.com/photos/20.jpg', 'Inviting outdoor seating area', '2024-08-10 18:30:00');
```

Property

Stores information about properties listed by hosts. The SQL statement CREATE TABLE is used to define the structure of the of property table.

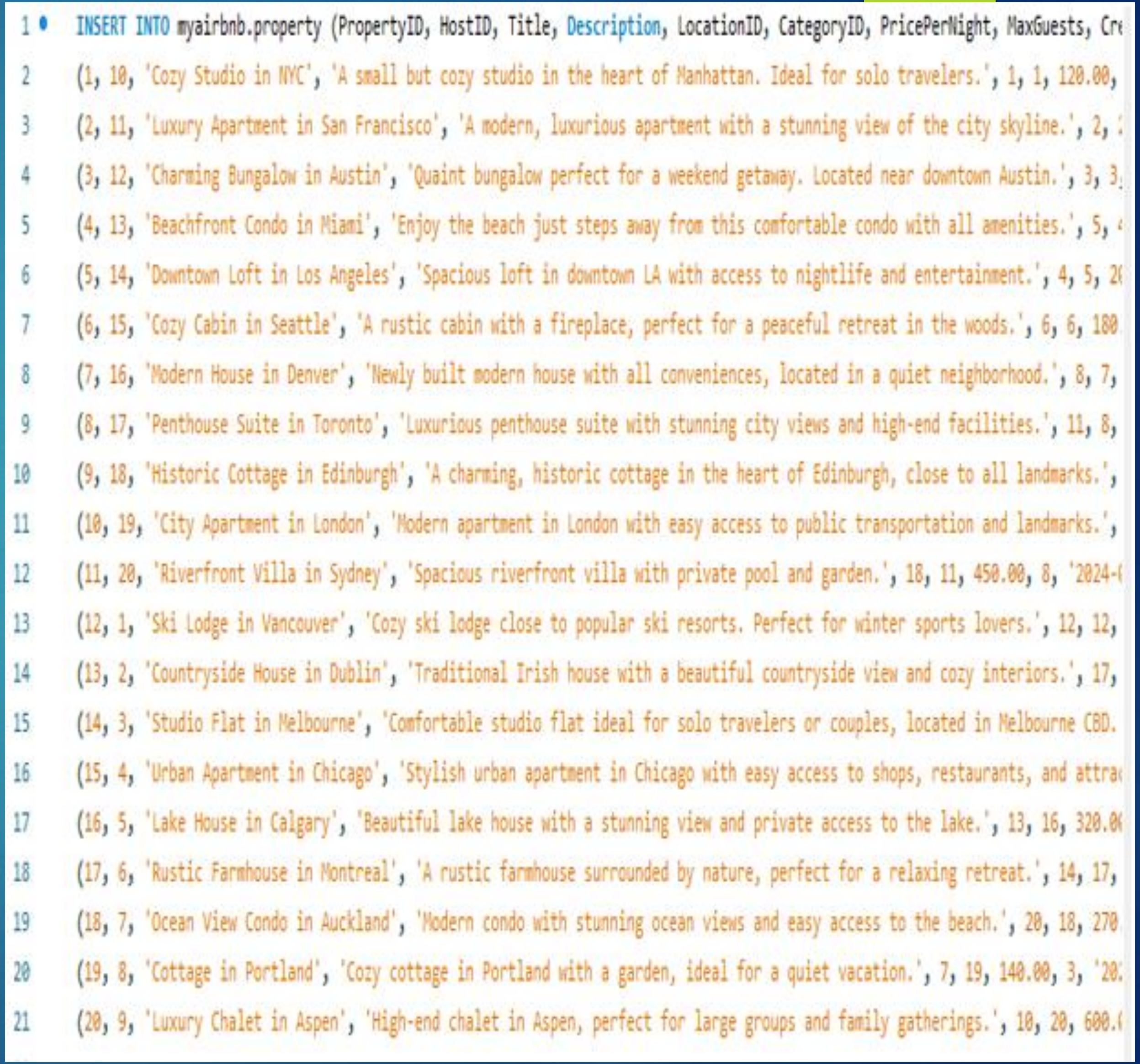


```sql
CREATE TABLE property (
    PropertyID INT AUTO_INCREMENT PRIMARY KEY,
    HostID INT,
    Title VARCHAR(255),
    Description TEXT,
    LocationID INT,
    CategoryID INT,
    PricePerNight DECIMAL(10,2),
    MaxGuests INT,
    CreatedAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (HostID) REFERENCES host(HostID),
    FOREIGN KEY (LocationID) REFERENCES location(LocationID),
    FOREIGN KEY (CategoryID) REFERENCES category(CategoryID)
);
```



PropertyID	HostID	Title	Description	LocationID	CategoryID	PricePerNight	MaxGuests	CreatedAt
1	10	Cozy Studio in NYC	A small but cozy studio in the heart of Manhatta...	1	1	120.00	2	2024-08-01 1
2	11	Luxury Apartment in San Francisco	A modern, luxurious apartment with a stunning ...	2	2	250.00	4	2024-08-02 1
3	12	Charming Bungalow in Austin	Quaint bungalow perfect for a weekend getawa...	3	3	150.00	3	2024-08-03 1
4	13	Beachfront Condo in Miami	Enjoy the beach just steps away from this comf...	5	4	300.00	5	2024-08-04 0
5	14	Downtown Loft in Los Angeles	Spacious loft in downtown LA with access to nig...	4	5	200.00	4	2024-08-05 1
6	15	Cozy Cabin in Seattle	A rustic cabin with a fireplace, perfect for a pea...	6	6	180.00	4	2024-08-06 1
7	16	Modern House in Denver	Newly built modern house with all conveniences...	8	7	220.00	6	2024-08-07 1
8	17	Penthouse Suite in Toronto	Luxurious penthouse suite with stunning city vie...	11	8	350.00	6	2024-08-08 1
9	18	Historic Cottage in Edinburgh	A charming, historic cottage in the heart of Edin...	16	9	160.00	2	2024-08-09 1
10	19	City Apartment in London	Modern apartment in London with easy access t...	15	10	280.00	3	2024-08-10 1
11	20	Riverfront Villa in Sydney	Spacious riverfront villa with private pool and g...	18	11	450.00	8	2024-08-11 0
12	1	Ski Lodge in Vancouver	Cozy ski lodge close to popular ski resorts. Perf...	12	12	400.00	5	2024-08-12 1
13	2	Countryside House in Dublin	Traditional Irish house with a beautiful countrysi...	17	13	130.00	4	2024-08-13 1
14	3	Studio Flat in Melbourne	Comfortable studio flat ideal for solo travelers o...	19	14	110.00	2	2024-08-14 1
15	4	Urban Apartment in Chicago	Stylish urban apartment in Chicago with easy ac...	9	15	210.00	3	2024-08-15 1
16	5	Lake House in Calgary	Beautiful lake house with a stunning view and p...	13	16	320.00	7	2024-08-16 1
17	6	Rustic Farmhouse in Montreal	A rustic farmhouse surrounded by nature, perf...	14	17	175.00	5	2024-08-17 1
18	7	Ocean View Condo in Auckland	Modern condo with stunning ocean views and e...	20	18	270.00	4	2024-08-18 1
19	8	Cottage in Portland	Cozy cottage in Portland with a garden, ideal fo...	7	19	140.00	3	2024-08-19 0

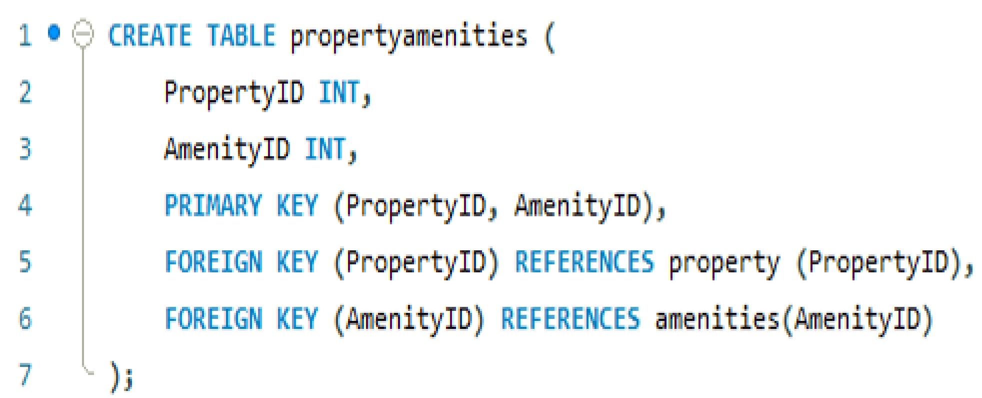
Property (e.g Bungalow) details can be inserted into the table using the INSERT INTO statement.



```sql
INSERT INTO myairbnb.property (PropertyID, HostID, Title, Description, LocationID, CategoryID, PricePerNight, MaxGuests, Cr
(1, 10, 'Cozy Studio in NYC', 'A small but cozy studio in the heart of Manhattan. Ideal for solo travelers.', 1, 1, 120.00,
(2, 11, 'Luxury Apartment in San Francisco', 'A modern, luxurious apartment with a stunning view of the city skyline.', 2,
(3, 12, 'Charming Bungalow in Austin', 'Quaint bungalow perfect for a weekend getaway. Located near downtown Austin.', 3, 3,
(4, 13, 'Beachfront Condo in Miami', 'Enjoy the beach just steps away from this comfortable condo with all amenities.', 5,
(5, 14, 'Downtown Loft in Los Angeles', 'Spacious loft in downtown LA with access to nightlife and entertainment.', 4, 5, 2
(6, 15, 'Cozy Cabin in Seattle', 'A rustic cabin with a fireplace, perfect for a peaceful retreat in the woods.', 6, 6, 180
(7, 16, 'Modern House in Denver', 'Newly built modern house with all conveniences, located in a quiet neighborhood.', 8, 7,
(8, 17, 'Penthouse Suite in Toronto', 'Luxurious penthouse suite with stunning city views and high-end facilities.', 11, 8,
(9, 18, 'Historic Cottage in Edinburgh', 'A charming, historic cottage in the heart of Edinburgh, close to all landmarks.',
(10, 19, 'City Apartment in London', 'Modern apartment in London with easy access to public transportation and landmarks.',
(11, 20, 'Riverfront Villa in Sydney', 'Spacious riverfront villa with private pool and garden.', 18, 11, 450.00, 8, '2024-
(12, 1, 'Ski Lodge in Vancouver', 'Cozy ski lodge close to popular ski resorts. Perfect for winter sports lovers.', 12, 12,
(13, 2, 'Countryside House in Dublin', 'Traditional Irish house with a beautiful countryside view and cozy interiors.', 17,
(14, 3, 'Studio Flat in Melbourne', 'Comfortable studio flat ideal for solo travelers or couples, located in Melbourne CBD.
(15, 4, 'Urban Apartment in Chicago', 'Stylish urban apartment in Chicago with easy access to shops, restaurants, and attra
(16, 5, 'Lake House in Calgary', 'Beautiful lake house with a stunning view and private access to the lake.', 13, 16, 320.0
(17, 6, 'Rustic Farmhouse in Montreal', 'A rustic farmhouse surrounded by nature, perfect for a relaxing retreat.', 14, 17,
(18, 7, 'Ocean View Condo in Auckland', 'Modern condo with stunning ocean views and easy access to the beach.', 20, 18, 270
(19, 8, 'Cottage in Portland', 'Cozy cottage in Portland with a garden, ideal for a quiet vacation.', 7, 19, 140.00, 3, '20
(20, 9, 'Luxury Chalet in Aspen', 'High-end chalet in Aspen, perfect for large groups and family gatherings.', 10, 20, 600.
```

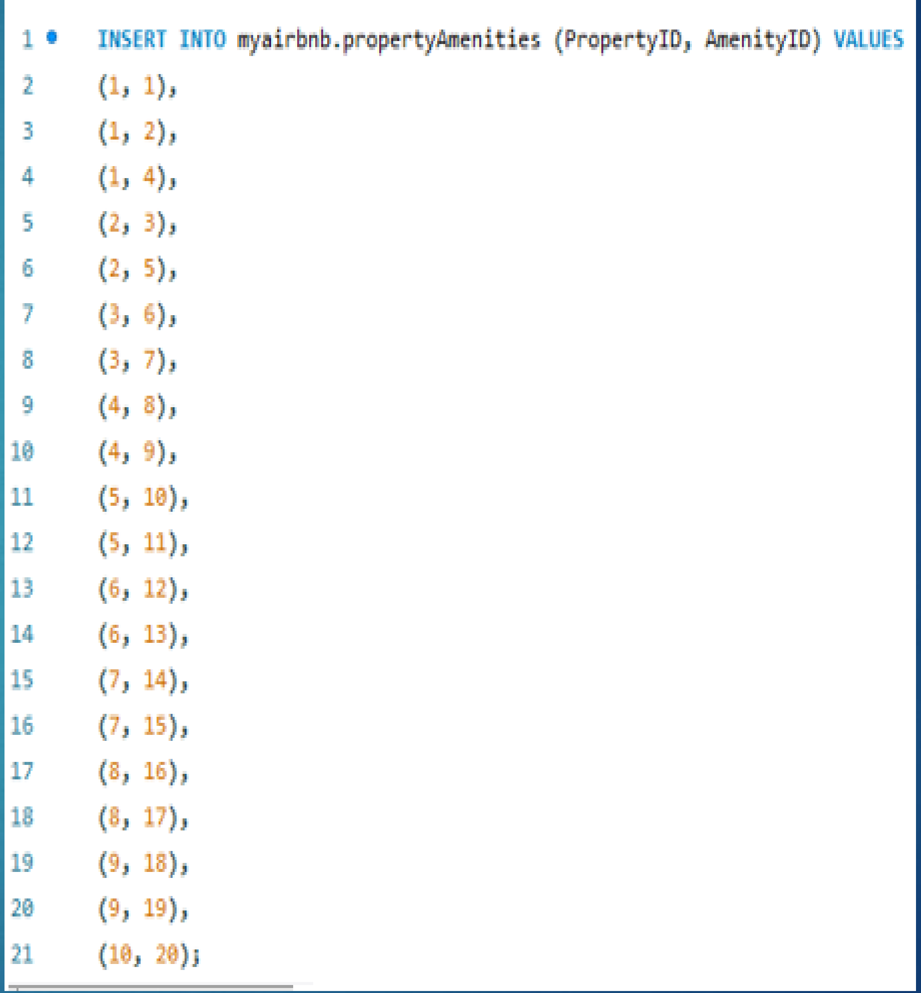
Propertyamenities

Composite table that stores the relationship between properties and their amenities. The SQL statement CREATE TABLE is used to define the structure of the of propertyamenities table.

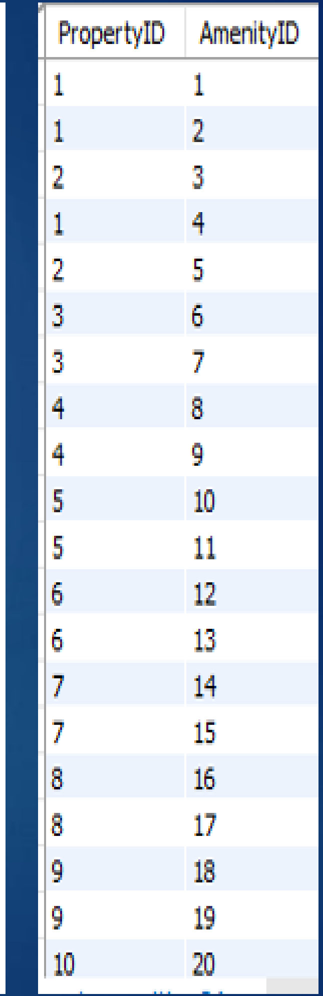


```
CREATE TABLE propertyamenities (
    PropertyID INT,
    AmenityID INT,
    PRIMARY KEY (PropertyID, AmenityID),
    FOREIGN KEY (PropertyID) REFERENCES property (PropertyID),
    FOREIGN KEY (AmenityID) REFERENCES amenities(AmenityID)
);
```

Property amenities details can be inserted into the table using the INSERT INTO statement.



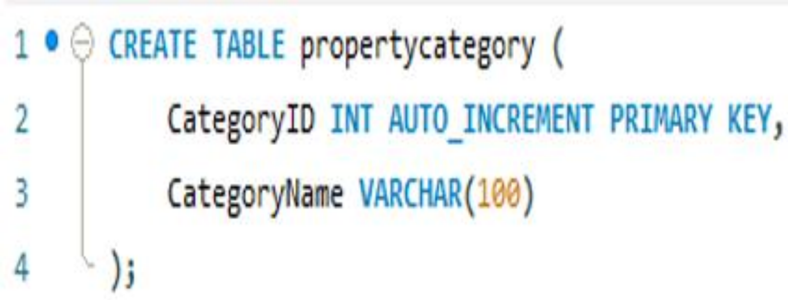
```
INSERT INTO myairbnb.propertyAmenities (PropertyID, AmenityID) VALUES
(1, 1),
(1, 2),
(1, 4),
(2, 3),
(2, 5),
(3, 6),
(3, 7),
(4, 8),
(4, 9),
(5, 10),
(5, 11),
(6, 12),
(6, 13),
(7, 14),
(7, 15),
(8, 16),
(8, 17),
(9, 18),
(9, 19),
(10, 20);
```



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

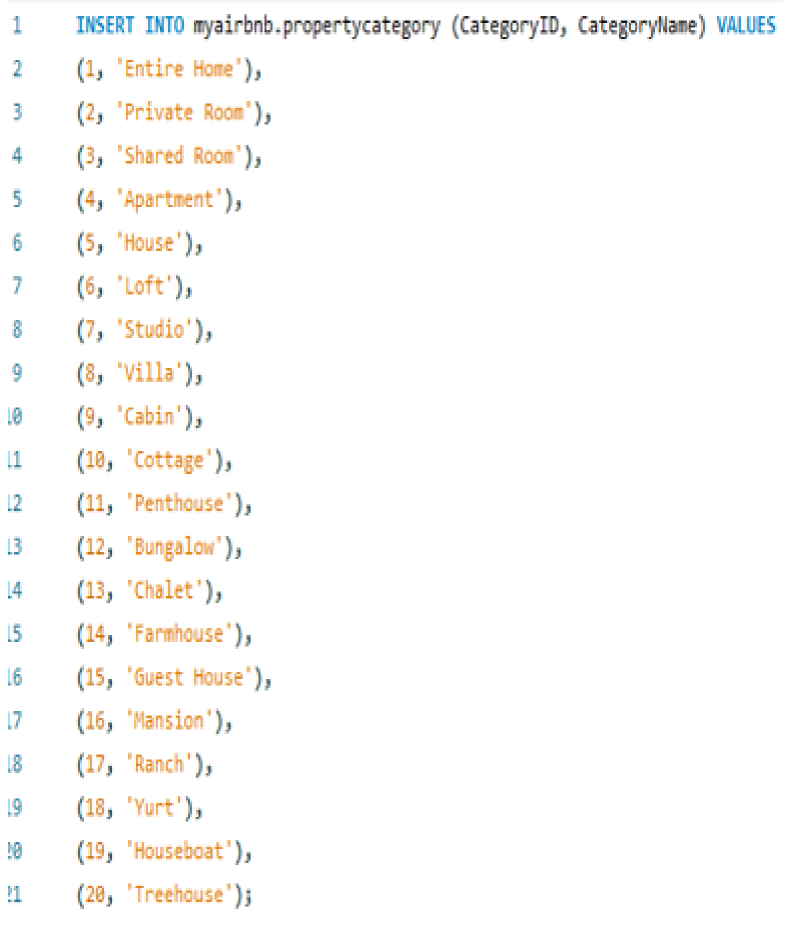
Propertycategory

Stores different categories of properties (e.g., apartment, house). The SQL statement CREATE TABLE is used to define the structure of the of propertcategory table.

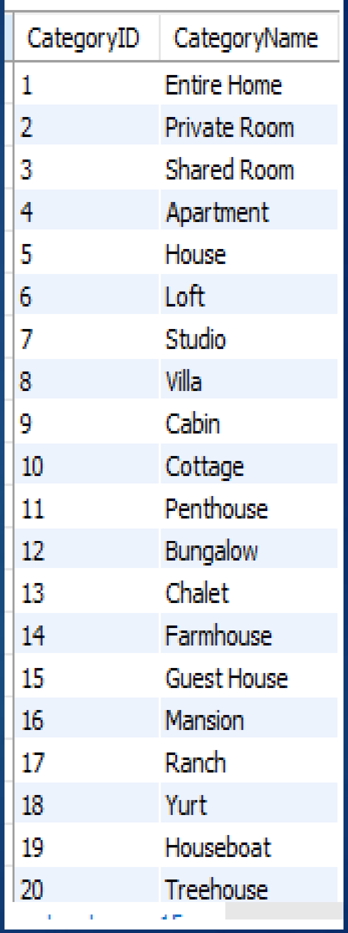


```sql
CREATE TABLE propertycategory (
      CategoryID INT AUTO_INCREMENT PRIMARY KEY,
      CategoryName VARCHAR(100)
);
```

Property category (e.g apartment) can be inserted into the table using the INSERT INTO statement.



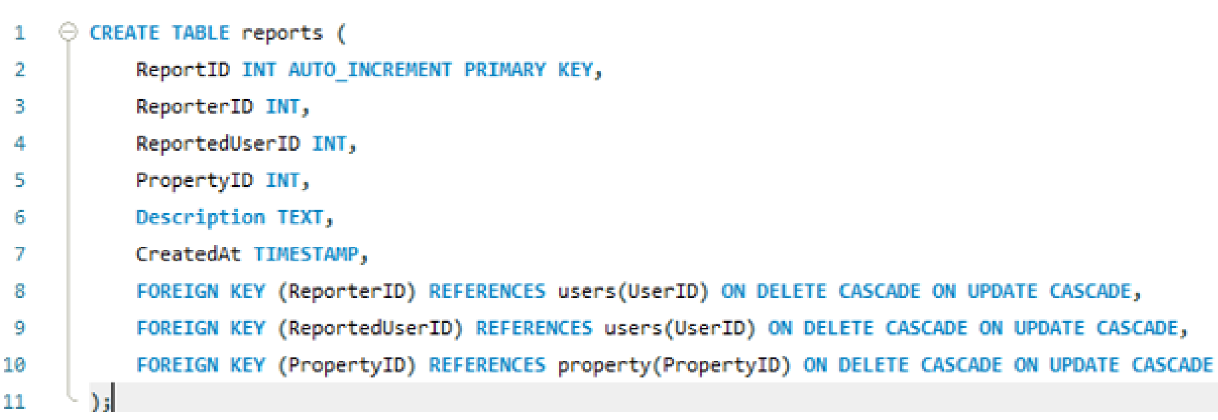
```sql
INSERT INTO myairbnb.propertycategory (CategoryID, CategoryName) VALUES
(1, 'Entire Home'),
(2, 'Private Room'),
(3, 'Shared Room'),
(4, 'Apartment'),
(5, 'House'),
(6, 'Loft'),
(7, 'Studio'),
(8, 'Villa'),
(9, 'Cabin'),
(10, 'Cottage'),
(11, 'Penthouse'),
(12, 'Bungalow'),
(13, 'Chalet'),
(14, 'Farmhouse'),
(15, 'Guest House'),
(16, 'Mansion'),
(17, 'Ranch'),
(18, 'Yurt'),
(19, 'Houseboat'),
(20, 'Treehouse');
```



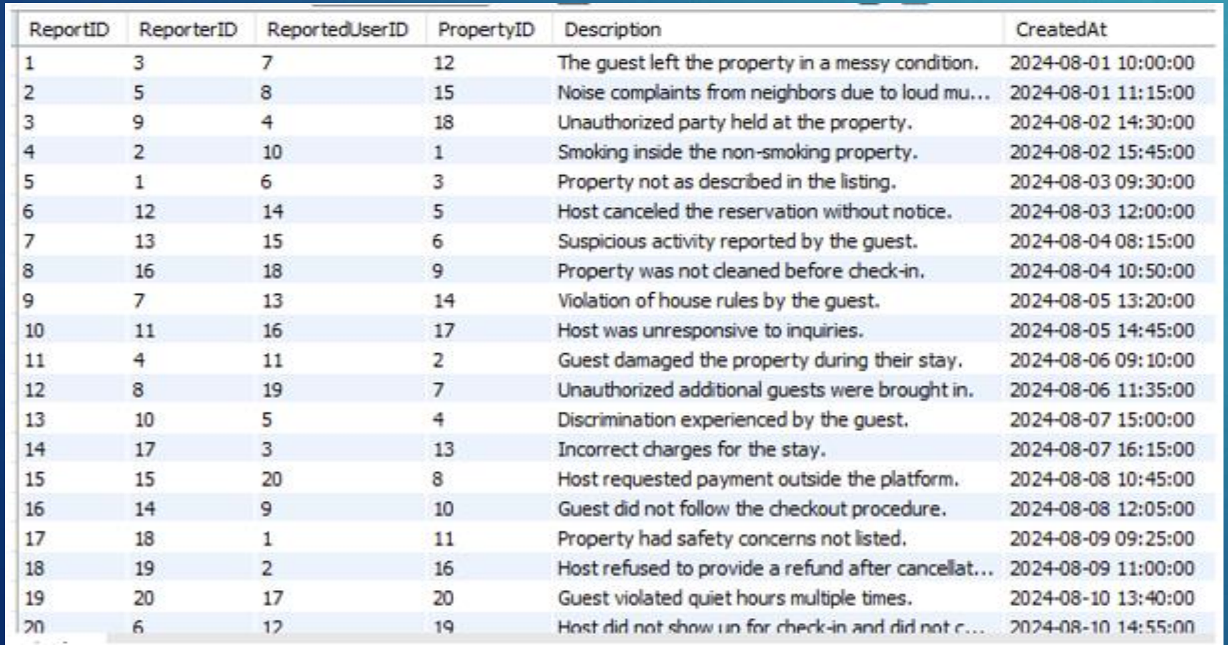
CategoryID	CategoryName
1	Entire Home
2	Private Room
3	Shared Room
4	Apartment
5	House
6	Loft
7	Studio
8	Villa
9	Cabin
10	Cottage
11	Penthouse
12	Bungalow
13	Chalet
14	Farmhouse
15	Guest House
16	Mansion
17	Ranch
18	Yurt
19	Houseboat
20	Treehouse

Reports

Stores reports of inappropriate behaviour or issues related to users or properties. The SQL statement CREATE TABLE is used to define the structure of the of reports table.

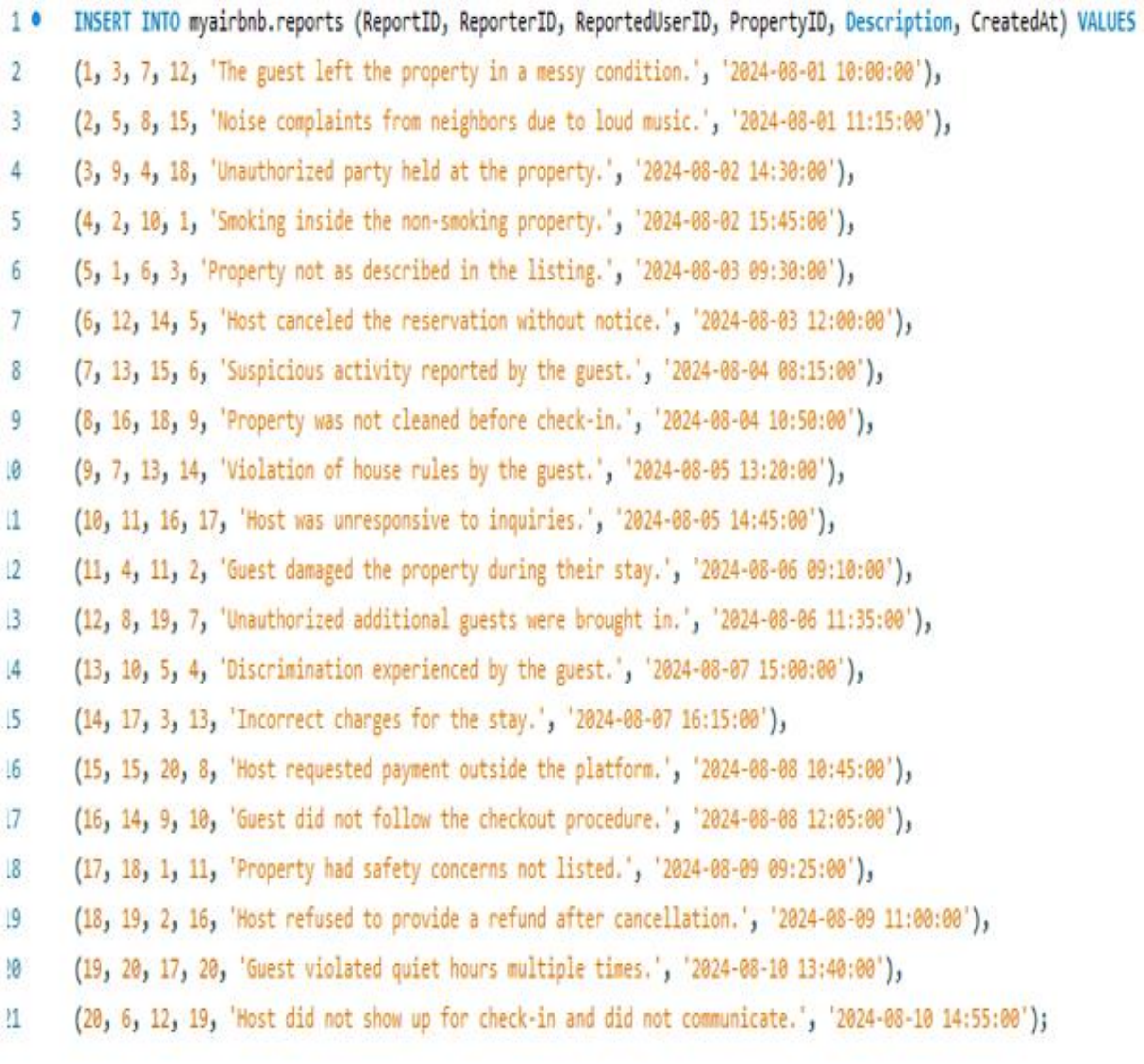


```sql
CREATE TABLE reports (
    ReportID INT AUTO_INCREMENT PRIMARY KEY,
    ReporterID INT,
    ReportedUserID INT,
    PropertyID INT,
    Description TEXT,
    CreatedAt TIMESTAMP,
    FOREIGN KEY (ReporterID) REFERENCES users(UserID) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (ReportedUserID) REFERENCES users(UserID) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (PropertyID) REFERENCES property(PropertyID) ON DELETE CASCADE ON UPDATE CASCADE
);
```



ReportID	ReporterID	ReportedUserID	PropertyID	Description	CreatedAt
1	3	7	12	The guest left the property in a messy condition.	2024-08-01 10:00:00
2	5	8	15	Noise complaints from neighbors due to loud mu...	2024-08-01 11:15:00
3	9	4	18	Unauthorized party held at the property.	2024-08-02 14:30:00
4	2	10	1	Smoking inside the non-smoking property.	2024-08-02 15:45:00
5	1	6	3	Property not as described in the listing.	2024-08-03 09:30:00
6	12	14	5	Host canceled the reservation without notice.	2024-08-03 12:00:00
7	13	15	6	Suspicious activity reported by the guest.	2024-08-04 08:15:00
8	16	18	9	Property was not cleaned before check-in.	2024-08-04 10:50:00
9	7	13	14	Violation of house rules by the guest.	2024-08-05 13:20:00
10	11	16	17	Host was unresponsive to inquiries.	2024-08-05 14:45:00
11	4	11	2	Guest damaged the property during their stay.	2024-08-06 09:10:00
12	8	19	7	Unauthorized additional guests were brought in.	2024-08-06 11:35:00
13	10	5	4	Discrimination experienced by the guest.	2024-08-07 15:00:00
14	17	3	13	Incorrect charges for the stay.	2024-08-07 16:15:00
15	15	20	8	Host requested payment outside the platform.	2024-08-08 10:45:00
16	14	9	10	Guest did not follow the checkout procedure.	2024-08-08 12:05:00
17	18	1	11	Property had safety concerns not listed.	2024-08-09 09:25:00
18	19	2	16	Host refused to provide a refund after cancellat...	2024-08-09 11:00:00
19	20	17	20	Guest violated quiet hours multiple times.	2024-08-10 13:40:00
20	6	12	19	Host did not show up for check-in and did not c...	2024-08-10 14:55:00

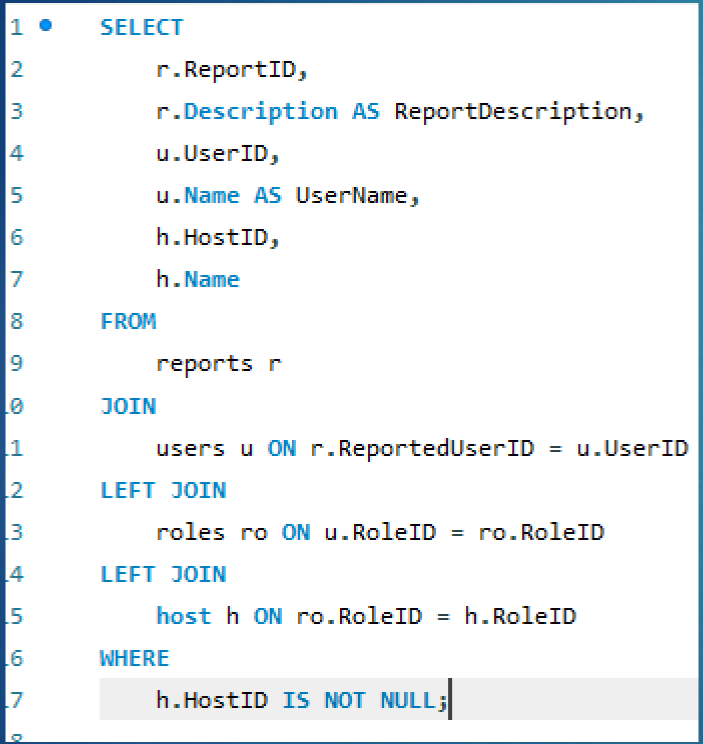
Reports can be inserted into the table using the INSERT INTO statement.



```sql
INSERT INTO myairbnb.reports (ReportID, ReporterID, ReportedUserID, PropertyID, Description, CreatedAt) VALUES
(1, 3, 7, 12, 'The guest left the property in a messy condition.', '2024-08-01 10:00:00'),
(2, 5, 8, 15, 'Noise complaints from neighbors due to loud music.', '2024-08-01 11:15:00'),
(3, 9, 4, 18, 'Unauthorized party held at the property.', '2024-08-02 14:30:00'),
(4, 2, 10, 1, 'Smoking inside the non-smoking property.', '2024-08-02 15:45:00'),
(5, 1, 6, 3, 'Property not as described in the listing.', '2024-08-03 09:30:00'),
(6, 12, 14, 5, 'Host canceled the reservation without notice.', '2024-08-03 12:00:00'),
(7, 13, 15, 6, 'Suspicious activity reported by the guest.', '2024-08-04 08:15:00'),
(8, 16, 18, 9, 'Property was not cleaned before check-in.', '2024-08-04 10:50:00'),
(9, 7, 13, 14, 'Violation of house rules by the guest.', '2024-08-05 13:20:00'),
(10, 11, 16, 17, 'Host was unresponsive to inquiries.', '2024-08-05 14:45:00'),
(11, 4, 11, 2, 'Guest damaged the property during their stay.', '2024-08-06 09:10:00'),
(12, 8, 19, 7, 'Unauthorized additional guests were brought in.', '2024-08-06 11:35:00'),
(13, 10, 5, 4, 'Discrimination experienced by the guest.', '2024-08-07 15:00:00'),
(14, 17, 3, 13, 'Incorrect charges for the stay.', '2024-08-07 16:15:00'),
(15, 15, 20, 8, 'Host requested payment outside the platform.', '2024-08-08 10:45:00'),
(16, 14, 9, 10, 'Guest did not follow the checkout procedure.', '2024-08-08 12:05:00'),
(17, 18, 1, 11, 'Property had safety concerns not listed.', '2024-08-09 09:25:00'),
(18, 19, 2, 16, 'Host refused to provide a refund after cancellation.', '2024-08-09 11:00:00'),
(19, 20, 17, 20, 'Guest violated quiet hours multiple times.', '2024-08-10 13:40:00'),
(20, 6, 12, 19, 'Host did not show up for check-in and did not communicate.', '2024-08-10 14:55:00');
```

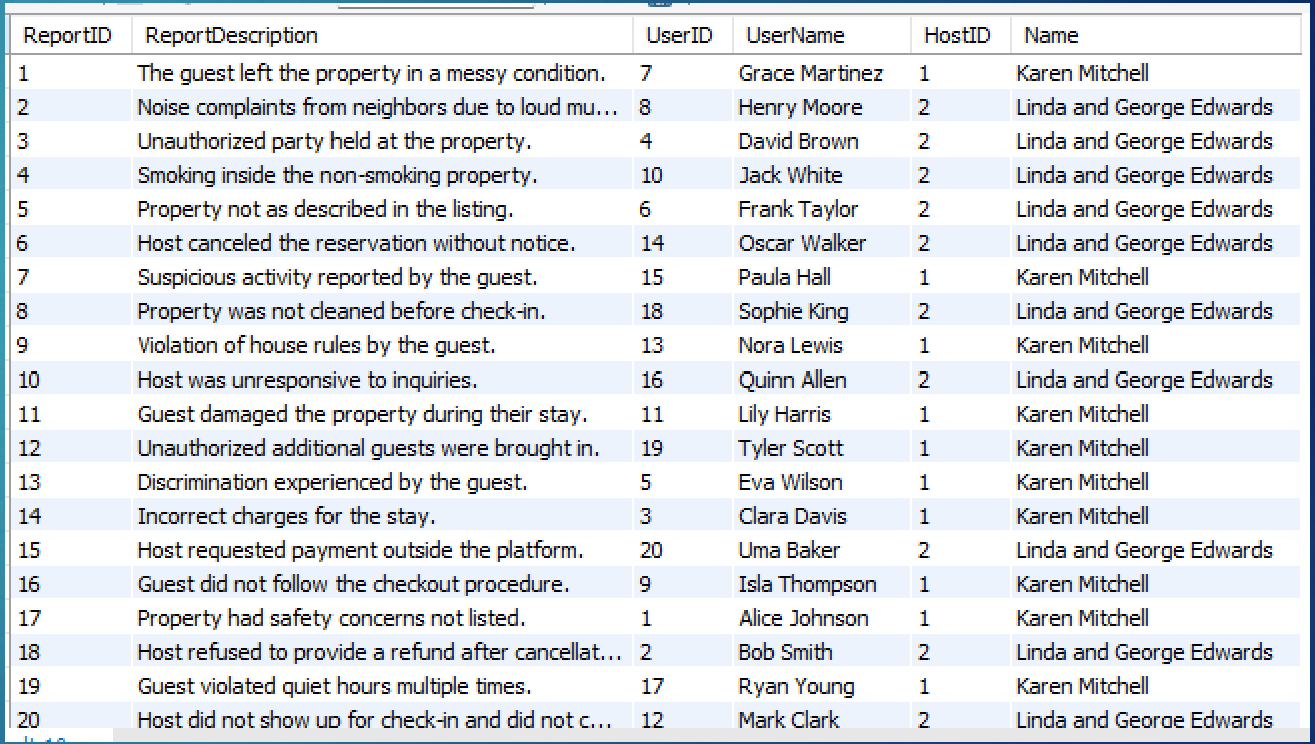
Reports Testcase: Showing reports made by several hosts

This query retrieves reports data by joining four tables: reports, users, roles and hosts



```sql
SELECT
    r.ReportID,
    r.Description AS ReportDescription,
    u.UserID,
    u.Name AS UserName,
    h.HostID,
    h.Name
FROM
    reports r
JOIN
    users u ON r.ReportedUserID = u.UserID
LEFT JOIN
    roles ro ON u.RoleID = ro.RoleID
LEFT JOIN
    host h ON ro.RoleID = h.RoleID
WHERE
    h.HostID IS NOT NULL;
```

Result



ReportID	ReportDescription	UserID	UserName	HostID	Name
1	The guest left the property in a messy condition.	7	Grace Martinez	1	Karen Mitchell
2	Noise complaints from neighbors due to loud mu...	8	Henry Moore	2	Linda and George Edwards
3	Unauthorized party held at the property.	4	David Brown	2	Linda and George Edwards
4	Smoking inside the non-smoking property.	10	Jack White	2	Linda and George Edwards
5	Property not as described in the listing.	6	Frank Taylor	2	Linda and George Edwards
6	Host canceled the reservation without notice.	14	Oscar Walker	2	Linda and George Edwards
7	Suspicious activity reported by the guest.	15	Paula Hall	1	Karen Mitchell
8	Property was not cleaned before check-in.	18	Sophie King	2	Linda and George Edwards
9	Violation of house rules by the guest.	13	Nora Lewis	1	Karen Mitchell
10	Host was unresponsive to inquiries.	16	Quinn Allen	2	Linda and George Edwards
11	Guest damaged the property during their stay.	11	Lily Harris	1	Karen Mitchell
12	Unauthorized additional guests were brought in.	19	Tyler Scott	1	Karen Mitchell
13	Discrimination experienced by the guest.	5	Eva Wilson	1	Karen Mitchell
14	Incorrect charges for the stay.	3	Clara Davis	1	Karen Mitchell
15	Host requested payment outside the platform.	20	Uma Baker	2	Linda and George Edwards
16	Guest did not follow the checkout procedure.	9	Isla Thompson	1	Karen Mitchell
17	Property had safety concerns not listed.	1	Alice Johnson	1	Karen Mitchell
18	Host refused to provide a refund after cancellat...	2	Bob Smith	2	Linda and George Edwards
19	Guest violated quiet hours multiple times.	17	Ryan Young	1	Karen Mitchell
20	Host did not show up for check-in and did not c...	12	Mark Clark	2	Linda and George Edwards

SELECT Clause: r.ReportID selects the ReportID from the reports table. This column uniquely identifies each report.
r.Description AS ReportDescription selects the Description from the reports table and renames it to ReportDescription in the result set.
u.UserID selects the UserID from the users table, which identifies the user associated with the report.
u.Name AS UserName selects the Name from the users table and renames it to UserName in the result set.
h.HostID selects the HostID from the host table. This is included to show host details if applicable.
h.Name selects the Name from the host table, which provides the name of the host associated with the report.

FROM Clause: reports r specifies the reports table as the primary table in the query, and assigns it an alias r. This table contains the report information.

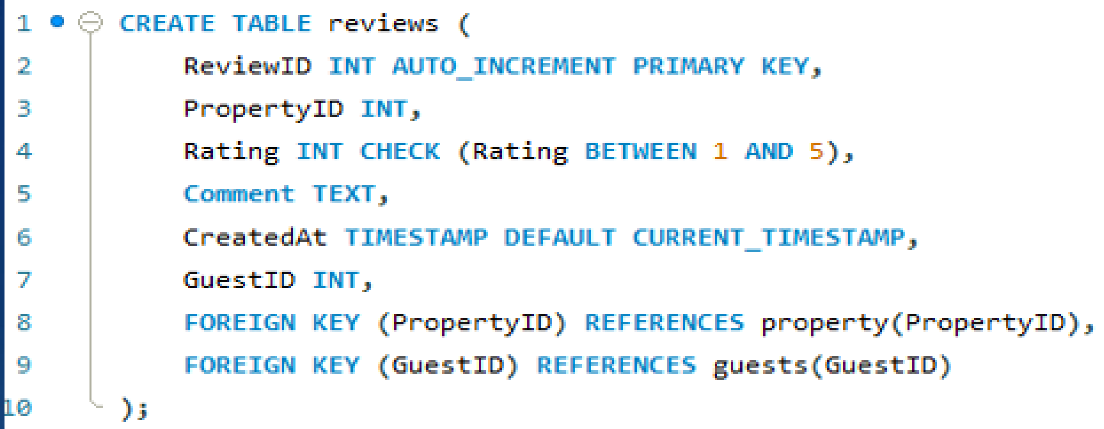
JOIN Clause: Joins users u ON r.ReportedUserID = u.UserID performs an inner join between the reports table and the users table on the ReportedUserID column from the reports table and the UserID column from the users table. This join ensures that only reports with a corresponding user are included in the result. It pulls user details for each report.

LEFT JOIN Clause: Left join roles ro ON u.RoleID = ro.RoleID performs a left join between the users table and the roles table on the RoleID column. This join includes all users from the users table and adds role information if available. If a user has no associated role, the resulting columns from roles will be NULL. While the **Left join host h ON ro.RoleID = h.RoleID** performs a left join between the roles table and the host table on the RoleID column. This join includes all roles and adds host details if the role corresponds to a host. If the role does not match a host, the host columns will be NULL.

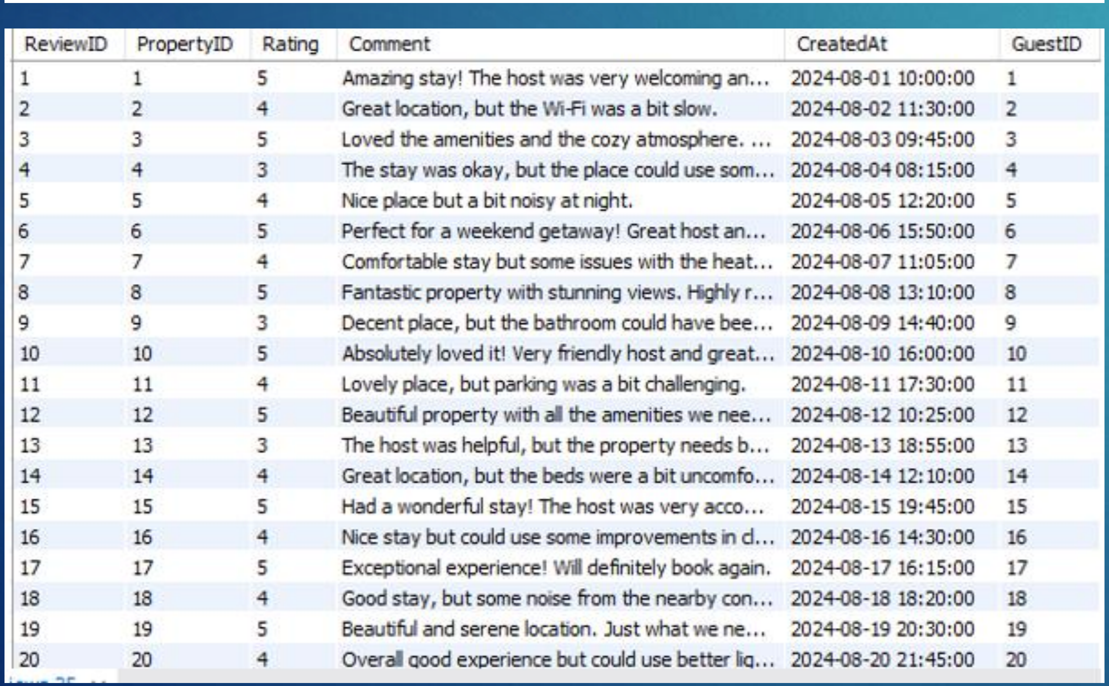
WHERE Clause: h.HostID IS NOT NULL filters the result set to include only rows where the HostID from the host table is not NULL.

Reviews

Stores reviews given by guests for properties they have stayed in. The SQL statement CREATE TABLE is used to define the structure of the of reviews table.

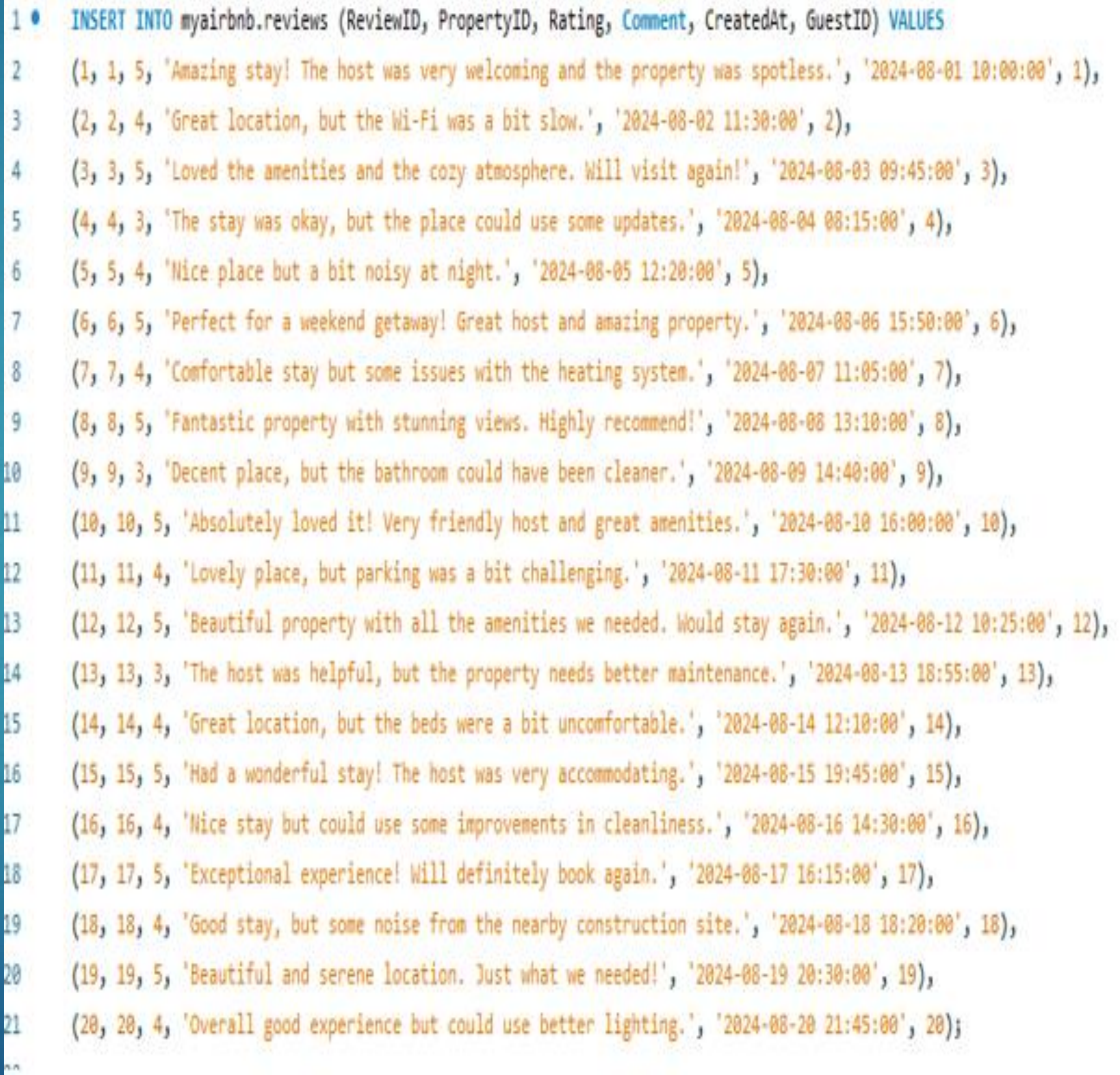


```sql
CREATE TABLE reviews (
    ReviewID INT AUTO_INCREMENT PRIMARY KEY,
    PropertyID INT,
    Rating INT CHECK (Rating BETWEEN 1 AND 5),
    Comment TEXT,
    CreatedAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    GuestID INT,
    FOREIGN KEY (PropertyID) REFERENCES property(PropertyID),
    FOREIGN KEY (GuestID) REFERENCES guests(GuestID)
);
```



ReviewID	PropertyID	Rating	Comment	CreatedAt	GuestID
1	1	5	Amazing stay! The host was very welcoming an...	2024-08-01 10:00:00	1
2	2	4	Great location, but the Wi-Fi was a bit slow.	2024-08-02 11:30:00	2
3	3	5	Loved the amenities and the cozy atmosphere. ...	2024-08-03 09:45:00	3
4	4	3	The stay was okay, but the place could use som...	2024-08-04 08:15:00	4
5	5	4	Nice place but a bit noisy at night.	2024-08-05 12:20:00	5
6	6	5	Perfect for a weekend getaway! Great host an...	2024-08-06 15:50:00	6
7	7	4	Comfortable stay but some issues with the heat...	2024-08-07 11:05:00	7
8	8	5	Fantastic property with stunning views. Highly r...	2024-08-08 13:10:00	8
9	9	3	Decent place, but the bathroom could have bee...	2024-08-09 14:40:00	9
10	10	5	Absolutely loved it! Very friendly host and great...	2024-08-10 16:00:00	10
11	11	4	Lovely place, but parking was a bit challenging.	2024-08-11 17:30:00	11
12	12	5	Beautiful property with all the amenities we nee...	2024-08-12 10:25:00	12
13	13	3	The host was helpful, but the property needs b...	2024-08-13 18:55:00	13
14	14	4	Great location, but the beds were a bit uncomfo...	2024-08-14 12:10:00	14
15	15	5	Had a wonderful stay! The host was very acco...	2024-08-15 19:45:00	15
16	16	4	Nice stay but could use some improvements in cl...	2024-08-16 14:30:00	16
17	17	5	Exceptional experience! Will definitely book again.	2024-08-17 16:15:00	17
18	18	4	Good stay, but some noise from the nearby con...	2024-08-18 18:20:00	18
19	19	5	Beautiful and serene location. Just what we ne...	2024-08-19 20:30:00	19
20	20	4	Overall good experience but could use better lig...	2024-08-20 21:45:00	20

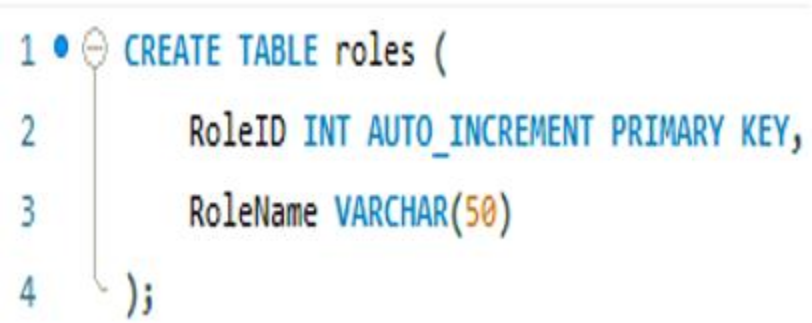
Reviews can be inserted into the table using the INSERT INTO statement.



```sql
INSERT INTO myairbnb.reviews (ReviewID, PropertyID, Rating, Comment, CreatedAt, GuestID) VALUES
(1, 1, 5, 'Amazing stay! The host was very welcoming and the property was spotless.', '2024-08-01 10:00:00', 1),
(2, 2, 4, 'Great location, but the Wi-Fi was a bit slow.', '2024-08-02 11:30:00', 2),
(3, 3, 5, 'Loved the amenities and the cozy atmosphere. Will visit again!', '2024-08-03 09:45:00', 3),
(4, 4, 3, 'The stay was okay, but the place could use some updates.', '2024-08-04 08:15:00', 4),
(5, 5, 4, 'Nice place but a bit noisy at night.', '2024-08-05 12:20:00', 5),
(6, 6, 5, 'Perfect for a weekend getaway! Great host and amazing property.', '2024-08-06 15:50:00', 6),
(7, 7, 4, 'Comfortable stay but some issues with the heating system.', '2024-08-07 11:05:00', 7),
(8, 8, 5, 'Fantastic property with stunning views. Highly recommend!', '2024-08-08 13:10:00', 8),
(9, 9, 3, 'Decent place, but the bathroom could have been cleaner.', '2024-08-09 14:40:00', 9),
(10, 10, 5, 'Absolutely loved it! Very friendly host and great amenities.', '2024-08-10 16:00:00', 10),
(11, 11, 4, 'Lovely place, but parking was a bit challenging.', '2024-08-11 17:30:00', 11),
(12, 12, 5, 'Beautiful property with all the amenities we needed. Would stay again.', '2024-08-12 10:25:00', 12),
(13, 13, 3, 'The host was helpful, but the property needs better maintenance.', '2024-08-13 18:55:00', 13),
(14, 14, 4, 'Great location, but the beds were a bit uncomfortable.', '2024-08-14 12:10:00', 14),
(15, 15, 5, 'Had a wonderful stay! The host was very accommodating.', '2024-08-15 19:45:00', 15),
(16, 16, 4, 'Nice stay but could use some improvements in cleanliness.', '2024-08-16 14:30:00', 16),
(17, 17, 5, 'Exceptional experience! Will definitely book again.', '2024-08-17 16:15:00', 17),
(18, 18, 4, 'Good stay, but some noise from the nearby construction site.', '2024-08-18 18:20:00', 18),
(19, 19, 5, 'Beautiful and serene location. Just what we needed!', '2024-08-19 20:30:00', 19),
(20, 20, 4, 'Overall good experience but could use better lighting.', '2024-08-20 21:45:00', 20);
```

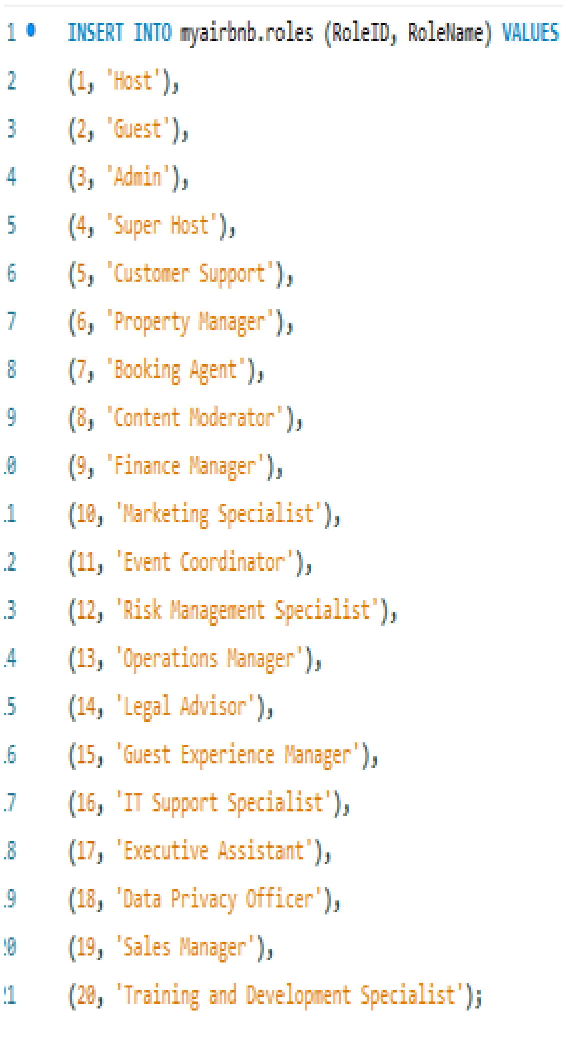
Roles

Stores different roles that users can have (e.g., guest, host, admin). The SQL statement CREATE TABLE is used to define the structure of the of roles table.

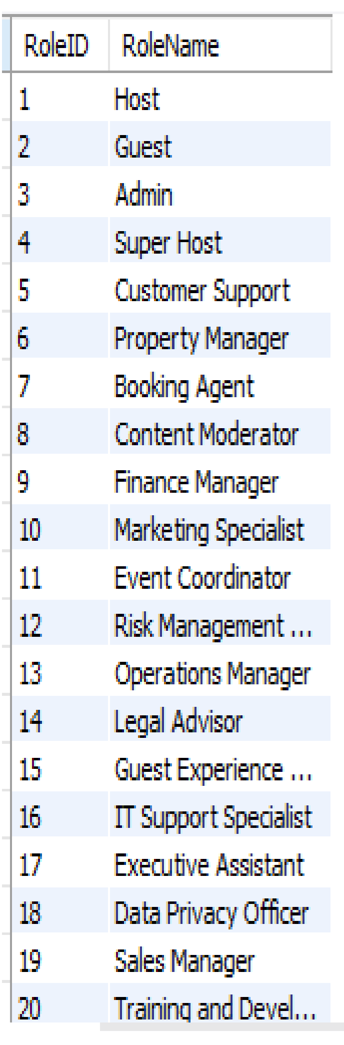


```sql
CREATE TABLE roles (
    RoleID INT AUTO_INCREMENT PRIMARY KEY,
    RoleName VARCHAR(50)
);
```

Roles can be inserted into the table using the INSERT INTO statement.



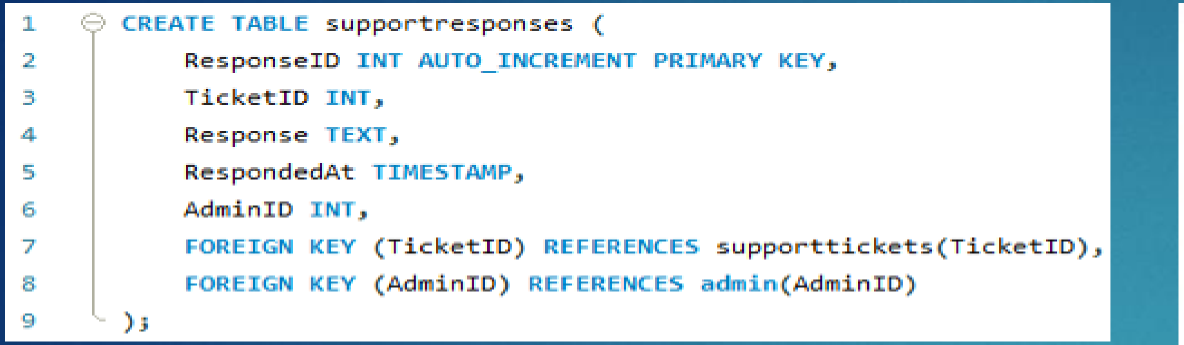
```sql
INSERT INTO myairbnb.roles (RoleID, RoleName) VALUES
(1, 'Host'),
(2, 'Guest'),
(3, 'Admin'),
(4, 'Super Host'),
(5, 'Customer Support'),
(6, 'Property Manager'),
(7, 'Booking Agent'),
(8, 'Content Moderator'),
(9, 'Finance Manager'),
(10, 'Marketing Specialist'),
(11, 'Event Coordinator'),
(12, 'Risk Management Specialist'),
(13, 'Operations Manager'),
(14, 'Legal Advisor'),
(15, 'Guest Experience Manager'),
(16, 'IT Support Specialist'),
(17, 'Executive Assistant'),
(18, 'Data Privacy Officer'),
(19, 'Sales Manager'),
(20, 'Training and Development Specialist');
```



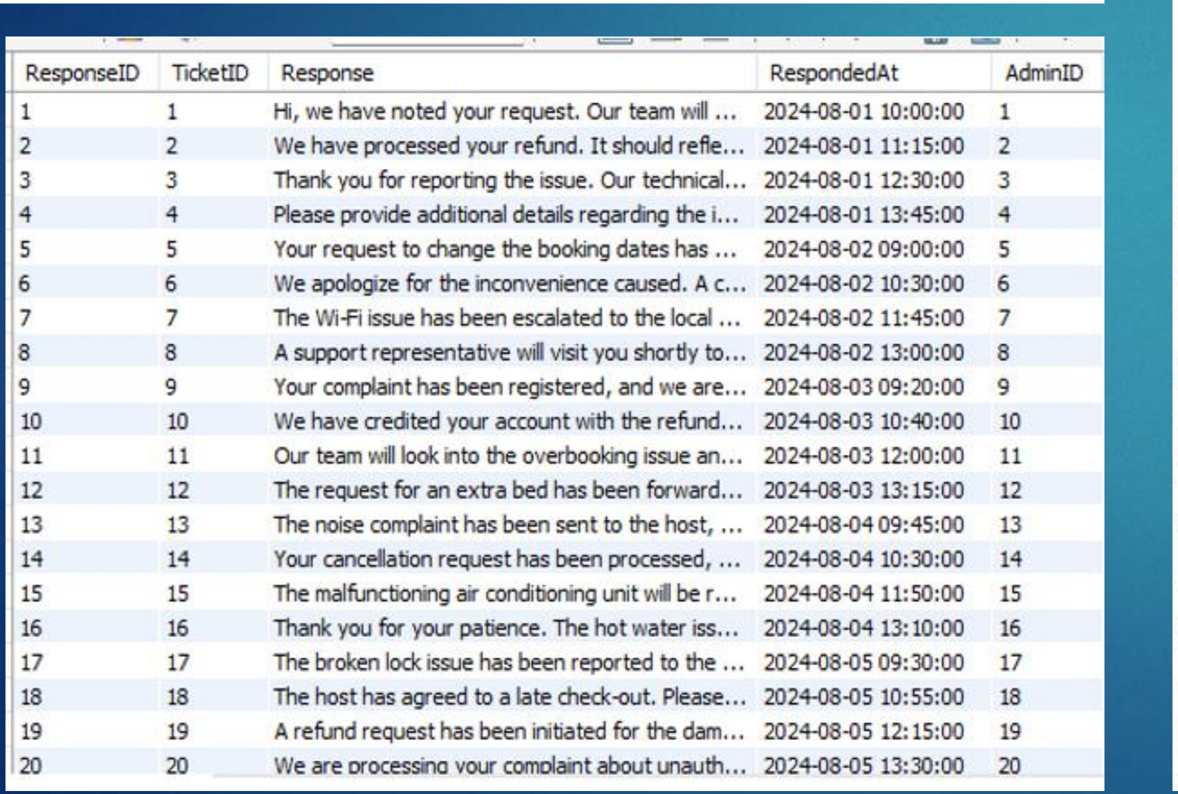
RoleID	RoleName
1	Host
2	Guest
3	Admin
4	Super Host
5	Customer Support
6	Property Manager
7	Booking Agent
8	Content Moderator
9	Finance Manager
10	Marketing Specialist
11	Event Coordinator
12	Risk Management ...
13	Operations Manager
14	Legal Advisor
15	Guest Experience ...
16	IT Support Specialist
17	Executive Assistant
18	Data Privacy Officer
19	Sales Manager
20	Training and Devel...

Supportresponses

Stores responses to support tickets. The SQL statement CREATE TABLE is used to define the structure of the of supportresponses table.

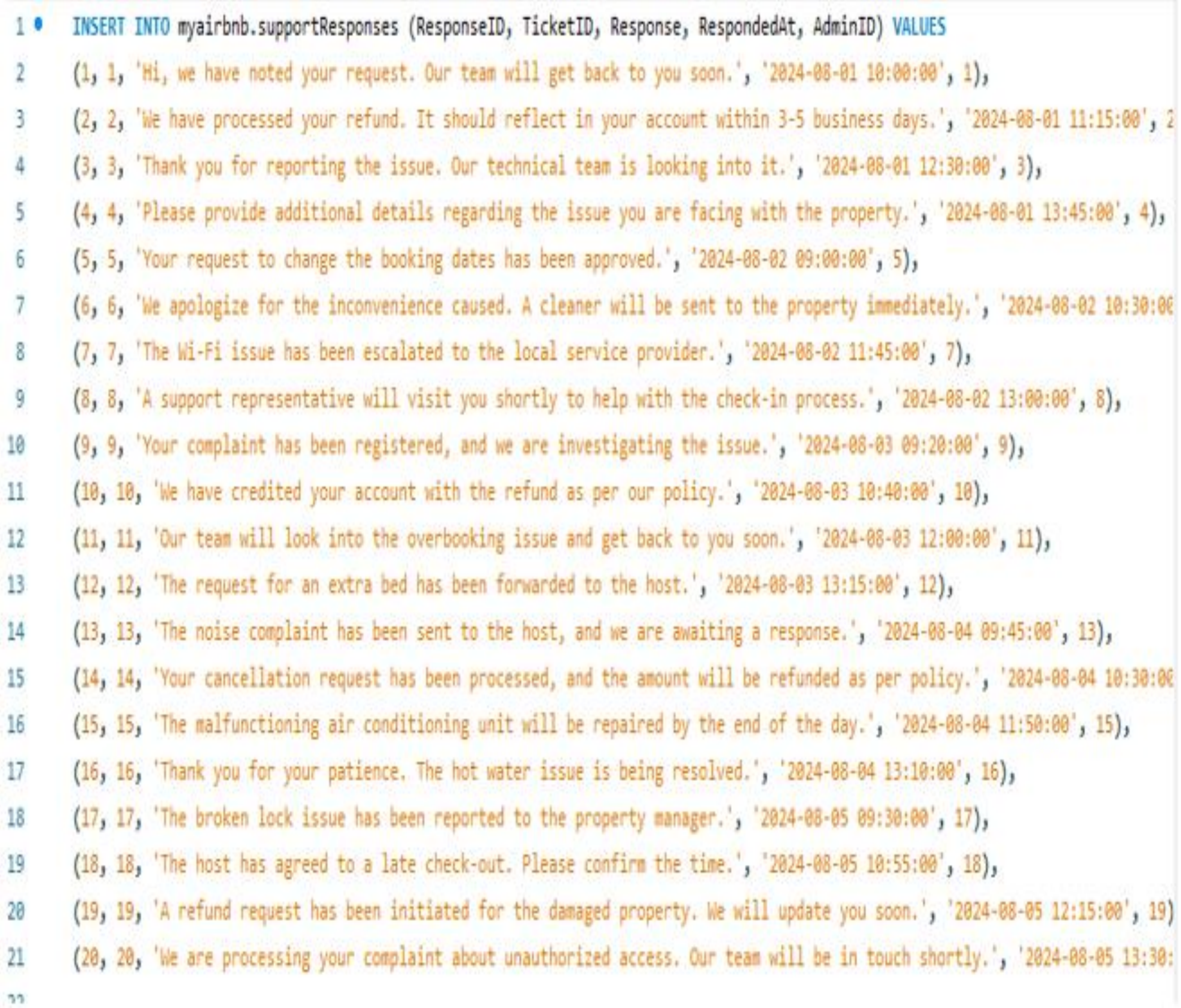


```sql
CREATE TABLE supportresponses (
    ResponseID INT AUTO_INCREMENT PRIMARY KEY,
    TicketID INT,
    Response TEXT,
    RespondedAt TIMESTAMP,
    AdminID INT,
    FOREIGN KEY (TicketID) REFERENCES supporttickets(TicketID),
    FOREIGN KEY (AdminID) REFERENCES admin(AdminID)
);
```



ResponseID	TicketID	Response	RespondedAt	AdminID
1	1	Hi, we have noted your request. Our team will ...	2024-08-01 10:00:00	1
2	2	We have processed your refund. It should refle...	2024-08-01 11:15:00	2
3	3	Thank you for reporting the issue. Our technical...	2024-08-01 12:30:00	3
4	4	Please provide additional details regarding the i...	2024-08-01 13:45:00	4
5	5	Your request to change the booking dates has ...	2024-08-02 09:00:00	5
6	6	We apologize for the inconvenience caused. A c...	2024-08-02 10:30:00	6
7	7	The Wi-Fi issue has been escalated to the local ...	2024-08-02 11:45:00	7
8	8	A support representative will visit you shortly to...	2024-08-02 13:00:00	8
9	9	Your complaint has been registered, and we are...	2024-08-03 09:20:00	9
10	10	We have credited your account with the refund...	2024-08-03 10:40:00	10
11	11	Our team will look into the overbooking issue an...	2024-08-03 12:00:00	11
12	12	The request for an extra bed has been forward...	2024-08-03 13:15:00	12
13	13	The noise complaint has been sent to the host, ...	2024-08-04 09:45:00	13
14	14	Your cancellation request has been processed, ...	2024-08-04 10:30:00	14
15	15	The malfunctioning air conditioning unit will be r...	2024-08-04 11:50:00	15
16	16	Thank you for your patience. The hot water iss...	2024-08-04 13:10:00	16
17	17	The broken lock issue has been reported to the ...	2024-08-05 09:30:00	17
18	18	The host has agreed to a late check-out. Please...	2024-08-05 10:55:00	18
19	19	A refund request has been initiated for the dam...	2024-08-05 12:15:00	19
20	20	We are processing your complaint about unauth...	2024-08-05 13:30:00	20

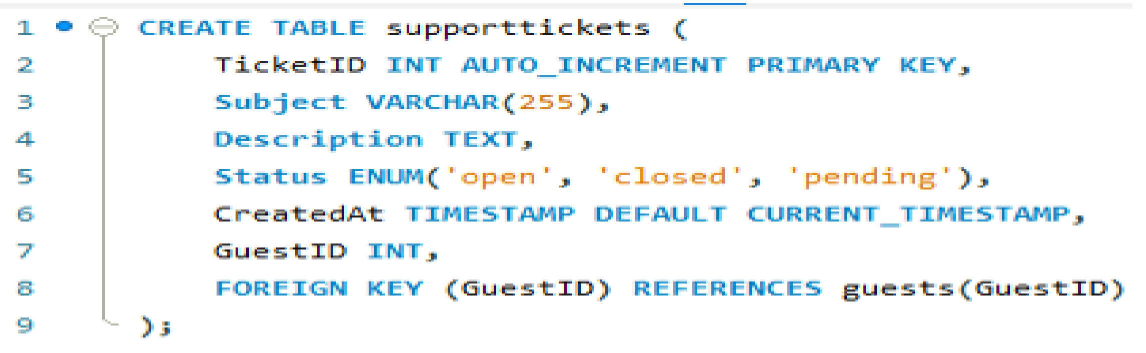
Support responses can be inserted into the table using the INSERT INTO statement.



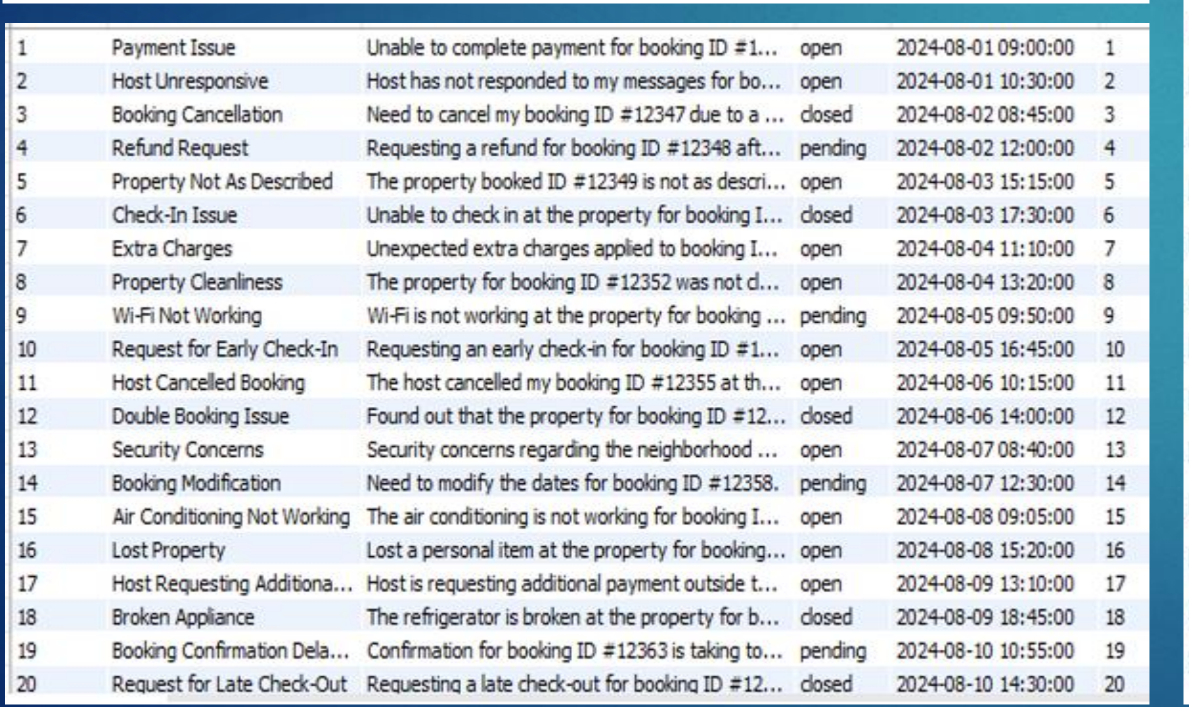
```sql
INSERT INTO myairbnb.supportResponses (ResponseID, TicketID, Response, RespondedAt, AdminID) VALUES
(1, 1, 'Hi, we have noted your request. Our team will get back to you soon.', '2024-08-01 10:00:00', 1),
(2, 2, 'We have processed your refund. It should reflect in your account within 3-5 business days.', '2024-08-01 11:15:00', 2
(3, 3, 'Thank you for reporting the issue. Our technical team is looking into it.', '2024-08-01 12:30:00', 3),
(4, 4, 'Please provide additional details regarding the issue you are facing with the property.', '2024-08-01 13:45:00', 4),
(5, 5, 'Your request to change the booking dates has been approved.', '2024-08-02 09:00:00', 5),
(6, 6, 'We apologize for the inconvenience caused. A cleaner will be sent to the property immediately.', '2024-08-02 10:30:0
(7, 7, 'The Wi-Fi issue has been escalated to the local service provider.', '2024-08-02 11:45:00', 7),
(8, 8, 'A support representative will visit you shortly to help with the check-in process.', '2024-08-02 13:00:00', 8),
(9, 9, 'Your complaint has been registered, and we are investigating the issue.', '2024-08-03 09:20:00', 9),
(10, 10, 'We have credited your account with the refund as per our policy.', '2024-08-03 10:40:00', 10),
(11, 11, 'Our team will look into the overbooking issue and get back to you soon.', '2024-08-03 12:00:00', 11),
(12, 12, 'The request for an extra bed has been forwarded to the host.', '2024-08-03 13:15:00', 12),
(13, 13, 'The noise complaint has been sent to the host, and we are awaiting a response.', '2024-08-04 09:45:00', 13),
(14, 14, 'Your cancellation request has been processed, and the amount will be refunded as per policy.', '2024-08-04 10:30:0
(15, 15, 'The malfunctioning air conditioning unit will be repaired by the end of the day.', '2024-08-04 11:50:00', 15),
(16, 16, 'Thank you for your patience. The hot water issue is being resolved.', '2024-08-04 13:10:00', 16),
(17, 17, 'The broken lock issue has been reported to the property manager.', '2024-08-05 09:30:00', 17),
(18, 18, 'The host has agreed to a late check-out. Please confirm the time.', '2024-08-05 10:55:00', 18),
(19, 19, 'A refund request has been initiated for the damaged property. We will update you soon.', '2024-08-05 12:15:00', 19)
(20, 20, 'We are processing your complaint about unauthorized access. Our team will be in touch shortly.', '2024-08-05 13:30:
```

Supporttickets

Stores support tickets submitted by users. The SQL statement CREATE TABLE is used to define the structure of the of supporttickets table.

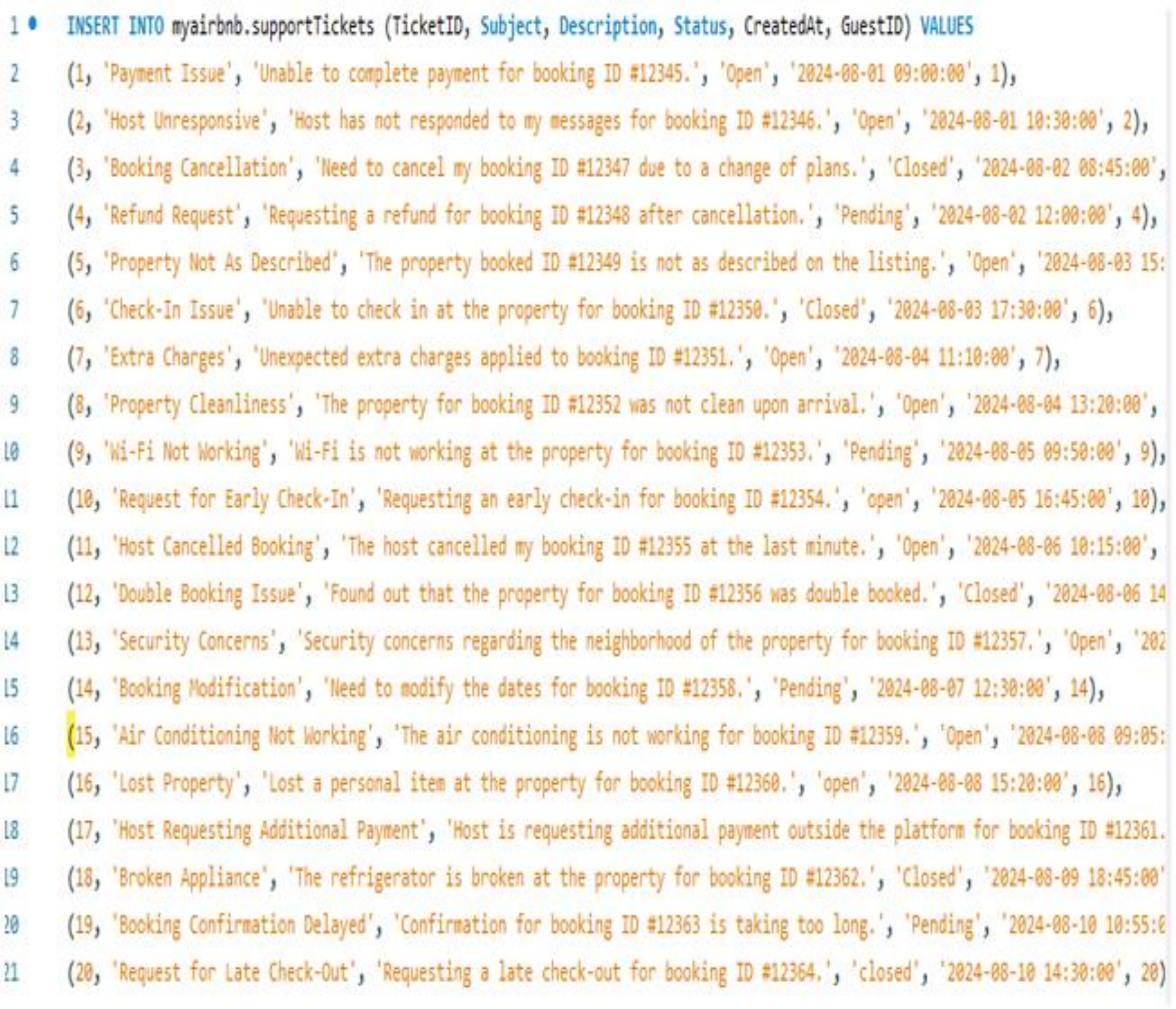


```
CREATE TABLE supporttickets (
    TicketID INT AUTO_INCREMENT PRIMARY KEY,
    Subject VARCHAR(255),
    Description TEXT,
    Status ENUM('open', 'closed', 'pending'),
    CreatedAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    GuestID INT,
    FOREIGN KEY (GuestID) REFERENCES guests(GuestID)
);
```



1	Payment Issue	Unable to complete payment for booking ID #1...	open	2024-08-01 09:00:00	1
2	Host Unresponsive	Host has not responded to my messages for bo...	open	2024-08-01 10:30:00	2
3	Booking Cancellation	Need to cancel my booking ID #12347 due to a ...	closed	2024-08-02 08:45:00	3
4	Refund Request	Requesting a refund for booking ID #12348 aft...	pending	2024-08-02 12:00:00	4
5	Property Not As Described	The property booked ID #12349 is not as descri...	open	2024-08-03 15:15:00	5
6	Check-In Issue	Unable to check in at the property for booking I...	closed	2024-08-03 17:30:00	6
7	Extra Charges	Unexpected extra charges applied to booking I...	open	2024-08-04 11:10:00	7
8	Property Cleanliness	The property for booking ID #12352 was not cl...	open	2024-08-04 13:20:00	8
9	Wi-Fi Not Working	Wi-Fi is not working at the property for booking ...	pending	2024-08-05 09:50:00	9
10	Request for Early Check-In	Requesting an early check-in for booking ID #1...	open	2024-08-05 16:45:00	10
11	Host Cancelled Booking	The host cancelled my booking ID #12355 at th...	open	2024-08-06 10:15:00	11
12	Double Booking Issue	Found out that the property for booking ID #12...	closed	2024-08-06 14:00:00	12
13	Security Concerns	Security concerns regarding the neighborhood ...	open	2024-08-07 08:40:00	13
14	Booking Modification	Need to modify the dates for booking ID #12358.	pending	2024-08-07 12:30:00	14
15	Air Conditioning Not Working	The air conditioning is not working for booking I...	open	2024-08-08 09:05:00	15
16	Lost Property	Lost a personal item at the property for booking...	open	2024-08-08 15:20:00	16
17	Host Requesting Additiona...	Host is requesting additional payment outside t...	open	2024-08-09 13:10:00	17
18	Broken Appliance	The refrigerator is broken at the property for b...	closed	2024-08-09 18:45:00	18
19	Booking Confirmation Dela...	Confirmation for booking ID #12363 is taking to...	pending	2024-08-10 10:55:00	19
20	Request for Late Check-Out	Requesting a late check-out for booking ID #12...	closed	2024-08-10 14:30:00	20

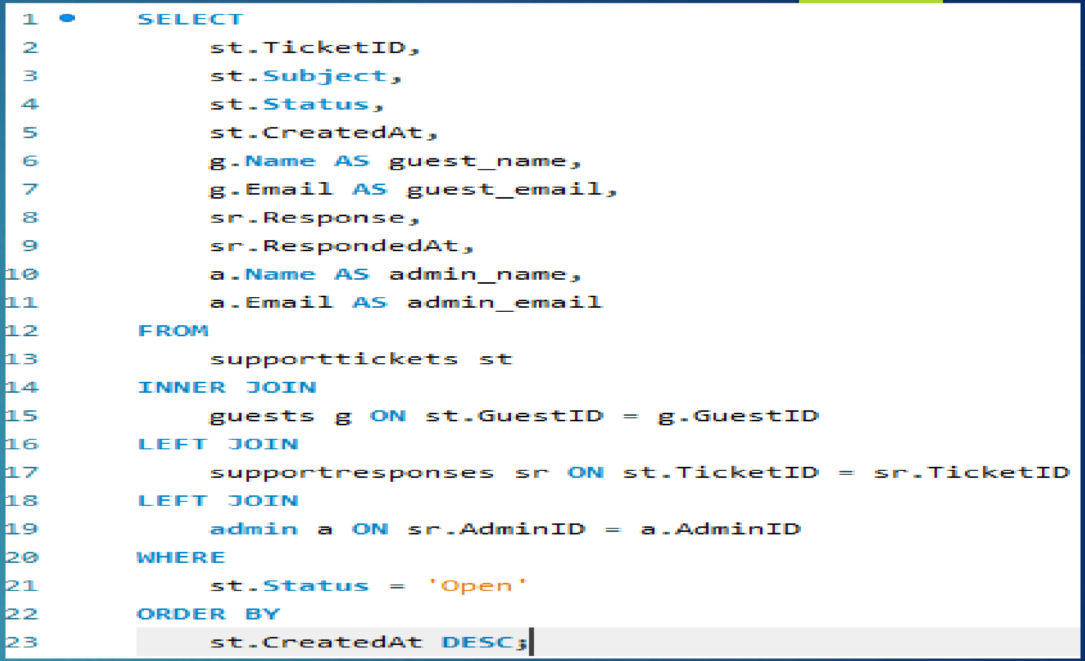
Supporttickets (e.g payment issue) can be inserted into the table using the INSERT INTO statement.



```
INSERT INTO myairbnb.supportTickets (TicketID, Subject, Description, Status, CreatedAt, GuestID) VALUES
(1, 'Payment Issue', 'Unable to complete payment for booking ID #12345.', 'Open', '2024-08-01 09:00:00', 1),
(2, 'Host Unresponsive', 'Host has not responded to my messages for booking ID #12346.', 'Open', '2024-08-01 10:30:00', 2),
(3, 'Booking Cancellation', 'Need to cancel my booking ID #12347 due to a change of plans.', 'Closed', '2024-08-02 08:45:00',
(4, 'Refund Request', 'Requesting a refund for booking ID #12348 after cancellation.', 'Pending', '2024-08-02 12:00:00', 4),
(5, 'Property Not As Described', 'The property booked ID #12349 is not as described on the listing.', 'Open', '2024-08-03 15:
(6, 'Check-In Issue', 'Unable to check in at the property for booking ID #12350.', 'Closed', '2024-08-03 17:30:00', 6),
(7, 'Extra Charges', 'Unexpected extra charges applied to booking ID #12351.', 'Open', '2024-08-04 11:10:00', 7),
(8, 'Property Cleanliness', 'The property for booking ID #12352 was not clean upon arrival.', 'Open', '2024-08-04 13:20:00',
(9, 'Wi-Fi Not Working', 'Wi-Fi is not working at the property for booking ID #12353.', 'Pending', '2024-08-05 09:50:00', 9),
(10, 'Request for Early Check-In', 'Requesting an early check-in for booking ID #12354.', 'open', '2024-08-05 16:45:00', 10),
(11, 'Host Cancelled Booking', 'The host cancelled my booking ID #12355 at the last minute.', 'Open', '2024-08-06 10:15:00',
(12, 'Double Booking Issue', 'Found out that the property for booking ID #12356 was double booked.', 'Closed', '2024-08-06 14
(13, 'Security Concerns', 'Security concerns regarding the neighborhood of the property for booking ID #12357.', 'Open', '202
(14, 'Booking Modification', 'Need to modify the dates for booking ID #12358.', 'Pending', '2024-08-07 12:30:00', 14),
(15, 'Air Conditioning Not Working', 'The air conditioning is not working for booking ID #12359.', 'Open', '2024-08-08 09:05:
(16, 'Lost Property', 'Lost a personal item at the property for booking ID #12360.', 'open', '2024-08-08 15:20:00', 16),
(17, 'Host Requesting Additional Payment', 'Host is requesting additional payment outside the platform for booking ID #12361.
(18, 'Broken Appliance', 'The refrigerator is broken at the property for booking ID #12362.', 'Closed', '2024-08-09 18:45:00'
(19, 'Booking Confirmation Delayed', 'Confirmation for booking ID #12363 is taking too long.', 'Pending', '2024-08-10 10:55:
(20, 'Request for Late Check-Out', 'Requesting a late check-out for booking ID #12364.', 'closed', '2024-08-10 14:30:00', 20)
```

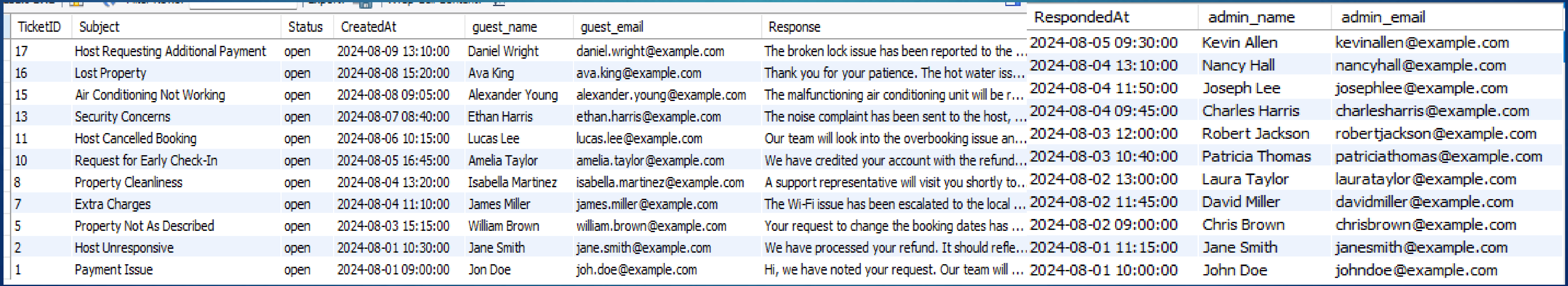
Supporttickets: Test Case: detailed view of support tickets and their responses

This query retrieves data from Four tables supporttickets , admin, guests and supportresponses.



```
SELECT
    st.TicketID,
    st.Subject,
    st.Status,
    st.CreatedAt,
    g.Name AS guest_name,
    g.Email AS guest_email,
    sr.Response,
    sr.RespondedAt,
    a.Name AS admin_name,
    a.Email AS admin_email
FROM
    supporttickets st
INNER JOIN
    guests g ON st.GuestID = g.GuestID
LEFT JOIN
    supportresponses sr ON st.TicketID = sr.TicketID
LEFT JOIN
    admin a ON sr.AdminID = a.AdminID
WHERE
    st.Status = 'Open'
ORDER BY
    st.CreatedAt DESC;
```

Result



TicketID	Subject	Status	CreatedAt	guest_name	guest_email	Response	RespondedAt	admin_name	admin_email
17	Host Requesting Additional Payment	open	2024-08-09 13:10:00	Daniel Wright	daniel.wright@example.com	The broken lock issue has been reported to the ...	2024-08-05 09:30:00	Kevin Allen	kevinallen@example.com
16	Lost Property	open	2024-08-08 15:20:00	Ava King	ava.king@example.com	Thank you for your patience. The hot water iss...	2024-08-04 13:10:00	Nancy Hall	nancyhall@example.com
15	Air Conditioning Not Working	open	2024-08-08 09:05:00	Alexander Young	alexander.young@example.com	The malfunctioning air conditioning unit will be r...	2024-08-04 11:50:00	Joseph Lee	josephlee@example.com
13	Security Concerns	open	2024-08-07 08:40:00	Ethan Harris	ethan.harris@example.com	The noise complaint has been sent to the host, ...	2024-08-04 09:45:00	Charles Harris	charlesharris@example.com
11	Host Cancelled Booking	open	2024-08-06 10:15:00	Lucas Lee	lucas.lee@example.com	Our team will look into the overbooking issue an...	2024-08-03 12:00:00	Robert Jackson	robertjackson@example.com
10	Request for Early Check-In	open	2024-08-05 16:45:00	Amelia Taylor	amelia.taylor@example.com	We have credited your account with the refund...	2024-08-03 10:40:00	Patricia Thomas	patriciathomas@example.com
8	Property Cleanliness	open	2024-08-04 13:20:00	Isabella Martinez	isabella.martinez@example.com	A support representative will visit you shortly to...	2024-08-02 13:00:00	Laura Taylor	laurataylor@example.com
7	Extra Charges	open	2024-08-04 11:10:00	James Miller	james.miller@example.com	The Wi-Fi issue has been escalated to the local ...	2024-08-02 11:45:00	David Miller	davidmiller@example.com
5	Property Not As Described	open	2024-08-03 15:15:00	William Brown	william.brown@example.com	Your request to change the booking dates has ...	2024-08-02 09:00:00	Chris Brown	chrisbrown@example.com
2	Host Unresponsive	open	2024-08-01 10:30:00	Jane Smith	jane.smith@example.com	We have processed your refund. It should refle...	2024-08-01 11:15:00	Jane Smith	janesmith@example.com
1	Payment Issue	open	2024-08-01 09:00:00	Jon Doe	joh.doe@example.com	Hi, we have noted your request. Our team will ...	2024-08-01 10:00:00	John Doe	johndoe@example.com

SELECT Clause: st.TicketID selects the unique identifier (TicketID) for each support ticket from the supporttickets table. **st.Subject** retrieves the Subject field of the support ticket. st**.Status** retrieves the current Status of the support ticket (e.g., 'Open', 'Closed', 'Pending')**. st.CreatedAt** retrieves the CreatedAt timestamp for the support ticket. g.Name AS guest_name retrieves the name of guest and aliases it for clarity, g.Email AS guest_email retrieves the guest email, **sr.Response**: Retrieves the actual Response text provided for a support ticket. **sr.RespondedAt**: Retrieves the timestamp (RespondedAt) indicating when the response was made. a.Name AS admin_name and a.Email AS admin_email retrieves the admin name and email, it aliases them as indicated for clarity,

FROM Clause: The query starts by selecting data from the supporttickets table, which contains information about the support tickets.

INNER JOIN CLause: Combines rows from supporttickets st and guests g with only rows matching in both tables included in the results.

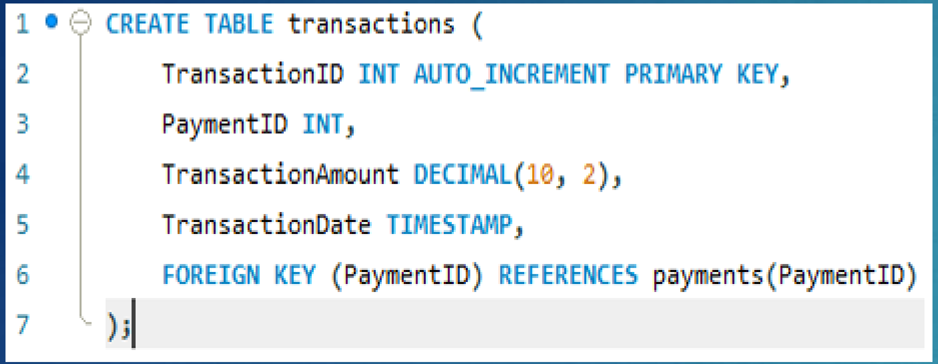
LEFT JOIN Clause: The first LEFT JOIN is used to combine and return all records from the supporttickets table that matches records from the supportresponses table. If there is no match, the result is NULL for columns from the supportresponses table. **ON st.TicketID = sr.TicketID**: This join condition specifies that rows in supporttickets should be matched with rows in supportresponses where the TicketID is the same. Essentially, this connects support tickets to their respective responses. The second LEFT JOIN joins supportresponses and admin on AdminID

WHERE Clause: Filters the result to only include suppottickets with status, st.Status = 'Open’

ORDER BY Clause: ORDER BY st.CreatedAt DESC sorts the results primarily by the TicketID in descending order. This groups all entries by each created at date.

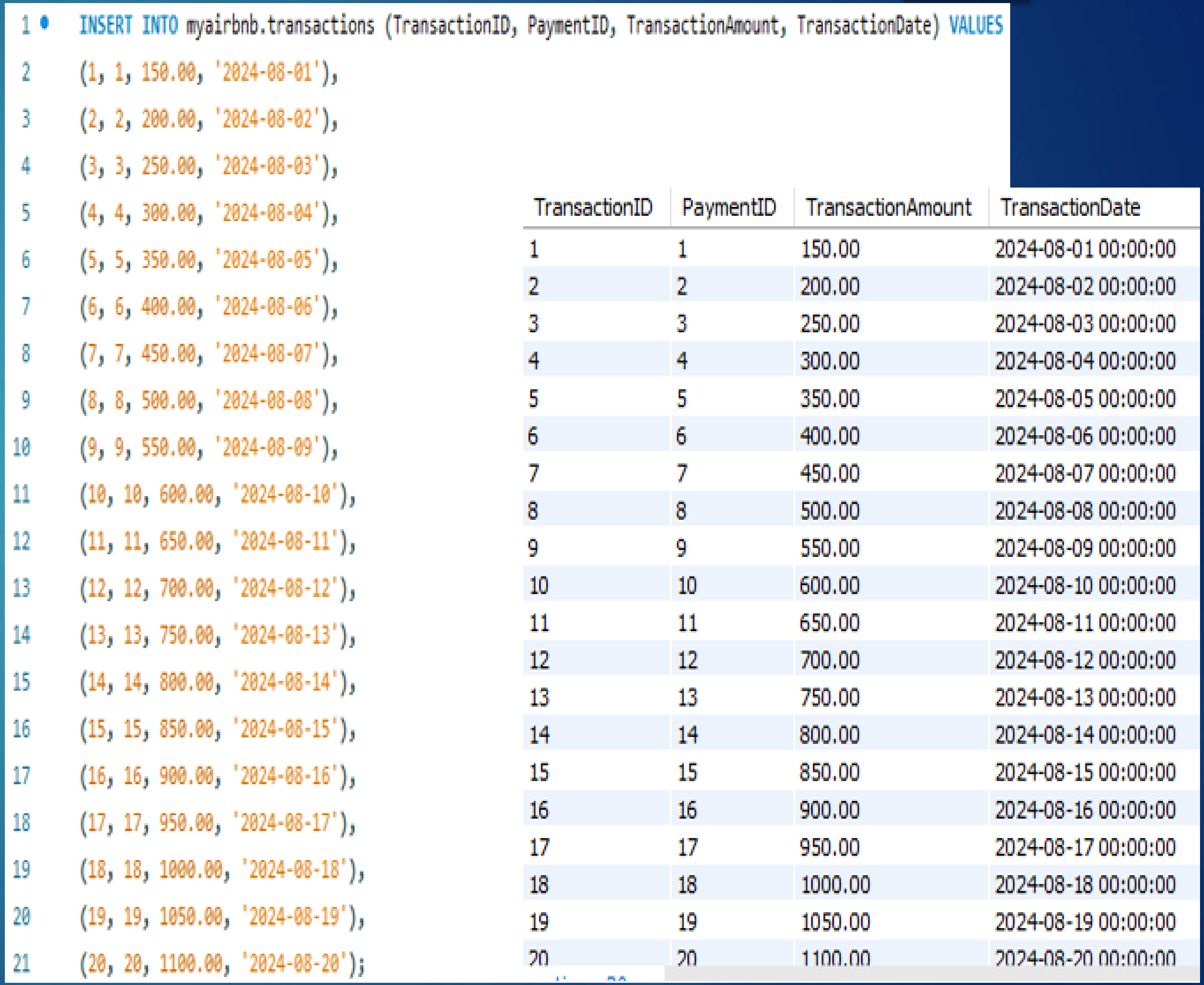
Transactions

Stores financial transactions related to bookings, payments, etc. The SQL statement CREATE TABLE is used to define the structure of the of transaction table.



```sql
CREATE TABLE transactions (
    TransactionID INT AUTO_INCREMENT PRIMARY KEY,
    PaymentID INT,
    TransactionAmount DECIMAL(10, 2),
    TransactionDate TIMESTAMP,
    FOREIGN KEY (PaymentID) REFERENCES payments(PaymentID)
);
```

Transaction can be inserted into the table using the INSERT INTO statement.

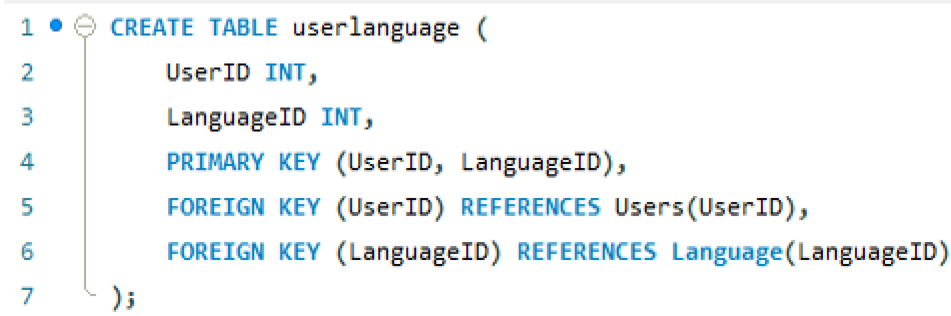


```sql
INSERT INTO myairbnb.transactions (TransactionID, PaymentID, TransactionAmount, TransactionDate) VALUES
(1, 1, 150.00, '2024-08-01'),
(2, 2, 200.00, '2024-08-02'),
(3, 3, 250.00, '2024-08-03'),
(4, 4, 300.00, '2024-08-04'),
(5, 5, 350.00, '2024-08-05'),
(6, 6, 400.00, '2024-08-06'),
(7, 7, 450.00, '2024-08-07'),
(8, 8, 500.00, '2024-08-08'),
(9, 9, 550.00, '2024-08-09'),
(10, 10, 600.00, '2024-08-10'),
(11, 11, 650.00, '2024-08-11'),
(12, 12, 700.00, '2024-08-12'),
(13, 13, 750.00, '2024-08-13'),
(14, 14, 800.00, '2024-08-14'),
(15, 15, 850.00, '2024-08-15'),
(16, 16, 900.00, '2024-08-16'),
(17, 17, 950.00, '2024-08-17'),
(18, 18, 1000.00, '2024-08-18'),
(19, 19, 1050.00, '2024-08-19'),
(20, 20, 1100.00, '2024-08-20');
```

TransactionID	PaymentID	TransactionAmount	TransactionDate
1	1	150.00	2024-08-01 00:00:00
2	2	200.00	2024-08-02 00:00:00
3	3	250.00	2024-08-03 00:00:00
4	4	300.00	2024-08-04 00:00:00
5	5	350.00	2024-08-05 00:00:00
6	6	400.00	2024-08-06 00:00:00
7	7	450.00	2024-08-07 00:00:00
8	8	500.00	2024-08-08 00:00:00
9	9	550.00	2024-08-09 00:00:00
10	10	600.00	2024-08-10 00:00:00
11	11	650.00	2024-08-11 00:00:00
12	12	700.00	2024-08-12 00:00:00
13	13	750.00	2024-08-13 00:00:00
14	14	800.00	2024-08-14 00:00:00
15	15	850.00	2024-08-15 00:00:00
16	16	900.00	2024-08-16 00:00:00
17	17	950.00	2024-08-17 00:00:00
18	18	1000.00	2024-08-18 00:00:00
19	19	1050.00	2024-08-19 00:00:00
20	20	1100.00	2024-08-20 00:00:00

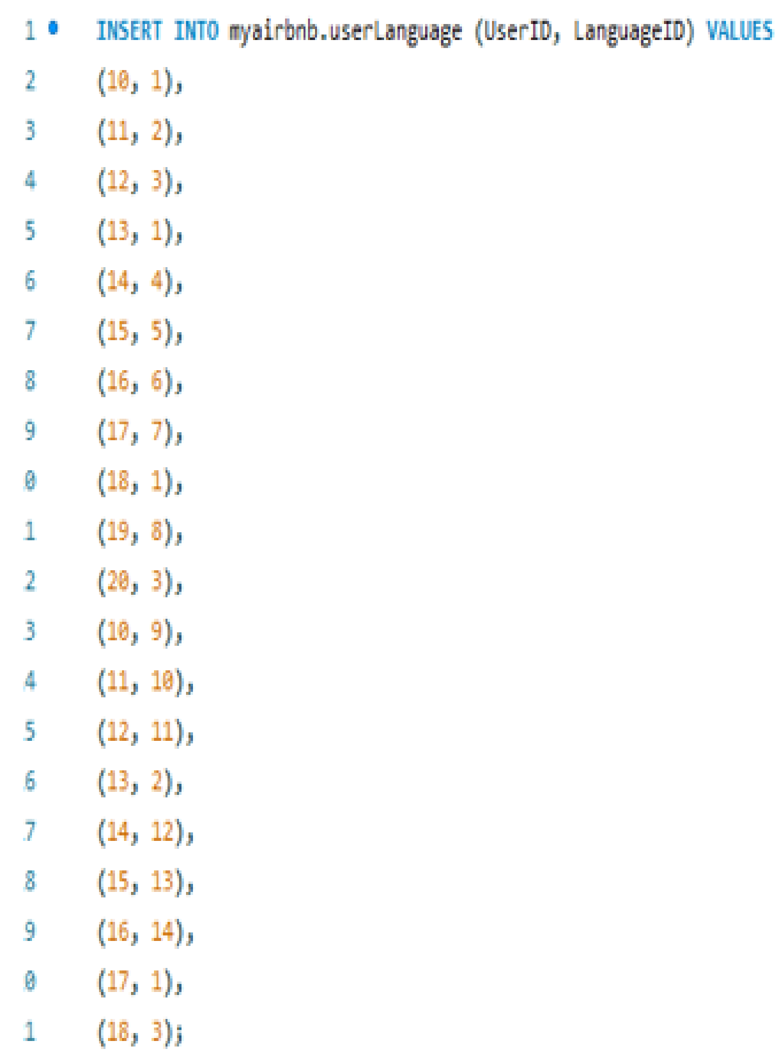
Userlanguage

Stores the relationship between users and the languages they speak or prefer. The SQL statement CREATE TABLE is used to define the structure of the of userlanguage table.

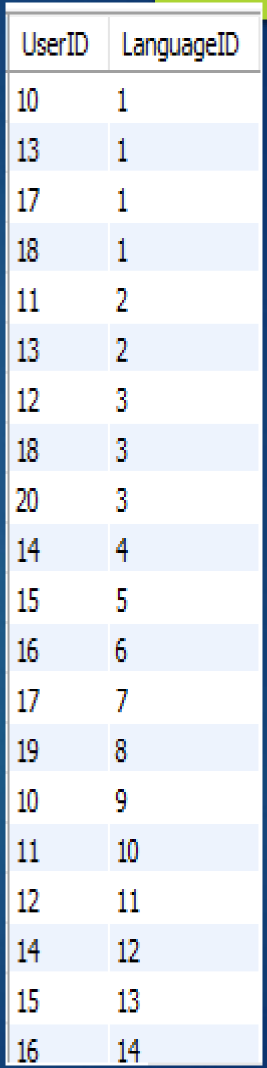


```
CREATE TABLE userlanguage (
    UserID INT,
    LanguageID INT,
    PRIMARY KEY (UserID, LanguageID),
    FOREIGN KEY (UserID) REFERENCES Users(UserID),
    FOREIGN KEY (LanguageID) REFERENCES Language(LanguageID)
);
```

Userlanguage can be inserted into the table using the INSERT INTO statement.



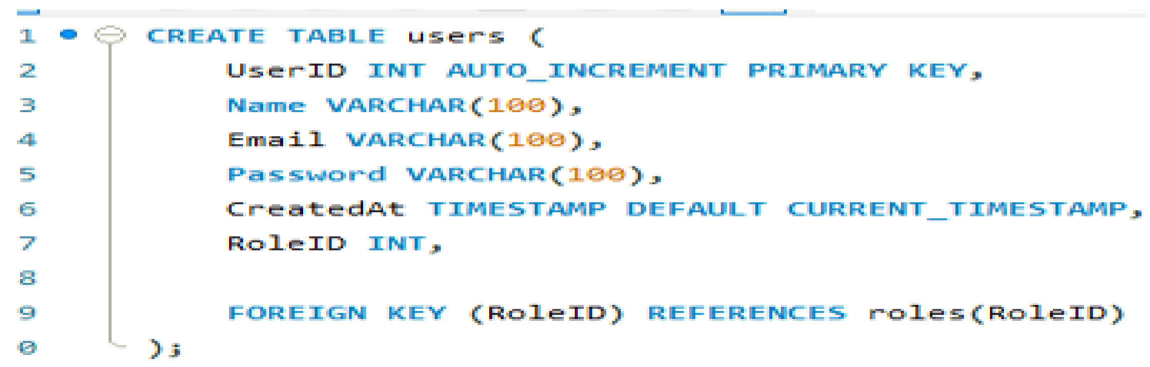
```
INSERT INTO myairbnb.userLanguage (UserID, LanguageID) VALUES
(10, 1),
(11, 2),
(12, 3),
(13, 1),
(14, 4),
(15, 5),
(16, 6),
(17, 7),
(18, 1),
(19, 8),
(20, 3),
(10, 9),
(11, 10),
(12, 11),
(13, 2),
(14, 12),
(15, 13),
(16, 14),
(17, 1),
(18, 3);
```



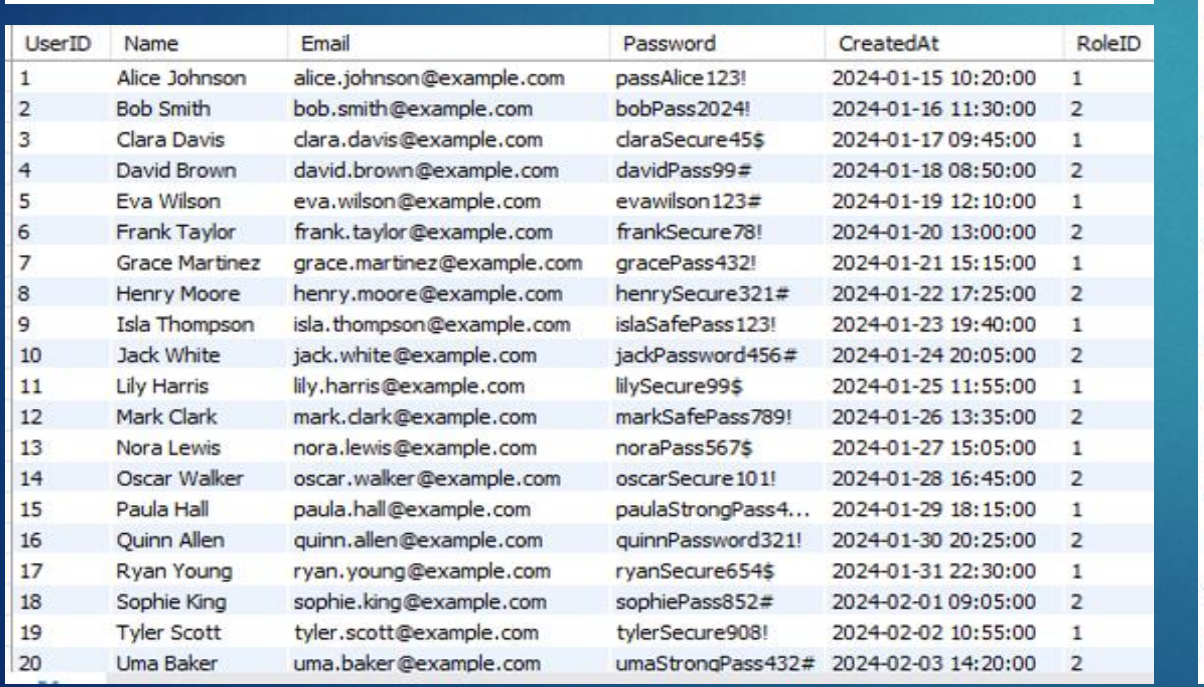
UserID	LanguageID
10	1
13	1
17	1
18	1
11	2
13	2
12	3
18	3
20	3
14	4
15	5
16	6
17	7
19	8
10	9
11	10
12	11
14	12
15	13
16	14

Users

Stores information about all users, both guests, and hosts. The SQL statement CREATE TABLE is used to define the structure of the of users table.

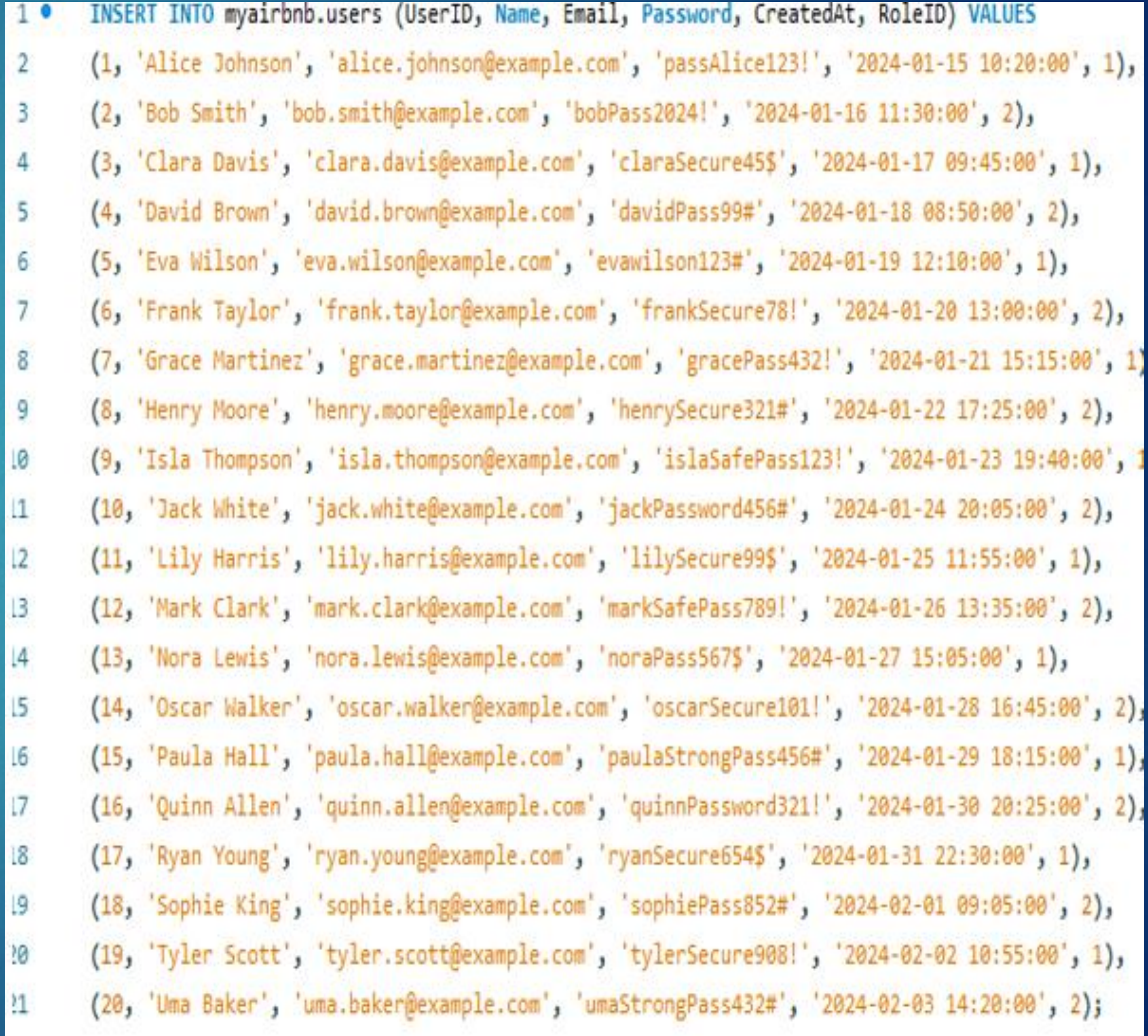


```
CREATE TABLE users (
        UserID INT AUTO_INCREMENT PRIMARY KEY,
        Name VARCHAR(100),
        Email VARCHAR(100),
        Password VARCHAR(100),
        CreatedAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
        RoleID INT,

        FOREIGN KEY (RoleID) REFERENCES roles(RoleID)
    );
```



UserID	Name	Email	Password	CreatedAt	RoleID
1	Alice Johnson	alice.johnson@example.com	passAlice123!	2024-01-15 10:20:00	1
2	Bob Smith	bob.smith@example.com	bobPass2024!	2024-01-16 11:30:00	2
3	Clara Davis	clara.davis@example.com	claraSecure45$	2024-01-17 09:45:00	1
4	David Brown	david.brown@example.com	davidPass99#	2024-01-18 08:50:00	2
5	Eva Wilson	eva.wilson@example.com	evawilson123#	2024-01-19 12:10:00	1
6	Frank Taylor	frank.taylor@example.com	frankSecure78!	2024-01-20 13:00:00	2
7	Grace Martinez	grace.martinez@example.com	gracePass432!	2024-01-21 15:15:00	1
8	Henry Moore	henry.moore@example.com	henrySecure321#	2024-01-22 17:25:00	2
9	Isla Thompson	isla.thompson@example.com	islaSafePass123!	2024-01-23 19:40:00	1
10	Jack White	jack.white@example.com	jackPassword456#	2024-01-24 20:05:00	2
11	Lily Harris	lily.harris@example.com	lilySecure99$	2024-01-25 11:55:00	1
12	Mark Clark	mark.clark@example.com	markSafePass789!	2024-01-26 13:35:00	2
13	Nora Lewis	nora.lewis@example.com	noraPass567$	2024-01-27 15:05:00	1
14	Oscar Walker	oscar.walker@example.com	oscarSecure101!	2024-01-28 16:45:00	2
15	Paula Hall	paula.hall@example.com	paulaStrongPass4...	2024-01-29 18:15:00	1
16	Quinn Allen	quinn.allen@example.com	quinnPassword321!	2024-01-30 20:25:00	2
17	Ryan Young	ryan.young@example.com	ryanSecure654$	2024-01-31 22:30:00	1
18	Sophie King	sophie.king@example.com	sophiePass852#	2024-02-01 09:05:00	2
19	Tyler Scott	tyler.scott@example.com	tylerSecure908!	2024-02-02 10:55:00	1
20	Uma Baker	uma.baker@example.com	umaStrongPass432#	2024-02-03 14:20:00	2

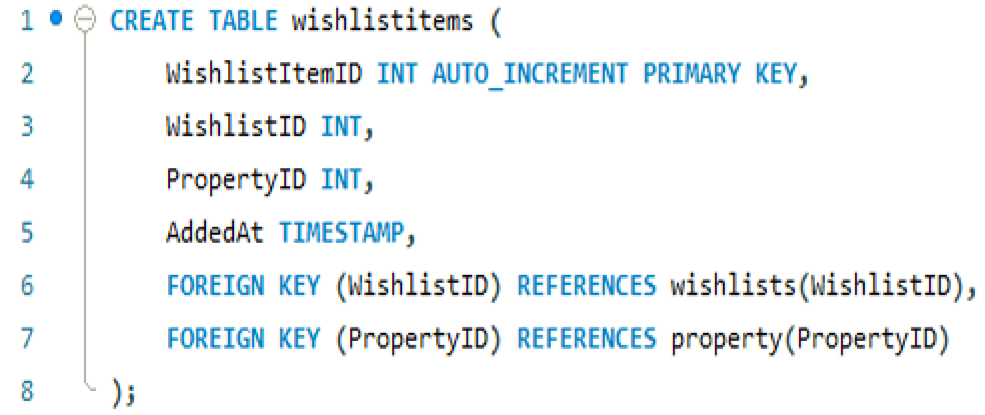
Users (e.g guests) information can be inserted into the table using the INSERT INTO statement.



```
INSERT INTO myairbnb.users (UserID, Name, Email, Password, CreatedAt, RoleID) VALUES
(1, 'Alice Johnson', 'alice.johnson@example.com', 'passAlice123!', '2024-01-15 10:20:00', 1),
(2, 'Bob Smith', 'bob.smith@example.com', 'bobPass2024!', '2024-01-16 11:30:00', 2),
(3, 'Clara Davis', 'clara.davis@example.com', 'claraSecure45$', '2024-01-17 09:45:00', 1),
(4, 'David Brown', 'david.brown@example.com', 'davidPass99#', '2024-01-18 08:50:00', 2),
(5, 'Eva Wilson', 'eva.wilson@example.com', 'evawilson123#', '2024-01-19 12:10:00', 1),
(6, 'Frank Taylor', 'frank.taylor@example.com', 'frankSecure78!', '2024-01-20 13:00:00', 2),
(7, 'Grace Martinez', 'grace.martinez@example.com', 'gracePass432!', '2024-01-21 15:15:00', 1)
(8, 'Henry Moore', 'henry.moore@example.com', 'henrySecure321#', '2024-01-22 17:25:00', 2),
(9, 'Isla Thompson', 'isla.thompson@example.com', 'islaSafePass123!', '2024-01-23 19:40:00', 1
(10, 'Jack White', 'jack.white@example.com', 'jackPassword456#', '2024-01-24 20:05:00', 2),
(11, 'Lily Harris', 'lily.harris@example.com', 'lilySecure99$', '2024-01-25 11:55:00', 1),
(12, 'Mark Clark', 'mark.clark@example.com', 'markSafePass789!', '2024-01-26 13:35:00', 2),
(13, 'Nora Lewis', 'nora.lewis@example.com', 'noraPass567$', '2024-01-27 15:05:00', 1),
(14, 'Oscar Walker', 'oscar.walker@example.com', 'oscarSecure101!', '2024-01-28 16:45:00', 2),
(15, 'Paula Hall', 'paula.hall@example.com', 'paulaStrongPass456#', '2024-01-29 18:15:00', 1),
(16, 'Quinn Allen', 'quinn.allen@example.com', 'quinnPassword321!', '2024-01-30 20:25:00', 2),
(17, 'Ryan Young', 'ryan.young@example.com', 'ryanSecure654$', '2024-01-31 22:30:00', 1),
(18, 'Sophie King', 'sophie.king@example.com', 'sophiePass852#', '2024-02-01 09:05:00', 2),
(19, 'Tyler Scott', 'tyler.scott@example.com', 'tylerSecure908!', '2024-02-02 10:55:00', 1),
(20, 'Uma Baker', 'uma.baker@example.com', 'umaStrongPass432#', '2024-02-03 14:20:00', 2);
```

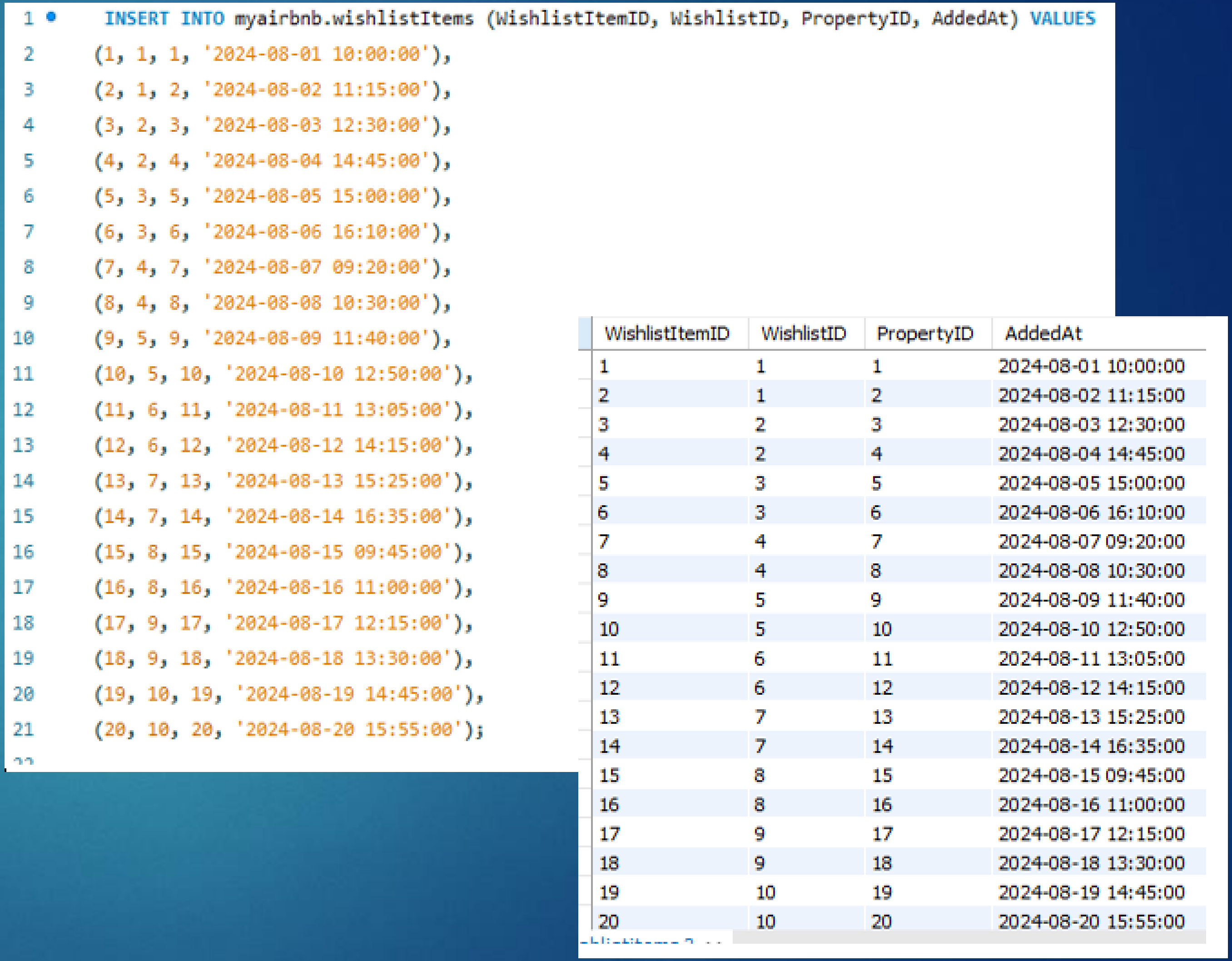
Wishlistitems

Stores properties added to a guest's wishlist.The SQL statement CREATE TABLE is used to define the structure of the of wishlistitems table.



```sql
CREATE TABLE wishlistitems (
    WishlistItemID INT AUTO_INCREMENT PRIMARY KEY,
    WishlistID INT,
    PropertyID INT,
    AddedAt TIMESTAMP,
    FOREIGN KEY (WishlistID) REFERENCES wishlists(WishlistID),
    FOREIGN KEY (PropertyID) REFERENCES property(PropertyID)
);
```

Wishlist item can be inserted into the table using the INSERT INTO statement.

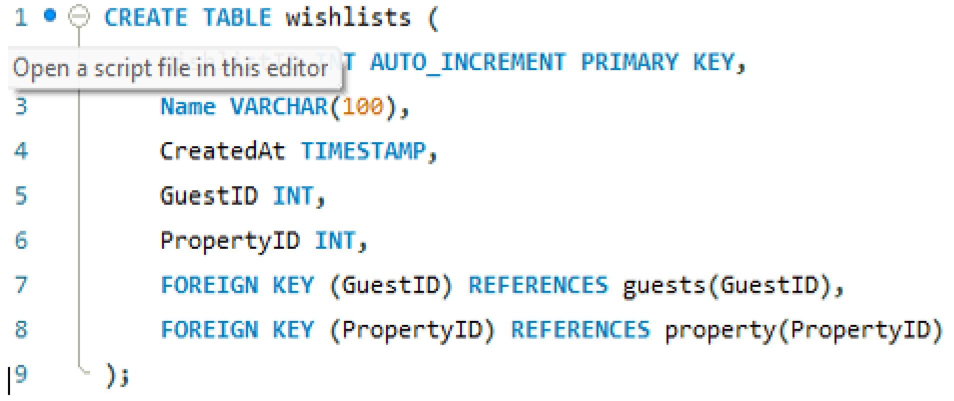


```sql
 INSERT INTO myairbnb.wishlistItems (WishlistItemID, WishlistID, PropertyID, AddedAt) VALUES
(1, 1, 1, '2024-08-01 10:00:00'),
(2, 1, 2, '2024-08-02 11:15:00'),
(3, 2, 3, '2024-08-03 12:30:00'),
(4, 2, 4, '2024-08-04 14:45:00'),
(5, 3, 5, '2024-08-05 15:00:00'),
(6, 3, 6, '2024-08-06 16:10:00'),
(7, 4, 7, '2024-08-07 09:20:00'),
(8, 4, 8, '2024-08-08 10:30:00'),
(9, 5, 9, '2024-08-09 11:40:00'),
(10, 5, 10, '2024-08-10 12:50:00'),
(11, 6, 11, '2024-08-11 13:05:00'),
(12, 6, 12, '2024-08-12 14:15:00'),
(13, 7, 13, '2024-08-13 15:25:00'),
(14, 7, 14, '2024-08-14 16:35:00'),
(15, 8, 15, '2024-08-15 09:45:00'),
(16, 8, 16, '2024-08-16 11:00:00'),
(17, 9, 17, '2024-08-17 12:15:00'),
(18, 9, 18, '2024-08-18 13:30:00'),
(19, 10, 19, '2024-08-19 14:45:00'),
(20, 10, 20, '2024-08-20 15:55:00');
```

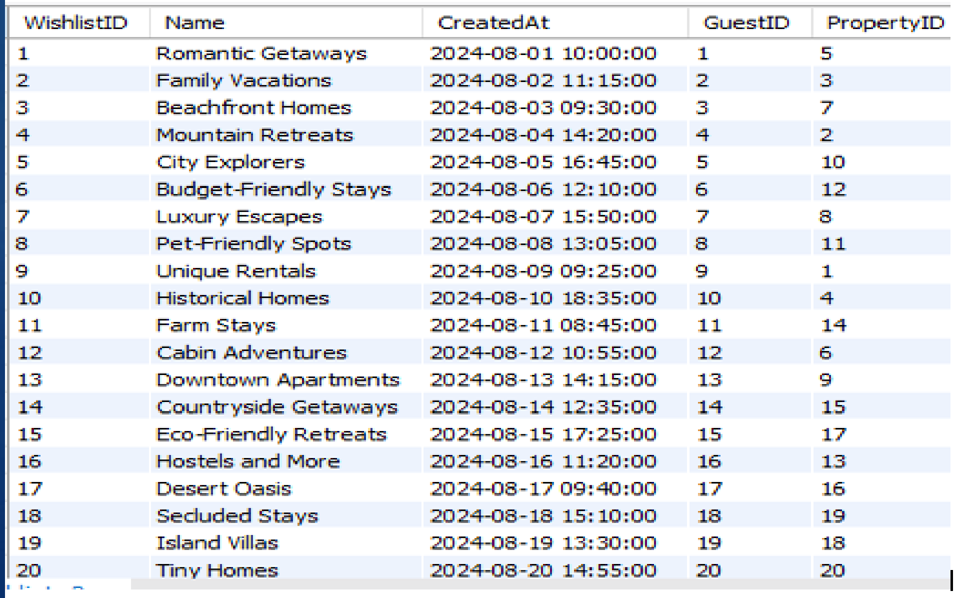
WishlistItemID	WishlistID	PropertyID	AddedAt
1	1	1	2024-08-01 10:00:00
2	1	2	2024-08-02 11:15:00
3	2	3	2024-08-03 12:30:00
4	2	4	2024-08-04 14:45:00
5	3	5	2024-08-05 15:00:00
6	3	6	2024-08-06 16:10:00
7	4	7	2024-08-07 09:20:00
8	4	8	2024-08-08 10:30:00
9	5	9	2024-08-09 11:40:00
10	5	10	2024-08-10 12:50:00
11	6	11	2024-08-11 13:05:00
12	6	12	2024-08-12 14:15:00
13	7	13	2024-08-13 15:25:00
14	7	14	2024-08-14 16:35:00
15	8	15	2024-08-15 09:45:00
16	8	16	2024-08-16 11:00:00
17	9	17	2024-08-17 12:15:00
18	9	18	2024-08-18 13:30:00
19	10	19	2024-08-19 14:45:00
20	10	20	2024-08-20 15:55:00

Wishlists

Stores wishlist details for guests. The SQL statement CREATE TABLE is used to define the structure of the of wishlists table.

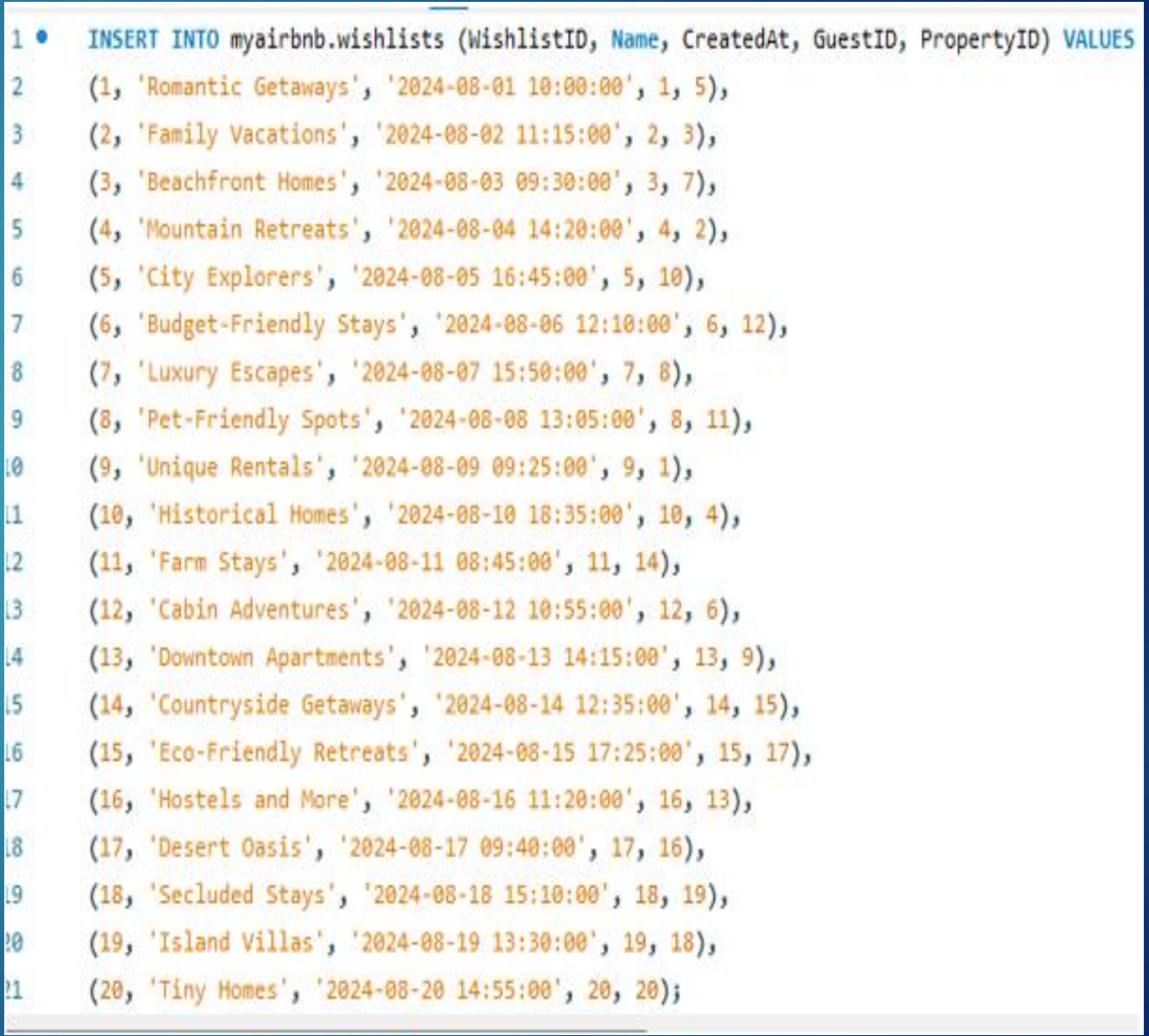


```
CREATE TABLE wishlists (
Open a script file in this editor T AUTO_INCREMENT PRIMARY KEY,
    Name VARCHAR(100),
    CreatedAt TIMESTAMP,
    GuestID INT,
    PropertyID INT,
    FOREIGN KEY (GuestID) REFERENCES guests(GuestID),
    FOREIGN KEY (PropertyID) REFERENCES property(PropertyID)
);
```



WishlistID	Name	CreatedAt	GuestID	PropertyID
1	Romantic Getaways	2024-08-01 10:00:00	1	5
2	Family Vacations	2024-08-02 11:15:00	2	3
3	Beachfront Homes	2024-08-03 09:30:00	3	7
4	Mountain Retreats	2024-08-04 14:20:00	4	2
5	City Explorers	2024-08-05 16:45:00	5	10
6	Budget-Friendly Stays	2024-08-06 12:10:00	6	12
7	Luxury Escapes	2024-08-07 15:50:00	7	8
8	Pet-Friendly Spots	2024-08-08 13:05:00	8	11
9	Unique Rentals	2024-08-09 09:25:00	9	1
10	Historical Homes	2024-08-10 18:35:00	10	4
11	Farm Stays	2024-08-11 08:45:00	11	14
12	Cabin Adventures	2024-08-12 10:55:00	12	6
13	Downtown Apartments	2024-08-13 14:15:00	13	9
14	Countryside Getaways	2024-08-14 12:35:00	14	15
15	Eco-Friendly Retreats	2024-08-15 17:25:00	15	17
16	Hostels and More	2024-08-16 11:20:00	16	13
17	Desert Oasis	2024-08-17 09:40:00	17	16
18	Secluded Stays	2024-08-18 15:10:00	18	19
19	Island Villas	2024-08-19 13:30:00	19	18
20	Tiny Homes	2024-08-20 14:55:00	20	20

Wishlist details can be inserted into the table using the INSERT INTO statement.



```
INSERT INTO myairbnb.wishlists (WishlistID, Name, CreatedAt, GuestID, PropertyID) VALUES
(1, 'Romantic Getaways', '2024-08-01 10:00:00', 1, 5),
(2, 'Family Vacations', '2024-08-02 11:15:00', 2, 3),
(3, 'Beachfront Homes', '2024-08-03 09:30:00', 3, 7),
(4, 'Mountain Retreats', '2024-08-04 14:20:00', 4, 2),
(5, 'City Explorers', '2024-08-05 16:45:00', 5, 10),
(6, 'Budget-Friendly Stays', '2024-08-06 12:10:00', 6, 12),
(7, 'Luxury Escapes', '2024-08-07 15:50:00', 7, 8),
(8, 'Pet-Friendly Spots', '2024-08-08 13:05:00', 8, 11),
(9, 'Unique Rentals', '2024-08-09 09:25:00', 9, 1),
(10, 'Historical Homes', '2024-08-10 18:35:00', 10, 4),
(11, 'Farm Stays', '2024-08-11 08:45:00', 11, 14),
(12, 'Cabin Adventures', '2024-08-12 10:55:00', 12, 6),
(13, 'Downtown Apartments', '2024-08-13 14:15:00', 13, 9),
(14, 'Countryside Getaways', '2024-08-14 12:35:00', 14, 15),
(15, 'Eco-Friendly Retreats', '2024-08-15 17:25:00', 15, 17),
(16, 'Hostels and More', '2024-08-16 11:20:00', 16, 13),
(17, 'Desert Oasis', '2024-08-17 09:40:00', 17, 16),
(18, 'Secluded Stays', '2024-08-18 15:10:00', 18, 19),
(19, 'Island Villas', '2024-08-19 13:30:00', 19, 18),
(20, 'Tiny Homes', '2024-08-20 14:55:00', 20, 20);
```

Summary

The aim of this phase is to develop a database management system tailored to the Airbnb use case. This involves creating a SQL data file that includes all the necessary SQL statements to establish the database schema, define table relationships, and populate the database with a minimum set of entries. The implementation is based on an Entity-Relationship (ER) Diagram that represents the conceptual data model for the Airbnb system, ensuring a solid structure that aligns with the application's functional requirements. By following these steps, a well-organized, effective, and reliable database management system is built for Airbnb, forming a strong basis for further application development and data handling.