UNIVERSITETI I EVROPËS JUGLINDORE
УНИВЕРЗИТЕТ НА ЈУГОИСТОЧНА ЕВРОПА
SOUTH EAST EUROPEAN UNIVERSITY

Faculty of Contemporary Science and Technologies

Tetovo

Database

(4th Sem. 2023)

Project:

Hotel Database Management System

Student:

Dilnoza Yodalieva

Mentor:

Prof. Dr.Jaumin Ajdari

May 2023, Tetovo

Abstract.

The purpose of this project is to create a comprehensive system that can assist hotels in managing their day-to-day operations. The system will be built using MySQL, Oracle, HTML, CSS, and Bootstrap technologies, and will utilise a robust database to store and organise hotel-related data, such as guest information, room inventory, reservations, financial transactions, reports, etc.

The Hotel Management System will cover several essential aspects of hotel management, including guest management, room management, reservation, etc. Throughout the development process, there may be a need to modify or remove certain aspects of the system, depending on the further progress. Overall, the Hotel Database Management System will be a valuable tool for hotels seeking to streamline their operations, improve their efficiency, and enhance the overall guest experience.

Key words: Hotel Management System, Database, Oracle, MySQL, HTML, CSS (or Ms Access).

Table of Components

1. Problem Description

Hotels face numerous operational challenges and managing them manually can be a tedious and error-prone task. One of the biggest challenges hotels faces is managing room inventory efficiently. Hotels must balance the number of rooms available to guests with the number of rooms that need to be stored for free for maintenance and cleaning.

Without an automated system, hotel staff would have to manually update room inventory whenever a room becomes available or runs out. This can be a time-consuming task, especially during high season when rooms are in high demand. Additionally, manual updates can lead to errors and discrepancies in room inventory, resulting in overbooking and double-booking.

Overbooking can lead to guest dissatisfaction, damage a hotels reputation, and lead to lost revenue. Double-booking can create confusion and inconvenience for both hotel staff and guests. A hotel management system can solve this problem by automating the process of managing room inventory.

The system keeps an up-to-date record of room availability and ensures that rooms are not double-booked or overbooked. This allows hotels to optimise their room inventory and ensure that they have enough rooms to meet guest demand without having too many rooms unnecessarily vacant.

By implementing a hotel management system, hotels can streamline their room inventory management process, reduce errors, and improve guest satisfaction, ultimately leading to increased revenue and a better overall guest experience. Increase.

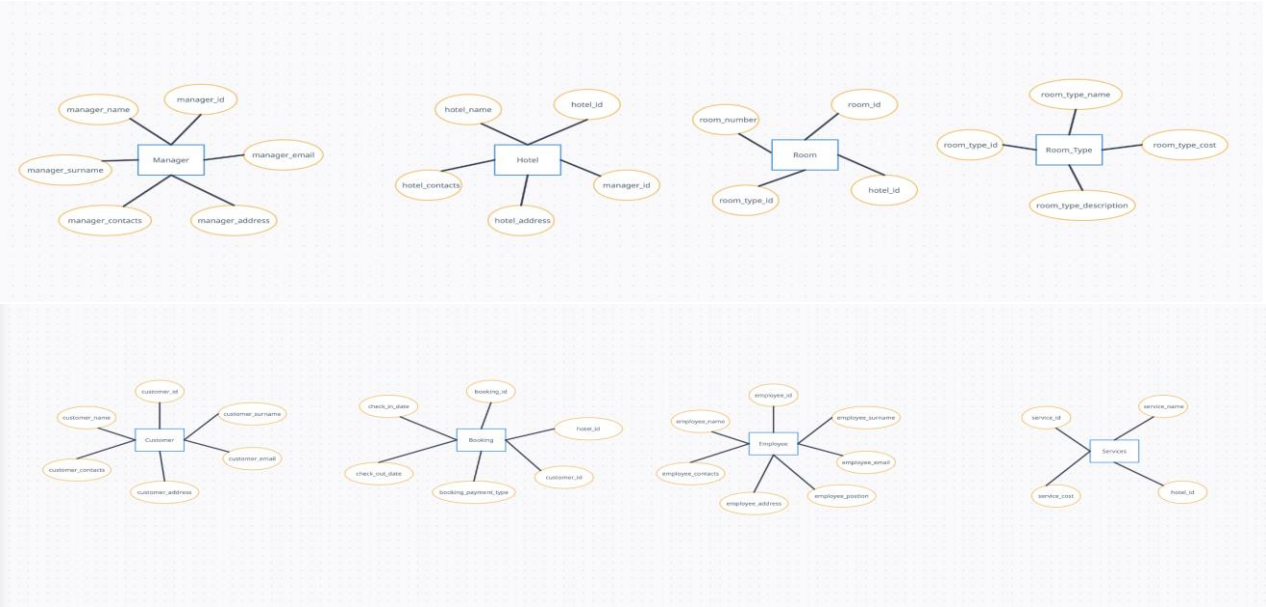
2. **Business Rules**

- One Manager can have manager one hotel
- Hotel offers multiple services, and these services are done in one hotel
- Hotel can have many employees working there
- Many employees can work in one hotel
- Employes prepare the room and goods for the customer
- Hotel can have many bookings done by customers
- The customer makes many bookings, but a certain room can be booked by one customer.
- Booking will contain the exact date and time

3. **Conceptual Design**

Initilay I started with finding the entity and attribute and draw the entity and relationship diagram (E-R) diagram for the Hotel Database Management System. I break down the solution into few steps.

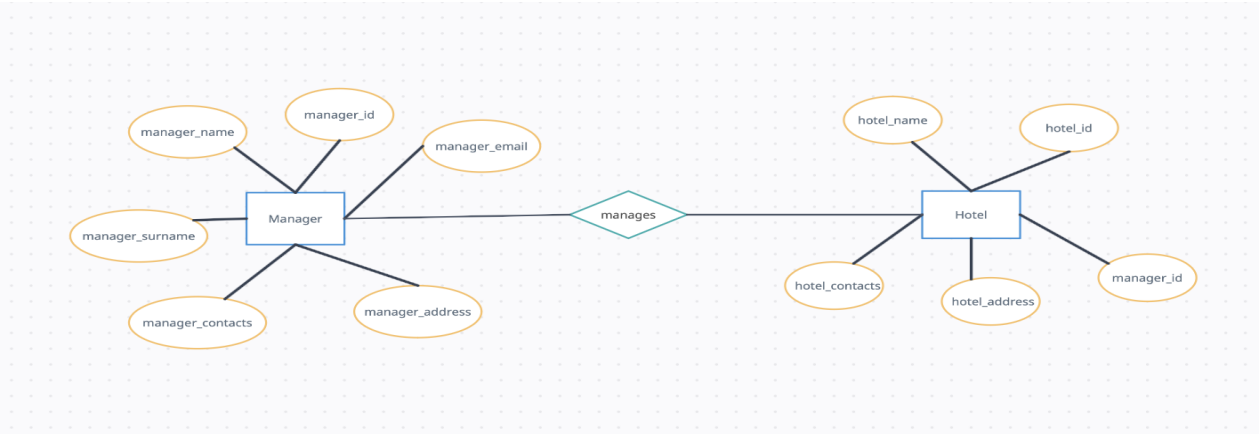
Step 1: I found the entity type and for each entity type I found the attributes.



This is how I found the entity type and attribute for each entity type. I hope everything is perfectly clear in the photo.

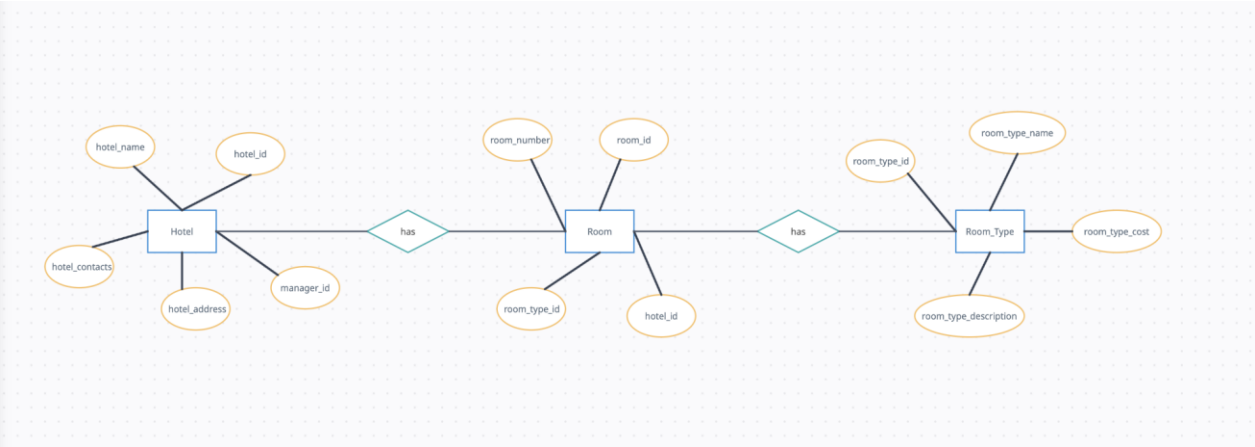
Step 2: After that I found the relationship for each entity and connection between them.

First, I found relationship between Manager and Hotel. The relationship will be 1:1, because one manager can only manage one hotel, and a hotel can be managed just by one person.

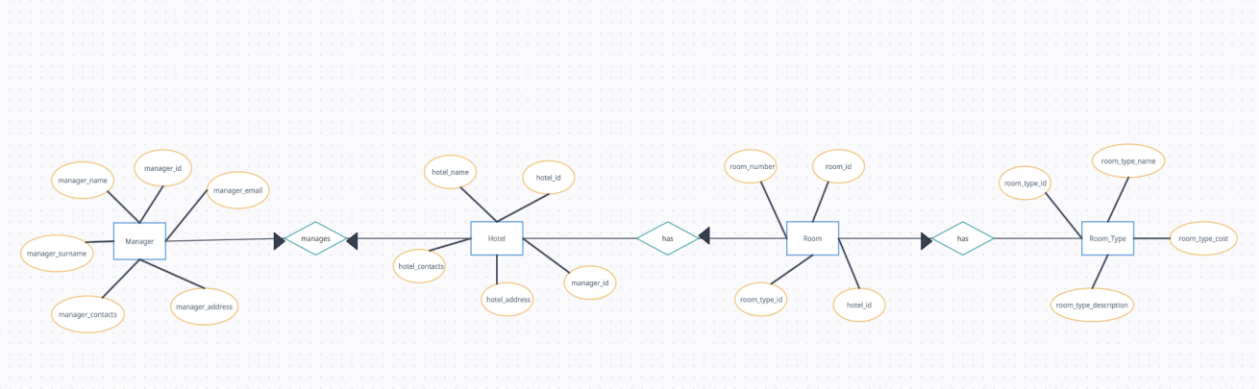


Relation between Manager and Hotel (1: 1)

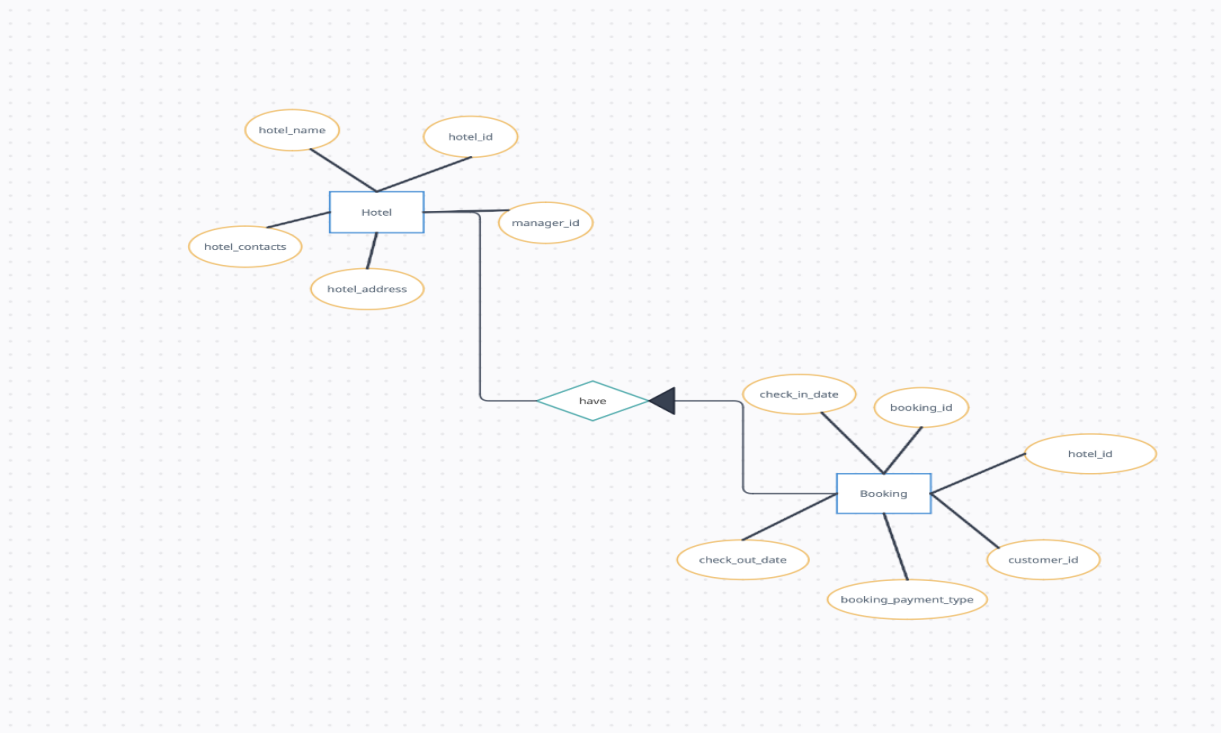
Then we can see that hotel has many rooms inside it. And each room has its own type of room. In case, there will be relationships between Entity Hotel, Room and Room_Type. Relationship between Entity and Room will be 1:M, because one hotel can have many rooms and each room is inside one hotel. Relation between Room and Room_Type will be 1:1, because each room have only one type of room assign to it.



Relation between Hotel and Room (1:M), Room and Room_Type (M: 1)

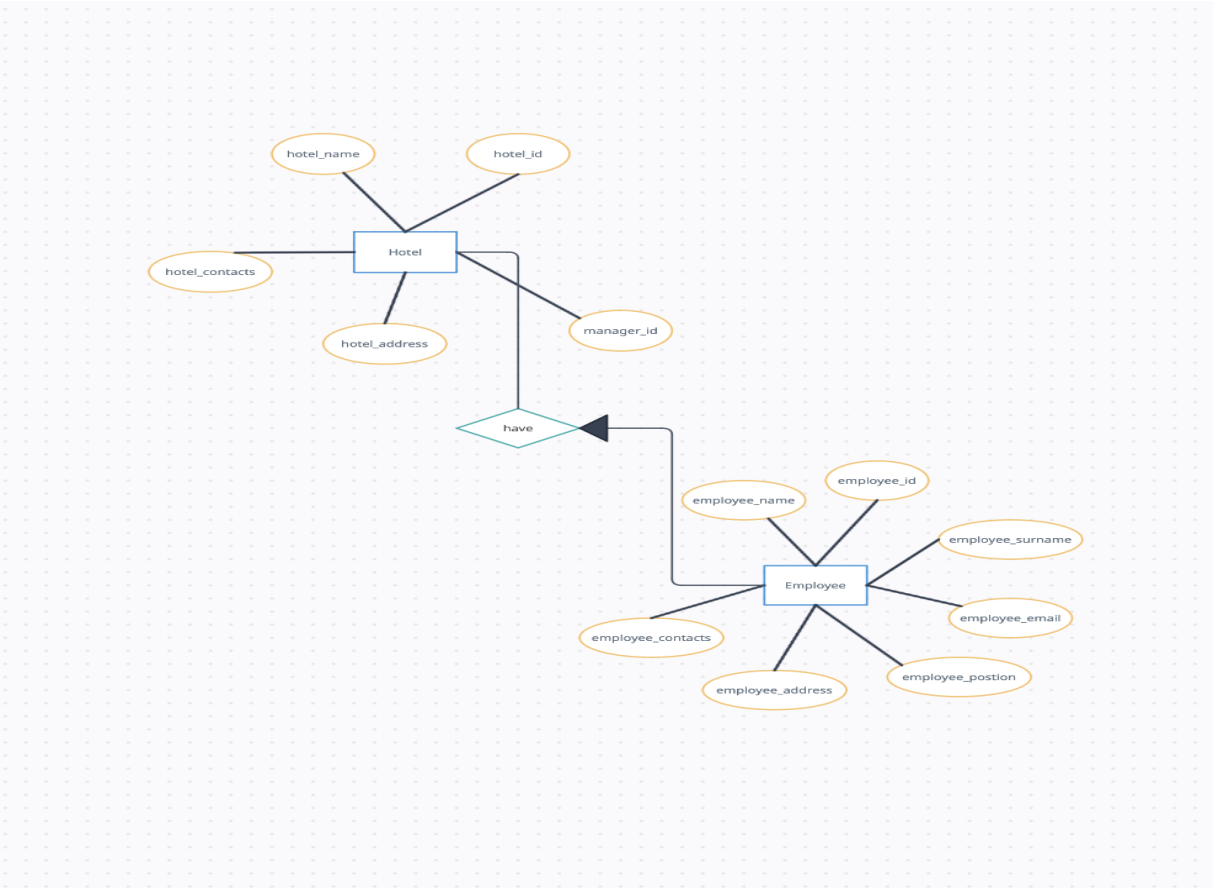


Next step will be the relation between Hotel and Booking entity (1:M). This means that one hotel can have many bookings associated with it, but each booking is associated with only one hotel.



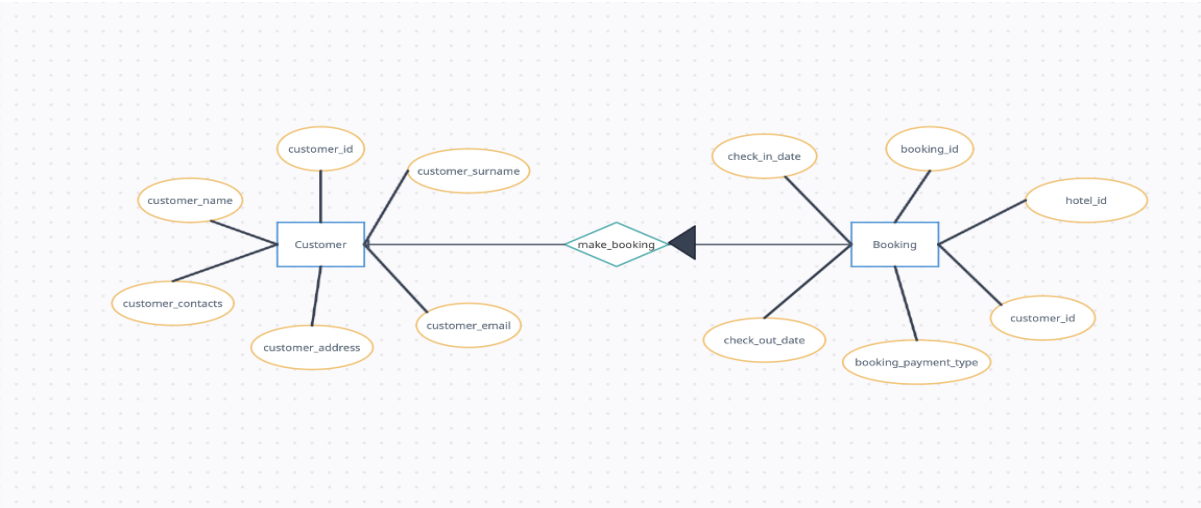
Relation between Hotel and Booking (1:M)

After that I found the relation between Hotel and Employee (1:M). This means that one hotel can have many employees working for it, but each employee is associated with only one hotel.



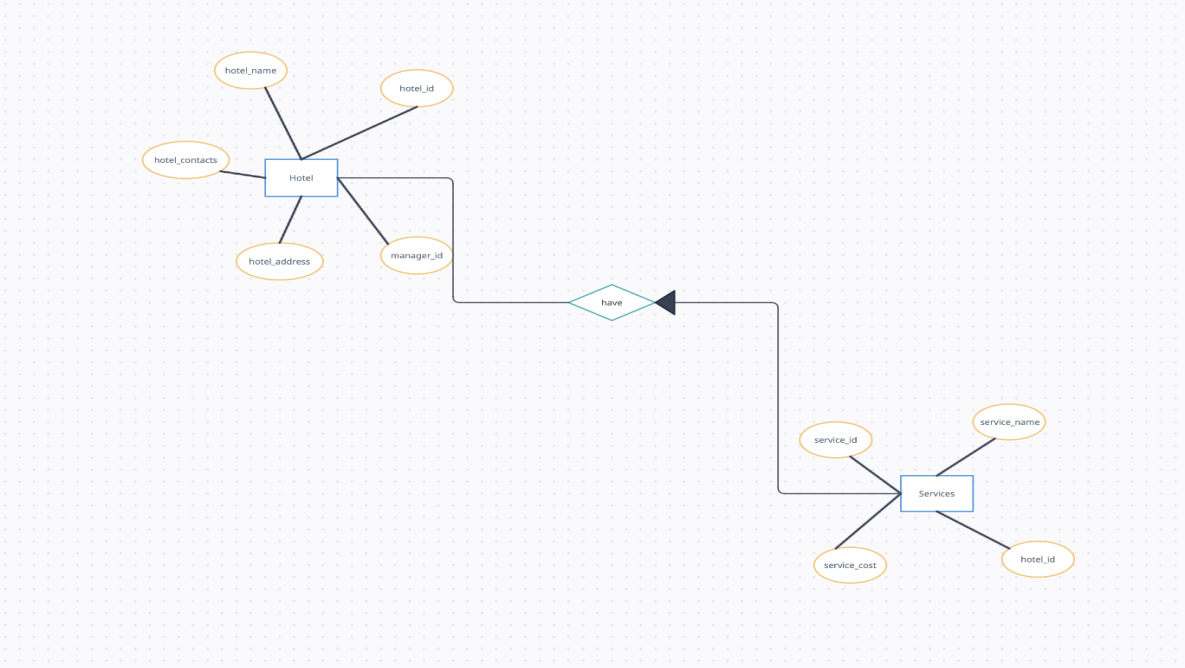
Relation between Hotel and Employee (1: M)

Other relation which I found was between Customer and Booking (1:M), meaning each customer can make many bookings, but each booking is associated with only one customer. I think this makes sense because a customer can make multiple bookings for different rooms and dates, but each booking is specific to a particular customer.



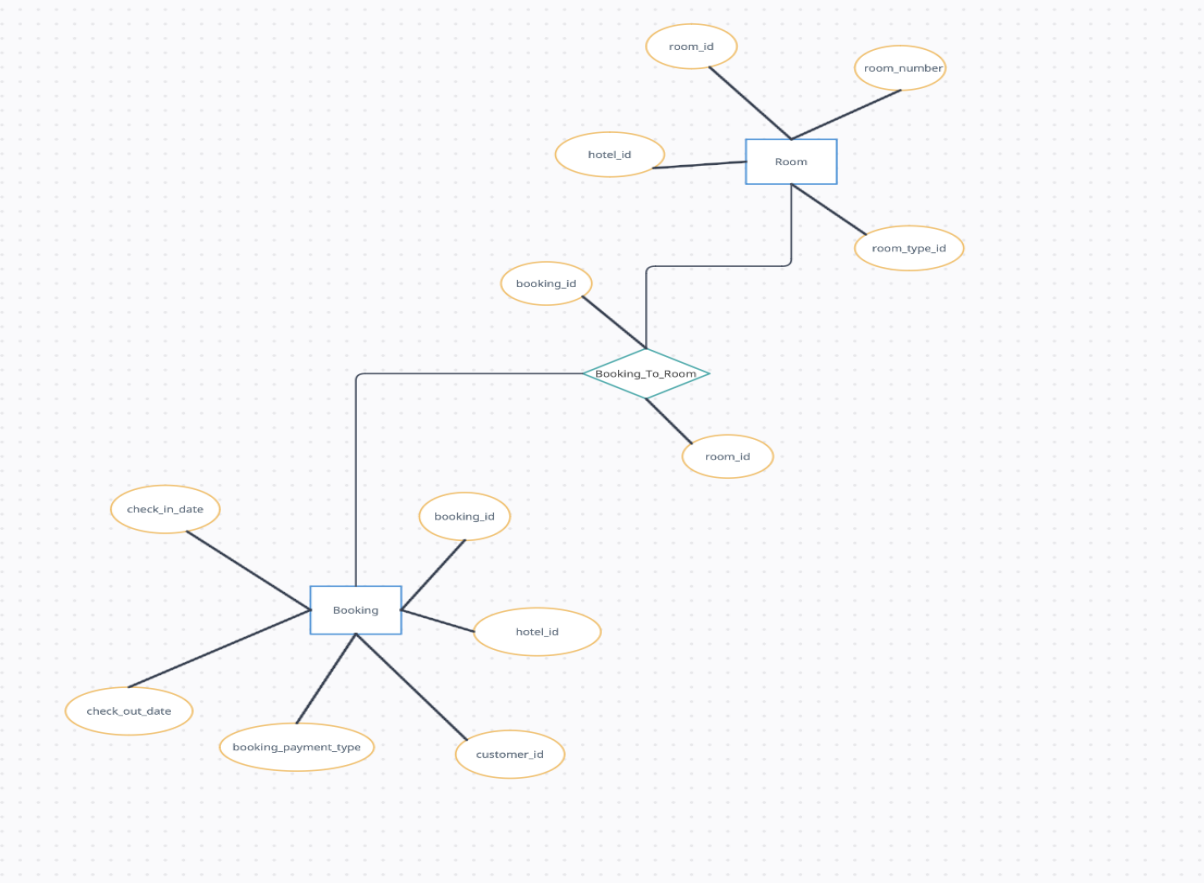
Relation between Customer and Booking (1:M)

The other relation is between Hotel and Service(1:M). Hotel can offer multiple services, but each service is specific to a particular hotel.



Relation between Hotel and Service (1:M)

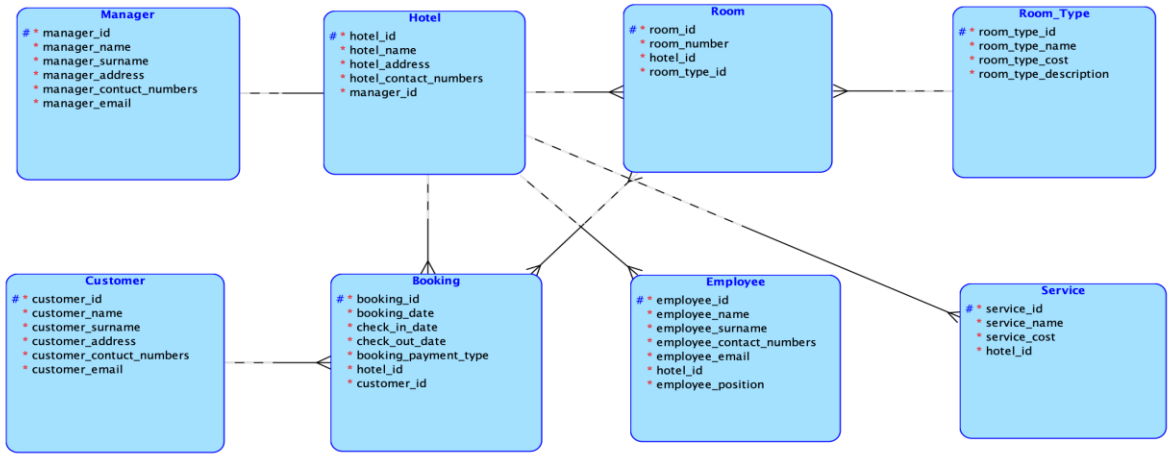
The last relationship is between Booking and Room (M: M), as it is many to many relationships, we will create one more table for that.



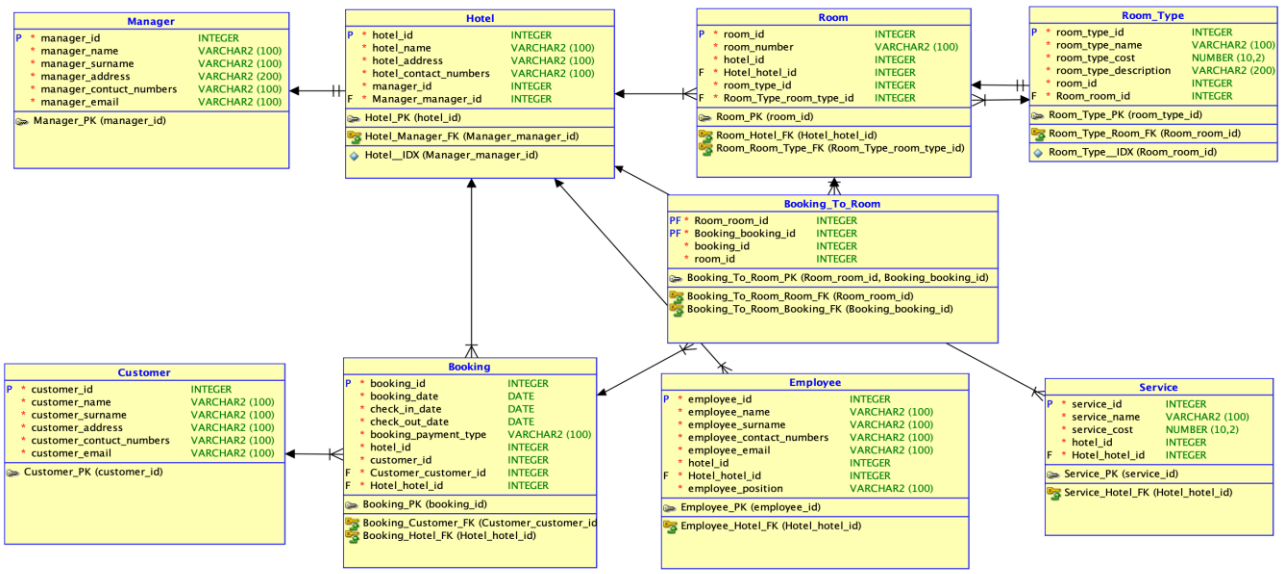
Relation between Booking and Room (M:M)

Solution by use of Oracle SQL Developer

- Conceptual Modelling (E – R Diagram, Logical Design)



- Relational Model

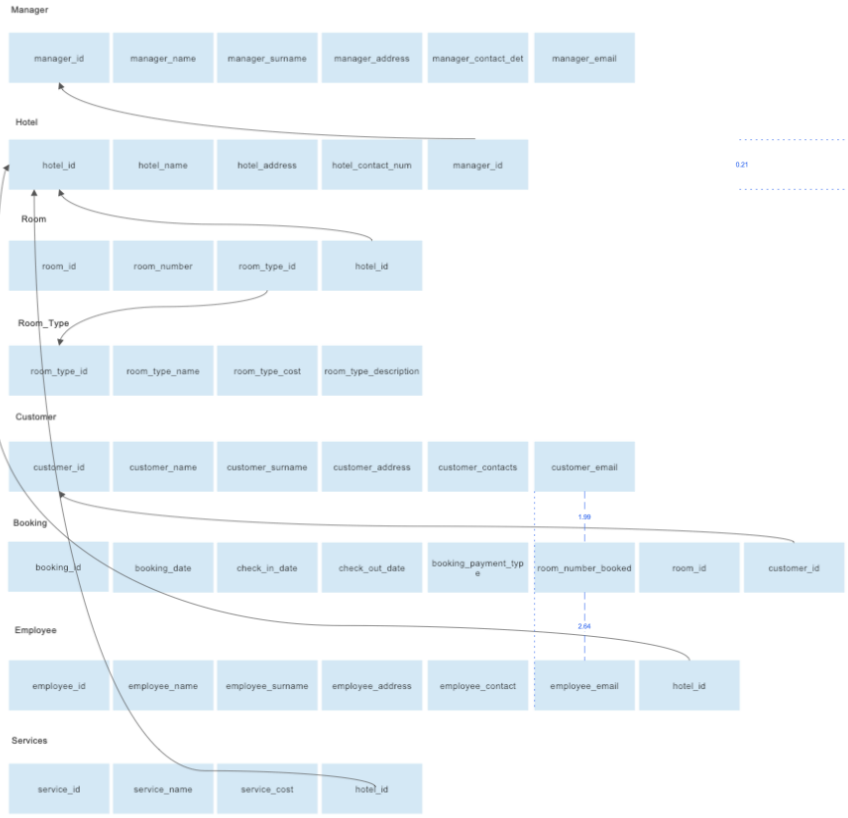


4. **Relational Design**

Using the relational mapping rules we convert the E-R Diagram to relational model (relational diagram). Here is how I solved this.

- Entity types, mapping of the entity type

)



5.**Schema normalization**

According to above relational diagram the entities with relations are:

Manager (manager_id, manager_name, manager_surname, manager_address, manager_cont, manager_email)

Hotel (hotel_id, hotel_name, hotel_address, hotel_cont, manager_id)

Room (room_id, room_number, room_type_id, hotel_id , room_type_id)

Room_Type (room_type_id, room_type_name, room_type_cost, room_type_description)

Customer (customer_id, customer_name, customer_surname, customer_email, customer_address, customer_cont)

Booking (booking_id, booking_date, check_in_date, check_out_date, booking_payment_type, room_number_booked, room_id, customer_id , hotel_id)

Employee (employee_id, employee_name, employee_surname, employee_address, employee_cont, employee_email, hotel_id)

Services (service_id, service_name, service_cost, hotel_id)

6. **SQL Code and Implementation**

```
--Database creation

create database Hotel_Management_Studio
```

```
--Table creation

create table Hotel
(
    --finding the attributes for this table

    hotel_id int primary key not null,
    hotel_name varchar(100) not null,
    hotel_contact_number varchar(100),
    hotel_address varchar(100) not null,
    manager_id int not null,


);
go


create table Manager
(
    --finding the attributes and setting them to not null
    --setting primary key manager_id
    manager_id int primary key not null,
    manager_name varchar(100) not null,
    manager_address varchar(200) not null,
    manager_contact_number varchar(100),
    manager_surname varchar(100) not null,
    manager_email varchar(100) not null


);
go

create table Room_Type
(
    room_type_id int primary key not null,
    room_type_name varchar(100),
    room_type_cost decimal (10, 2) not null,
    room_type_description varchar(300) not null,


);
```

```
create table Room
(

    room_id int primary key not null,
    room_number varchar(100) not null,
    hotel_id int not null,
    room_type_id int not null,
    constraint Hotel_To_Room_FK foreign key (hotel_id) references Hotel(hotel_id),
    constraint Room_Type_to_Room_FK foreign key (room_type_id) references Room_Type (room_type_id)

);
go



go


create table Customer
(
    customer_id int primary key not null,
    customer_name varchar(100) not null,
    customer_surname varchar(100),
    customer_address varchar(100) not null,
    customer_contact_number varchar(100),
    customer_email varchar(100) not null
);


go


create table Booking
(
    booking_id int primary key not null,
    booking_date datetime not null,
    check_in_date datetime not null,
    check_out_date datetime not null,
    booking_payment_type varchar(100) not null,
    customer_id int not null,
    hotel_id int not null,
    constraint Booking_to_Hotel foreign key (hotel_id) references  Hotel (hotel_id),
    constraint Booking_to_Customer_FK foreign key (customer_id) references Customer(customer_id)
);
```

```sql
create table Bookin_To_Room
(
    booking_id int not null,
    room_id int not null,
    constraint Booking_To_Room_Fk foreign key (booking_id) references Booking(booking_id),
    constraint Booking_To_Room_Fk1 foreign key (room_id) references Room(room_id),
    primary key (booking_id, room_id)

);

go


create table Employee
(
    employee_id int primary key not null,
    employee_name varchar(100) not null,
    employee_surname varchar(100),
    employee_address varchar(100)not null,
    employee_contact_number varchar(100),
    employee_position varchar(100) not null,
    employee_email varchar(100),
    hotel_id int not null,
    constraint Employee_to_Hotel_FK foreign key  (hotel_id) references Hotel(hotel_id)
);

go

create table Service1
(
    service_id int primary key not null,
    service_name varchar(100) not null,
    service_cost decimal(10, 2) not null,
    hotel_id int not null,
    constraint Service1_to_Hotel_FK foreign key (hotel_id) references Hotel (hotel_id)
);

```

```
go

alter table Hotel
--creating a foreign key constraint to named Manager_to_Hotel_FK
add constraint Manager_to_Hotel_FK foreign key (manager_id) references Manager (manager_id);
```

Code Implementation

```sql
insert into Manager (manager_id, manager_name, manager_surname, manager_address, manager_contact_number, manager_email)
values
( 8950, 'Alexander', 'Mitchell', '789 Oak Avenue, Cityville, USA', ' +1 (555) 123-4567', 'alexandermichell@gmail.com')



insert into Hotel (hotel_id, hotel_name, hotel_contact_number, hotel_address, manager_id)
values
( 1902, 'Sunset View', '+1 (555) 123-4567' , '456 Park Avenue, Townsville, USA', 8950 )


insert into Customer (customer_id, customer_name, customer_surname, customer_address, customer_contact_number, customer_email)
values
( 0001, 'Antoni',   'Levis',    '123 Main Street, Anytown, USA',      '+1 (555) 123-4567', 'antonilevis@gmail.com'),
( 0002, 'Emma',     'Johnson',  '456 Elm Avenue, Springfield, USA',   '+1 (555) 987-6543', 'emmajohnson@gmail.com'),
( 0003, 'Michael',  'Davis',    '789 Oak Road, Cityville, USA',       '+1 (555) 246-8102', 'michaeldevis@gmail.com'),
( 0004, 'Olivia',   'Williams', '321 Pine Street, Metroville, USA',   '+1 (555) 135-7910', 'oliviawilliams@gmail.com'),
( 0005, 'William',  'Brown',    '567 Cedar Lane, Townsville, USA',    '+1 (555) 369-2580', 'williambrown@gmail.com'),
( 0006, 'Ava',      'Jones',    '890 Walnut Drive, Villageland, USA', '+1 (555) 802-4671', 'ayajones@gmail.com'),
( 0007, 'James',    'Wilson',   '234 Maple Avenue, Countyville, USA', '+1 (555) 624-3579', 'jameswilson@gmail.com'),
( 0008, 'Sophia ',  'Martinez', '678 Birch Street, Suburbia, USA',    '+1 (555) 917-2468', 'sophiamartinez@gmail.com'),
( 0009, 'Benjamin', 'Anderson', '901 Willow Road, Countryside, USA',  '+1 (555) 372-9150', 'benjaminanderson@gmail.com'),
( 0010, 'Mia',      'Taylor',   '345 Ash Lane, Townscape, USA',       '+1 (555) 689-2345', 'miataylor@gmail.com')



insert into Employee (employee_id, employee_name, employee_surname, employee_address, employee_contact_number, employee_position, hotel_id)
values
( 001, 'Sarah', 'Johnson', '123 Maple Street, Cityville, USA',   '+1 (555) 123-4567', 'Front Desk Receptionist',    1902),
( 002, ' Ethan', 'Williams', '456 Oak Avenue, Townsville, USA',  '+1 (555) 987-6543', 'Concierge',                  1902),
( 003, 'Olivia', 'Smith', '789 Elm Road, Countryside, USA',      '+1 (555) 246-8102', 'Housekeeping Staff' ,        1902),
( 004, 'Noah', 'Anderson', '234 Pine Street, Metroville, USA',   '+1 (555) 135-7910', 'Bellhop',                    1902),
( 005, 'Ava', 'Brown', '567 Cedar Lane, Suburbia, USA',          '+1 (555) 369-2580', 'Restaurant Server',          1902),
( 006, 'William', 'Davis', '890 Walnut Drive, Villageland, USA', '+1 (555) 802-4671', 'Bartender',                  1902),
( 007, 'Sophia', 'Clark', '345 Oak Road, Anytown, USA',          '+1 (555) 624-3579', 'Event Coordinator',          1902),
( 008, 'Jackson', 'Thomas', '678 Pine Avenue, Countyville, USA', '+1 (555) 917-2468', 'Security Officer',           1902),
( 009, 'Emma', 'Martinez', '901 Cedar Lane, Cityville, USA',     '+1 (555) 802-4671', 'Spa Therapist',              1902),
( 010, 'Lucas', 'Walker', ' 123 Elm Street, Townsville, USA',    '+1 (555) 624-3579', 'Sales and Marketing Manager',1902)
```

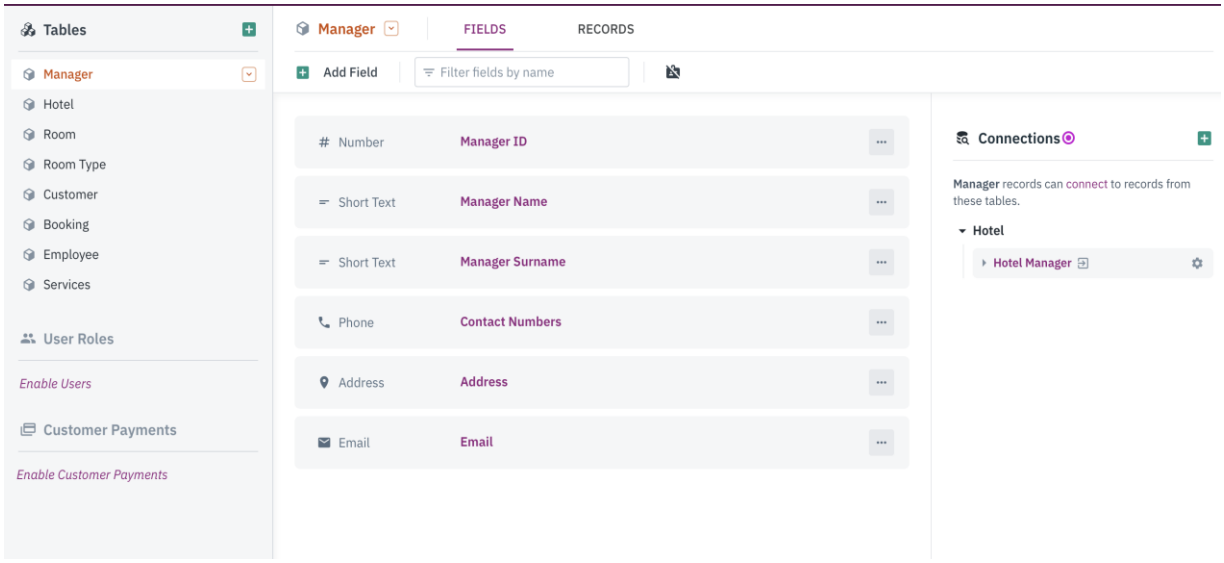
```sql
insert into Booking (booking_id, booking_date, check_in_date, check_out_date, booking_payment_type, customer_id,  hotel_id)
values
(1001 , '2023-05-01', '2023-05-05', '2023-05-10', 'Credit Card',   0001,  1902 ),
(1002 , '2023-05-02', '2023-05-06', '2023-05-09', 'Cash',          0002,  1902 ),
(1003 , '2023-05-03', '2023-05-07', '2023-05-11', 'Debit Card',    0003,  1902 ),
(1004 , '2023-05-04', '2023-05-08', '2023-05-12', 'Credit Card',   0004,  1902 ),
(1005 , '2023-05-05', '2023-05-09', '2023-05-13', 'Cash',          0005,  1902 ),
(1006 , '2023-05-06', '2023-05-10', '2023-05-15', 'Credit Card',   0006,  1902 ),
(1007 , '2023-05-07', '2023-05-11', '2023-05-14', 'Debit Card',    0007,  1902 ),
(1008 , '2023-05-08', '2023-05-12', '2023-05-16', 'Cash',          0008,  1902 ),
(1009 , '2023-05-09', '2023-05-13', '2023-05-17', 'Credit Card',   0009,  1902 ),
(1010 , '2023-05-09', '2023-05-14', '2023-05-18', 'Debit Card',    0010,  1902 )


insert into Room_Type (room_type_id, room_type_name, room_type_cost, room_type_description )
values
(1, 'Standard Room', 100, 'Cozy and comfortable room with basic amenities' ),
(2, 'Deluxe Room',   150, 'Spacious room with modern amenities and a stunning view' ),
(3, 'Suite Room',   150, 'Luxurious suite with separate living area and deluxe facilities')


insert into Room (room_id, room_number, hotel_id, room_type_id)
values
(101, 202, 1902,  1 ),
(102, 203, 1902,  2),
(103, 204, 1902,  3 ),
(104, 205, 1902,  2 ),
(105, 206, 1902,  2 ),
(106, 207, 1902,  1 ),
(107, 208, 1902,  3 ),
(108, 209, 1902,  1 ),
(109, 210, 1902,  3 ),
(110, 211, 1902,  3 ),
(111, 212, 1902,  1 ),
(112, 213, 1902,  2 ),
(113, 214, 1902,  3 ),
(114, 215, 1902,  1 ),
(115, 216, 1902,  2 ),
(116, 217, 1902,  1 ),
(117, 218, 1902,  2 ),
(118, 219, 1902,  2 ),
(119, 220, 1902,  3 ),
(120, 221, 1902,  1 )
```

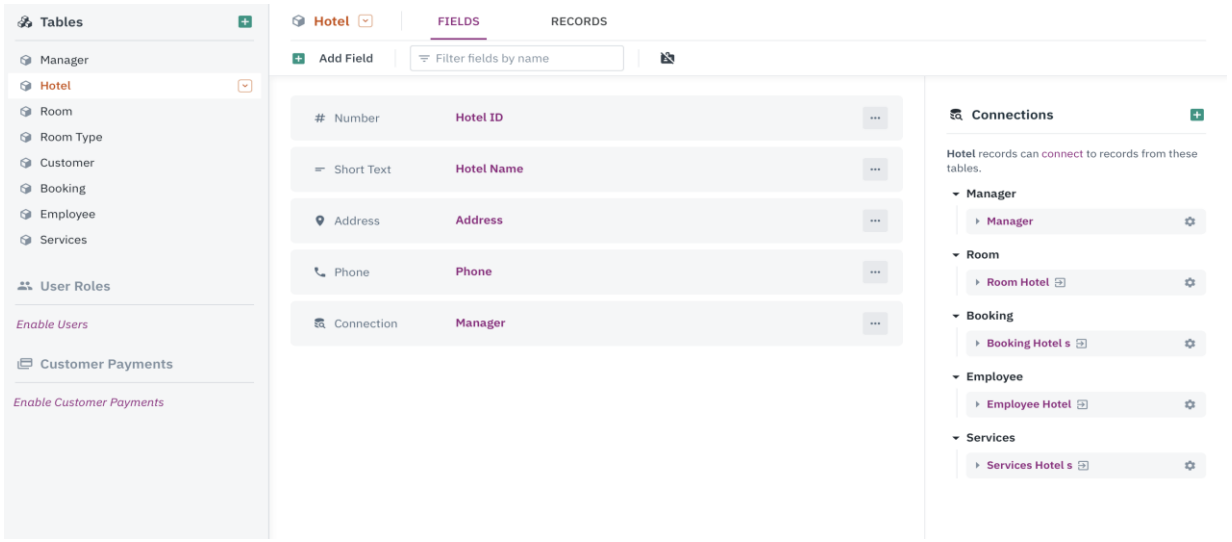
```
insert into Service1 (service_id,[service_name], service_cost, hotel_id)
values
( 01 , 'Room Service',          20, 1902),
( 02 , 'Housekeeping',          10, 1902),
( 03 , 'Laundry Service',       15, 1902),
( 04 , 'Concierge Assistence',  10, 1902),
( 05 , 'Fitness Center Access', 0,  1902),
( 06 , 'Spa Service',           50, 1902),
( 07 , 'Valet Parking',         20, 1902),
( 08 , 'Business Center',       0,  1902),
( 09 , 'Tour Desk',             10, 1902),
( 10 , 'Airport Shuttle',       30, 1902)
```

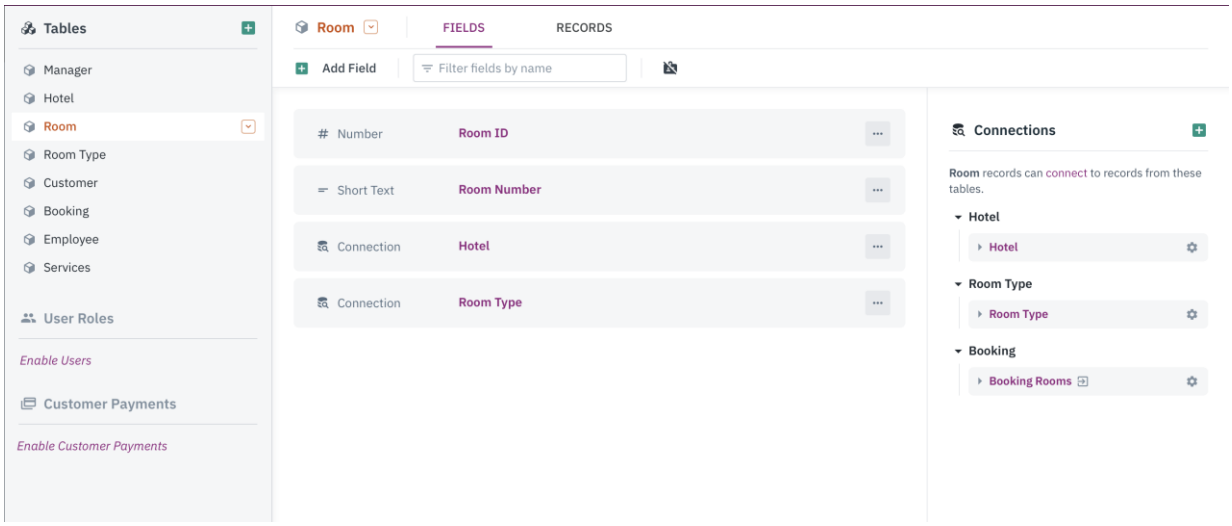
7. The Interface

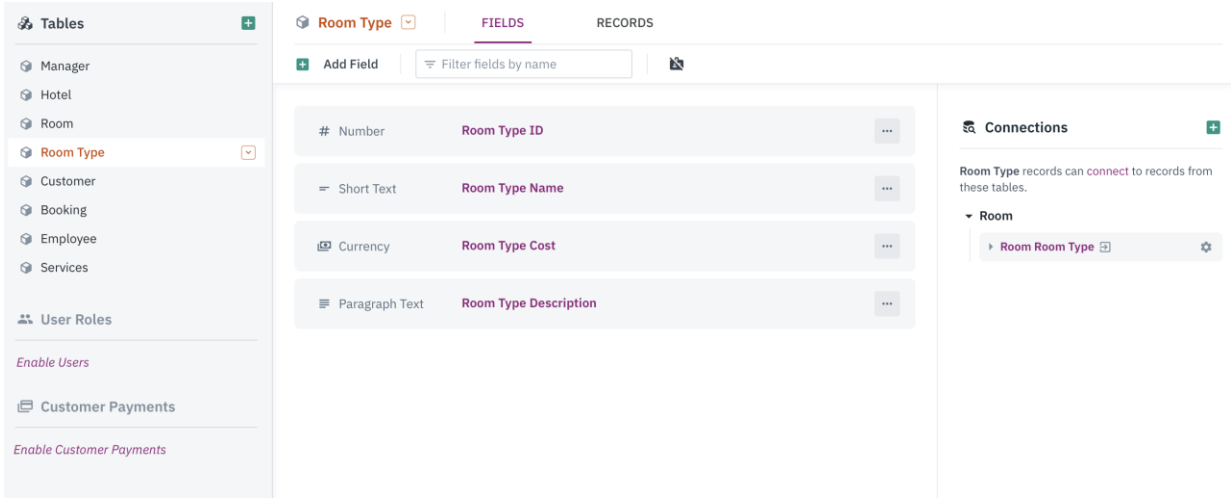
The database was stored into MS SQL Server. I created the interface using Knack software.

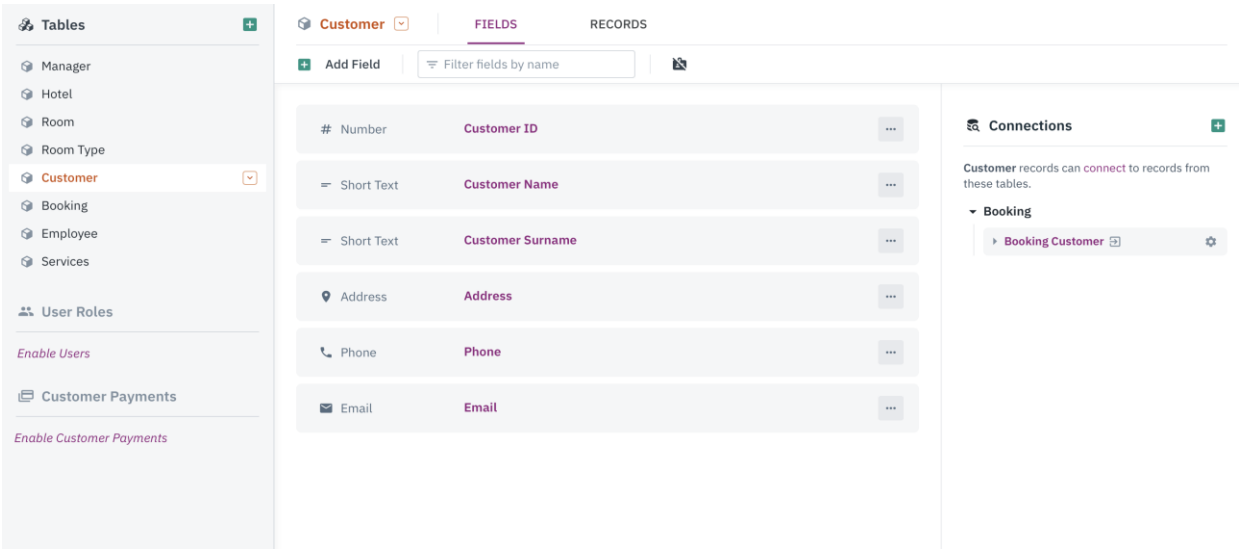
Data Interstation

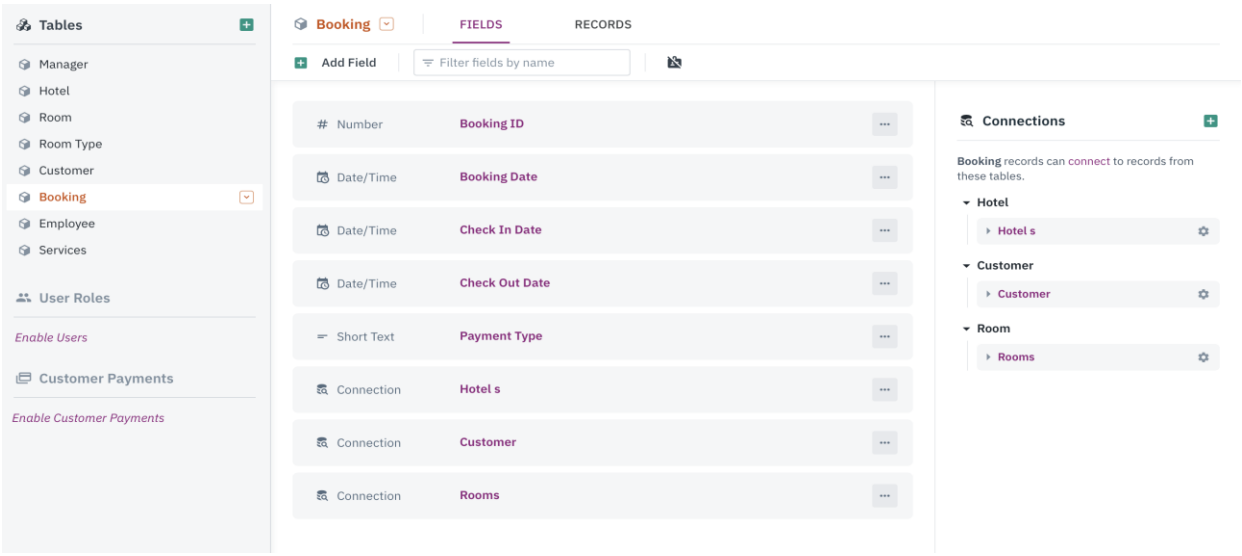


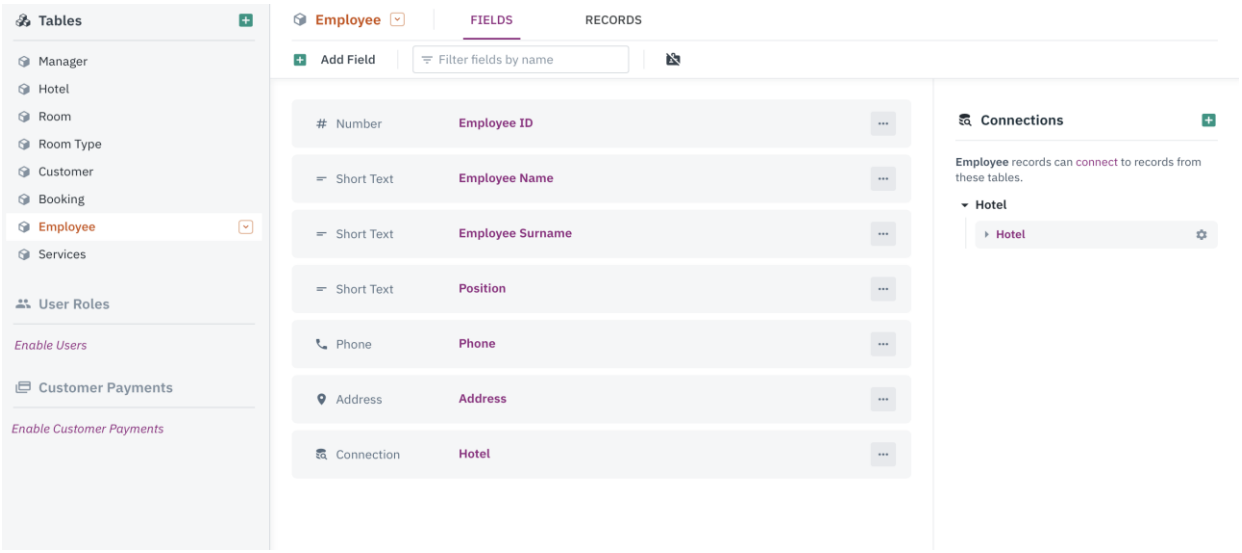

Tables
Manager
Hotel
Room
Room Type
Customer
Booking
Employee
Services
User Roles
Enable Users
Customer Payments
Enable Customer Payments
Hotel
FIELDS
RECORDS
Add Field
Filter fields by name
Number
Hotel ID
Short Text
Hotel Name
Address
Address
Phone
Phone
Connection
Manager
Connections
Hotel records can connect to records from these tables.
Manager
Manager
Room
Room Hotel
Booking
Booking Hotel s
Employee
Employee Hotel
Services
Services Hotel s

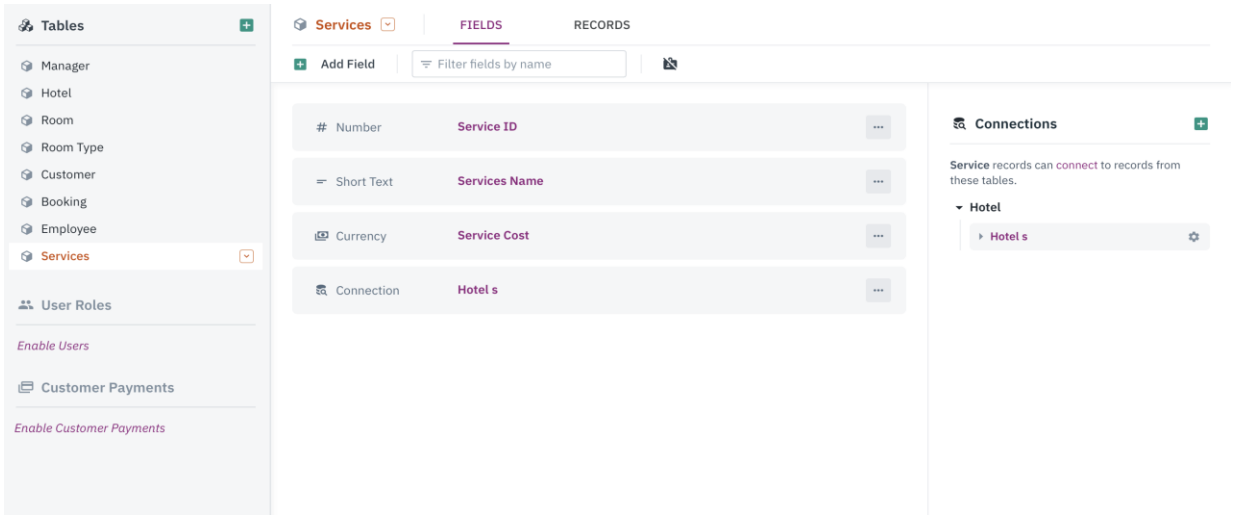

Tables
Manager
Hotel
Room
Room Type
Customer
Booking
Employee
Services
User Roles
Enable Users
Customer Payments
Enable Customer Payments
Room
FIELDS
RECORDS
Add Field
Filter fields by name
Number
Room ID
Short Text
Room Number
Connection
Hotel
Connection
Room Type
Connections
Room records can connect to records from these tables.
Hotel
Hotel
Room Type
Room Type
Booking
Booking Rooms


Tables
Manager
Hotel
Room
Room Type
Customer
Booking
Employee
Services
User Roles
Enable Users
Customer Payments
Enable Customer Payments
Room Type
FIELDS
RECORDS
Add Field
Filter fields by name
Number
Room Type ID
Short Text
Room Type Name
Currency
Room Type Cost
Paragraph Text
Room Type Description
Connections
Room Type records can connect to records from these tables.
Room
Room Room Type

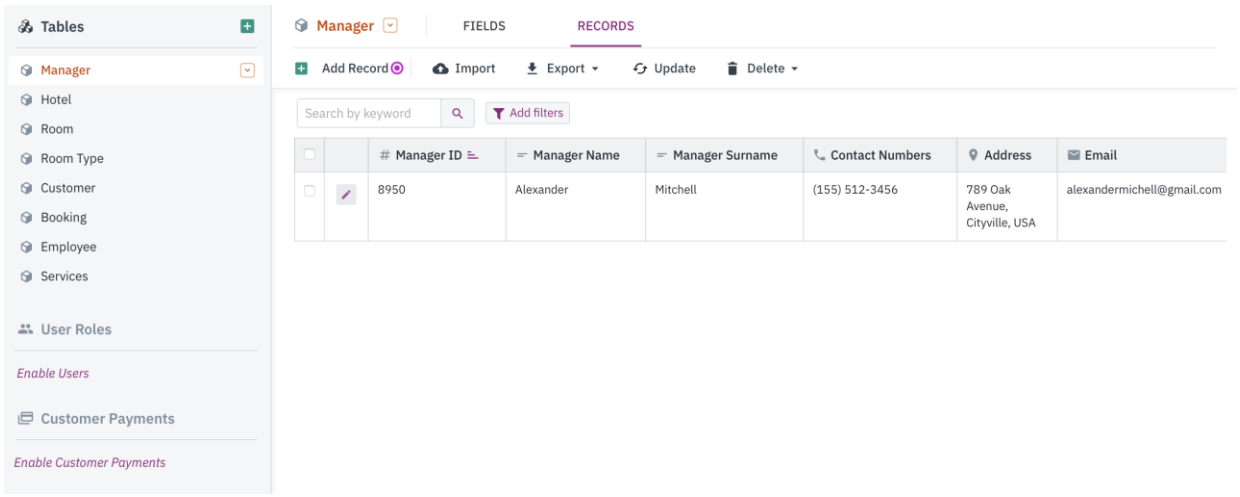

Tables
Manager
Hotel
Room
Room Type
Customer
Booking
Employee
Services
User Roles
Enable Users
Customer Payments
Enable Customer Payments
Customer
FIELDS
RECORDS
Add Field
Filter fields by name
Number
Customer ID
Short Text
Customer Name
Short Text
Customer Surname
Address
Address
Phone
Phone
Email
Email
Connections
Customer records can connect to records from these tables.
Booking
Booking Customer

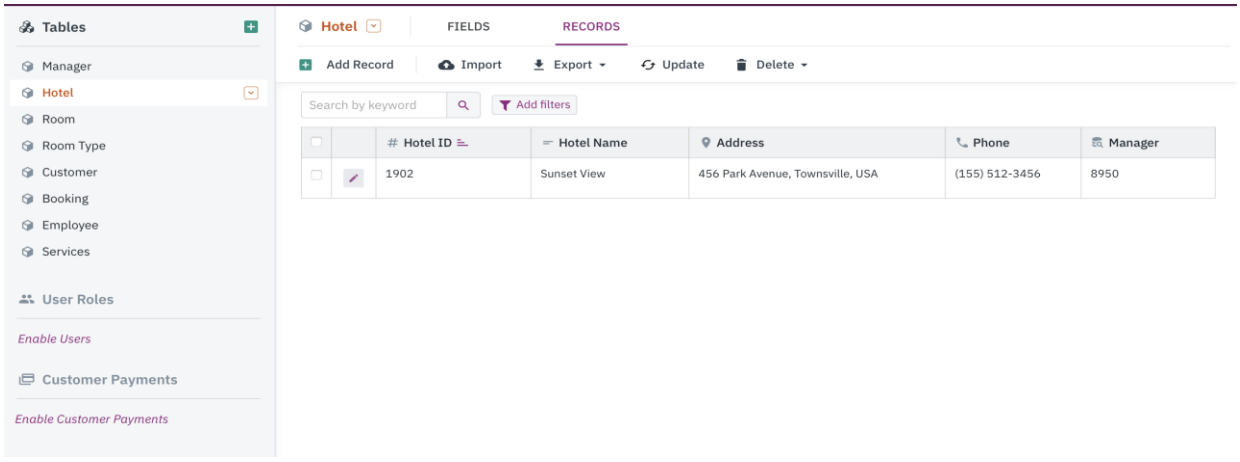

Tables
Manager
Hotel
Room
Room Type
Customer
Booking
Employee
Services
User Roles
Enable Users
Customer Payments
Enable Customer Payments
Booking
FIELDS
RECORDS
Add Field
Filter fields by name
Number
Booking ID
Date/Time
Booking Date
Date/Time
Check In Date
Date/Time
Check Out Date
Short Text
Payment Type
Connection
Hotel s
Connection
Customer
Connection
Rooms
Connections
Booking records can connect to records from these tables.
Hotel
Hotel s
Customer
Customer
Room
Rooms

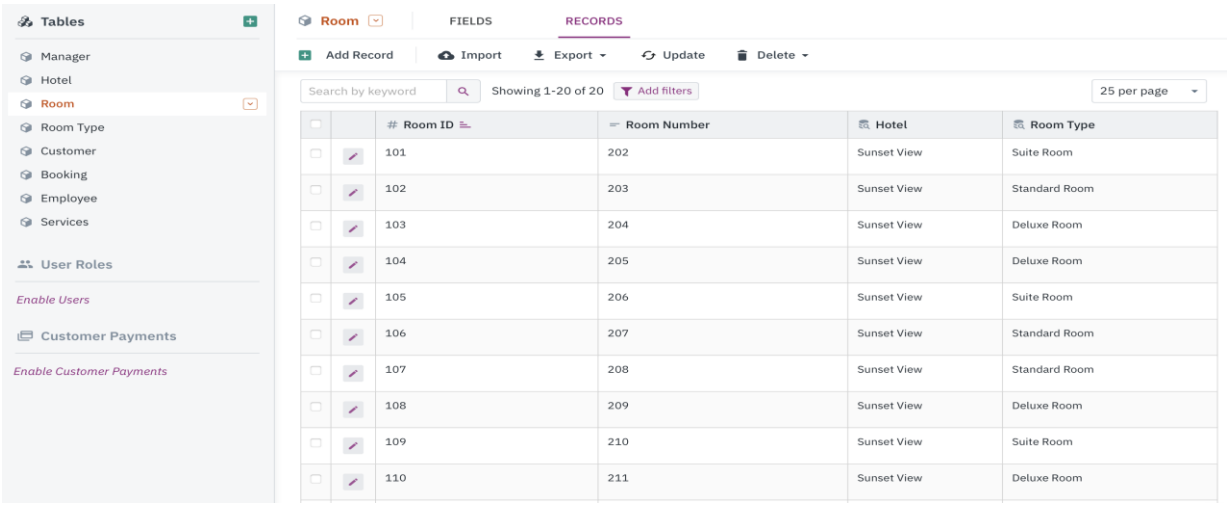

Tables
Manager
Hotel
Room
Room Type
Customer
Booking
Employee
Services
User Roles
Enable Users
Customer Payments
Enable Customer Payments
Employee
FIELDS
RECORDS
Add Field
Filter fields by name
Number
Employee ID
Short Text
Employee Name
Short Text
Employee Surname
Short Text
Position
Phone
Phone
Address
Address
Connection
Hotel
Connections
Employee records can connect to records from these tables.
Hotel
Hotel

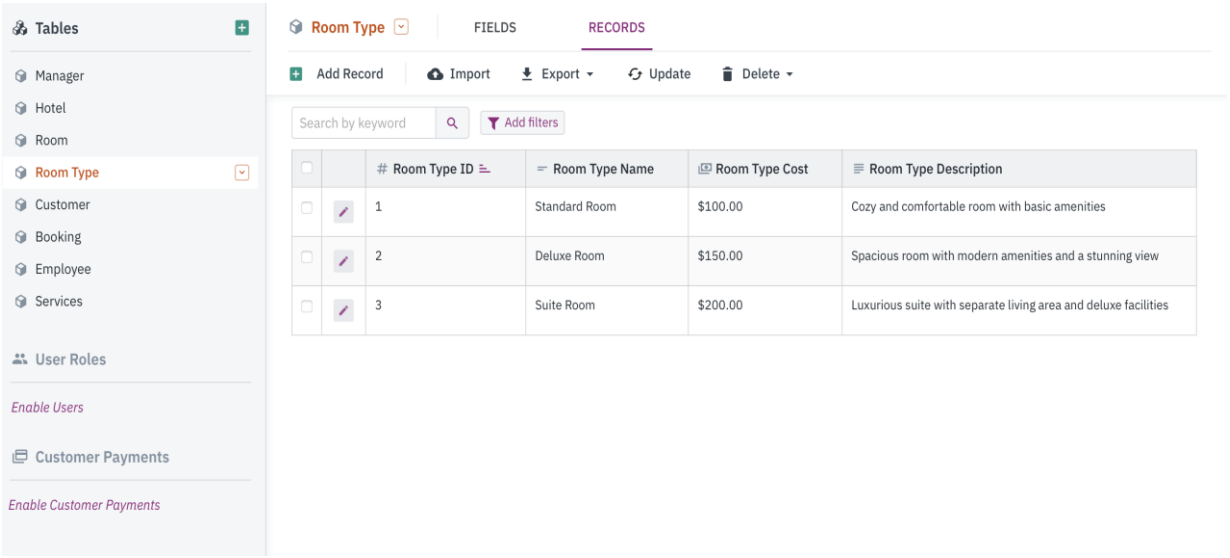


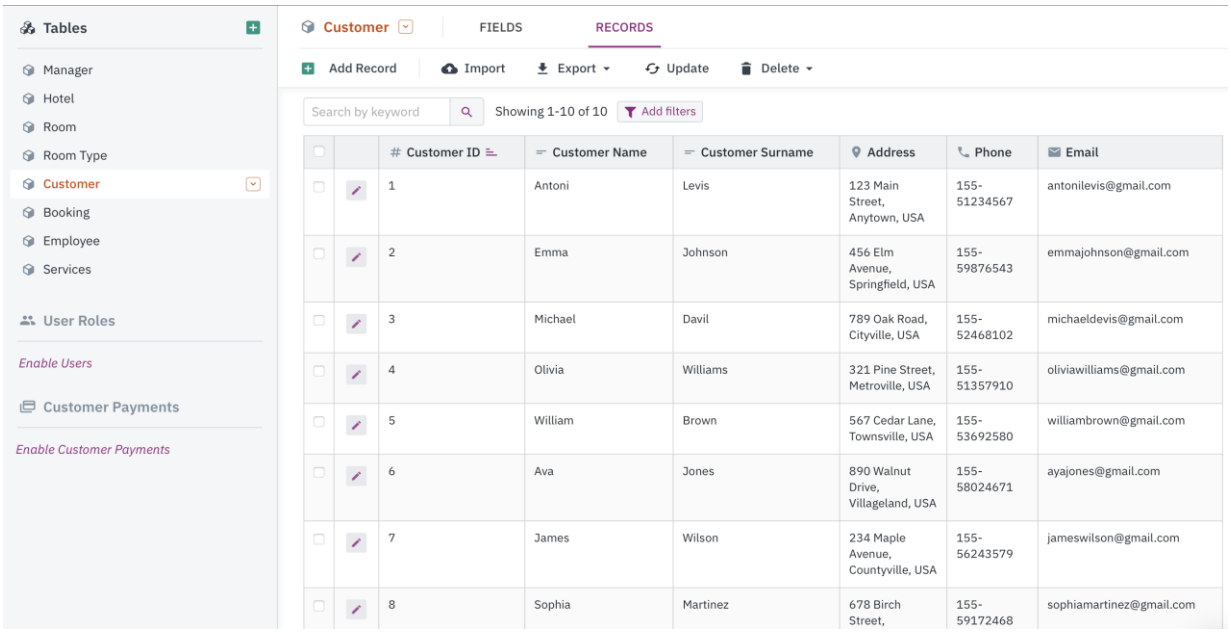
Tables with articles

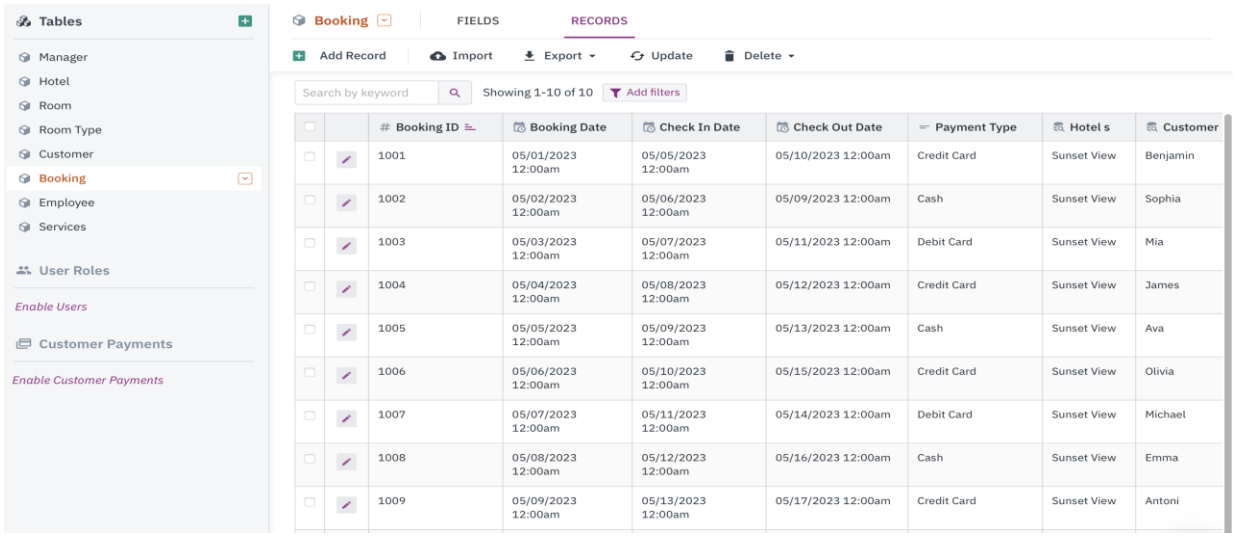


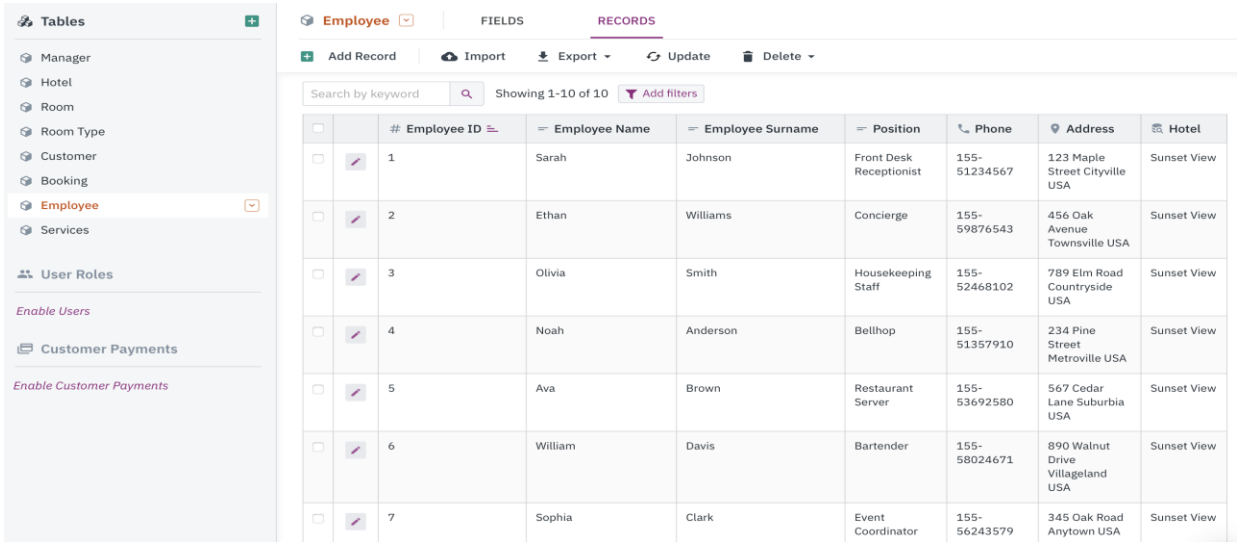


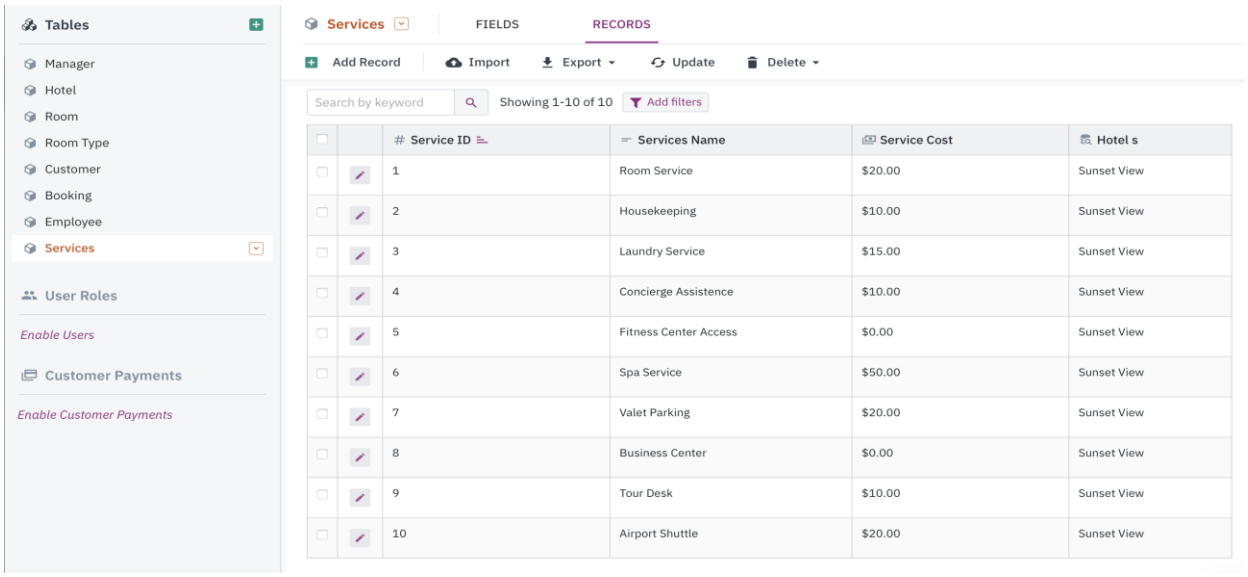

Tables
Manager
Hotel
Room
Room Type
Customer
Booking
Employee
Services
User Roles
Enable Users
Customer Payments
Enable Customer Payments
Room
FIELDS
RECORDS
Add Record
Import
Export
Update
Delete
Search by keyword
Showing 1-20 of 20
Add filters
25 per page
Room ID
Room Number
Hotel
Room Type
101 202 Sunset View Suite Room
102 203 Sunset View Standard Room
103 204 Sunset View Deluxe Room
104 205 Sunset View Deluxe Room
105 206 Sunset View Suite Room
106 207 Sunset View Standard Room
107 208 Sunset View Standard Room
108 209 Sunset View Deluxe Room
109 210 Sunset View Suite Room
110 211 Sunset View Deluxe Room


Tables
Manager
Hotel
Room
Room Type
Customer
Booking
Employee
Services
User Roles
Enable Users
Customer Payments
Enable Customer Payments
Room Type
FIELDS
RECORDS
Add Record
Import
Export
Update
Delete
Search by keyword
Add filters
Room Type ID
Room Type Name
Room Type Cost
Room Type Description
1 Standard Room $100.00 Cozy and comfortable room with basic amenities
2 Deluxe Room $150.00 Spacious room with modern amenities and a stunning view
3 Suite Room $200.00 Luxurious suite with separate living area and deluxe facilities

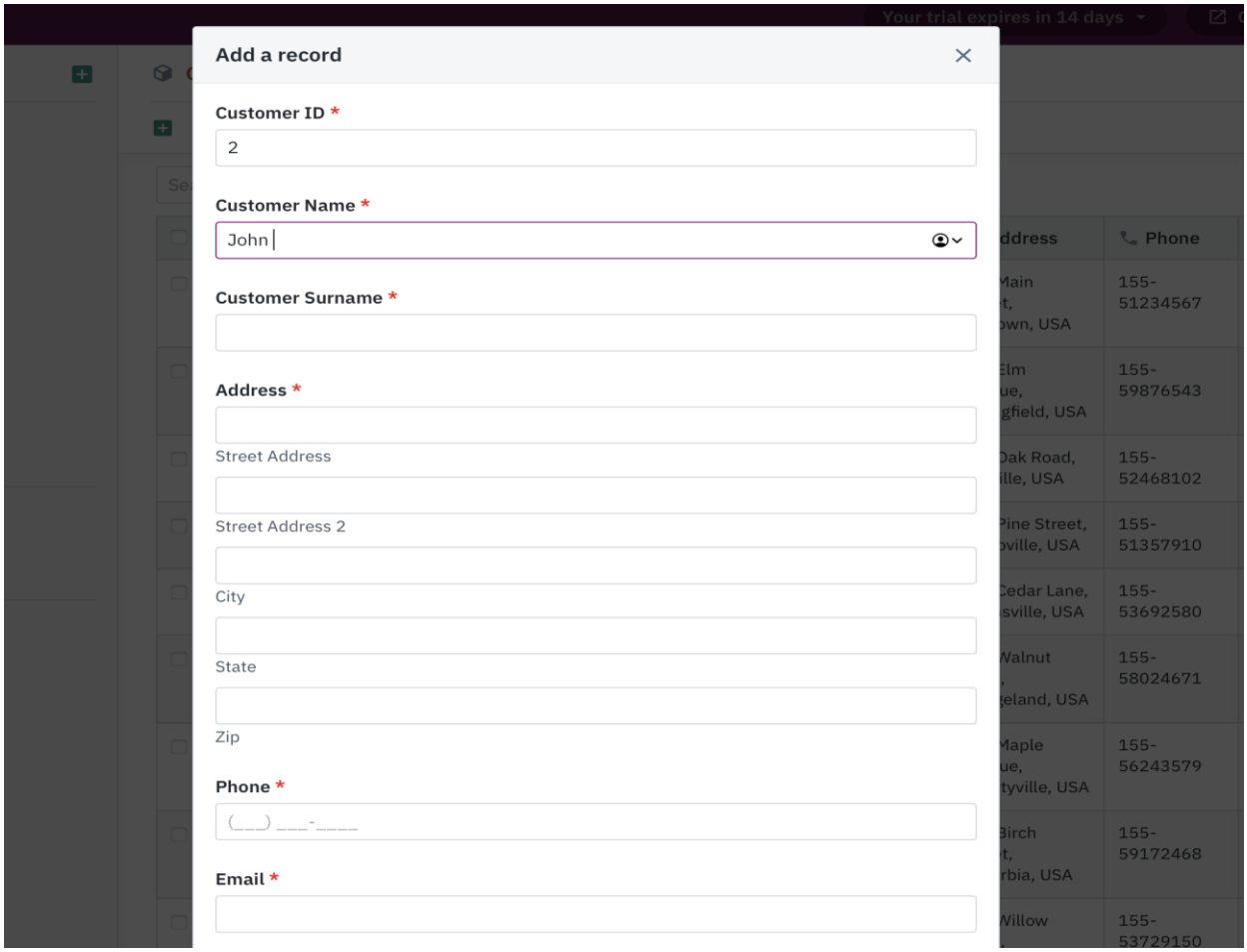

Tables
Manager
Hotel
Room
Room Type
Customer
Booking
Employee
Services
User Roles
Enable Users
Customer Payments
Enable Customer Payments
Customer
FIELDS
RECORDS
Add Record
Import
Export
Update
Delete
Search by keyword
Showing 1-10 of 10
Add filters
Customer ID
Customer Name
Customer Surname
Address
Phone
Email
1 Antoni Levis 123 Main Street, Anytown, USA 155-51234567 antonilevis@gmail.com
2 Emma Johnson 456 Elm Avenue, Springfield, USA 155-59876543 emmajohnson@gmail.com
3 Michael Davil 789 Oak Road, Cityville, USA 155-52468102 michaeldevis@gmail.com
4 Olivia Williams 321 Pine Street, Metroville, USA 155-51357910 oliviawilliams@gmail.com
5 William Brown 567 Cedar Lane, Townsville, USA 155-53692580 williambrown@gmail.com
6 Ava Jones 890 Walnut Drive, Villageland, USA 155-58024671 ayajones@gmail.com
7 James Wilson 234 Maple Avenue, Countyville, USA 155-56243579 jameswilson@gmail.com
8 Sophia Martinez 678 Birch Street, 155-59172468 sophiamartinez@gmail.com







Data insertion example



Data searching (example in SQL)

```
1   use Hotel_Management_Studio
2
3   select customer_id, customer_name
4   from Customer c
5   where c.customer_name = 'Mia'
```

Results | Messages

	customer_id	customer_name
1	10	Mia

This was all my project, during interstation process, creating the data model and so on, I have made some changes, other than that I have checked and the database works perfectly.