Database Project

Friona Pocari

BINF II A

November 13 2022

Project Topic: *Designing a database for a Hotel Management System.*

- Short description and introduction of the business I am going to work with:

Hani i Xheblatit is listed on the UNESCO site as first category house, and is

protected by the Directory of Cultural Monumental. Hotel Hani i Xheblatit is

located in Berat(in the old quarter of Mangalem), just a 5-min walk from the city center. This small family-owned hotel has 10 rooms, 2 familiar suites (for 4 persons), 2 standard rooms,3 triple rooms, and 3 double or twin rooms.

We are going to represent the problem we identified later on but as an introduction, it is important to say that we are going to work with the whole management part of the hotel.

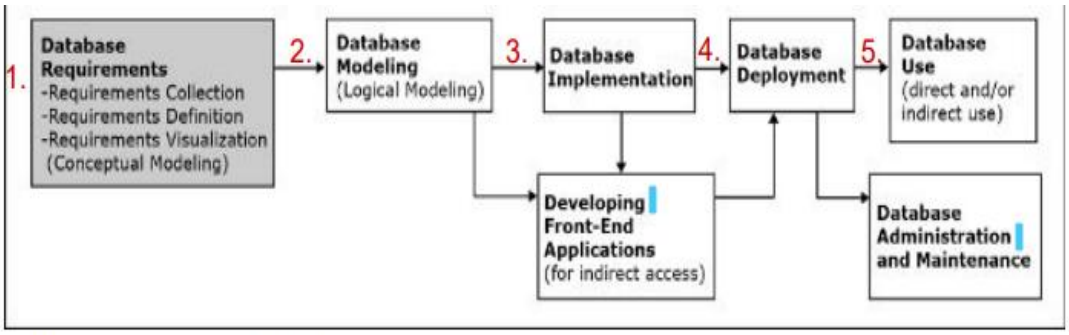
Describe the scope of your project:

This project is for the development of a hotel database management system. This database will assist employees inside the hotel to organize the services provided to customers, including payments, bookings(reservations), check-in and check-out dates, and so on. Its primary purpose is to provide hotel workers with a clear understanding of general data regarding their clients and avoid different problems such as double booking. The first interface will log the user in as an admin(staff) or a client. Option 1: If the user is going to be logged in as an admin a menu bar is going to be shown in which

the user can choose to see clients, and reservations, or check for room availability. Admins can perform actions like editing (insert/delete clients) and search through the database. Option 2: If the user is going to be logged in as a client, aside from personal information, his room number and days left in the hotel will be available for use.

The Situation: As a growing business the Hotel employees and managers are struggling to keep track of clients' reservations, and the availability of rooms. This problem has caused double booking and delays which has consequently affected the relationship with current and next customers.

IMPORTANT PART: THE DEVELOPMENT OF THE DATABASE SYSTEM



Steps in the development of database systems

1. The first step is the **Database Requirements**, which can be determined by the questions below. *After we (our team) analyzed the Hotel, we got a full understanding of their needs to put together the best system for their business.*

- What kind of data needs to be stored and why it is needed to be stored?

The data needed to be stored in the Database is data related to Clients, Clients' Reservations, and Rooms so the users can see which room is occupied with which client and which room is available to avoid problems like double booking.

- How is data going to be accessed?

Each table of the Database will be connected through the Primary Keys of each relation which will mainly be the Id since is unique for every person. The relationship between the data sets is very important, so we are making sure to design the best **entity relationship diagram-relational schema** to have a **clear visualization** of the Database.

- Who is going to be accessing this data?

This Data will only be accessible to users and by Hotel employees.

- How will we protect the data?

Data protection is very important the **strategy we are going to use is storing the Database in different locations.**

The database will not be complex as it is a small family-owned business.

List the descriptions of requirements, in terms of actions that a user could perform:

What actions can the user perform?

The management system will be like a front-end application with a few (4-5) interfaces. **1)**To log in the user must input the username and the password at that moment his information will be available for use. **2)**After that the main form will appear, on the menu bar the user can choose if he is 1) logging in as staff(admin) in this case these options will appear: check the clients, reservations, or room availability OR b) Log in as a client and access his personal and room data such as (id, first name, last name, phone no, address, and room number). **3)** Each of the three forms in the admin account has its own interfaces and database tables. (Current update: we have established the entities and will discuss later their specific attributes). Specific options on what the users can do with the clients are to be discussed but we are thinking o*f including options like searching, selecting, inserting, and updating the database* and consequently the program.

As for the program, all the elements of the database are going to be shown such as client info, hotel info, room info and price, check-in and check-out date, etc.

2. **Database Modelling** (Second Step)

- Definition: Data modelling is the process of diagramming data flows.

This term is used to refer to the creation of the database model, also known as a logical database model, that is implementable by the DBMS software.

Our project illustrates the management system of a family-owned hotel. Two level users will be used: administrator & customer. We will use ER diagrams to visualize the requirements and then continue with the process of mapping into a relational schema, to later conclude in a Database with its own tables.

The main entities of the management system are presented in the table below:

Administrator Level: 1. Room management - RoomID (PK) - Room NO - Name - Fees - Vacancy	2. Bookings/Reservation - BookingID (PK) - Check in date - Check out date - Status	3. Customer registration - CostumerID (PK) - Name - Surname - Phone NO - Email - Address

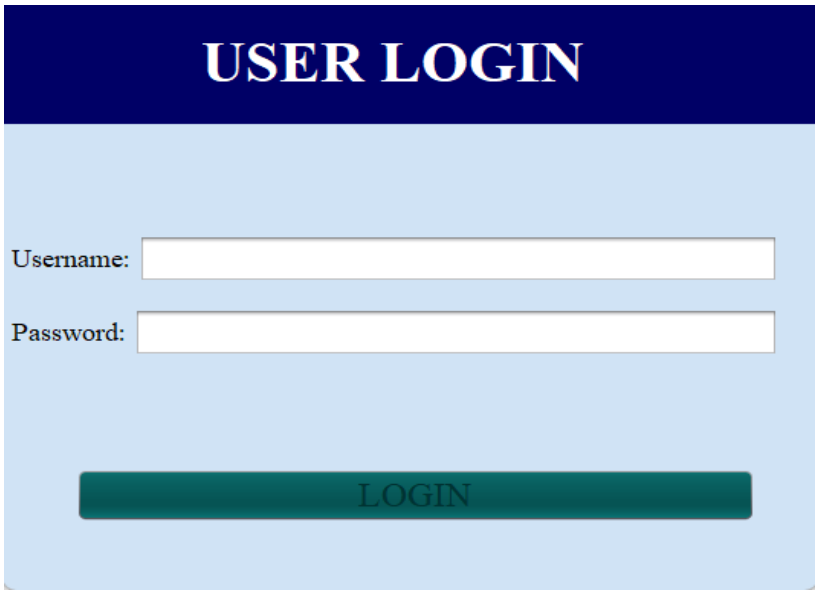
4. SYSTEM LOGIN - AdminID (PK) - Name - Surname - Email - Password	Costumer Level: 5. LOGIN / REGISTRATION - UserID (PK) - Name - Surname - Email - Password	

3. Database Implementation

After developing the Database, the next step is implementing it. In this step, DBMS software is needed. Our team will use the DBMS functionalities to implement the model as an actual database. Since we are going to model the Database as a relational Database, we will also use an RDBMS software package. The language we are going to use is SQL. SQL includes commands like ("SELECT * FROM (database table name) WHERE (a row of the table) =? AND (another row of the table) =?") etc...

4. Front-end application

The development of this application refers to designing and creating an application that can help end users to access indirectly the database. Our front-end application will be easy and simple so users can get a full understanding of the data they need to access. It will include all the different interfaces mentioned above. We have provided a small glimpse of the login interface:



5. Database Deployment

When the Database and its front-end application are finished we will deploy the program for use by the end users of the Hotel.

6. Database use:

After the deployment of the DB at this last step the end users are able to engage in the database use. The database system can be used **indirectly** (by the front-end application users they will have the right to perform actions like searching the database and updating or deleting a customer) or can be used **directly** through the DBMS in SQL Query (our team) can write a line of code to insert, delete, update and modify the database.

Conclusion: These commands mentioned above can be issued by front-end applications (indirect use) or by (direct use).

7. **Database Administration and Maintenance**

- Definition: Database Administration consists of everything required to manage a database and make it available as needed.

We will have one or more administrators who will operate the hotel management system. They will be in charge of monitoring data flows, keeping it updated, consistent, and easily accessible, providing security to prevent data breaches, and ensuring the best and most efficient services to our customers.

Key Note: As mentioned at the beginning *Data protection is very important to us the* ***strategy, we are going to use is storing the Database in different locations.***

List the descriptions of requirements, in terms of actions that a user could perform:

What actions can the user perform?

The management system will be like a front-end application with a few (4-5) interfaces. **1)**To log in the user must input the username and the password at that moment his information will be available for use. **2)**After that the main form will appear, on the menu bar the user can choose if he is 1) logging in as staff(admin) in this case these options will appear: check the clients, reservations, or room availability OR b) Log in as a client and access his personal and room data such as (id, first name, last name, phone no, address, and room number). **3)** Each of the three forms in the admin account has its own interfaces and database tables. (Current update: have established the entities and will discuss later their specific attributes). Specific options on what the users can do with the clients are to be discussed but we are thinking o*f including options like searching, selecting, inserting, and updating the database* and consequently the program.

(Intentionally in two places because this paragraph belongs to Database Requirements but at the outline was placed at the end)

PART II

Entity Relationship Diagram

i) Define ENTITIES

Entities - constructs that represent what the database keeps track of.

The basic building blocks of an ER diagram.

Represent various real-world notions, such as people, places, objects, events, items, and other concepts.

Within one ERD each entity must have a different name.

Entities that are part of our project: COSTUMER, ADMINISTRATOR, BOOKING, ROOM.

ii) Define attributes: please make sure to include some numerical and date fields as attributes, in order to use them in analysis step.

Attributes- depiction of a characteristic of an entity.

• Represents the details that will be recorded for each entity instance.

• Within one entity, each attribute must have a different name.

Numerical attributes included in our project: BookindID, RoomID, RoomNO, PhoneNO, CostumerID etc.

Date fields included in our project: CheckInDate, CheckOutDate.

iii) PK, FK?

(1) Describe Relationships:

PK: A primary key is the column or columns that contain values that uniquely identify each row in a table. Each relation must have a primary key. The name of the primary key column is underlined in order to distinguish it from the other columns in the relation

FK: A foreign key is a column in a relation that refers to a primary key column in another (referred) relation. A mechanism that is used to depict relationships in the relational database model. For every occurrence of a foreign key, the relational schema contains a line pointing from the foreign key to the corresponding primary key

Relationship between PK and FK:

A foreign key is a column or a set of columns in a table whose values correspond to the values of the primary key in another table. In order to add a row with a given foreign key value, there must exist a row in the related table with the same primary key value.

Administrator □ PK: AdminId

Room □ PK: RoomId; FK: BookingId

Booking □ PK: BookingId; FK: AdminId, CustomerIId

Costumer □ PK: CostumerId; FK: RoomId, AdminId, BookingId

iv) State TWO sentences that describe the relationship

(1) Cardinality and why?

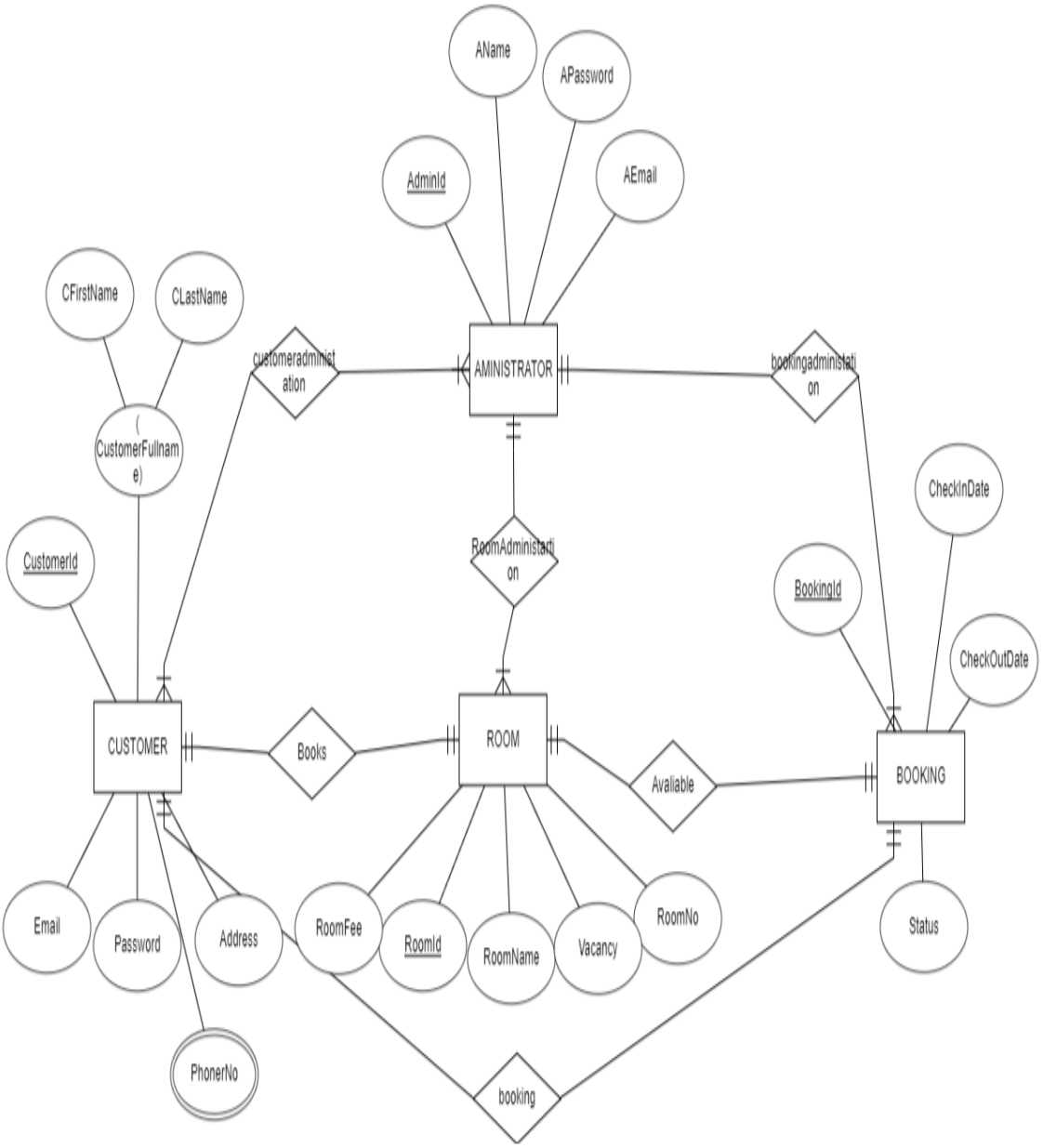
(2) Participation and why?

o Cardinality In a database, cardinality usually represents the relationship between the data in two different tables by highlighting how many times a specific entity occurs in comparison to another.
o Participation in a database, participation refers to whether an entity must participate in a relationship with another entity to exist.

Costumer Administration:
-COSTUMER mandatory participation (one-to-one)
-ADMINISTRATOR minimum cardinality is 1 and maximum cardinality is n
Room Administration:
-ROOM mandatory participation (one-to-one)
-ADMINISTRATOR minimum cardinality is 1 and maximum cardinality is n
Booking Administration:
-BOOKING mandatory participation (one-to-one)
-ADMINISTRATOR minimum cardinality is 1 and maximum cardinality is n
Books:
-COSTUMER mandatory participation (one-to-one)
-ROOM mandatory participation (one-to-one)
Available:
-BOOKING mandatory participation (one-to-one)
-ROOM mandatory participation (one-to-one)
Booking:
-COSTUMER mandatory participation (one-to-one)
-BOOKING mandatory participation (one-to-one)

v) Diagram using ERDPLUS as image (.png). Include your company or business name in the main entity name.

vi) Describe Entity Relationship Diagram (a sentence for each entity and each relationship).



Example:

 Each user has a unique user id, a unique username, a password, user score, credits, full name, age, phone number, email address, street address, city, state province, zip code and country. (entity)

 Each manager manages one or many buildings. Each building is managed by exactly one manager. (relationship)

ENTITIES:

ENTITY 1 (CUSTOMER) Each customer has a unique customer Id (PK), an email, a password, a phone number, an address and his/her full name (first and last name).

ENTITY 2 (ADMINISTRATOR) Each administrator has a unique admin Id (PK), a name, a password and an email.

ENTITY 3 (ROOM) Each room has its fee, a unique room Id (PK), a name, the vacancy and a number.

ENTITY 4 (BOOKING) Each booking has a booking Id (PK), the check in date, the check out date and its status.

RELATIONSHIPS:

Relationship 1 (CustomerAdministration) Each administrator administrates at least one (one or more) customer. Each customer is administrated by exactly at least one (one or more) administrator.

Relationship 2 (Books) Each customer books exactly one room. Each room is booked by exactly one customer.

Relationship 3 (RoomAdministration) Each administrator administrates at least one (one or more) room. Each room is administrated by exactly one administrator.

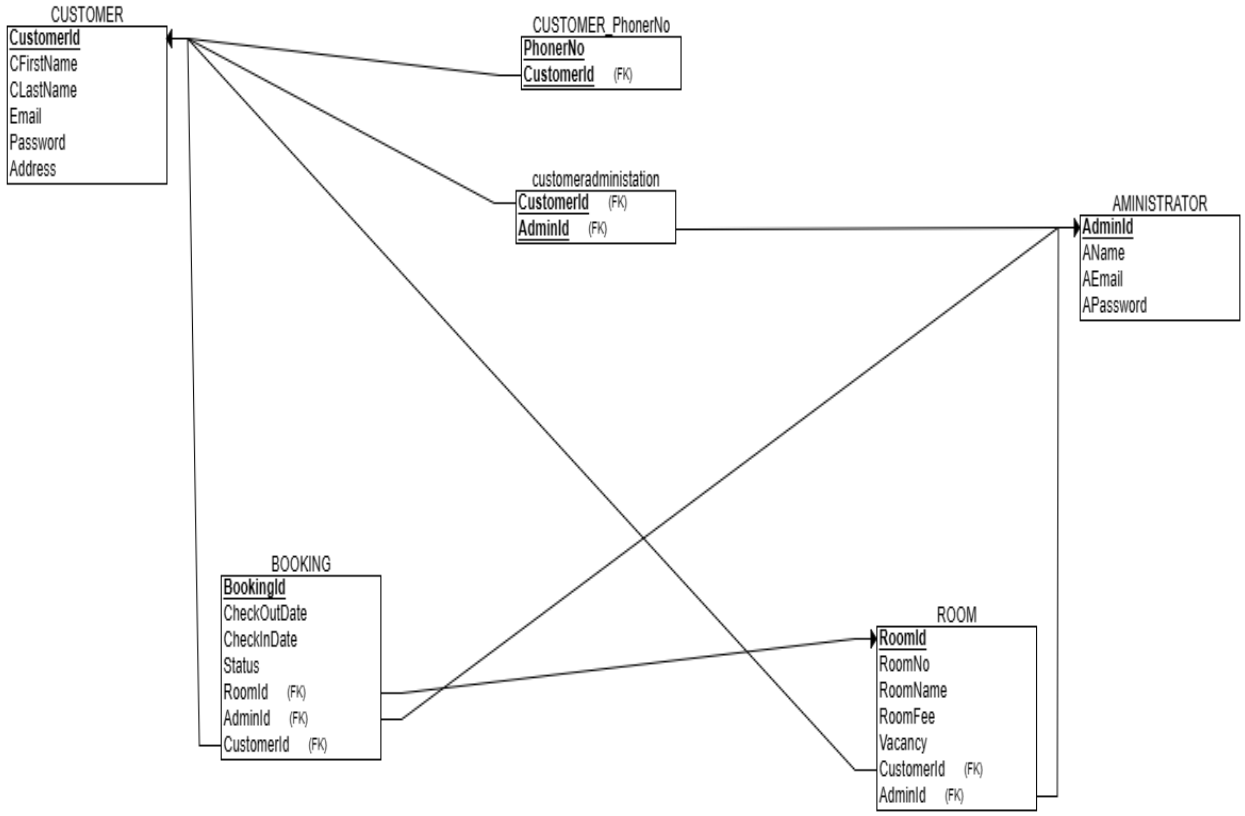
Relationship 4 (Available) Each room is available by exactly one booking. Each booking is available by exactly one room.

Relationship 5 (BookingAdministration) Each administrator administrates at least one (one or more) booking. Each booking is administrated by exactly one administrator.

Relationship 6 (Booking) Each customer makes exactly one booking. Each booking is made by exactly one customer.

PART III

THE RELATIONAL SCHEMA:



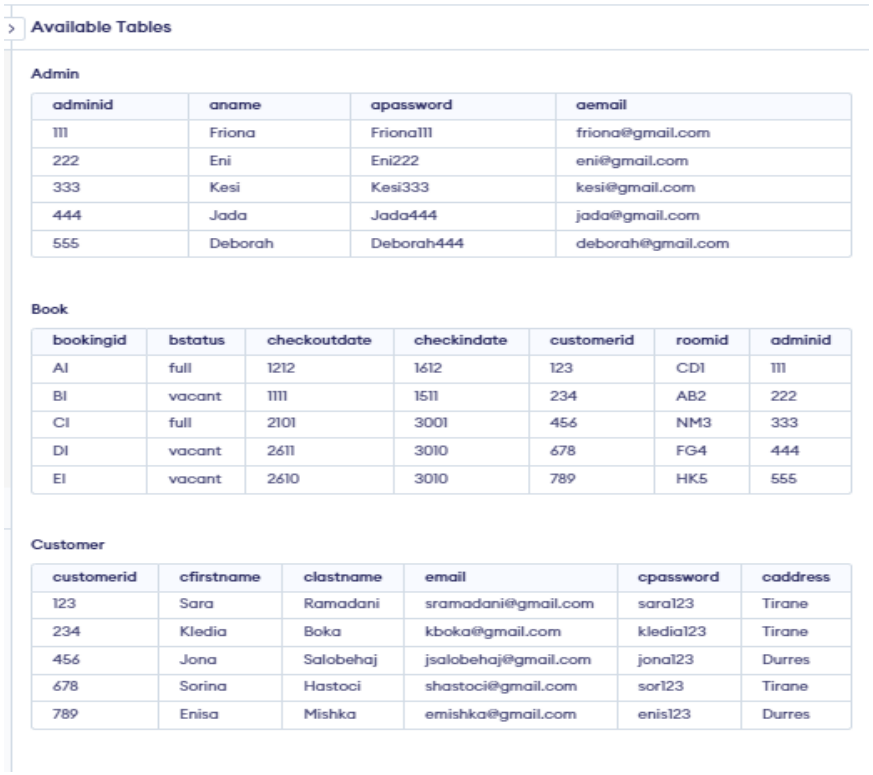
The SQL Quaries to Create and populate the tables that later will be used to slect the needed information from the database:

```
CREATE TABLE admin
(adminid INT(30) NOT NULL,
  aname CHAR(30) NOT NULL,
  apassword VARCHAR(30) NOT NULL,
  aemail VARCHAR(30) NOT NULL,
  PRIMARY KEY (adminid));


INSERT INTO admin VALUES (111, 'Friona', 'Friona111','friona@gmail.com');
INSERT INTO admin VALUES (222, 'Eni', 'Eni222', 'eni@gmail.com');
INSERT INTO admin VALUES (333, 'Kesi', 'Kesi333', 'kesi@gmail.com');
INSERT INTO admin VALUES (444, 'Jada', 'Jada444', 'jada@gmail.com');
INSERT INTO admin VALUES (555,'Deborah','Deborah444','deborah@gmail.com');


CREATE TABLE customer
(customerid INT(30) NOT NULL,
  cfirstname CHAR(30) NOT NULL,
```

```
  clastname CHAR(30) NOT NULL,
  email VARCHAR(30) NOT NULL,
  cpassword VARCHAR(30) NOT NULL,
  caddress CHAR(30) NOT NULL,
  PRIMARY KEY (customerid));

INSERT INTO customer VALUES (123, 'Sara', 'Ramadani', 'sramadani@gmail.com', 'sara123',
'Tirane');
INSERT INTO customer VALUES (234, 'Kledia', 'Boka', 'kboka@gmail.com' , 'kledia123', 'Tirane');
INSERT INTO customer VALUES (456, 'Jona', 'Salobehaj', 'jsalobehaj@gmail.com','jona123',
'Durres');
INSERT INTO customer VALUES (678, 'Sorina', 'Hastoci', 'shastoci@gmail.com','sor123', 'Tirane');
INSERT INTO customer VALUES (789, 'Enisa', 'Mishka', 'emishka@gmail.com','enis123', 'Durres');

CREATE TABLE room
(roomid VARCHAR(30) NOT NULL,
roomfee INT(30) NOT NULL,
roomname VARCHAR(30) NOT NULL,
vacancy VARCHAR(30) NOT NULL,
roomno INT(30) NOT NULL,
customerid INT REFERENCES customer(customerid),
CONSTRAINT roomidPK PRIMARY KEY (roomid));

INSERT INTO room VALUES ('CD1', 350, 'E301', 'empty', 301, 123, 111);
INSERT INTO room VALUES ('AB2', 400, 'E302', 'notempty', 302, 234, 222);
INSERT INTO room VALUES ('NM3', 450, 'E303', 'empty', 303, 456, 333);
INSERT INTO room VALUES ('FG4', 360, 'E304', 'notempty', 304, 678, 444);
INSERT INTO room VALUES ('HK5', 470, 'E305', 'empty', 305, 789, 555);

CREATE TABLE book
(bookingid CHAR(30) NOT NULL,
bstatus VARCHAR(30) NOT NULL,
checkoutdate INT(30) NOT NULL,
checkindate INT(30) NOT NULL,
customerid INT REFERENCES customer(customerid),
roomid INT REFERENCES room(roomid),
```

```
adminid INT REFERENCES admin(adminid),
CONSTRAINT bookingidPK PRIMARY KEY (bookingid));

INSERT INTO book VALUES ('AI', 'full', 1212, 1612, 123, 'CD1', 111);
INSERT INTO  book VALUES ('BI', 'vacant', 1111, 1511, 234,  'AB2', 222);
INSERT INTO  book VALUES ('CI', 'full', 2101, 3001, 456, 'NM3', 333 );
INSERT INTO  book VALUES ('DI', 'vacant', 2611, 3010, 678, 'FG4', 444);
INSERT INTO  book VALUES ('EI', 'vacant', 2610, 3010, 789, 'HK5', 555);

CREATE TABLE customeradministration
( customerid INT REFERENCES customer(customerid),
adminid INT REFERENCES admin(adminid));

INSERT INTO customeradministration VALUES (123, 111);
INSERT INTO customeradministration VALUES (234, 222);
 INSERT INTO customeradministration VALUES (456, 333);
 INSERT INTO customeradministration VALUES (678, 444);
 INSERT INTO customeradministration VALUES (789, 555);

CREATE TABLE phoneno
(phone INT(30) NOT NULL,
customerid INT REFERENCES customer(customerid),
CONSTRAINT phonePK PRIMARY KEY (phone));
```

Pictures From Programiz:



Available Tables

Admin

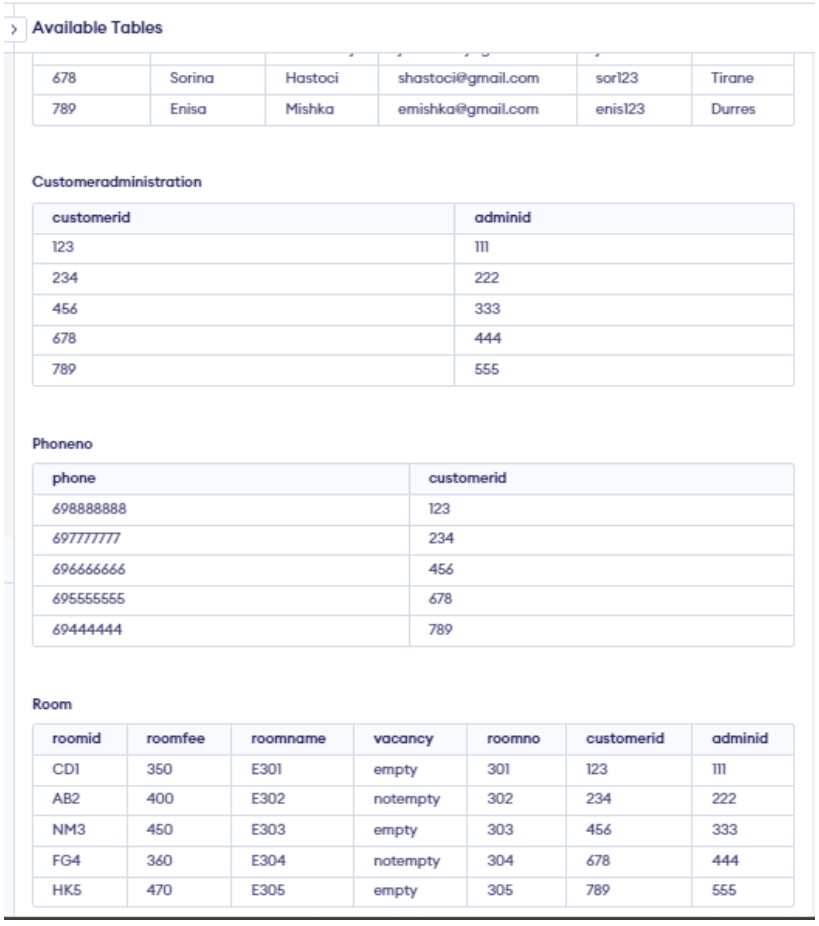
adminid	aname	apassword	aemail
111	Friona	Friona111	friona@gmail.com
222	Eni	Eni222	eni@gmail.com
333	Kesi	Kesi333	kesi@gmail.com
444	Jada	Jada444	jada@gmail.com
555	Deborah	Deborah444	deborah@gmail.com

Book

bookingid	bstatus	checkoutdate	checkindate	customerid	roomid	adminid
AI	full	1212	1612	123	CD1	111
BI	vacant	1111	1511	234	AB2	222
CI	full	2101	3001	456	NM3	333
DI	vacant	2611	3010	678	FG4	444
EI	vacant	2610	3010	789	HK5	555

Customer

customerid	cfirstname	clastname	email	cpassword	caddress
123	Sara	Ramadani	sramadani@gmail.com	sara123	Tirane
234	Kledia	Boka	kboka@gmail.com	kledia123	Tirane
456	Jona	Salobehaj	jsalobehaj@gmail.com	jona123	Durres
678	Sorina	Hastoci	shastoci@gmail.com	sor123	Tirane
789	Enisa	Mishka	emishka@gmail.com	enis123	Durres



Available Tables

678	Sorina	Hastoci	shastoci@gmail.com	sor123	Tirane
789	Enisa	Mishka	emishka@gmail.com	enis123	Durres

Customeradministration

customerid	adminid
123	111
234	222
456	333
678	444
789	555

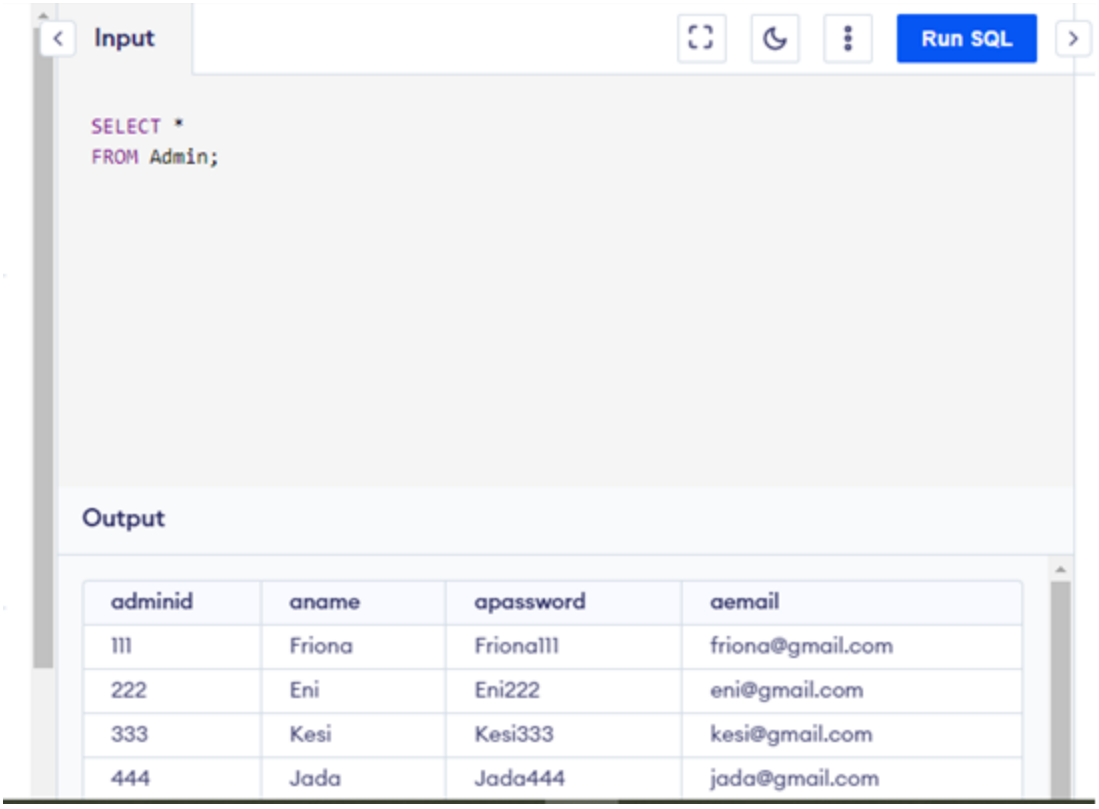
Phoneno

phone	customerid
698888888	123
697777777	234
696666666	456
695555555	678
69444444	789

Room

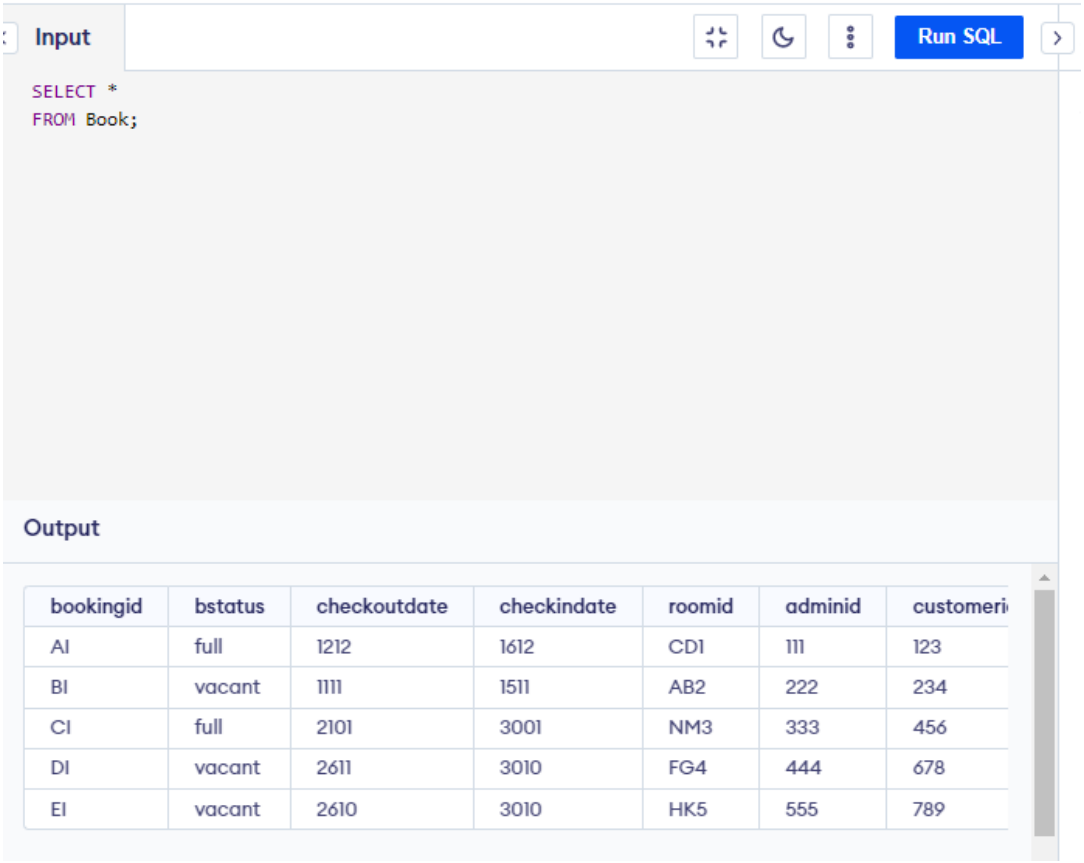
roomid	roomfee	roomname	vacancy	roomno	customerid	adminid
CD1	350	E301	empty	301	123	111
AB2	400	E302	notempty	302	234	222
NM3	450	E303	empty	303	456	333
FG4	360	E304	notempty	304	678	444
HK5	470	E305	empty	305	789	555

1. Display all the records in the table ADMIN.



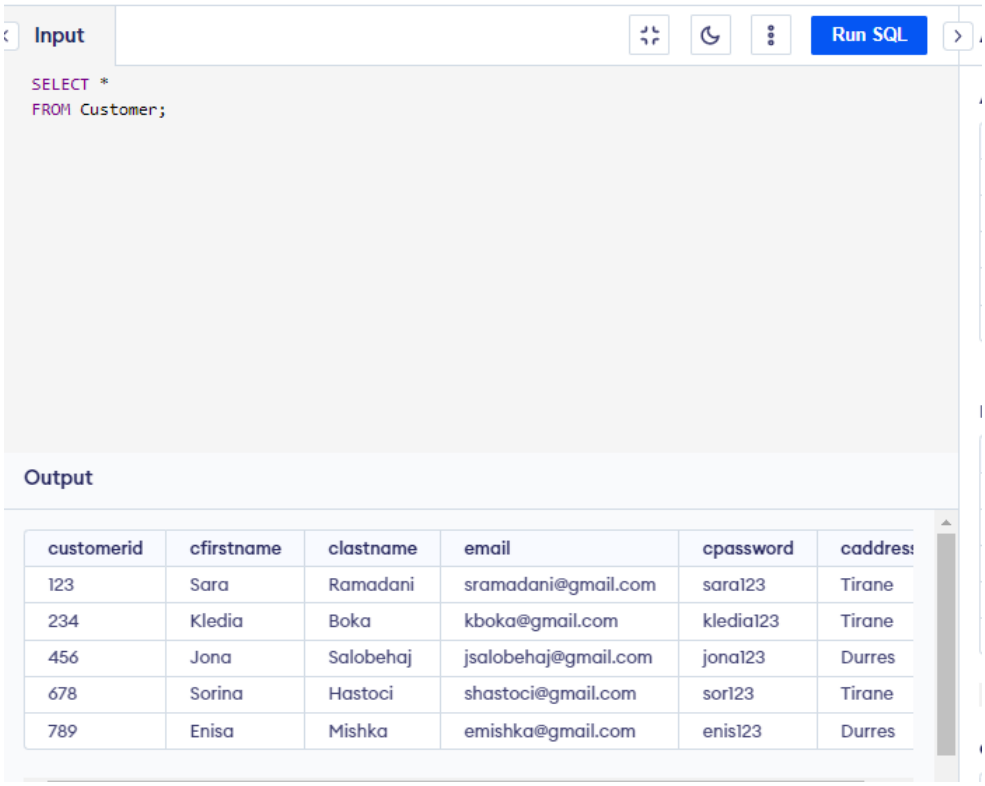
adminid	aname	apassword	aemail
111	Friona	Friona111	friona@gmail.com
222	Eni	Eni222	eni@gmail.com
333	Kesi	Kesi333	kesi@gmail.com
444	Jada	Jada444	jada@gmail.com

2. Display all the records in the table BOOK.



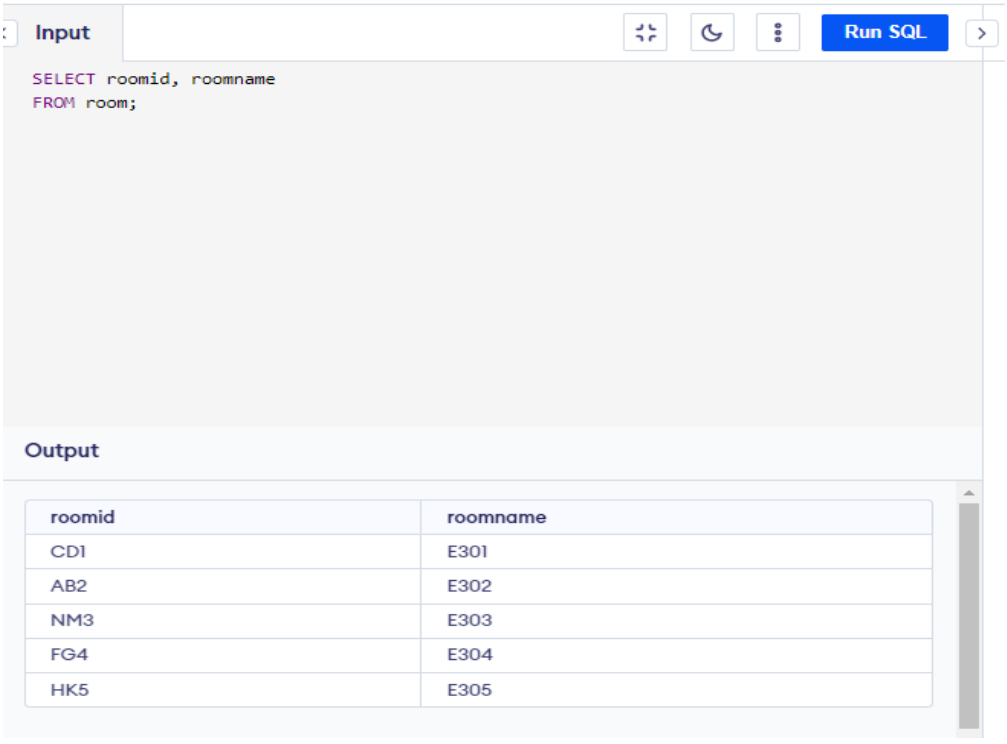
bookingid	bstatus	checkoutdate	checkindate	roomid	adminid	customeri
AI	full	1212	1612	CD1	111	123
BI	vacant	1111	1511	AB2	222	234
CI	full	2101	3001	NM3	333	456
DI	vacant	2611	3010	FG4	444	678
EI	vacant	2610	3010	HK5	555	789

3. Display all the records in the table CUSTOMER.



customerid	cfirstname	clastname	email	cpassword	caddres
123	Sara	Ramadani	sramadani@gmail.com	sara123	Tirane
234	Kledia	Boka	kboka@gmail.com	kledia123	Tirane
456	Jona	Salobehaj	jsalobehaj@gmail.com	jona123	Durres
678	Sorina	Hastoci	shastoci@gmail.com	sor123	Tirane
789	Enisa	Mishka	emishka@gmail.com	enis123	Durres

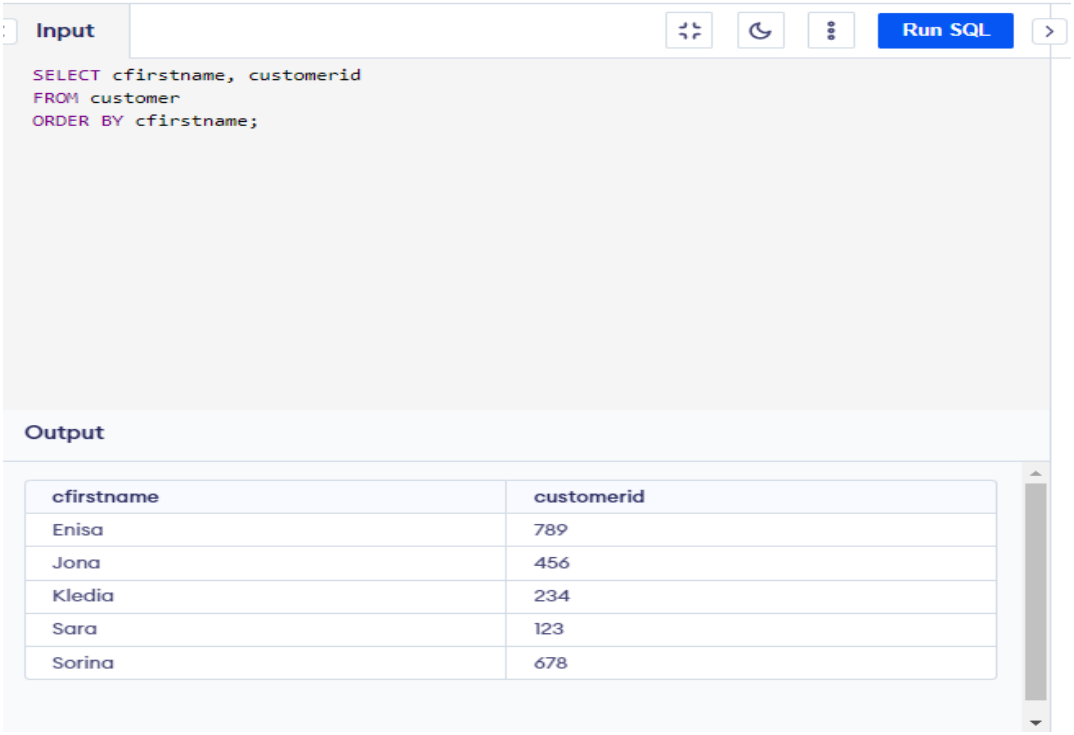
5. Display the RoomID and RoomName for all rooms.



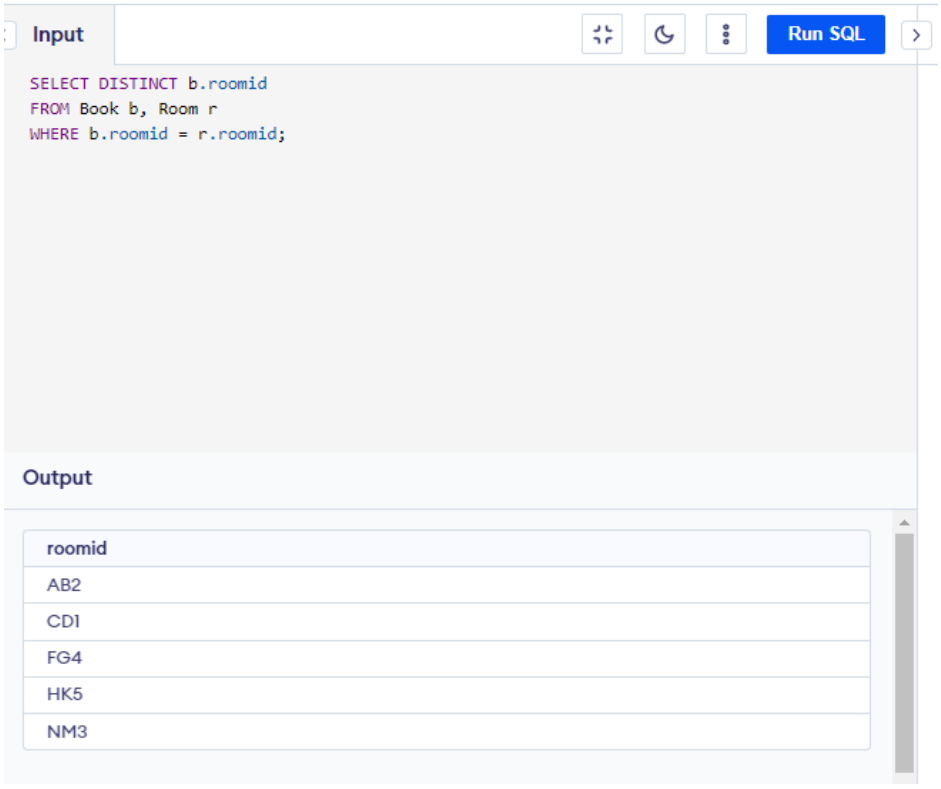
roomid	roomname
CD1	E301
AB2	E302
NM3	E303
FG4	E304
HK5	E305

6. Display the CustomerName and CustomerId for all customers, sorted alphabetically by

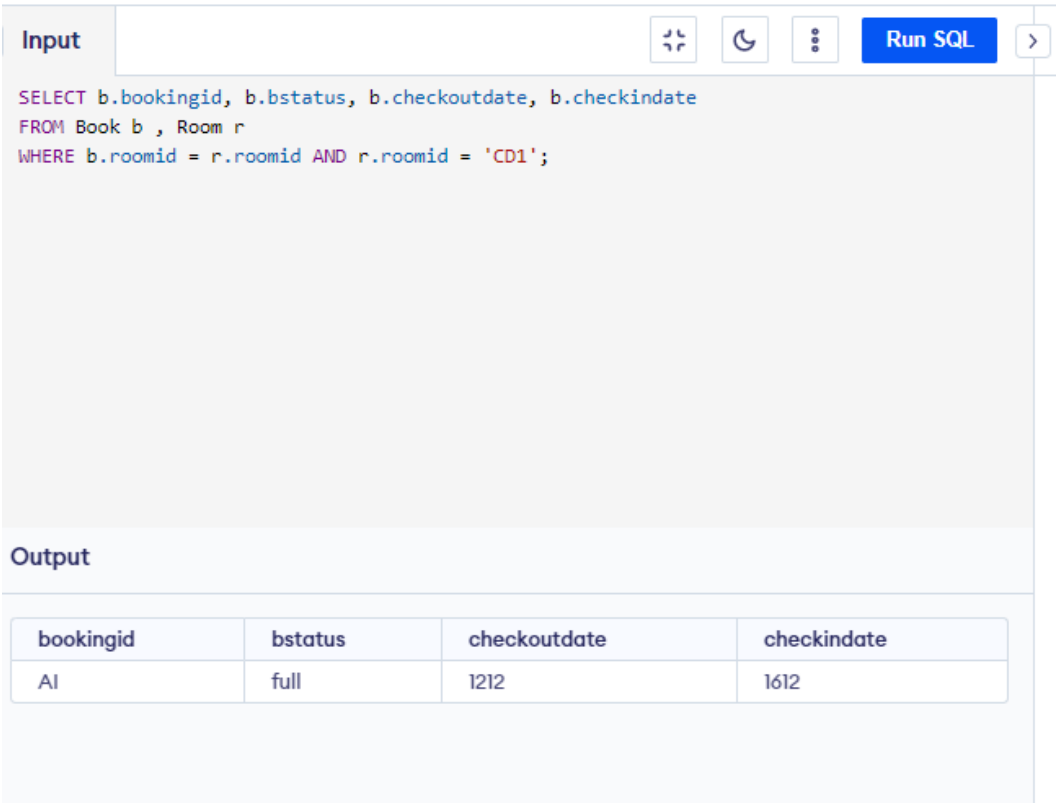
CustomerName.



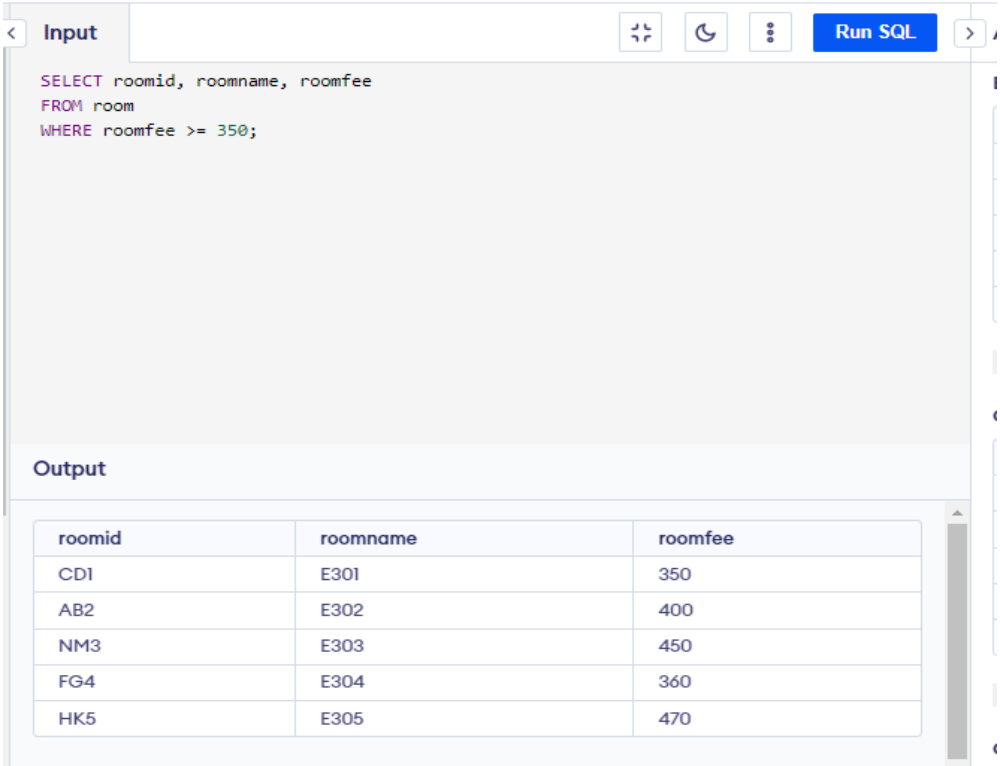
7. Display the IDs of room where we have bookings (use only table BOOKING and do not display the same information more than once).



8. Display all the information for all bookings whose RoomID value is CD1.



9. Display the RoomID, RoomName, and Roomfee for products with a Roomfee of $350 or higher.



10. Display Roomid, roomname, roomno and vacancy from room.

Input

Run SQL

```
SELECT roomid, roomname,vacancy,roomno
FROM room;
```

Output

roomid	roomname	vacancy	roomno
CD1	E301	empty	301
AB2	E302	notempty	302
NM3	E303	empty	303
FG4	E304	notempty	304
HK5	E305	empty	305

11. Display aname and adminid from admin , sorted alphabetically by aname

Input

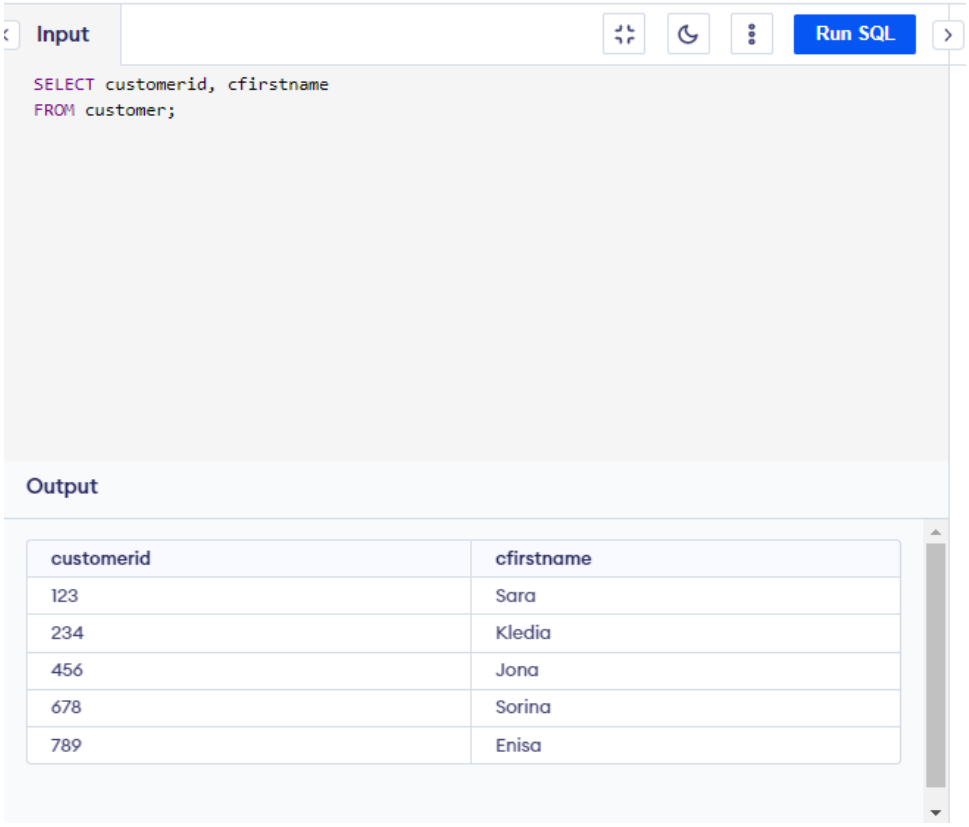
Run SQL

```
SELECT aname, adminid
FROM admin
ORDER BY aname;
```

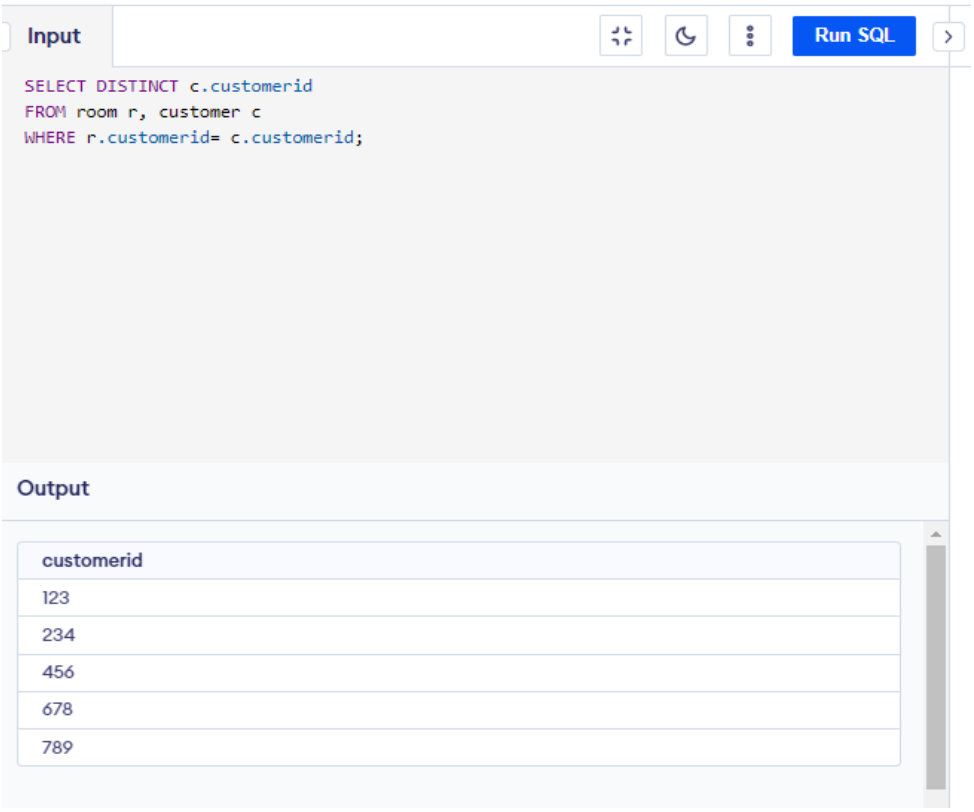
Output

aname	adminid
Deborah	555
Eni	222
Friona	111
Jada	444
Kesi	333

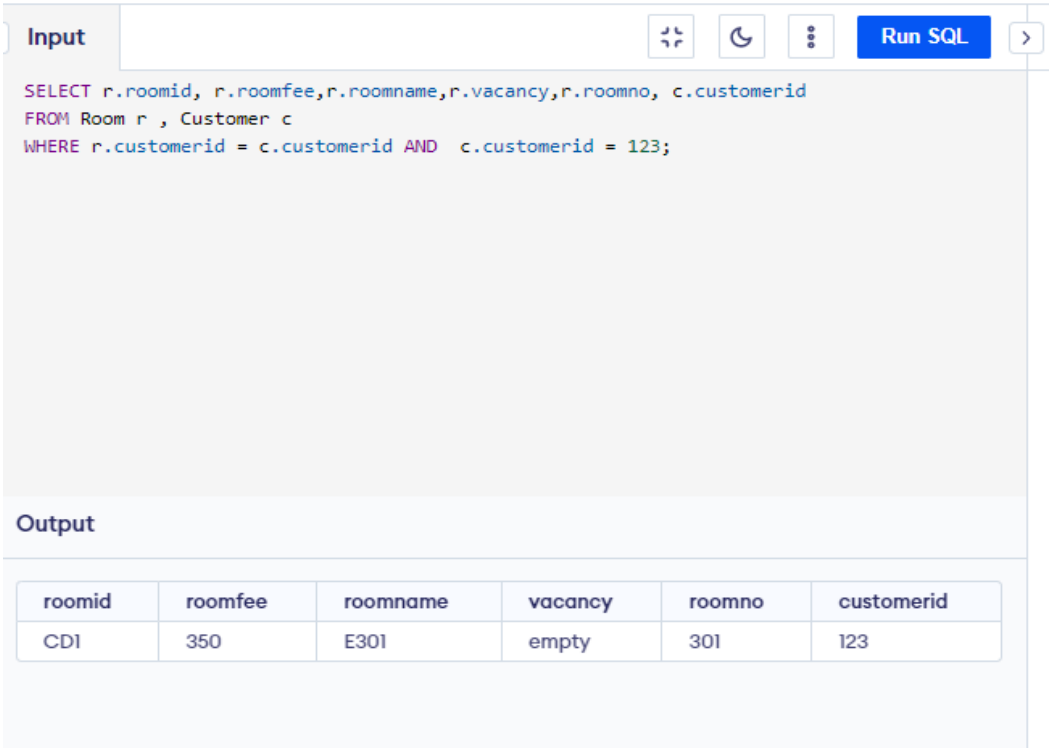
12.Display customerid and cfirstname from customer.



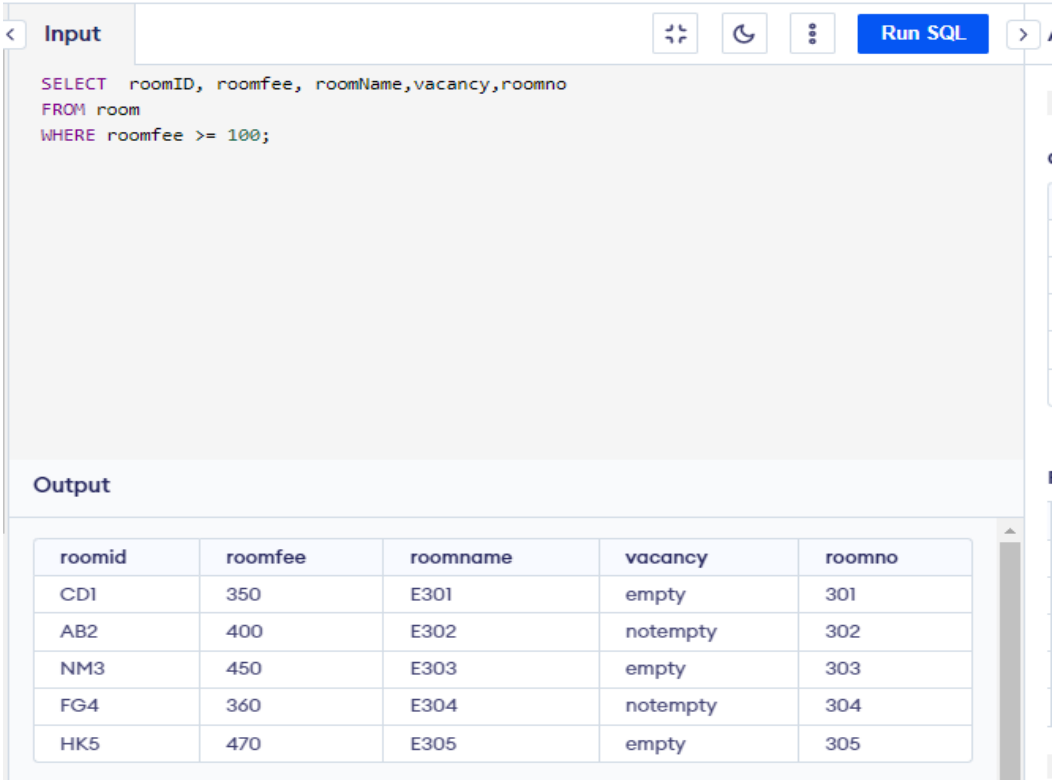
13. Display the IDs of customers where we have bookings (use only table BOOKING and do not display the same information more than once).



14. Display the RoomID, RoomName, Roomfee, vacancy, roomno and customerID for products with costumerID=123



15.Display roomID, roomfee, roomname, vacancy and roomno for rooms with roomfee equal to 100 or higher.



16.Display roomid, roomfee, roomname, vacancy, roomno and aname sorted by

Input

Run SQL

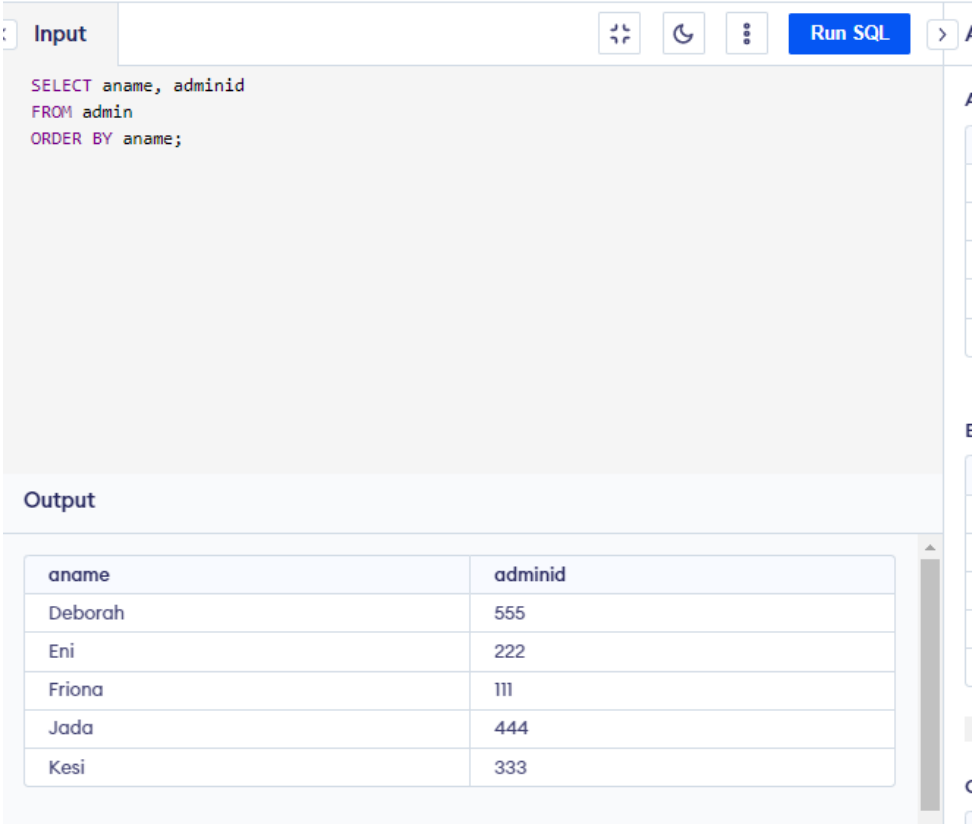
```
SELECT r.roomid, r.roomfee,r.roomname,r.vacancy,r.roomno,a.aname
FROM  room r, admin a
WHERE r.adminId = a.adminid
ORDER BY r.roomid;
```

Output

roomid	roomfee	roomname	vacancy	roomno	aname
AB2	400	E302	notempty	302	Eni
CD1	350	E301	empty	301	Friona
FG4	360	E304	notempty	304	Jada
HK5	470	E305	empty	305	Deborah
NM3	450	E303	empty	303	Kesi

roomid

Display aname and adminid sorted by aname.



18.Display bookigid, checkoutdate and checkindate.

Input

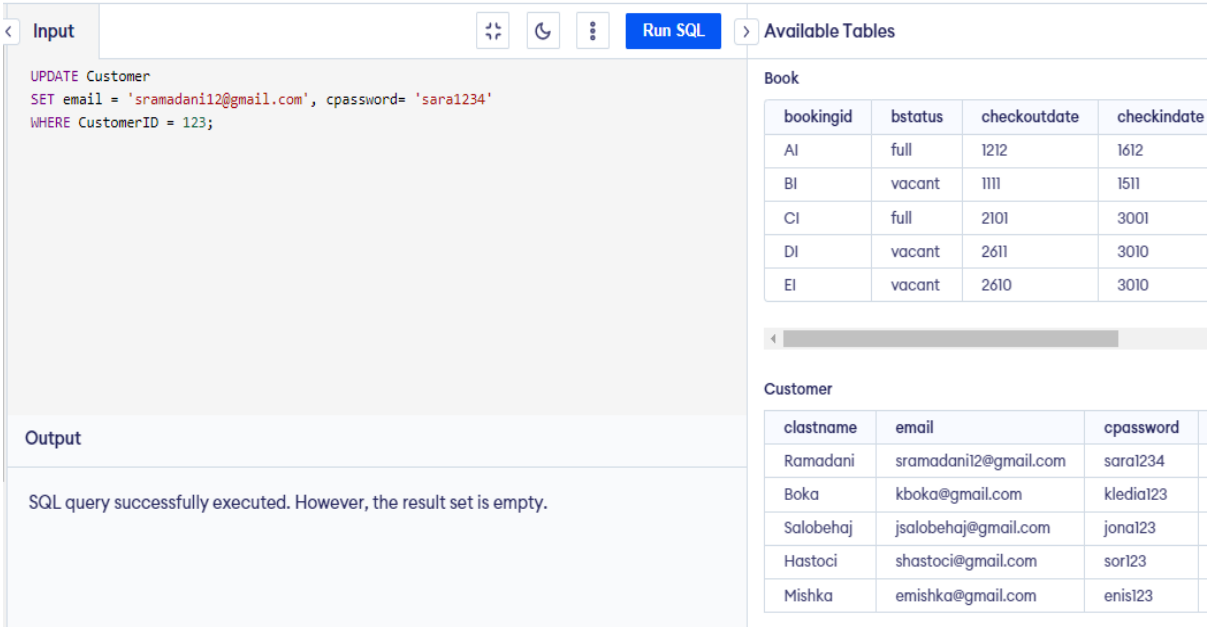
Run SQL

```
SELECT bookingid, checkoutdate, checkindate
FROM book;
```

Output

bookingid	checkoutdate	checkindate
AI	1212	1612
BI	1111	1511
CI	2101	3001
DI	2611	3010
EI	2610	3010

19. Display customer table updated by changing the email and password of one of the customers.



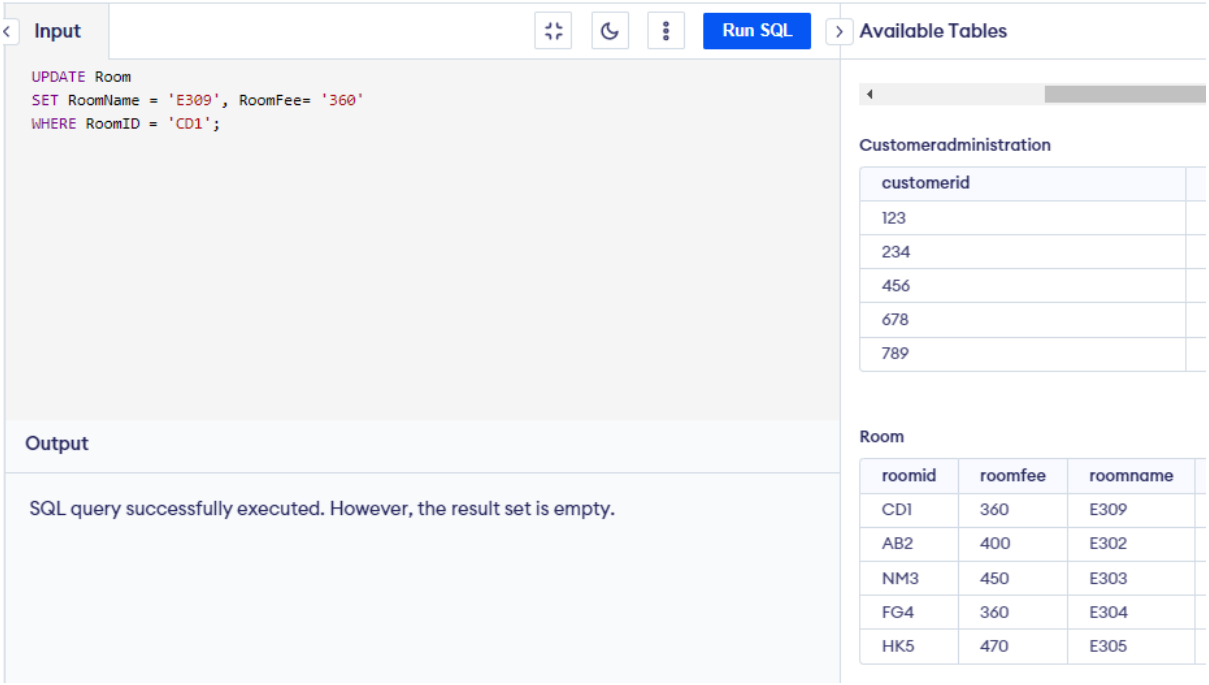
20.Display room table updated by changing the roomname and fee for all id=CD1

BEFORE

Room

roomname	vacancy	roomno	customerid	adminid
E301	empty	301	123	111
E302	notempty	302	234	222
E303	empty	303	456	333
E304	notempty	304	678	444
E305	empty	305	789	555

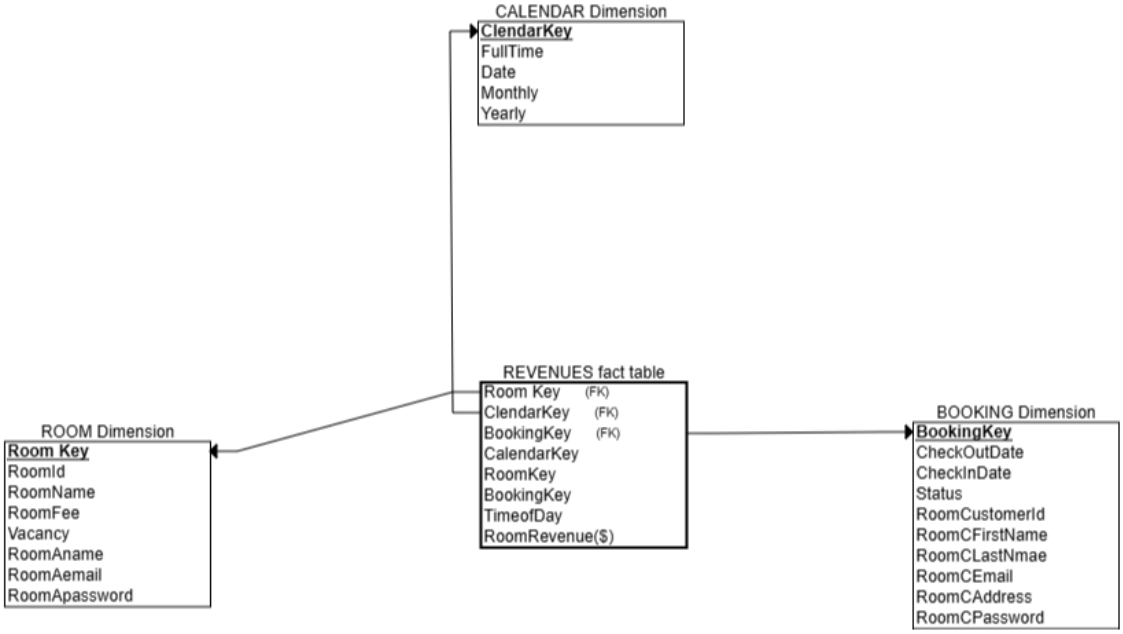
AFTER



PART IV:

STAR SCHEMA

Our Purpose: Create an analytical database (Datawarehouse) to analyse room booking and the revenues they bring in a specific period of time. So later we can analyse our busiest and most profitable months.


CALENDAR Dimension
ClendarKey
FullTime
Date
Monthly
Yearly
REVENUES fact table
Room Key (FK)
ClendarKey (FK)
BookingKey (FK)
CalendarKey
RoomKey
BookingKey
TimeofDay
RoomRevenue($)
ROOM Dimension
Room Key
RoomId
RoomName
RoomFee
Vacancy
RoomAname
RoomAemail
RoomApassword
BOOKING Dimension
BookingKey
CheckOutDate
CheckInDate
Status
RoomCustomerId
RoomCFirstName
RoomCLastNmae
RoomCEmail
RoomCAddress
RoomCPassword