DBMS LAB-OPEN ENDED

TOPIC:HOTEL MANAGEMENT DATABASE

TEAM MEMBERS:

HARSHITHA M M (4VV20CS049)

ESHANYE SRINIVAS (4VV20CS038)

H S SANJANA (4VV20CS045)

FRANKLIN (4VV20CS040)

PROBLEM STATEMENT

Consider a Hotel' website "The Hotel Palace" which has many rooms various categories of rooms like prime, normal. VIP Stay etc. Customers can signup/register to the website to book of their choice and pay online. Here are some assumptions with respect to this scenario. More than one customer can have the same name.

There may be many rooms available, but each can book only one room. Rooms are booked with ID proof, with details of the customers. Rooms can be booked by different customers on different dates for different fairs.

1. Design a database based on the scenario and details provided and write an SQL query for the following
2. Display the details of room along with customer who booked the room with the room no id 101.
3. Display the rooms that are not booked.
4. Display the details of room and its amenities along with the pay.

SCHEMA

- ROOM

ROOMID	ROOMTYPE	ROOMNUMBER	PRICE

- CUSTOMER

CUSTOMERID	CUSTOMERNAME	ADDRESS	IDPROFF

- BOOKING

BOOKINGID	CUSTIOMERID	ROOMID	BOOKINGDATE

- PAYMENT

PAYMENTID	BOOKINGID	CUSTOMERID

- AVAILABILITY

ROOMID	BOOKED

ER DIAGRAM

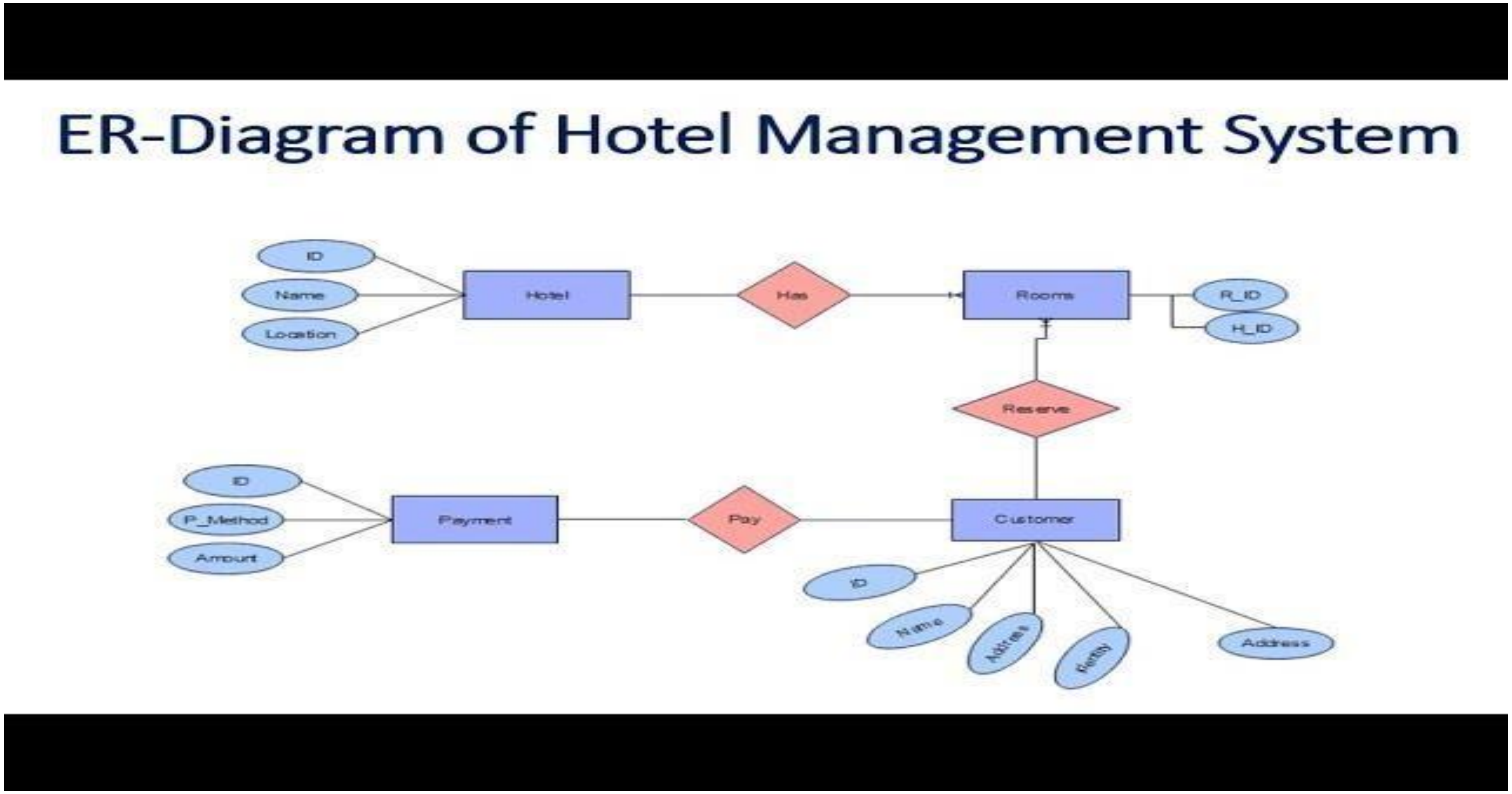
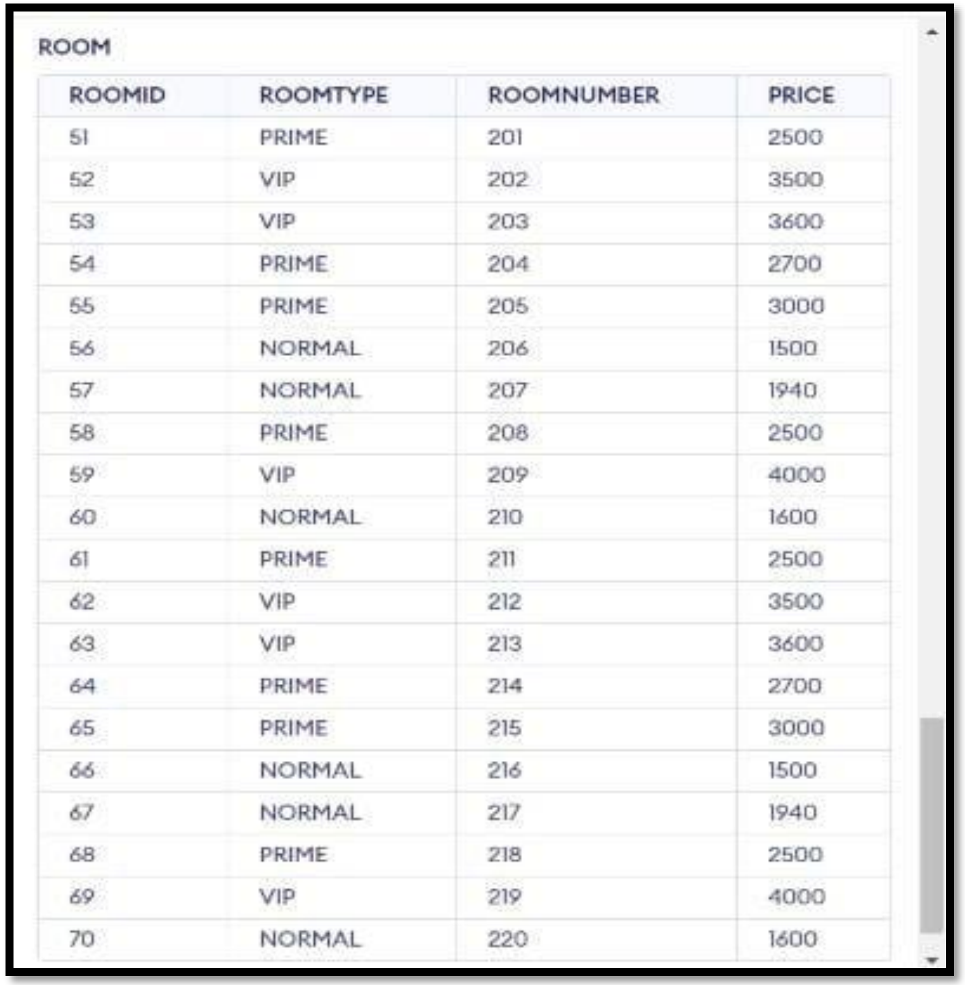

ER-Diagram of Hotel Management System
ID
Name
Location
Hotel
Has
Rooms
R_ID
H_ID
Reserve
ID
P_Method
Amount
Payment
Pay
Customer
ID
Name
Address
Identity
Address

TABLE CREATION

```
1. CREATE TABLE ROOM(
   ROOMID INT PRIMARY KEY,
   ROOMTYPE VARCHAR(10),
   ROOMNUMBER INT,
   PRICE INT);
```



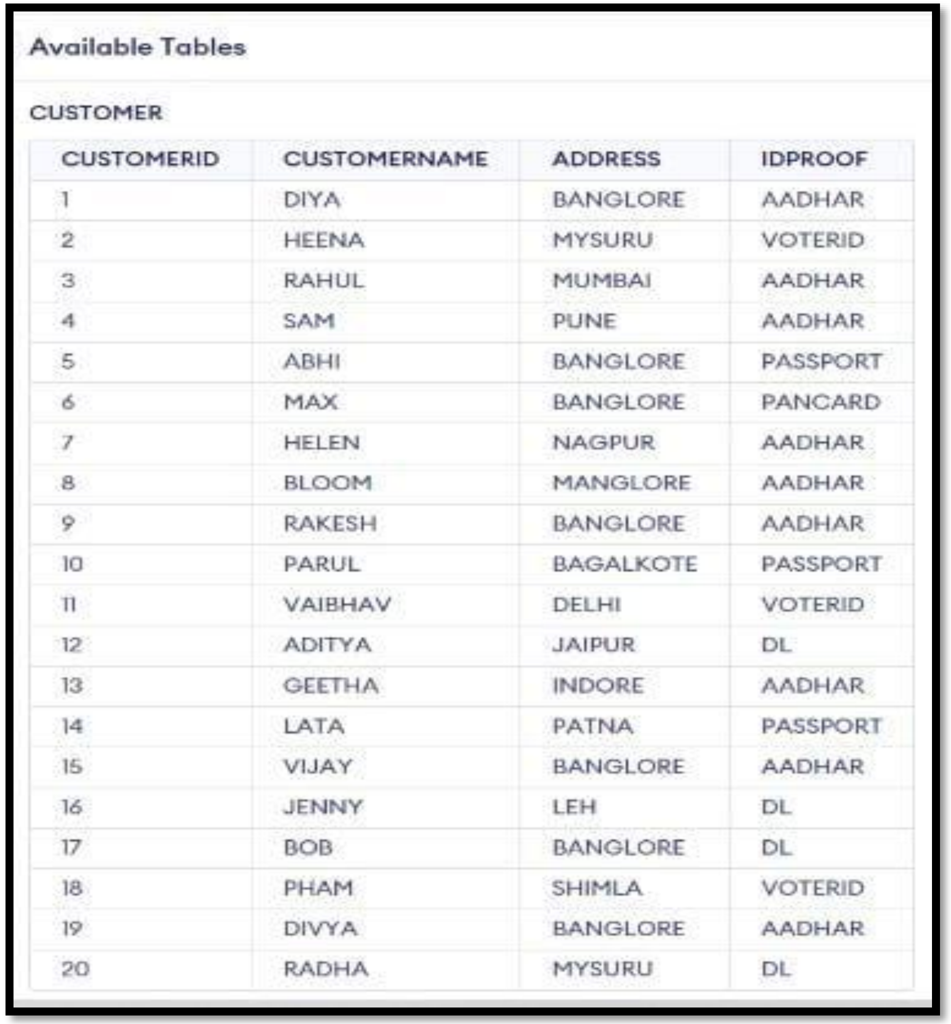
ROOM

ROOMID	ROOMTYPE	ROOMNUMBER	PRICE
51	PRIME	201	2500
52	VIP	202	3500
53	VIP	203	3600
54	PRIME	204	2700
55	PRIME	205	3000
56	NORMAL	206	1500
57	NORMAL	207	1940
58	PRIME	208	2500
59	VIP	209	4000
60	NORMAL	210	1600
61	PRIME	211	2500
62	VIP	212	3500
63	VIP	213	3600
64	PRIME	214	2700
65	PRIME	215	3000
66	NORMAL	216	1500
67	NORMAL	217	1940
68	PRIME	218	2500
69	VIP	219	4000
70	NORMAL	220	1600

```
INSERT INTO ROOM VALUES(&ROOMID,'&ROOMTYPE',&ROOMNUMBER,&PRICE);
```

TABLE CREATION

```
2. CREATE TABLE CUSTOMER(
    CUSTOMERID INT PRIMARY KEY,
    CUSTOMERNAME VARCHAR(10),
    ADDRESS VARCHAR(10),
    IDPROOF VARCHAR(10));
```



Available Tables

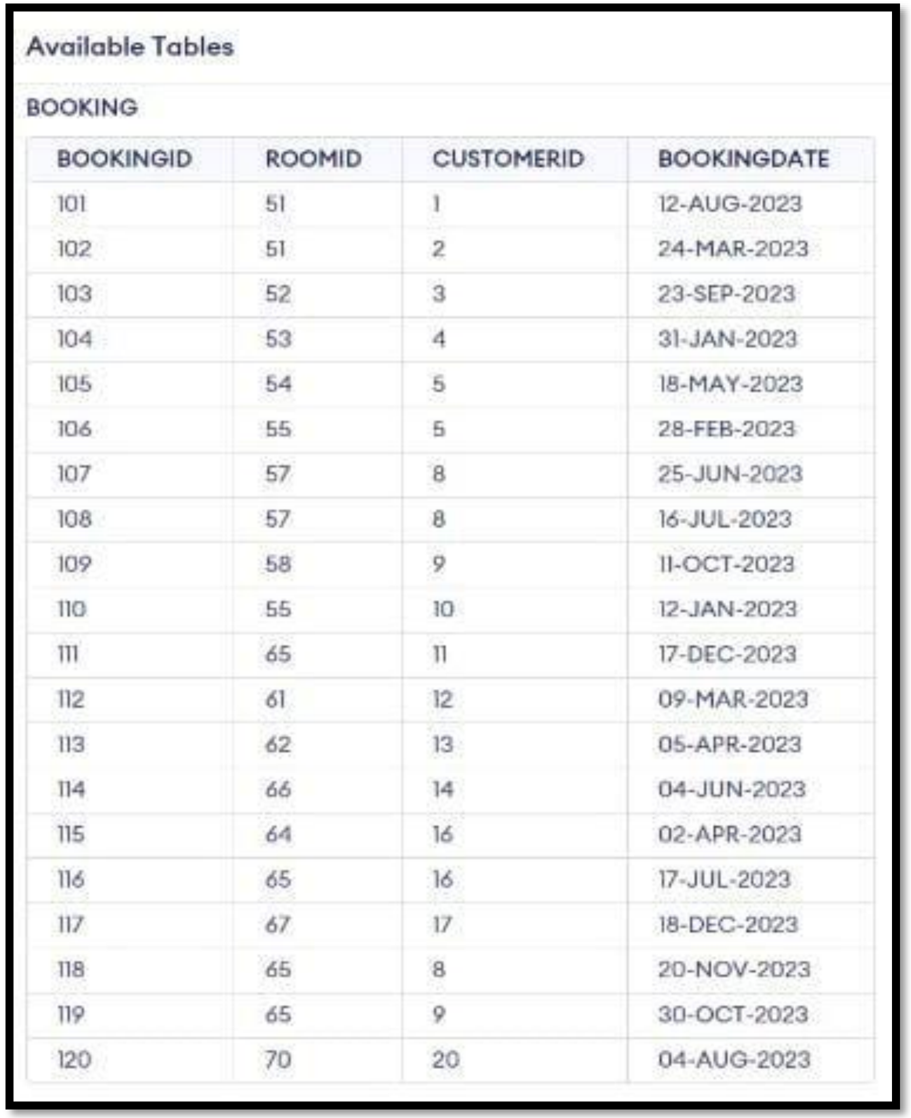
CUSTOMER

CUSTOMERID	CUSTOMERNAME	ADDRESS	IDPROOF
1	DIYA	BANGLORE	AADHAR
2	HEENA	MYSURU	VOTERID
3	RAHUL	MUMBAI	AADHAR
4	SAM	PUNE	AADHAR
5	ABHI	BANGLORE	PASSPORT
6	MAX	BANGLORE	PANCARD
7	HELEN	NAGPUR	AADHAR
8	BLOOM	MANGLORE	AADHAR
9	RAKESH	BANGLORE	AADHAR
10	PARUL	BAGALKOTE	PASSPORT
11	VAIBHAV	DELHI	VOTERID
12	ADITYA	JAIPUR	DL
13	GEETHA	INDORE	AADHAR
14	LATA	PATNA	PASSPORT
15	VIJAY	BANGLORE	AADHAR
16	JENNY	LEH	DL
17	BOB	BANGLORE	DL
18	PHAM	SHIMLA	VOTERID
19	DIVYA	BANGLORE	AADHAR
20	RADHA	MYSURU	DL

```
INSERT INTO CUSTOMER
VALUES(&CUSTOMERID,'&CUSTOMERNAME','&ADDRESS','&IDPROOF');
```

TABLE CREATION

```
3. CREATE TABLE BOOKING(
    BOOKINGID INT PRIMARY KEY,
    ROOMID REFERENCES ROOMID(ROOM)
    ON DELETE CASCADE,
   CUSTOMERID REFERENCES CUSTOMERID(CUSTOMER)
   ON DELETE CASCADE,
   BOOKINGDATE DATE);
```



Available Tables

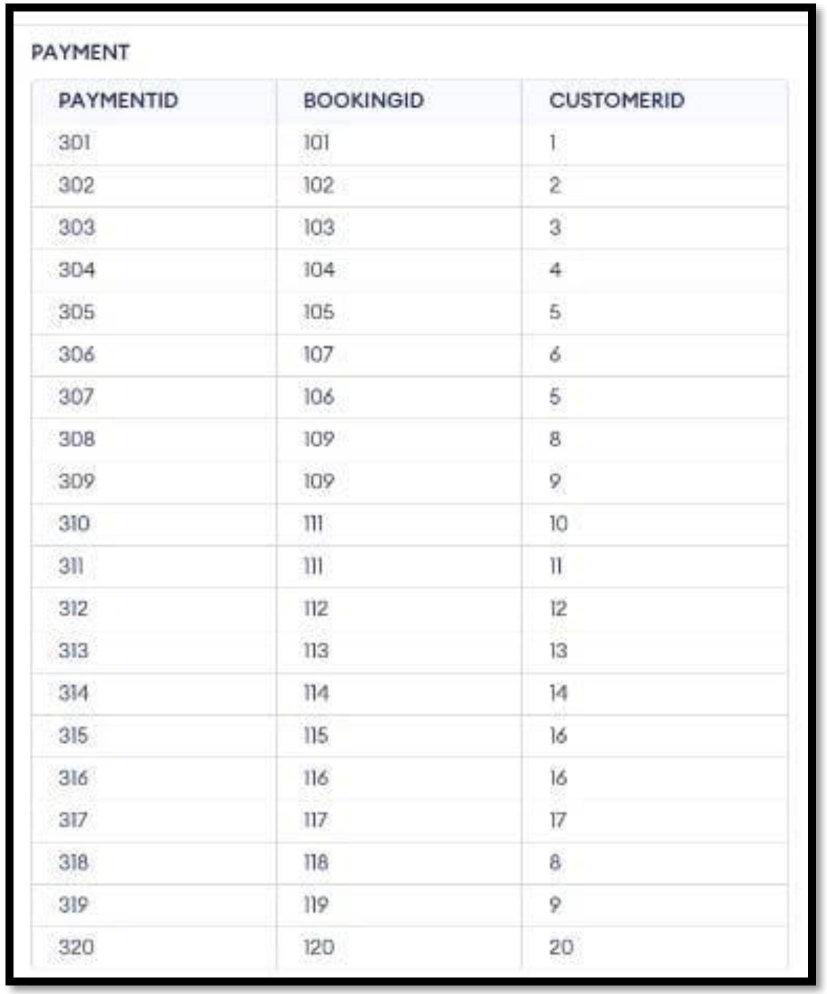
BOOKING

BOOKINGID	ROOMID	CUSTOMERID	BOOKINGDATE
101	51	1	12-AUG-2023
102	51	2	24-MAR-2023
103	52	3	23-SEP-2023
104	53	4	31-JAN-2023
105	54	5	18-MAY-2023
106	55	5	28-FEB-2023
107	57	8	25-JUN-2023
108	57	8	16-JUL-2023
109	58	9	11-OCT-2023
110	55	10	12-JAN-2023
111	65	11	17-DEC-2023
112	61	12	09-MAR-2023
113	62	13	05-APR-2023
114	66	14	04-JUN-2023
115	64	16	02-APR-2023
116	65	16	17-JUL-2023
117	67	17	18-DEC-2023
118	65	8	20-NOV-2023
119	65	9	30-OCT-2023
120	70	20	04-AUG-2023

```
INSERT INTO BOOKING
VALUES(&BOOKINGID,&ROOMID,&CUSTOMERID,'&BOOKINGDATE');
```

TABLE CREATION

```
4. CREATE TABLE PAYMENT(
    PAYMENTID INT PRIMARY KEY,
    BOOKINGID REFERENCES BOOKINGID(BOOKING)
    ON DELETE CASCADE,
    CUSTOMERID REFERENCES CUSTOMERID(CUSTOMER)
    ON DELETE CASCADE);
```



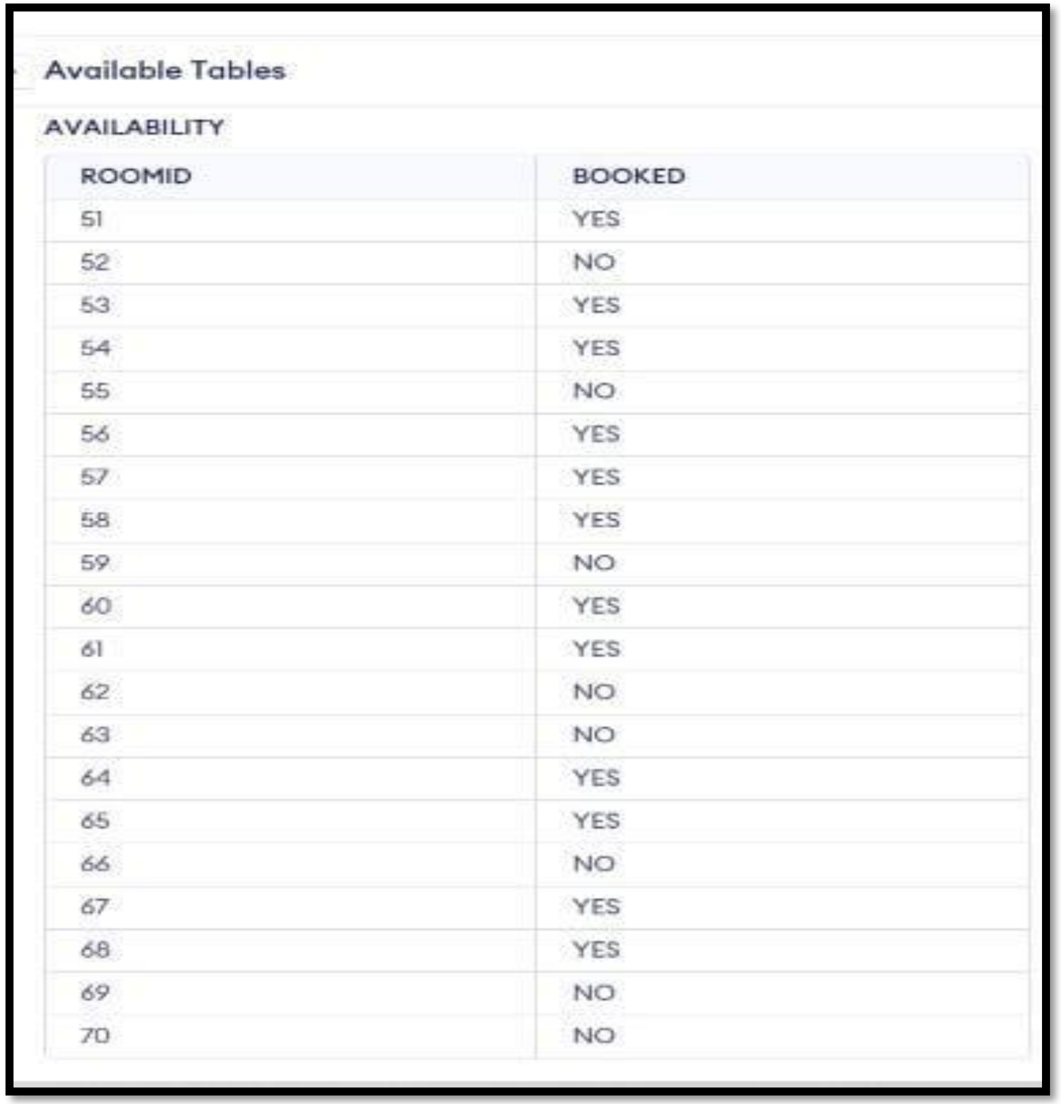
PAYMENT

PAYMENTID	BOOKINGID	CUSTOMERID
301	101	1
302	102	2
303	103	3
304	104	4
305	105	5
306	107	6
307	106	5
308	109	8
309	109	9
310	111	10
311	111	11
312	112	12
313	113	13
314	114	14
315	115	16
316	116	16
317	117	17
318	118	8
319	119	9
320	120	20

```
INSERT INTO PAYMENT VALUES(&PAYMENTID,&BOOKINGID,&CUSTOMERID);
```

TABLE CREATION

```
5. CREATE TABLE AVAILABILITY(
  ROOMID REFERENCES ROOM(ROOMID)
  ON DELETE SET NULL,
  BOOKED VARCHAR(6) );
```



Available Tables

AVAILABILITY

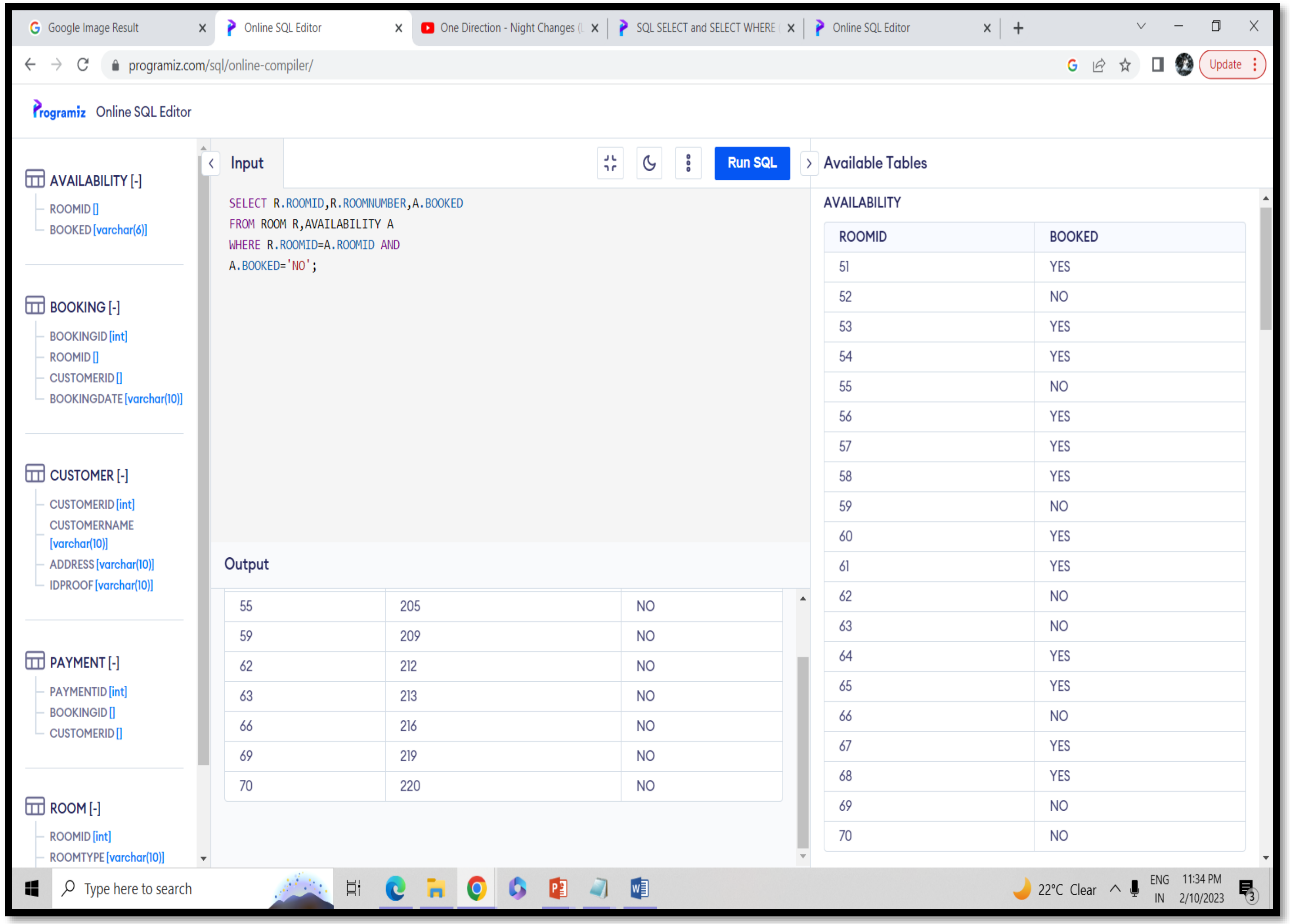
ROOMID	BOOKED
51	YES
52	NO
53	YES
54	YES
55	NO
56	YES
57	YES
58	YES
59	NO
60	YES
61	YES
62	NO
63	NO
64	YES
65	YES
66	NO
67	YES
68	YES
69	NO
70	NO

```
INSERT INTO AVAILABILITY VALUES(&ROOMID,'&BOOKED');
```

QUERIES

1.Display the details of room along with customer who booked the room with the room no id 101.

```
SELECT DISTINCT R.ROOMID,C.CUSTOMERID,
B.BOOKINGID,C.CUSTOMERNAME
FROM ROOM R,CUSTOMER C,BOOKING B,PAYMENT P
WHERE R.ROOMID=B.ROOMID AND
        C.CUSTOMERID=B.CUSTOMERID AND
        R.ROOMID='51';
```



AVAILABILITY [-]
- ROOMID []
- BOOKED [varchar(6)]

BOOKING [-]
- BOOKINGID [int]
- ROOMID []
- CUSTOMERID []
- BOOKINGDATE [varchar(10)]

CUSTOMER [-]
- CUSTOMERID [int]
- CUSTOMERNAME [varchar(10)]
- ADDRESS [varchar(10)]
- IDPROOF [varchar(10)]

PAYMENT [-]
- PAYMENTID [int]
- BOOKINGID []
- CUSTOMERID []

ROOM [-]
- ROOMID [int]
- ROOMTYPE [varchar(10)]

```
SELECT R.ROOMID,R.ROOMNUMBER,A.BOOKED
FROM ROOM R,AVAILABILITY A
WHERE R.ROOMID=A.ROOMID AND
A.BOOKED='NO';
```

Output

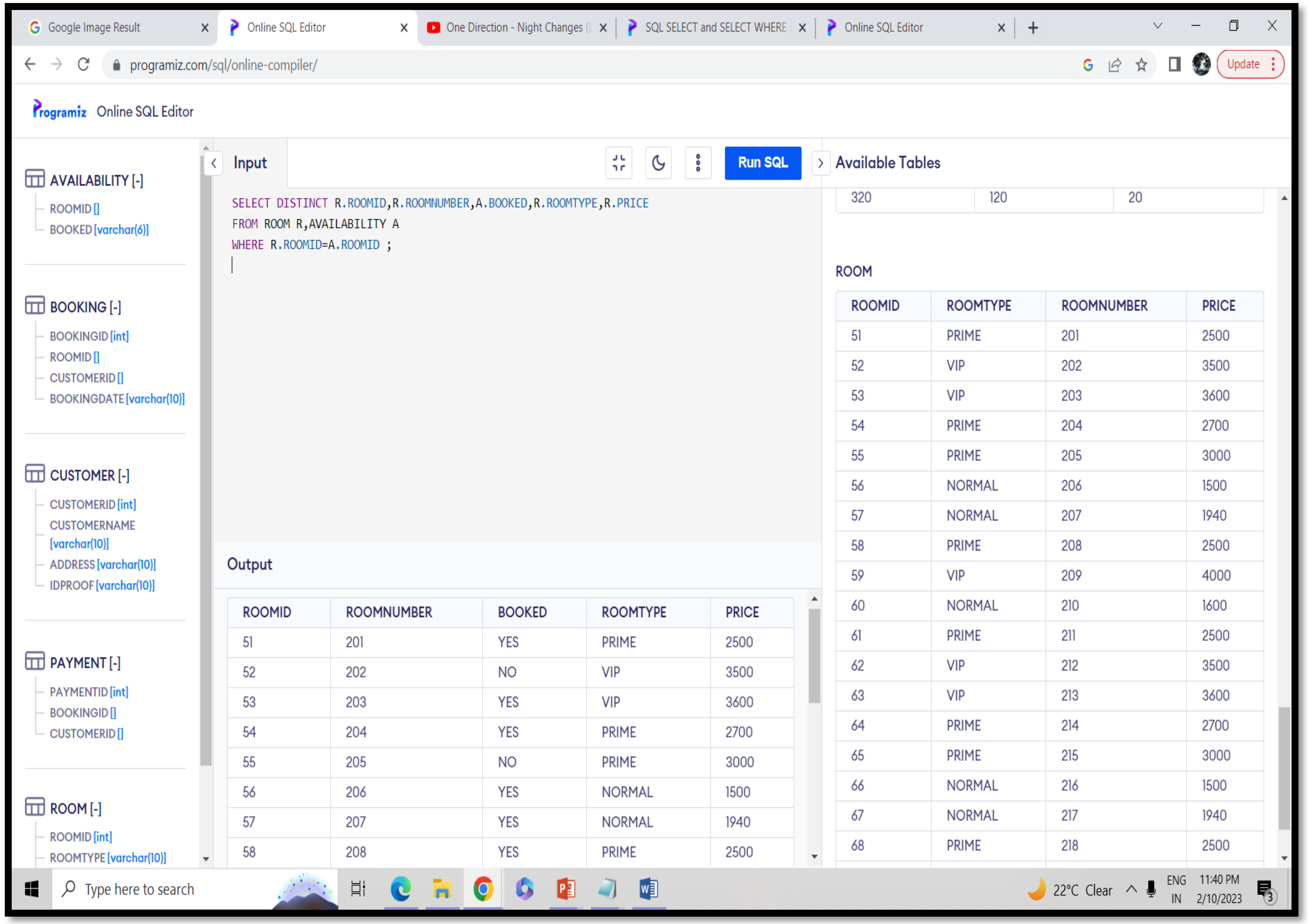
```
55    205    NO
59    209    NO
62    212    NO
63    213    NO
66    216    NO
69    219    NO
70    220    NO
```

AVAILABILITY

ROOMID	BOOKED
51	YES
52	NO
53	YES
54	YES
55	NO
56	YES
57	YES
58	YES
59	NO
60	YES
61	YES
62	NO
63	NO
64	YES
65	YES
66	NO
67	YES
68	YES
69	NO
70	NO

2. Display the rooms that are not booked.

```
SELECT R.ROOMID,R.ROOMNUMBER,A.BOOKED
FROM ROOM R,AVAILABILITY A
WHERE R.ROOMID=A.ROOMID AND
A.BOOKED='NO';
```



```
SELECT DISTINCT R.ROOMID,R.ROOMNUMBER,A.BOOKED,R.ROOMTYPE,R.PRICE
FROM ROOM R,AVAILABILITY A
WHERE R.ROOMID=A.ROOMID ;
```

Output

ROOMID	ROOMNUMBER	BOOKED	ROOMTYPE	PRICE
51	201	YES	PRIME	2500
52	202	NO	VIP	3500
53	203	YES	VIP	3600
54	204	YES	PRIME	2700
55	205	NO	PRIME	3000
56	206	YES	NORMAL	1500
57	207	YES	NORMAL	1940
58	208	YES	PRIME	2500

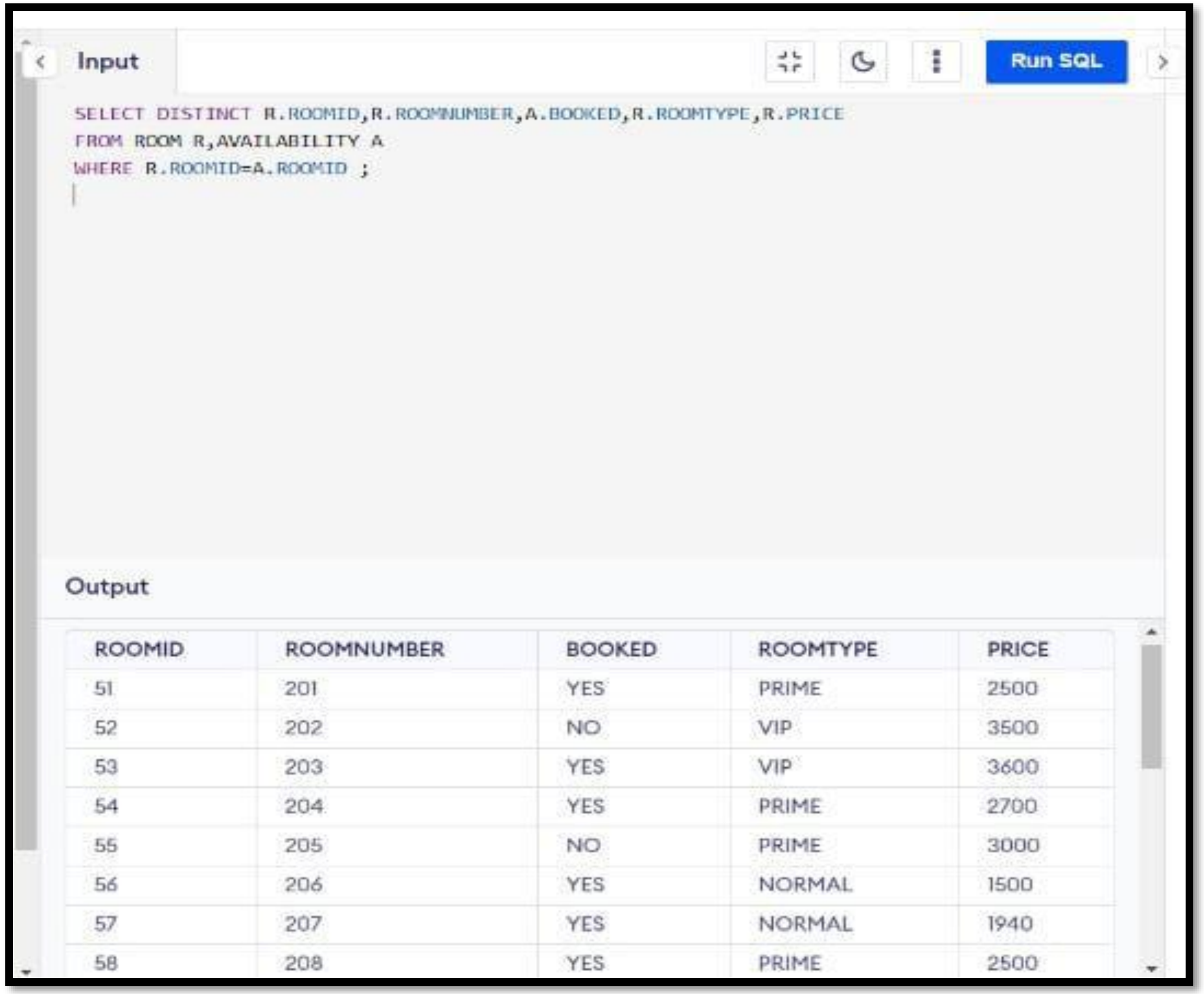
320	120	20

ROOM

ROOMID	ROOMTYPE	ROOMNUMBER	PRICE
51	PRIME	201	2500
52	VIP	202	3500
53	VIP	203	3600
54	PRIME	204	2700
55	PRIME	205	3000
56	NORMAL	206	1500
57	NORMAL	207	1940
58	PRIME	208	2500
59	VIP	209	4000
60	NORMAL	210	1600
61	PRIME	211	2500
62	VIP	212	3500
63	VIP	213	3600
64	PRIME	214	2700
65	PRIME	215	3000
66	NORMAL	216	1500
67	NORMAL	217	1940
68	PRIME	218	2500

3. Display the details of room and its amenities along with the pay.

```
SELECT DISTINCT R.ROOMID,R.ROOMNUMBER,
A.BOOKED,R.ROOMTYPE,R.PRICE
FROM ROOM R,AVAILABILITY A
WHERE R.ROOMID=A.ROOMID;
```



```
SELECT DISTINCT R.ROOMID,R.ROOMNUMBER,A.BOOKED,R.ROOMTYPE,R.PRICE
FROM ROOM R,AVAILABILITY A
WHERE R.ROOMID=A.ROOMID ;
```

Output

ROOMID	ROOMNUMBER	BOOKED	ROOMTYPE	PRICE
51	201	YES	PRIME	2500
52	202	NO	VIP	3500
53	203	YES	VIP	3600
54	204	YES	PRIME	2700
55	205	NO	PRIME	3000
56	206	YES	NORMAL	1500
57	207	YES	NORMAL	1940
58	208	YES	PRIME	2500

THANK YOU