Hotel Management System

By

Name	ID
Aisha Hamid Al-Anzi	421005345
Wijdan Musaad Al-Fayidi	391009823
Ghaida Naji Al-Fayidi	421004679
Rahaf Hamid Al-Anzi	411008796
Rana Muslim Al-Fayidi	421003934

Table Contents

Abstract

The hospitality and hotel management sector is considered one of the most complex and challenging industries in the business world. In addition to meeting guest needs, it is also crucial to efficiently manage internal hotel operations. To achieve this, an effective and reliable hotel management system is vital.

Our project aims to develop a comprehensive hotel management system that facilitates and organizes reservation processes, room management, attendance tracking, and payment control. The system will be designed to meet the needs of different hotels, whether small or large, and provide

advanced tools to improve operational efficiency and enhance the guest experience.

Entities and Attributes

1. Hotel:

   Entity Function: Represents hotel information such as its name and address.

   Attributes:

   - Hotel_ID (primary key): A unique identifier for each hotel.
   - Name: The name of the hotel.
   - Address: The address of the hotel.

   Relationships:

   - Each hotel can have multiple rooms associated with it through the foreign key relationship with the Room table.
   - It contains a foreign key (Staff_ID) that refers to the staff member in the Staff table.

2. Room:

   Entity Function: Represents room information within the hotel, such as room type, price, and availability.

   Attributes:

   - Room_ID (primary key): A unique identifier for each room.
   - Room_Type: The type or category of the room (e.g., standard, deluxe).
   - Price: The price of the room per night.
   - Availability: Indicates whether the room is currently available or not.
   - Hot_ID (foreign key referencing Hotel.Hotel_ID): The identifier of the hotel to which the room belongs.

   Relationships:

- Each room is associated with a hotel through a foreign key (Hot_ID) referencing the Hotel table.
- It can have multiple bookings associated with it through the foreign key relationship with the Booking table.

3. Booking:

   Entity Function: Represents the process of reserving a room in the hotel.

   Attributes:

   - Booking_ID (primary key): A unique identifier for each booking.
   - Checkindate: The date of check-in.
   - Checkoutdate: The date of check-out.
   - Payment_Status: Indicates the payment status of the booking.
   - Room_ID (foreign key referencing Room.Room_ID): The identifier of the booked room.
   - Client_ID (foreign key referencing Client.Client_ID): The identifier of the client who made the booking.

   Relationships:

   - Each booking is associated with a specific room through the foreign key (Room_ID) referencing the Room table.
   - It is also associated with a client through the foreign key (Client_ID) referencing the Client table.

4. Client:

   Entity Function: Represents the information of the client who reserves a room in the hotel.

   Attributes:

   - Client_ID (primary key): A unique identifier for each client.
   - Name: The name of the client.

- Sex: The gender of the client.
- PhoneNumber: The phone number of the client.

Relationships:

- Each client can have multiple bookings associated with them through the foreign key relationship with the Booking table.
- The Attendance and Cash tables have a foreign key (Client_ID) referencing the Client table, indicating the client's attendance and cash payments.

5. Attendance:

   Entity Function: Represents the registration of a client's attendance at the hotel on a specific date.

   Attributes:

   - Attend_ID (primary key): A unique identifier for each attendance record.
   - Date_book: The date of the attendance record.
   - Client_ID (foreign key referencing Client.Client_ID): The identifier of the attending client.

   Relationships:

   - Each attendance record is associated with a specific client through the foreign key (Client_ID) referencing the Client table.

6. Cash:

   Entity Function: Represents information about cash payments made by clients.

   Attributes:

   - Cash_ID (primary key): A unique identifier for each cash payment.
   - Pay_Price: The amount of the cash payment made by the client.
   - Client_ID (foreign key referencing Client.Client_ID): The identifier of the client who made the cash payment.

Relationships:

- Each cash payment is associated with a specific client through the foreign key (Client_ID) referencing the Client table.

7. Staff:

   Entity Function: Represents information about the staff members working at the hotel.

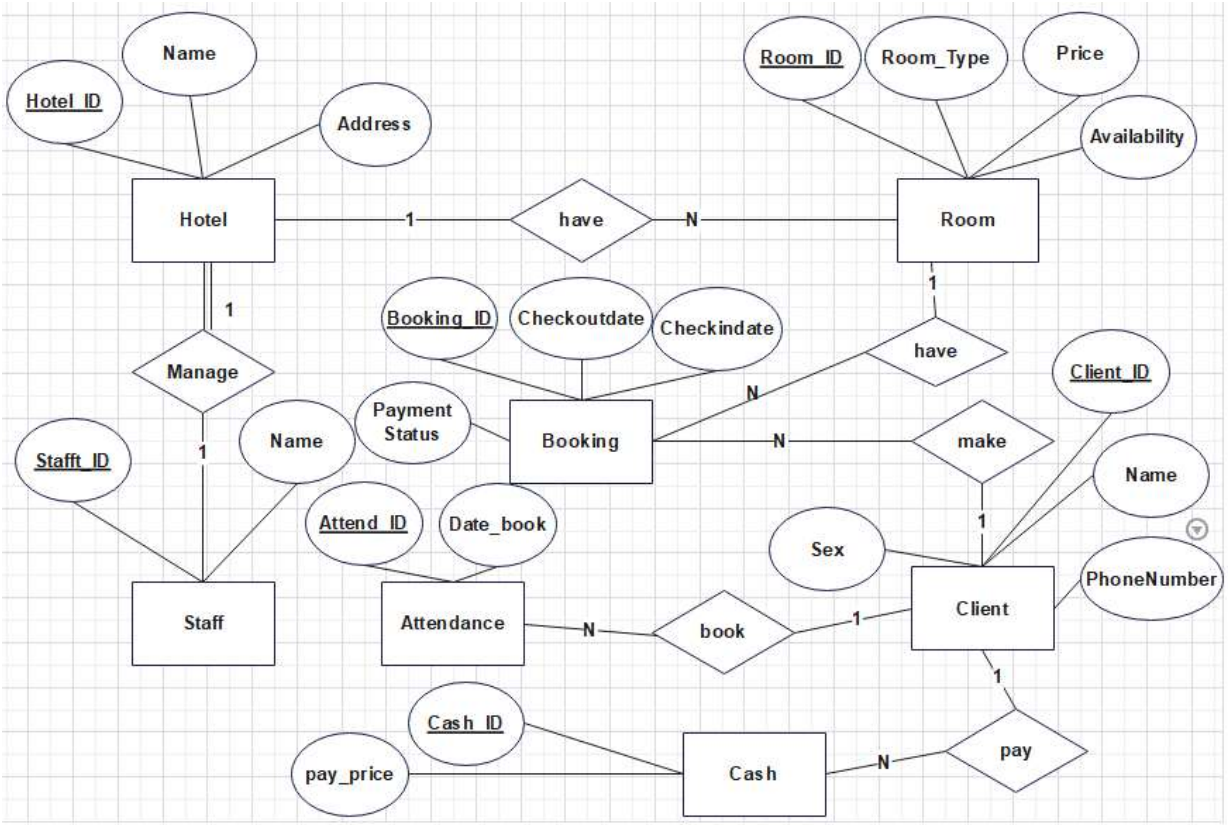
   Attributes:

   - Staff_ID (primary key): A unique identifier for each staff member.
   - Name: The name of the staff member.

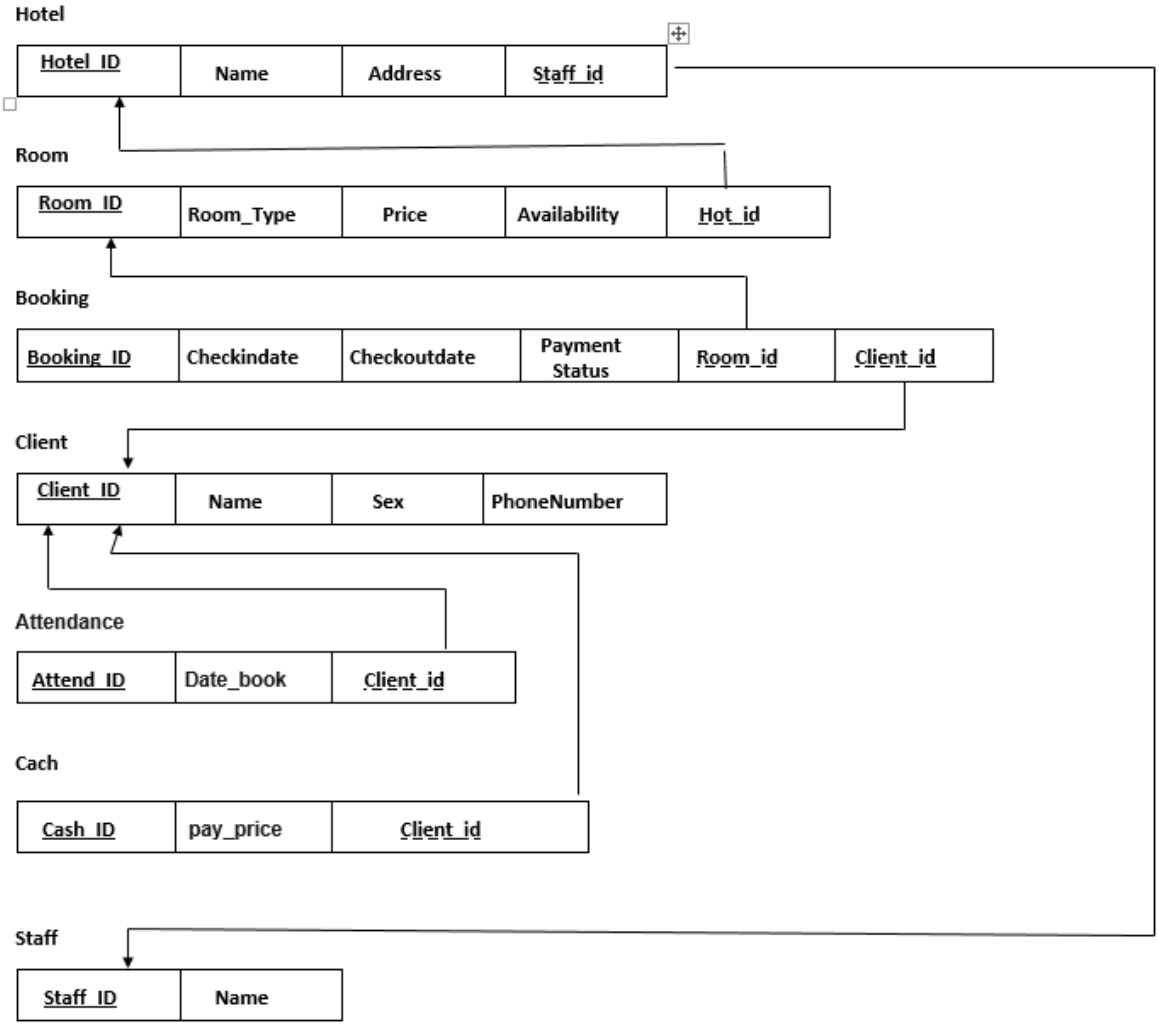
   Relationships:

   - Each staff member is associated with a specific hotel through the foreign key (Hotel_ID) referencing the Hotel table.

ER diagram:


Name
Hotel_ID
Address
Hotel
1
have
N
Room
Room_ID
Room_Type
Price
Availability
1
Manage
1
Booking_ID
Checkoutdate
Checkindate
have
N
Client_ID
Payment Status
Booking
N
make
Name
Stafft_ID
Name
1
Attend_ID
Date_book
Sex
1
Staff
Attendance
N
book
1
Client
PhoneNumber
1
Cash_ID
pay_price
Cash
N
pay

Database Schema(Mapping):



SQL Commands and Output:

Hotel(CREATE , INSERT ,SELECT):-

CREATE TABLE Hotel (Hotel_ID INT PRIMARY KEY,

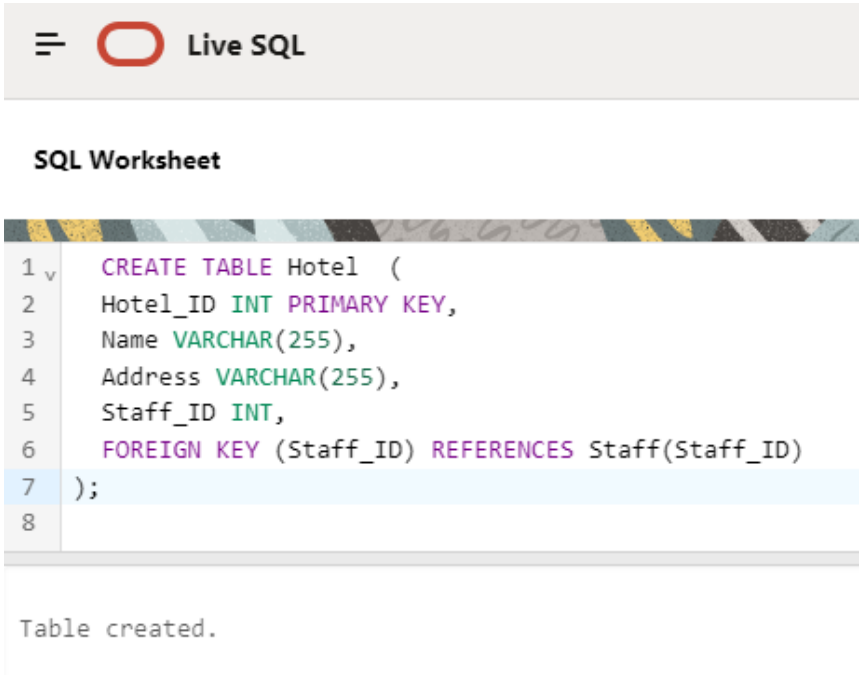
Name VARCHAR(255),

Address VARCHAR(255),

Staff_ID INT,

FOREIGN KEY (Staff_ID) REFERENCES Staff(Staff_ID)

);



Insert data into Hotel table --

INSERT INTO Hotel (Hotel_ID, Name, Address, Staff_ID)

VALUES (1, 'ABC Hotel', '123 Main Street, Riyadh', 1);

INSERT INTO Hotel (Hotel_ID, Name, Address, Staff_ID)

VALUES (2, 'Palm Hotel', 'King Fahd Street, Jeddah', 2);

INSERT INTO Hotel (Hotel_ID, Name, Address, Staff_ID)

VALUES (3, 'Address Hotel', 'Tahlia Street, Dammam', 3);

SQL Worksheet

```
1
2  INSERT INTO Hotel (Hotel_ID, Name, Address, Staff_ID)
3  VALUES (1, 'ABC Hotel', '123 Main Street, Riyadh', 1);
4
5  INSERT INTO Hotel (Hotel_ID, Name, Address, Staff_ID)
6  VALUES (2, 'Palm Hotel', 'King Fahd Street, Jeddah', 2);
7
8  INSERT INTO Hotel (Hotel_ID, Name, Address, Staff_ID)
9  VALUES (3, 'Address Hotel', 'Tahlia Street, Dammam', 3);
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

select * from Hotel;

SQL Worksheet

```
1  select * from Hotel;
```

HOTEL_ID	NAME	ADDRESS	STAFF_ID
1	ABC Hotel	123 Main Street, Riyadh	1
2	Palm Hotel	King Fahd Street, Jeddah	2
3	Address Hotel	Tahlia Street, Dammam	3

Download CSV

3 rows selected.

Room(CREATE , INSERT ,SELECT):

CREATE TABLE Room (

Room_ID INT PRIMARY KEY,

Room_Type VARCHAR(255),

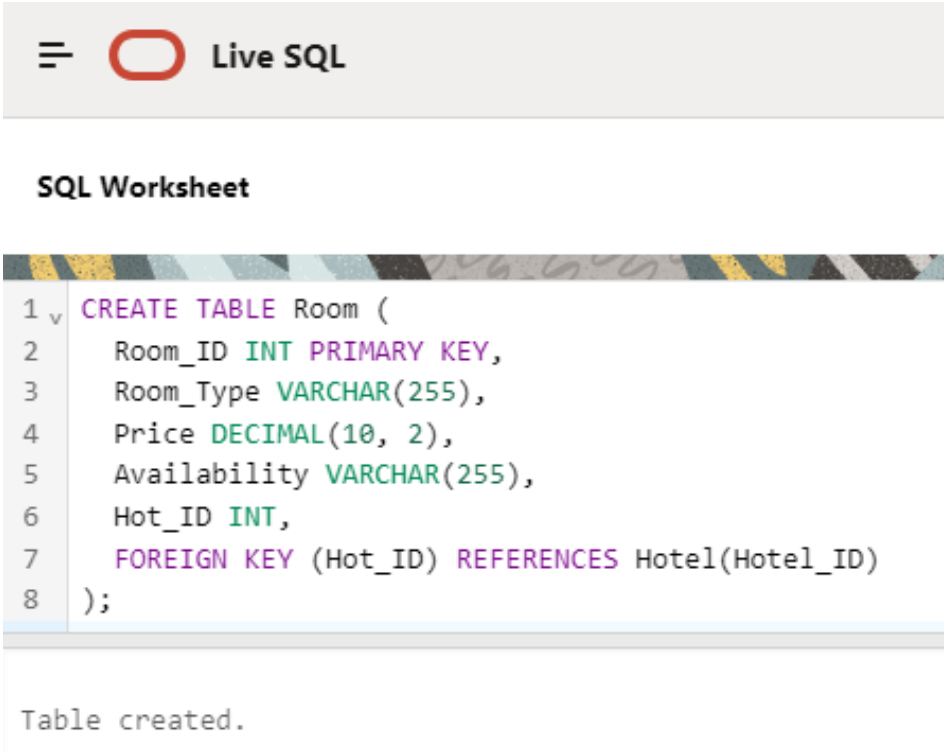
Price DECIMAL(10, 2),

Availability VARCHAR(255),

Hot_ID INT,

FOREIGN KEY (Hot_ID) REFERENCES Hotel(Hotel_ID)

;(



INSERT INTO Room (Room_ID, Room_Type, Price, Availability, Hot_ID) --

VALUES (1, 'Standard Room', 100.00, 'Available', 1);

INSERT INTO Room (Room_ID, Room_Type, Price, Availability, Hot_ID)

VALUES (2, 'Deluxe Room', 150.00, 'Available', 1);

INSERT INTO Room (Room_ID, Room_Type, Price, Availability, Hot_ID)

VALUES (3, 'Standard Room', 120.00, 'Not Available', 2);

INSERT INTO Room (Room_ID, Room_Type, Price, Availability, Hot_ID)

VALUES (4, 'Suite', 200.00, 'Available', 2);

SQL Worksheet

```
1  INSERT INTO Room (Room_ID, Room_Type, Price, Availability, Hot_ID)
2  VALUES (1, 'Standard Room', 100.00, 'Available', 1);
3
4  INSERT INTO Room (Room_ID, Room_Type, Price, Availability, Hot_ID)
5  VALUES (2, 'Deluxe Room', 150.00, 'Available', 1);
6
7  INSERT INTO Room (Room_ID, Room_Type, Price, Availability, Hot_ID)
8  VALUES (3, 'Standard Room', 120.00, 'Not Available', 2);
9
10 INSERT INTO Room (Room_ID, Room_Type, Price, Availability, Hot_ID)
11 VALUES (4, 'Suite', 200.00, 'Available', 2);
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

select * from Room;

SQL Worksheet

```
1  select * from Room;
```

ROOM_ID	ROOM_TYPE	PRICE	AVAILABILITY	HOT_ID
1	Standard Room	100	Available	1
2	Deluxe Room	150	Available	1
3	Standard Room	120	Not Available	2
4	Suite	200	Available	2

Download CSV

4 rows selected.

Booking(CREATE , INSERT ,SELECT):

CREATE TABLE Booking (

Booking_ID INT PRIMARY KEY,

Checkindate DATE,

Checkoutdate DATE,

Payment_Status VARCHAR(255),

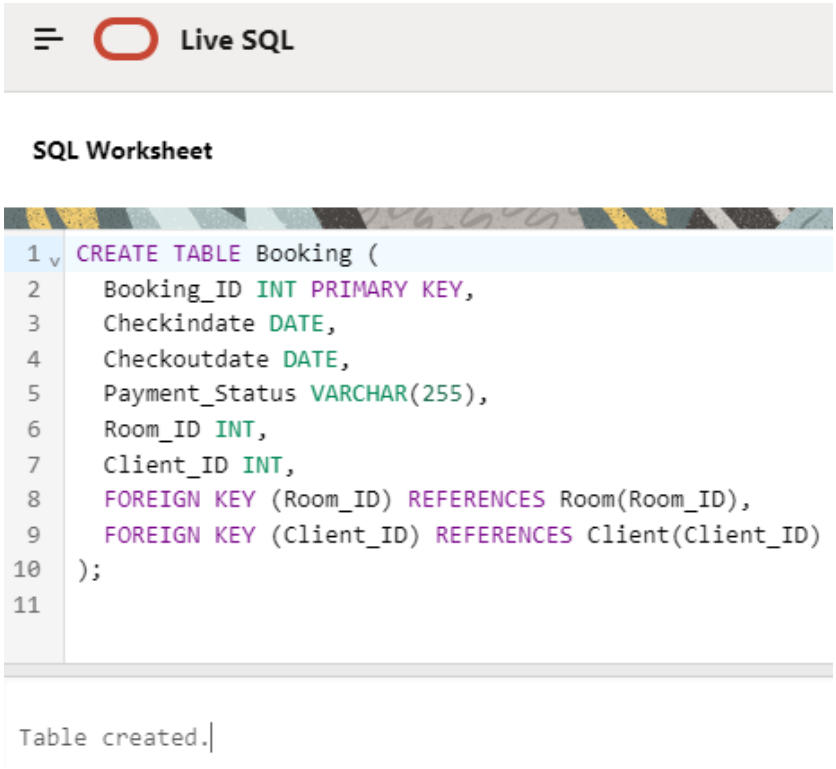
Room_ID INT,

Client_ID INT,

FOREIGN KEY (Room_ID) REFERENCES Room(Room_ID),

FOREIGN KEY (Client_ID) REFERENCES Client(Client_ID)

;(



INSERT INTO Booking (Booking_ID, Checkindate, Checkoutdate, Payment_Status, Room_ID, Client_ID)

VALUES (1, TO_DATE('2023-11-01', 'YYYY-MM-DD'), TO_DATE('2023-11-05', 'YYYY-MM-DD'), 'Paid', 1, 1);

INSERT INTO Booking (Booking_ID, Checkindate, Checkoutdate, Payment_Status, Room_ID, Client_ID)

VALUES (2, TO_DATE('2023-12-10', 'YYYY-MM-DD'), TO_DATE('2023-12-15', 'YYYY-MM-DD'), 'Pending', 2, 2);

INSERT INTO Booking (Booking_ID, Checkindate, Checkoutdate, Payment_Status, Room_ID, Client_ID)

VALUES (3, TO_DATE('2023-11-20', 'YYYY-MM-DD'), TO_DATE('2023-11-25', 'YYYY-MM-DD'), 'Paid', 3, 3);

SQL Worksheet

```
INSERT INTO Booking (Booking_ID, Checkindate, Checkoutdate, Payment_Status, Room_ID, Client_ID)
VALUES (1, TO_DATE('2023-11-01', 'YYYY-MM-DD'), TO_DATE('2023-11-05', 'YYYY-MM-DD'), 'Paid', 1, 1);

INSERT INTO Booking (Booking_ID, Checkindate, Checkoutdate, Payment_Status, Room_ID, Client_ID)
VALUES (2, TO_DATE('2023-12-10', 'YYYY-MM-DD'), TO_DATE('2023-12-15', 'YYYY-MM-DD'), 'Pending', 2, 2);

INSERT INTO Booking (Booking_ID, Checkindate, Checkoutdate, Payment_Status, Room_ID, Client_ID)
VALUES (3, TO_DATE('2023-11-20', 'YYYY-MM-DD'), TO_DATE('2023-11-25', 'YYYY-MM-DD'), 'Paid', 3, 3);
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

select * from Booking;

SQL Worksheet

```
select * from Booking;
```

BOOKING_ID	CHECKINDATE	CHECKOUTDATE	PAYMENT_STATUS	ROOM_ID	CLIENT_ID
1	01-NOV-23	05-NOV-23	Paid	1	1
2	10-DEC-23	15-DEC-23	Pending	2	2
3	20-NOV-23	25-NOV-23	Paid	3	3

Download CSV

3 rows selected.

Client(CREATE , INSERT ,SELECT):

CREATE TABLE Client (

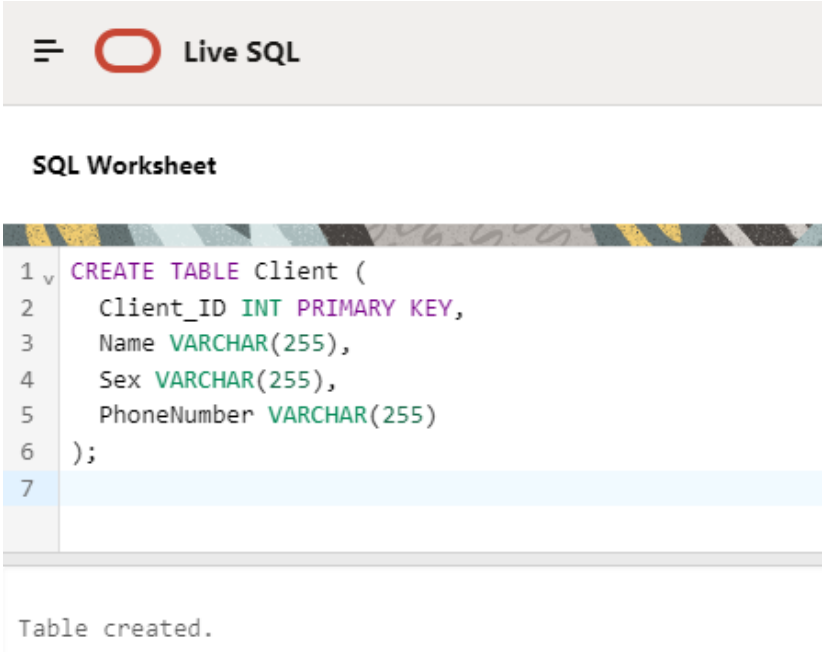
Client_ID INT PRIMARY KEY,

Name VARCHAR(255),

Sex VARCHAR(255),

PhoneNumber VARCHAR(255)

;(



INSERT INTO Client (Client_ID, Name, Sex, PhoneNumber)

VALUES (1, 'Ahmed Mohammed', 'Male', '123-456-7890');

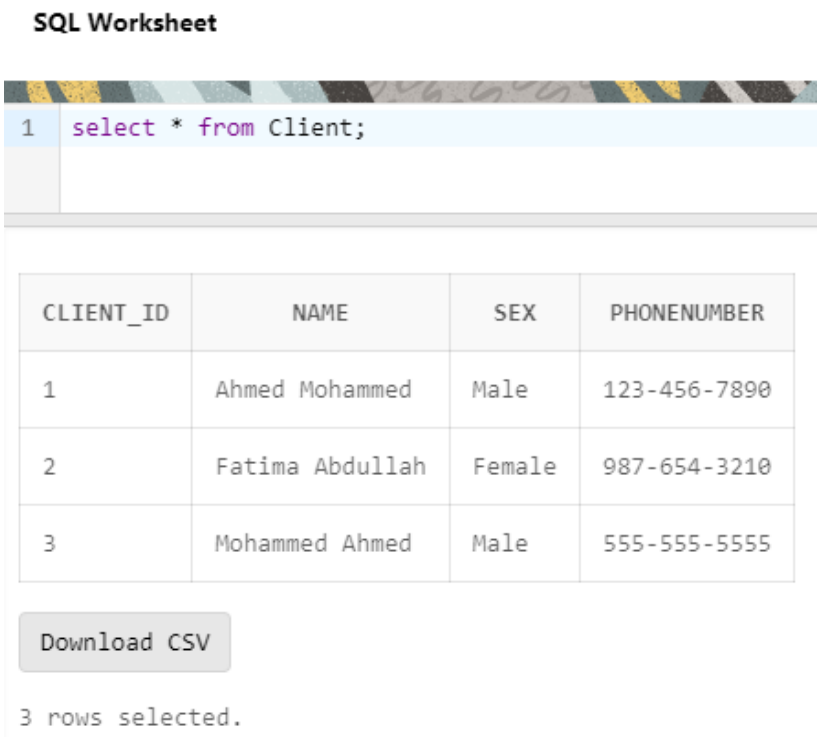
INSERT INTO Client (Client_ID, Name, Sex, PhoneNumber)

VALUES (2, 'Fatima Abdullah', 'Female', '987-654-3210');

INSERT INTO Client (Client_ID, Name, Sex, PhoneNumber)

VALUES (3, 'Mohammed Ahmed', 'Male', '555-555-5555');

select * from Client;



Attendance (CREATE , INSERT ,SELECT):

CREATE TABLE Attendance (

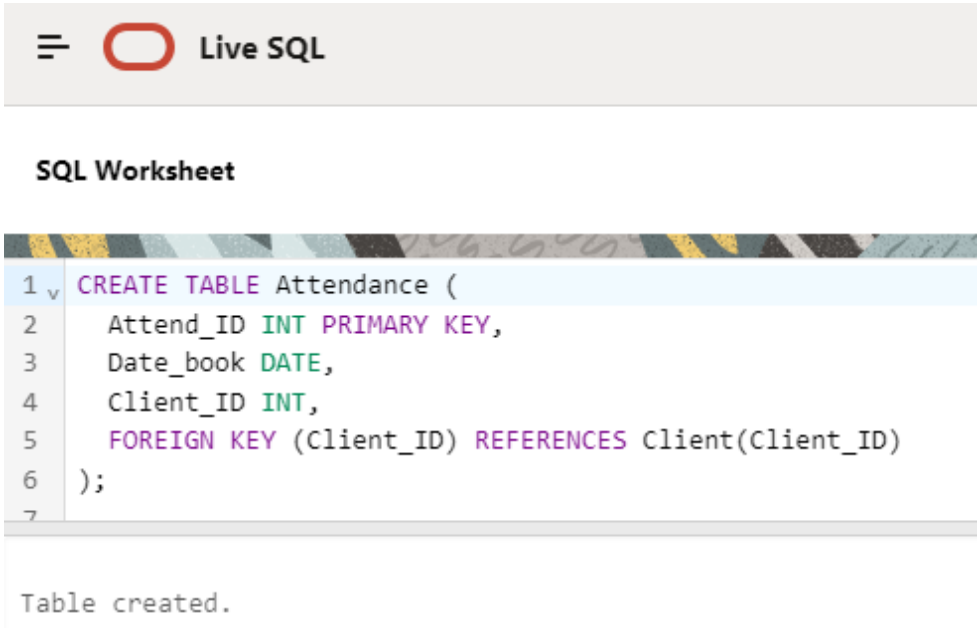
Attend_ID INT PRIMARY KEY,

Date_book DATE,

Client_ID INT,

FOREIGN KEY (Client_ID) REFERENCES Client(Client_ID)

;(



INSERT INTO Attendance (Attend_ID, Date_book, Client_ID) --

VALUES (1, TO_DATE('2023-11-01', 'YYYY-MM-DD'), 1);

INSERT INTO Attendance (Attend_ID, Date_book, Client_ID)

VALUES (2, TO_DATE('2023-12-10', 'YYYY-MM-DD'), 2);

INSERT INTO Attendance (Attend_ID, Date_book, Client_ID)

VALUES (3, TO_DATE('2023-11-20', 'YYYY-MM-DD'), 3);

Live SQL

SQL Worksheet

```
1  INSERT INTO Attendance (Attend_ID, Date_book, Client_ID)
2  VALUES (1, TO_DATE('2023-11-01', 'YYYY-MM-DD'), 1);
3
4  INSERT INTO Attendance (Attend_ID, Date_book, Client_ID)
5  VALUES (2, TO_DATE('2023-12-10', 'YYYY-MM-DD'), 2);
6
7  INSERT INTO Attendance (Attend_ID, Date_book, Client_ID)
8  VALUES (3, TO_DATE('2023-11-20', 'YYYY-MM-DD'), 3);
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

select * from Attendance ;

Cash (CREATE , INSERT ,SELECT):

CREATE TABLE Cash (

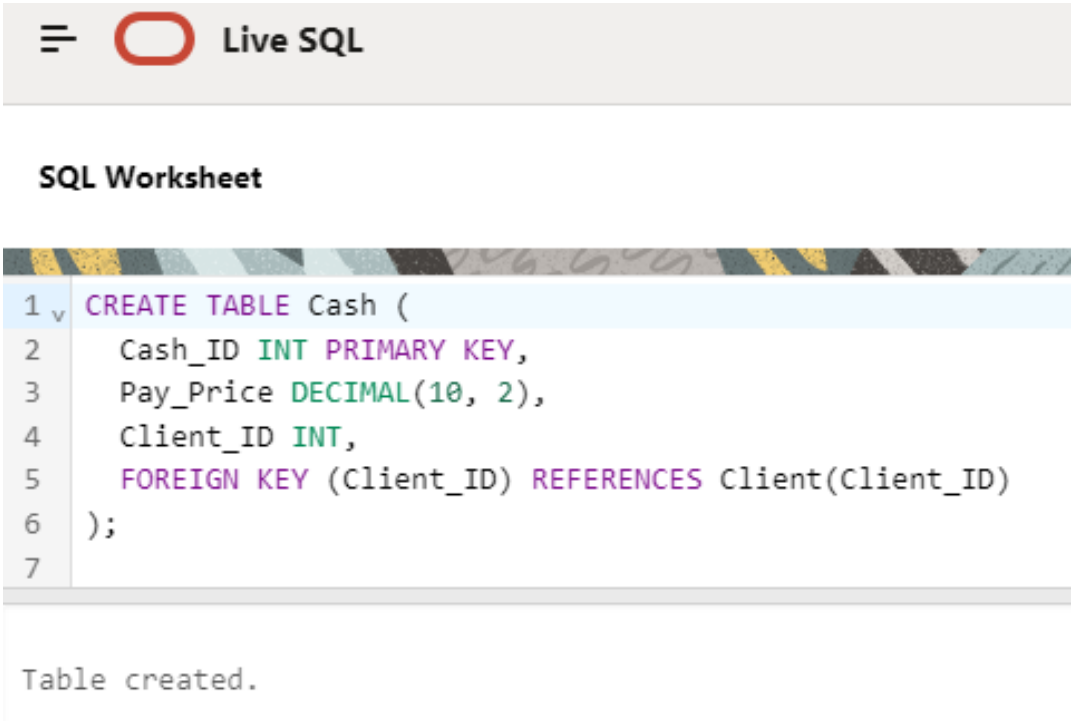
Cash_ID INT PRIMARY KEY,

Pay_Price DECIMAL(10, 2),

Client_ID INT,

FOREIGN KEY (Client_ID) REFERENCES Client(Client_ID)

;(



INSERT INTO Cash (Cash_ID, Pay_Price, Client_ID)

VALUES (1, 100.00, 1);

INSERT INTO Cash (Cash_ID, Pay_Price, Client_ID)

VALUES (2, 200.00, 2);

INSERT INTO Cash (Cash_ID, Pay_Price, Client_ID)

VALUES (3, 150.00, 3);

SQL Worksheet

```
1  INSERT INTO Cash (Cash_ID, Pay_Price, Client_ID)
2  VALUES (1, 100.00, 1);
3
4  INSERT INTO Cash (Cash_ID, Pay_Price, Client_ID)
5  VALUES (2, 200.00, 2);
6
7  INSERT INTO Cash (Cash_ID, Pay_Price, Client_ID)
8  VALUES (3, 150.00, 3);
```

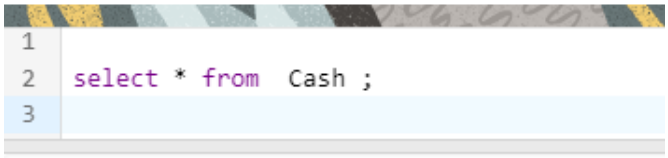
```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

select * from Cash ;

SQL Worksheet



```
1
2  select * from  Cash ;
3
```

CASH_ID	PAY_PRICE	CLIENT_ID
1	100	1
2	200	2
3	150	3

Download CSV

3 rows selected.

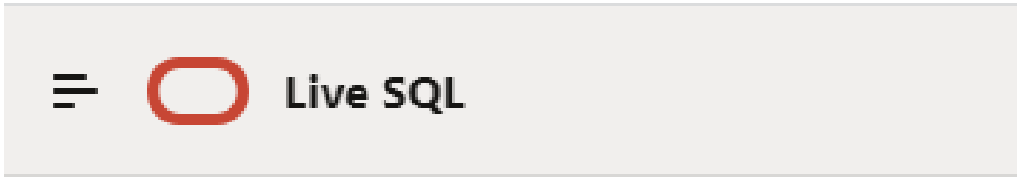
Staff (CREATE , INSERT ,SELECT):

CREATE TABLE Staff (

Staff_ID INT PRIMARY KEY,

Name VARCHAR(255)

;(



SQL Worksheet

```sql
-- Create Staff table
CREATE TABLE Staff (
  Staff_ID INT PRIMARY KEY,
  Name VARCHAR(255)
);
```

```
Table created.
```

Insert data into Staff table --

INSERT INTO Staff (Staff_ID, Name)

VALUES (1, 'Abdulrahman Khalid');

INSERT INTO Staff (Staff_ID, Name)

VALUES (2, 'Nora Mohammed');

INSERT INTO Staff (Staff_ID, Name)

VALUES (3, 'Ali Ahmed');

SQL Worksheet

```
1  INSERT INTO Staff (Staff_ID, Name)
2  VALUES (1, 'Abdulrahman Khalid');
3
4  INSERT INTO Staff (Staff_ID, Name)
5  VALUES (2, 'Nora Mohammed');
6
7  INSERT INTO Staff (Staff_ID, Name)
8  VALUES (3, 'Ali Ahmed');
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

select * from Staff ;

SQL Worksheet

```
1  select * from  Staff ;
```

STAFF_ID	NAME
1	Abdulrahman Khalid
2	Nora Mohammed
3	Ali Ahmed

Download CSV

3 rows selected.

Normalization:

1NF:

The original design already satisfies 1NF as each table has a unique primary key and there are no repeating groups.

2NF:

Booking Table: Identify the partial dependency. The Payment_Status attribute is dependent on the Room_ID but not the entire primary key (Room_ID, Client_ID).

Solution: Create a separate table called Room_Rates to store room-specific information, including Room_ID, Price, and Availability. Move the Price and Availability attributes from the Room table to the Room_Rates table. Update the Booking table to reference the Room_Rates table for Price and Availability instead of the Room table directly.

Original Booking Table

Column Name	Data Type	Description
Booking_ID	INT	Unique identifier for each booking (Primary Key)
Checkindate	DATE	Date of check-in
Checkoutdate	DATE	Date of check-out
Payment_Status	VARCHAR(255)	Indicates the payment status of the booking
Room_ID	INT	Foreign key referencing the Room table (identifies the room booked)
Client_ID	INT	Foreign key referencing the Client table (identifies the client who made the booking)
Price	DECIMAL(10, 2)	Price of the room per night (Partially Dependent on Room_ID)
Availability	VARCHAR(255)	Indicates whether the room is available or not (Partially Dependent on Room_ID)

Solution Table - Room_Rates:

Column Name	Data Type	Description
Room_ID	INT	Unique identifier for each room (Primary Key)
Price	DECIMAL(10, 2)	Price of the room per night
Availability	VARCHAR(255)	Indicates whether the room is available or not

Updated Booking Table (After Normalization):

Column Name	Data Type	Description
Booking_ID	INT	Unique identifier for each booking (Primary Key)
Checkindate	DATE	Date of check-in
Checkoutdate	DATE	Date of check-out
Payment_Status	VARCHAR(255)	Indicates the payment status of the booking
Room_ID	INT	Foreign key referencing the Room table (identifies the room booked)
Client_ID	INT	Foreign key referencing the Client table (identifies the client who made the booking)
Room_Rate_ID	INT	Foreign key referencing the Room_Rates table (identifies the room rate associated with the booking)

3NF:

- Room Table: Identify the transitive dependency. The Hotel_ID attribute is used indirectly to determine the Address of a room through the Hotel table.
- Solution: Add the Hotel_Address attribute to the Room table to directly store the address of the hotel associated with the room. This eliminates the transitive dependency on the Hotel table.

Original Room Table

Column Name	Data Type	Description
Room_ID	INT	Unique identifier for each room (Primary Key)
Room_Type	VARCHAR(255)	Type of room (e.g., standard, deluxe)
Hotel_ID	INT	Foreign key referencing the Hotel table (identifies the hotel the room belongs to)
Price	DECIMAL(10, 2)	Price of the room per night (Already moved to Room_Rates table for 2NF)
Availability	VARCHAR(255)	Indicates whether the room is available or not (Already moved to Room_Rates table for 2NF)

Hotel_ID has a transitive dependency on the address because to find the address of a room, you need to first find the hotel using the Hotel_ID and then access the address from the Hotel table.

Solution Table - Updated Room:

Column Name	Data Type	Description
Room_ID	INT	Unique identifier for each room (Primary Key)
Room_Type	VARCHAR(255)	Type of room (e.g., standard, deluxe)
Hotel_ID	INT	Foreign key referencing the Hotel table (identifies the hotel the room belongs to)
Hotel_Address	VARCHAR(255)	Address of the hotel associated with the room