# DBMS - MINI PROJECT
## TOPIC:Covid Bed Slot Booking

**Submitted By:**
Name:Shrutesh Reddy L
SRN:PES2UG20CS464
SECTION:H

**<u>Short  Description and Scope of the Project:</u>**
Covid Bed Slot Booking system enables a patient to book the bed through browser instead of coming to hospital.

In my project there will be admin, hospital user, user(patient).
The job of these roles are as follows:
1)Admin adds hospital user to the database.

2) Then hospital user logs in through his given credentials and adds hospital name, number of beds available for various types of beds namely:normal, HICU,ICU, ventilators beds.There is a section in my project called as operations where activities of hospitial user is tracked called as triggered data.

3) After that if patient has user account then he can login or else he has to sign up. User can then book bed by providing details like bed type,oxygen level,patient name, mobile number hospital code

**<u>Scope:</u>** As corona cases were increasing people will not be able to come to hospital to book bed slot. So there will be high demand for this application in future.

# ER Diagram
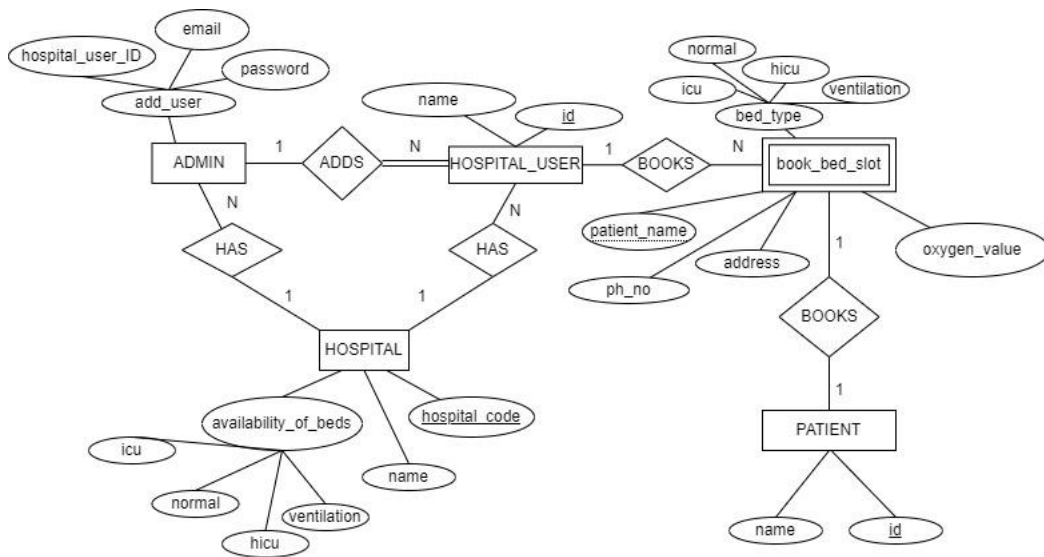


# Relational Schema

# SCHEMA DIAGRAM



PATIENT:

| name | patient_id |
|------|------------|

ADMIN:

| hospital_user_id | add_user | email | password | hosp_code |
|------------------|----------|-------|----------|-----------|

USER:

| name | id | hosp_code |
|------|-----|-----------|

HOSPITAL:

| name | hospital_code | availability_of_beds |
|------|---------------|----------------------|

BOOK_BED_SLOT:

| id | patient_id | ph_no | address | oxy_level | bed_type | patient_name |
|-----|------------|-------|---------|-----------|----------|--------------|

# DDL STATEMENTS - BUILDING THE DATABASE

## Create command:

## 1)Command for creating "bookingpatient" table:

```
CREATE TABLE `bookingpatient` (
  `id` int(11) NOT NULL,
  `srfid` varchar(50) NOT NULL,
  `bedtype` varchar(50) NOT NULL,
  `hcode` varchar(50) NOT NULL,
  `spo2` int(11) NOT NULL,
  `pname` varchar(50) NOT NULL,
  `pphone` varchar(12) NOT NULL,
  `paddress` text NOT NULL
)
```

## 2) Command for creating "hospitaldata" table:

```
CREATE TABLE `hospitaldata` (
  `id` int(11) NOT NULL,
  `hcode` varchar(200) NOT NULL,
  `hname` varchar(200) NOT NULL,
  `normalbed` int(11) NOT NULL,
  `hicubed` int(11) NOT NULL,
  `icubed` int(11) NOT NULL,
  `vbed` int(11) NOT NULL
)
```

## 3) Command for creating "hospitaluser" table:

```
CREATE TABLE `hospitaluser` (
  `id` int(11) NOT NULL,
  `hcode` varchar(20) NOT NULL,
  `email` varchar(100) NOT NULL,
  `password` varchar(1000) NOT NULL
)
```

## 4) Command for creating "trig" table:

```
CREATE TABLE `trig` (
  `id` int(11) NOT NULL,
  `hcode` varchar(50) NOT NULL,
  `normalbed` int(11) NOT NULL,
  `hicubed` int(11) NOT NULL,
  `icubed` int(11) NOT NULL,
  `vbed` int(11) NOT NULL,
  `querys` varchar(50) NOT NULL,
  `date` date NOT NULL
)
```

## 5) Command for creating "test" table:

```sql
CREATE TABLE `test` (
  `id` int(11) NOT NULL,
  `name` varchar(50) NOT NULL
)
```

## ALTER command:

```sql
ALTER TABLE `bookingpatient`
  ADD PRIMARY KEY (`id`),
  ADD UNIQUE KEY `srfid` (`srfid`(20));

ALTER TABLE `hospitaldata`
  ADD PRIMARY KEY (`id`),
  ADD UNIQUE KEY `hcode` (`hcode`);

ALTER TABLE `hospitaluser`
  ADD PRIMARY KEY (`id`);

ALTER TABLE `test`
  ADD PRIMARY KEY (`id`);

ALTER TABLE `trig`
  ADD PRIMARY KEY (`id`);

ALTER TABLE `user`
  ADD PRIMARY KEY (`id`),
  ADD UNIQUE KEY `srfid` (`srfid`);
```

## Populating the Database

## INSERT commands:

```sql
INSERT INTO `trig` (`id`, `hcode`, `normalbed`, `hicubed`, `icubed`, `vbed`,
`querys`, `date`) VALUES
(1, 'BBH01', 50, 9, 2, 1, ' UPDATED', '2021-11-26'),
(2, 'BBH01', 50, 9, 2, 1, ' DELETED', '2021-11-26'),

INSERT INTO `test` (`id`, `name`) VALUES
(1, 'anees'),
(2, 'rehman');
```

phpMyAdmin

Server: 127.0.0.1 » Database: bedbooking » Table: hospitaluser

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Tracking | More

Showing rows 0 - 2 (3 total, Query took 0.0009 seconds.)

SELECT * FROM `hospitaluser`

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

| | | | id | hcode | email | password |
|---|---|---|---|---|---|---|
| Edit | Copy | Delete | 15 | 1 | shruteshreddy2001@gmail.com | pbkdf2:sha256:260000$HI8c9S8A0NxETqzf$734f7f0c955f... |
| Edit | Copy | Delete | 16 | 1 | shruteshreddy2001@gmail.com | pbkdf2:sha256:260000$nRh8Bi9d2yn2op8f$2a8e9310a871... |
| Edit | Copy | Delete | 17 | 1 | shruteshreddy2001@gmail.com | pbkdf2:sha256:260000$7x0oSrPqv1XFetdb$f0366066af0b... |

Check all | With selected: Edit | Copy | Delete | Export

---



phpMyAdmin

Server: 127.0.0.1 » Database: bedbooking » Table: hospitaldata

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Tracking | More

Showing rows 0 - 0 (1 total, Query took 0.0006 seconds.)

SELECT * FROM `hospitaldata`

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

| | | | id | hcode | hname | normalbed | hicubed | icubed | vbed |
|---|---|---|---|---|---|---|---|---|---|
| Edit | Copy | Delete | 7 | 1 | shrutesh hospital | 12 | 5 | 4 | 2 |

Check all | With selected: Edit | Copy | Delete | Export

---



phpMyAdmin

Server: 127.0.0.1 » Database: bedbooking » Table: bookingpatient

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Tracking | More

Showing rows 0 - 0 (1 total, Query took 0.0005 seconds.)

SELECT * FROM `bookingpatient`

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

| | | | id | srfid | bedtype | hcode | spo2 | pname | pphone | paddress |
|---|---|---|---|---|---|---|---|---|---|---|
| Edit | Copy | Delete | 7 | LSR | NormalBed | 1 | | 2 abc | 123 | abc |

Check all | With selected: Edit | Copy | Delete | Export

---



Server: 127.0.0.1 » Database: bedbooking » Table: user

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Tracking | More

Showing rows 0 - 0 (1 total, Query took 0.0014 seconds.)

SELECT * FROM `user`

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

| | | | id | srfid | email | dob |
|---|---|---|---|---|---|---|
| Edit | Copy | Delete | 13 | LSR | shruteshreddy2001@gmail.com | pbkdf2:sha256:260000$PBbSl7dOviuKDCuf$f5df13bfea0a... |

Check all | With selected: Edit | Copy | Delete | Export

# Join Queries

Showcase at least 4 join queries.

Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results
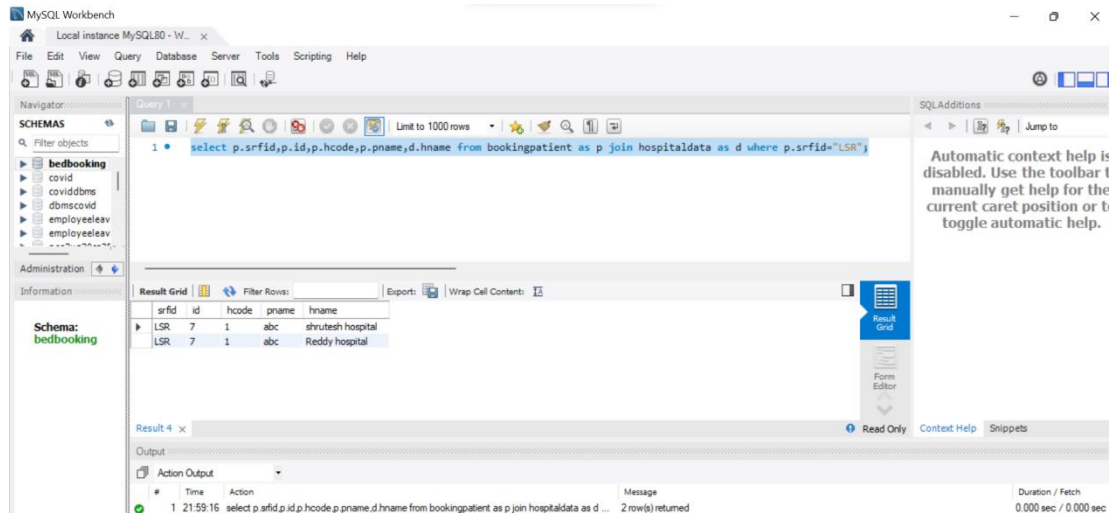
1) select p.hcode from bookingpatient as p join hospitaldata as d where p.id=d.id;

Selecting the columns from 2 tables where the attribute called ID is common among the two tables
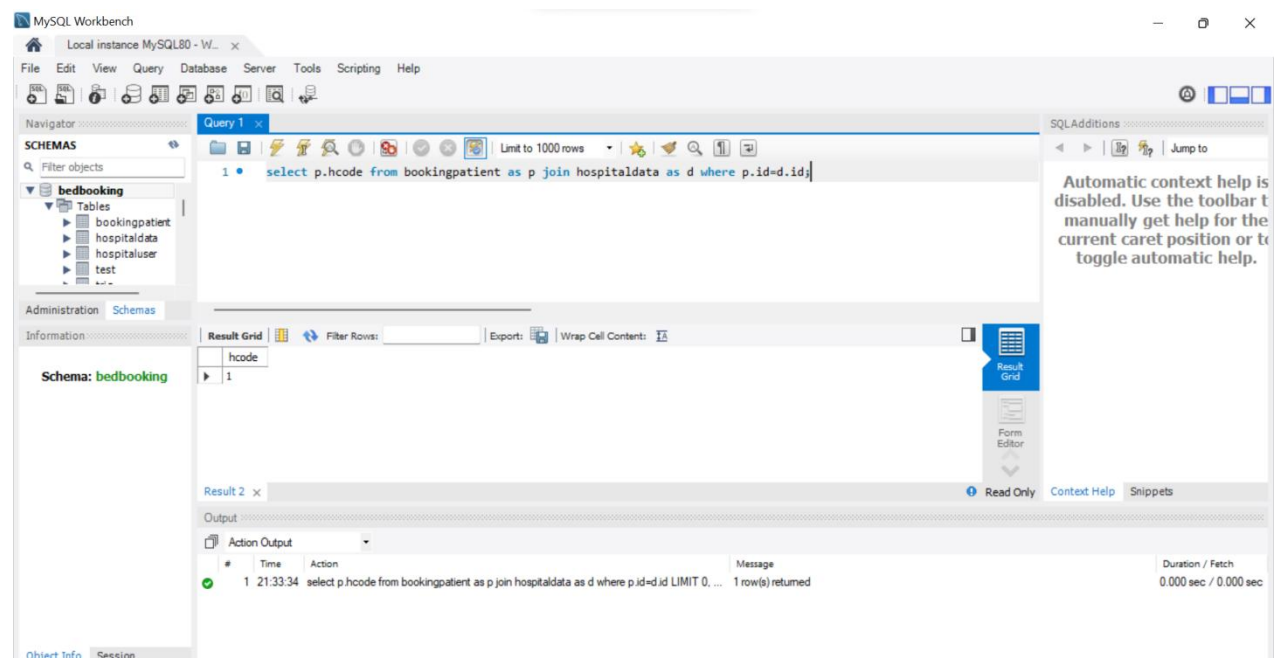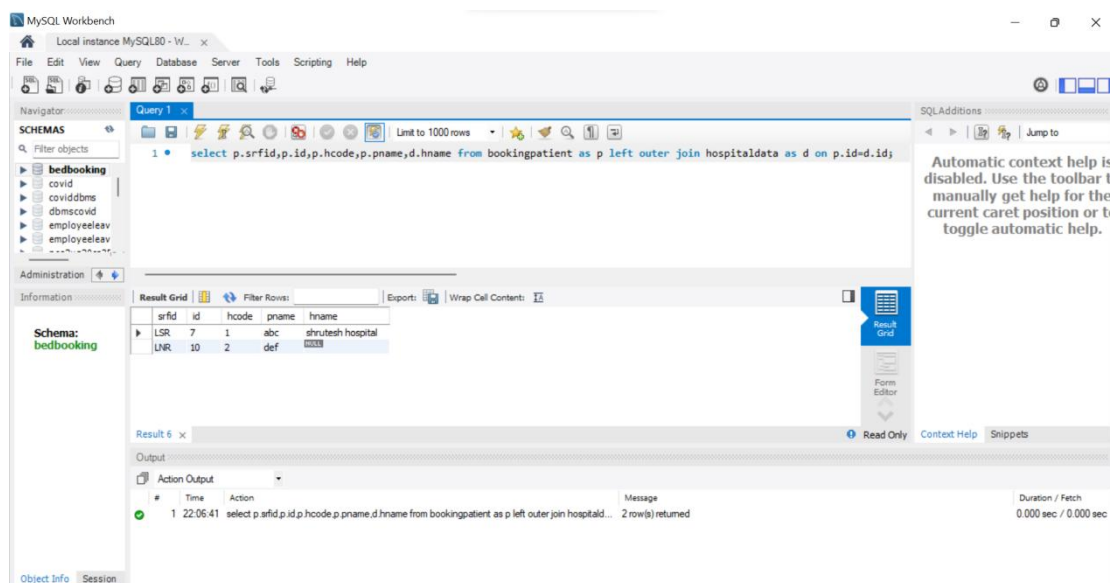


2) select p.srfid,p.id,p.hcode,p.pname,d.hname from bookingpatient as p join hospitaldata as d where p.srfid="LSR";

Selecting few attributes from both the tables where srfid=LSR.

3) select p.srfid,p.id,p.hcode,p.pname,d.hname from bookingpatient as p left outer join hospitaldata as d on p.id=d.id;

Selecting few attributes from the tables where ID is common among the two tables by performing left outer join.



4) select p.srfid,p.id,p.hcode,p.pname,d.hname from bookingpatient as p right outer join hospitaldata as d on p.id=d.id;

Selecting few attributes from the tables where ID is common among the two tables by performing right outer join.
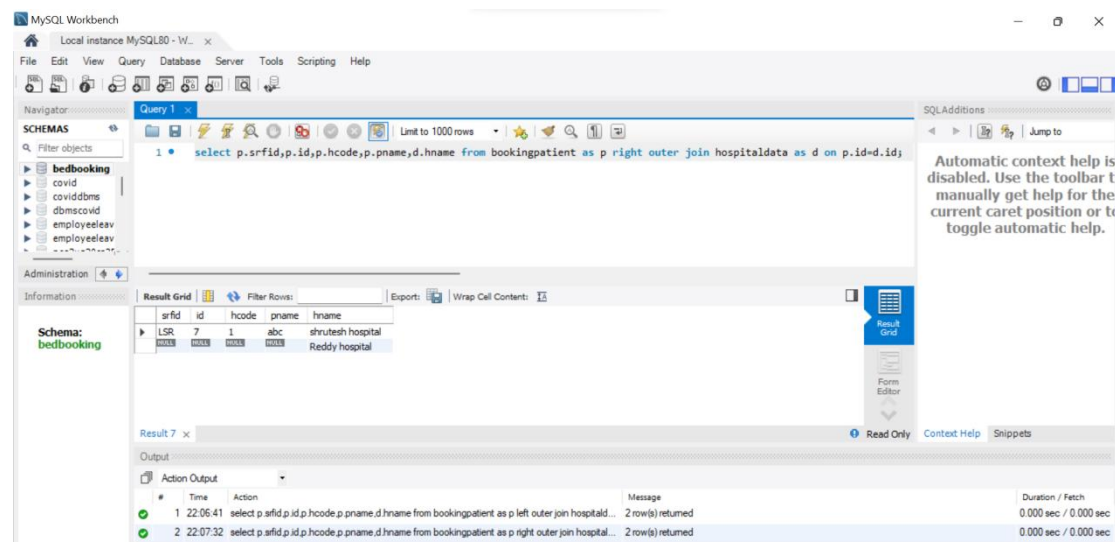


## Aggregate Functions

Showcase at least 4 Aggregate function queries
Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results.

1)   select id,avg(normalbed) from hospitaldata group by id;

Retrieving the average of the column called noemalbed.
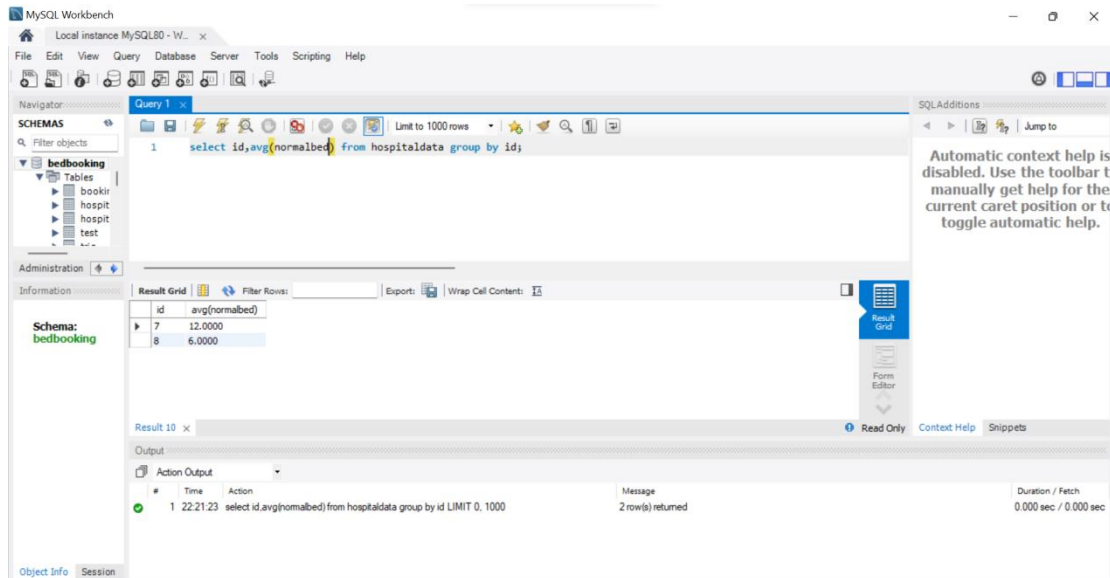
2) select id,bookingpatient.pname,count(*) from bookingpatient where id in(7,10) group by id;

Retrieving the attributes where ID lies in between 7 to 10 grouped by ID
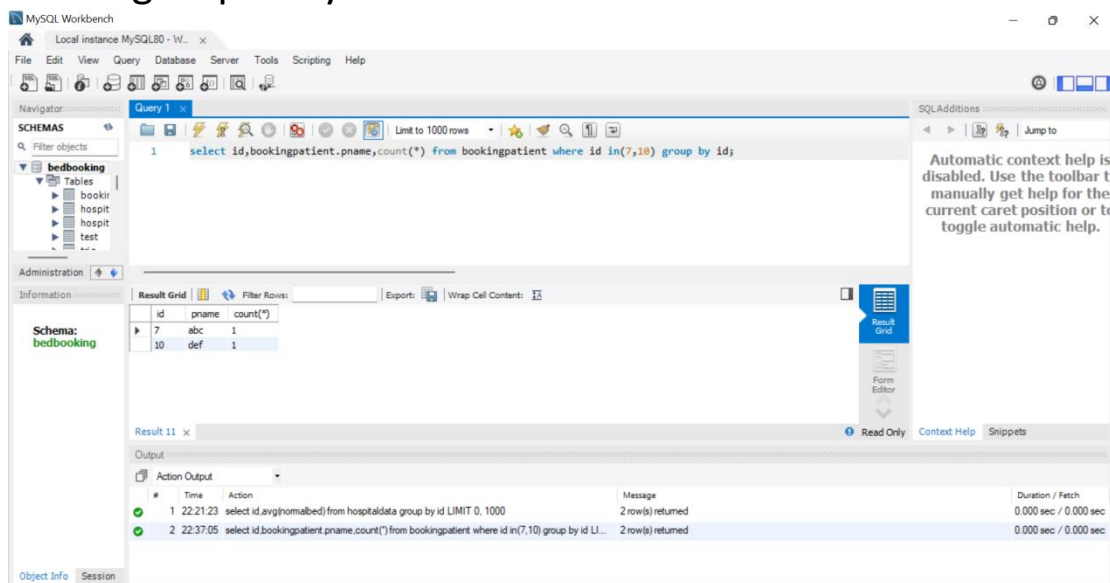


3) select hcode,hname,min(normalbed) from hospitaldata;

Retrieving the attribute called normalbed with minimum number of beds
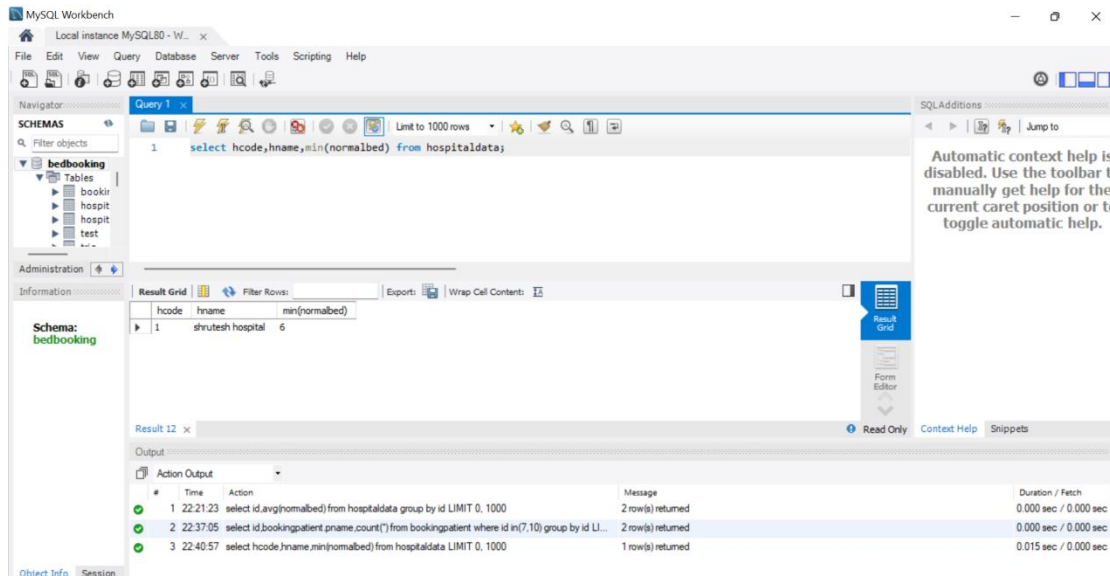
4) select hcode,hname,max(normalbed) from hospitaldata;

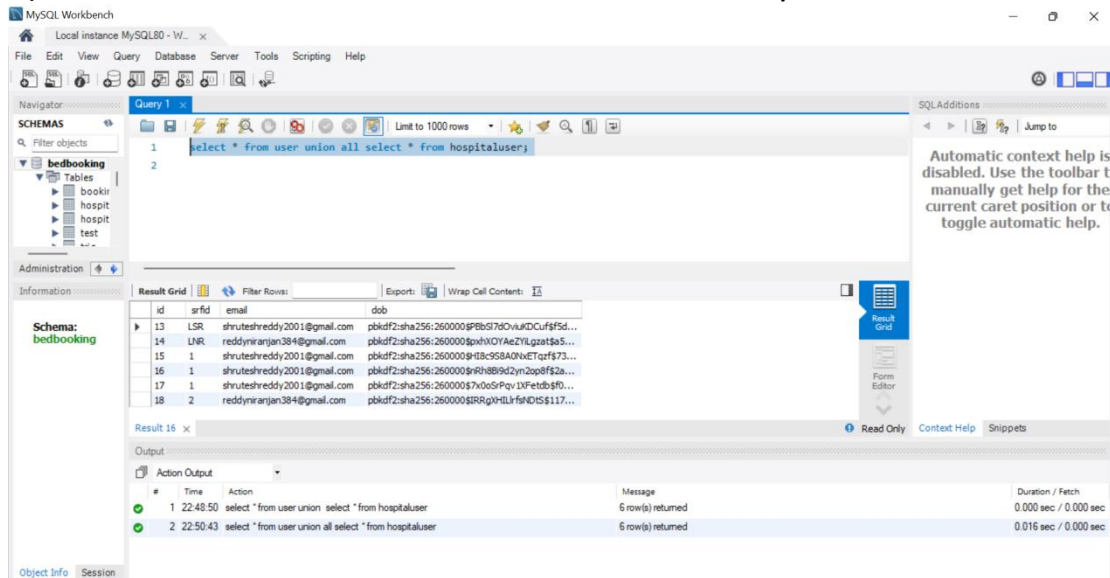Retrieving the attribute called normalbed with maximum number of beds
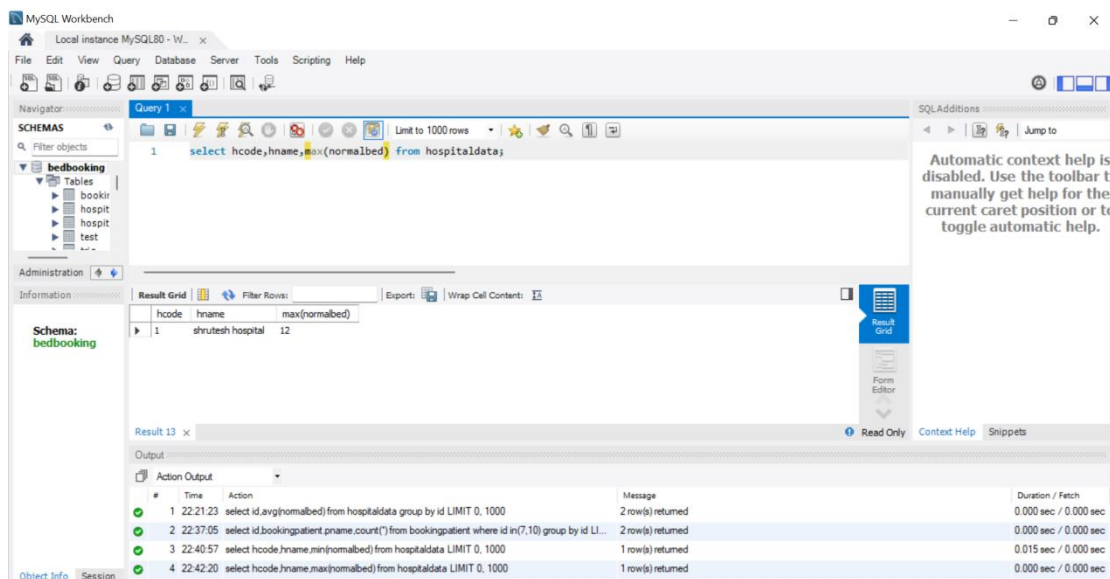


## Set Operations

Showcase at least 4 Set Operations queries
Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results

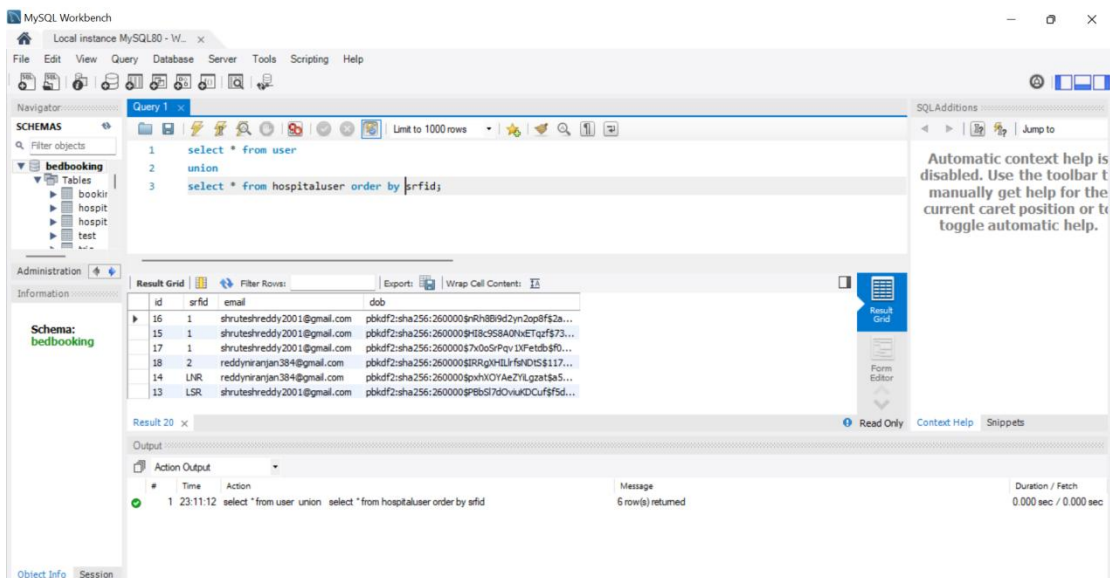1) select * from user union  select * from hospitaluser;
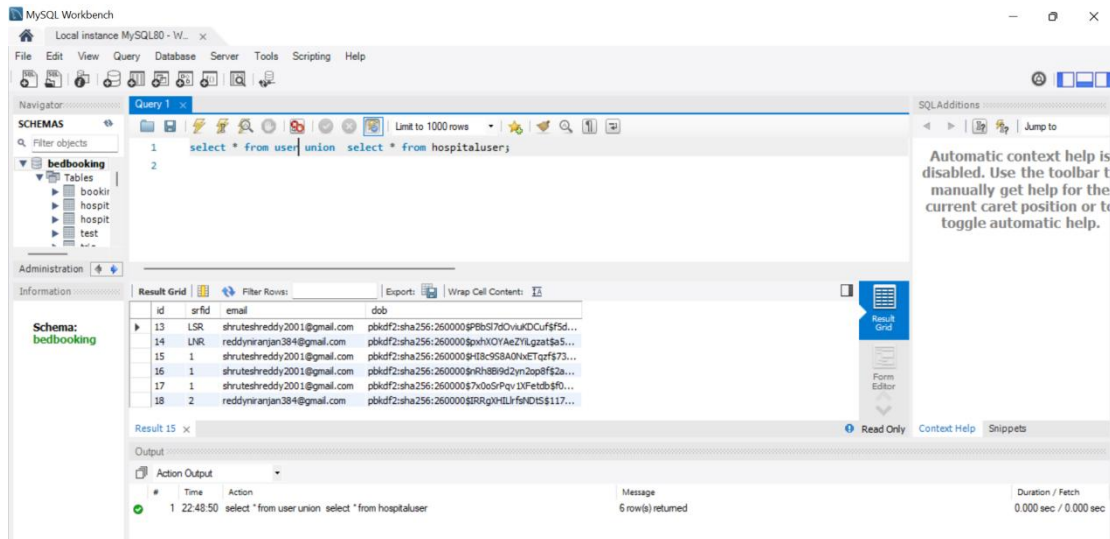
2) select * from user union all select * from hospitaluser;



3)SELECT * FROM user  INTERSECT  SELECT * FROM hospitaluser;
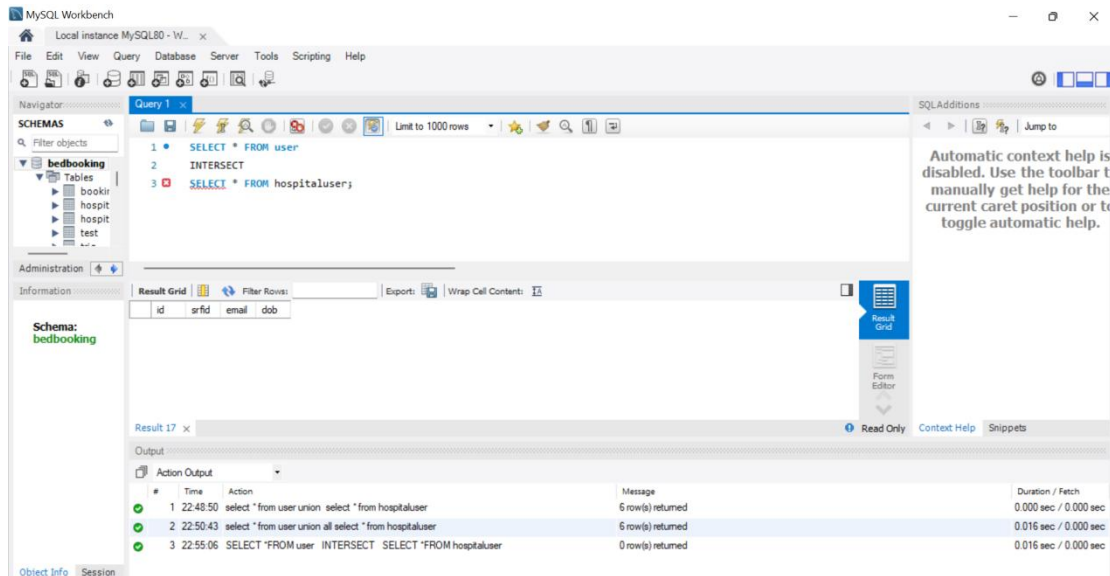
4)select * from user union  select * from hospitaluser order by srfid;



## **Functions and Procedures**

Create a Function and Procedure. State the objective of the function / Procedure. Run and display the results.

## **Functions:**

## Objective:

If count of normalbed is more than 0, then its displays the number of beds, else it displays 'Sorry, no beds are available'.

## Code:

```
CREATE DEFINER=`root`@`localhost` FUNCTION
`beds_info`(hospcode char(7)) RETURNS longtext CHARSET
utf8mb4
    DETERMINISTIC
BEGIN
DECLARE cnt INT;
DECLARE msg LONGTEXT;
set cnt = (Select normalbed FROM hospitaldata WHERE
hcode=hospcode);
IF cnt > 0 THEN
SET msg = CONCAT('Number of beds
available :',CONVERT(cnt,char));
ELSE
SET msg = CONCAT('Sorry, no beds are available');
END IF;
RETURN msg;
END
```
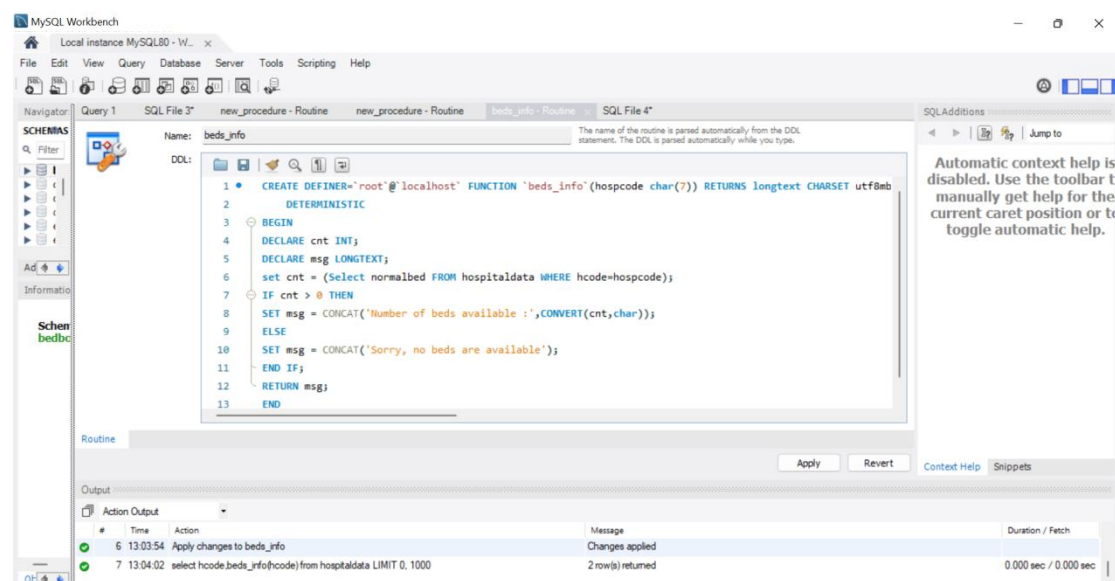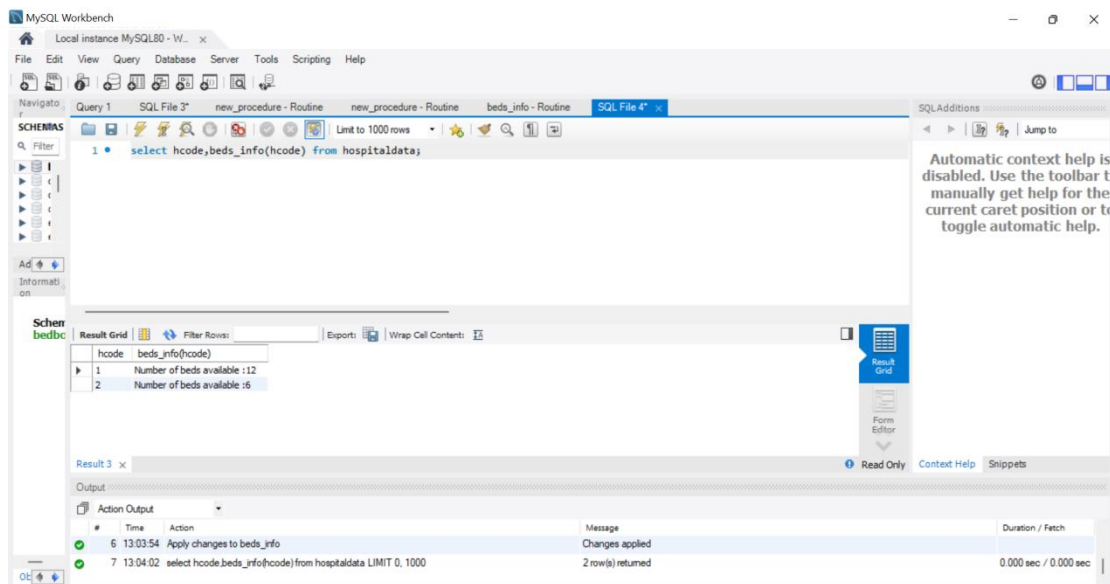
## Procedures:

**Objective:** here in this procedure if count of normalbed in The table hospitaldata is greater than 0, then message is set to "beds are available" else "beds are not available"

## Code:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE
`count_bed`(IN user_id int)
BEGIN
DECLARE cnt int;
DECLARE msg varchar(30);
set cnt= (SELECT normalbed from hospitaldata where
id=user_id);
IF cnt >0 THEN
set msg= 'Beds are available';
ELSE
set msg='Beds are not available';
END IF;
END
```
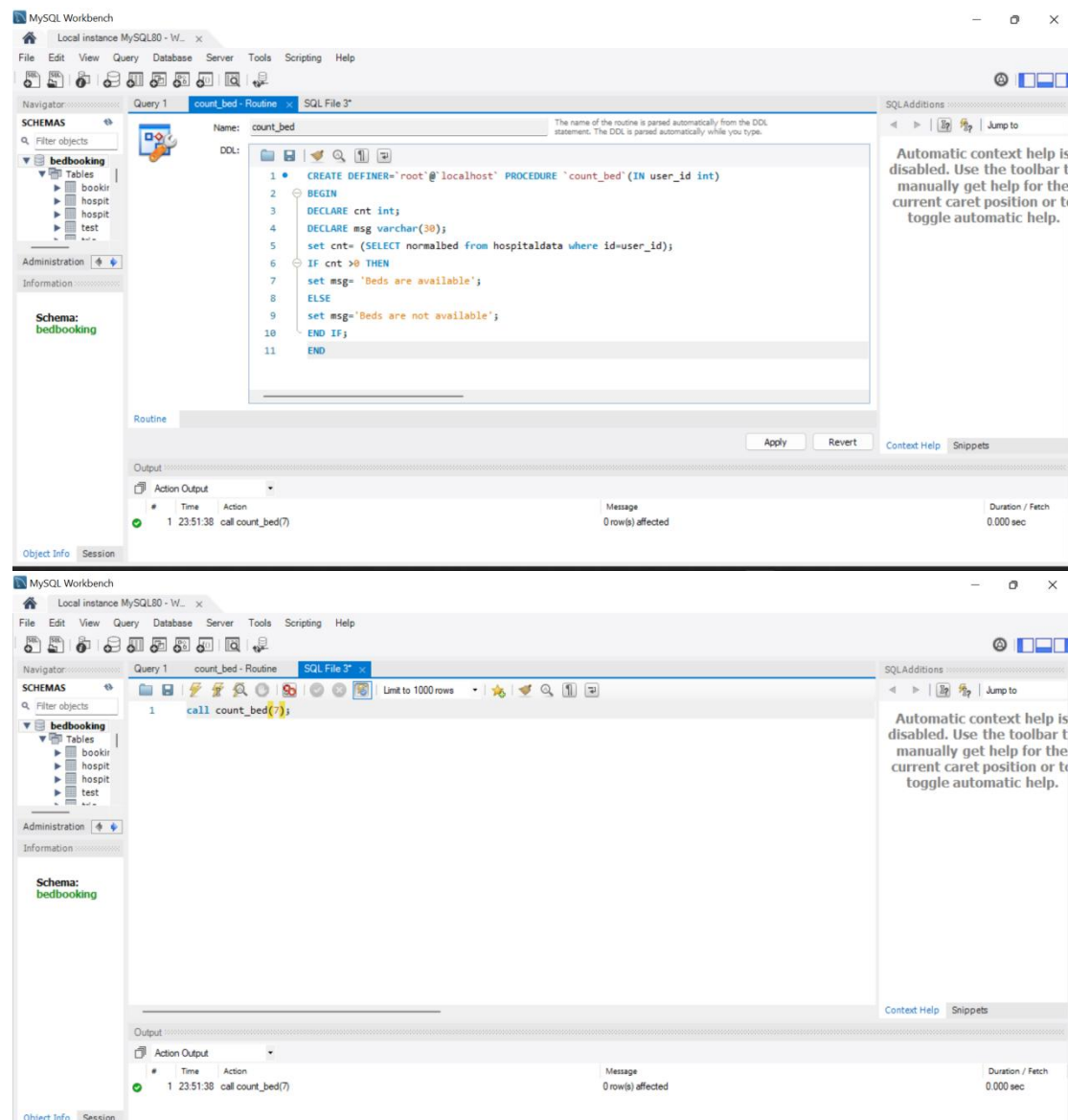
# **Triggers and Cursors**

Create a Trigger and a Cursor. State the objective. Run and display the results.

## **Trigger:**

**Objective:** There is a section in my project called as operations where activities of hospital user is tracked called as triggered data. If hospital user inserts, deletes or updates the data then all the details related to his activities will be displayed in operations sections. Here I have just showed a trigger for AFTER_INSERT.

## CODE:

```
CREATE TRIGGER `Insert` AFTER INSERT ON `hospitaldata` FOR
EACH ROW INSERT INTO trig
VALUES(null,NEW.hcode,NEW.normalbed,NEW.hicubed,NEW.icubed,NEW.vbed,' INSERTED',NOW())
```
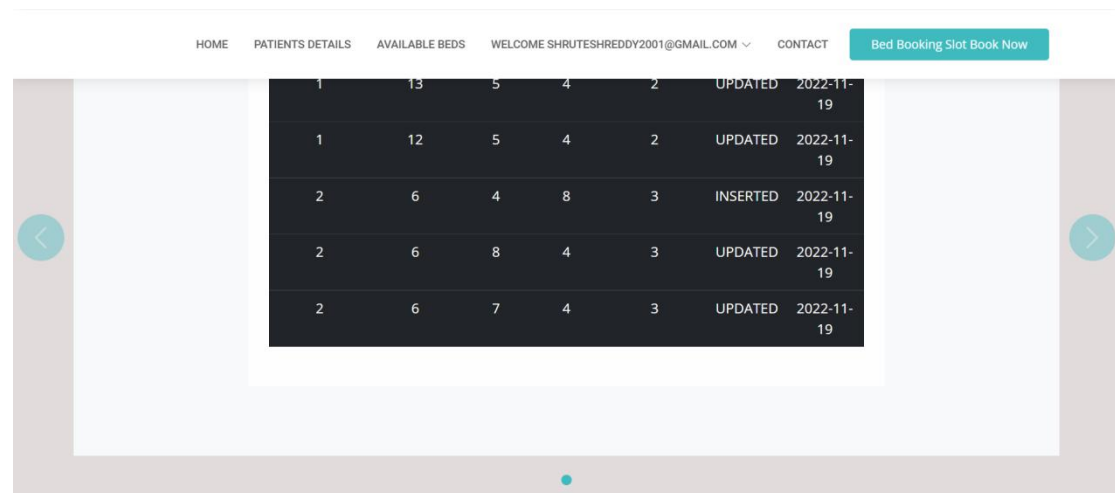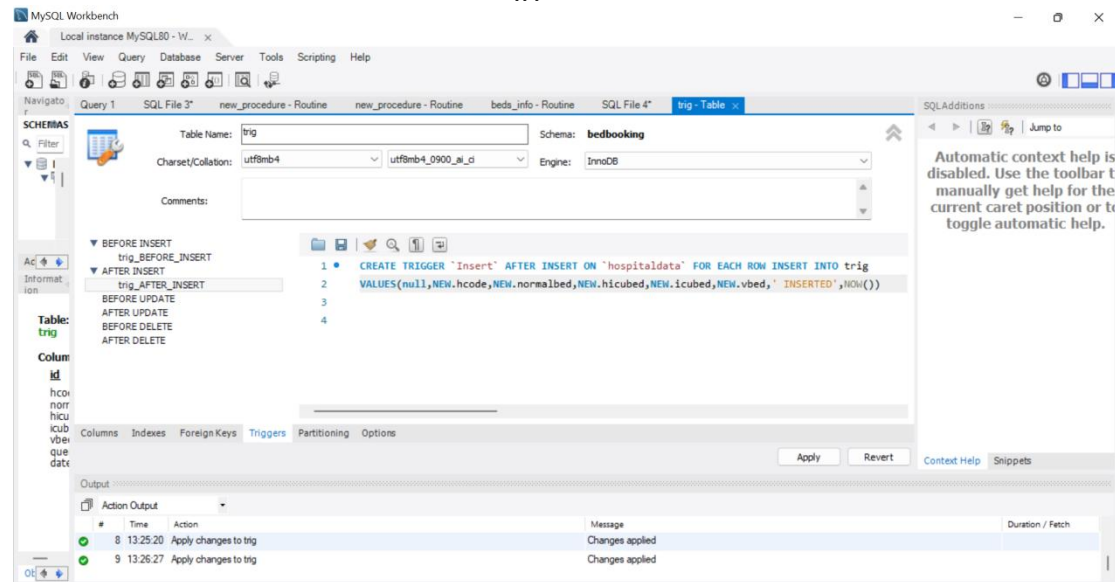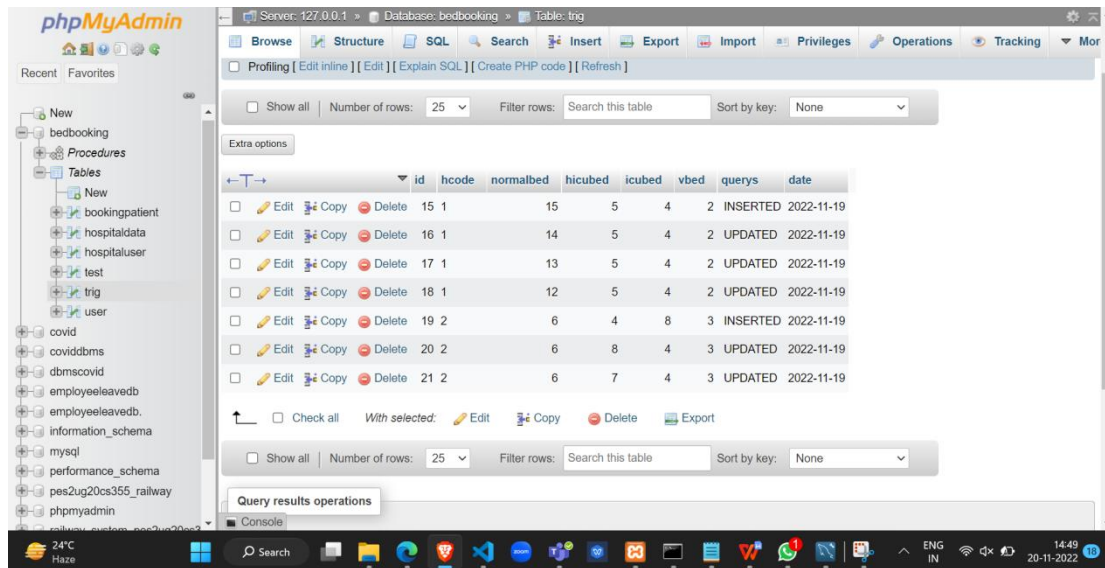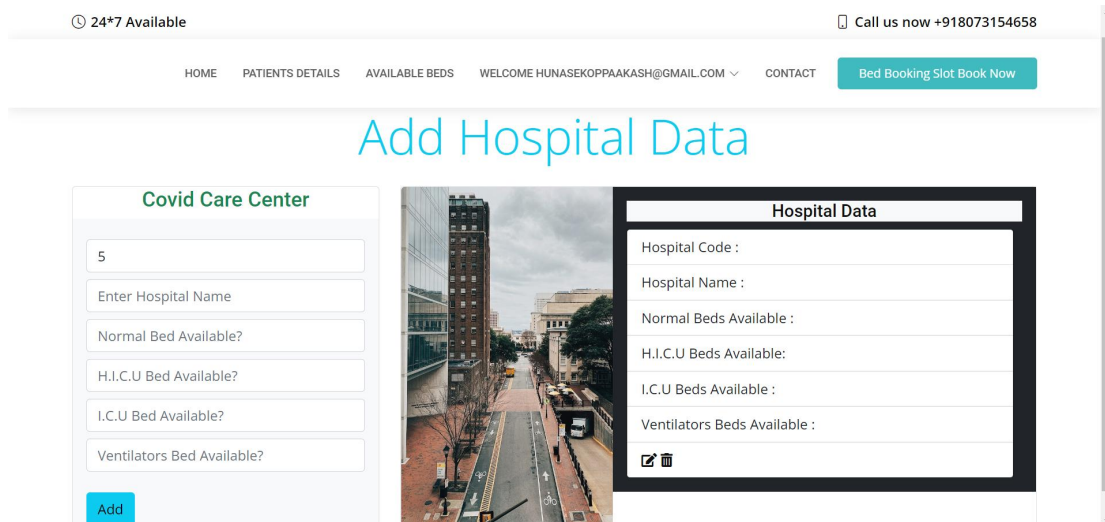
# Developing a Frontend

The frontend should support

1. Addition, Modification and Deletion of records from any chosen table
2. There should be an window to accept and run any SQL statement and display the result

## Adding Hospital data:
## Before adding hospital data:

## Add Hospital Data

### Covid Care Center

5

Shrutesh Reddy hospitals

7

4

6

10

Add

**Hospital Data**

Hospital Code : **5**

Hospital Name : **Shrutesh Reddy hospitals**

Normal Beds Available :**7**

H.I.C.U Beds Available: **4**

I.C.U Beds Available :**6**

Ventilators Beds Available : **10**

---

## Before and after updating the table hospitaldata:

## Add Hospital Data

### Covid Care Center

5

Shrutesh Reddy hospitals

7

4

6

10

Add

**Hospital Data**

Hospital Code : **5**

Hospital Name : **Shrutesh Reddy hospitals**

Normal Beds Available :**7**

H.I.C.U Beds Available: **4**

I.C.U Beds Available :**6**

Ventilators Beds Available : **10**

## Add Hospital Data

**Slot Updated**    ✕

### Covid Care Center

5

Enter Hospital Name

Normal Bed Available?

H.I.C.U Bed Available?

I.C.U Bed Available?

Ventilators Bed Available?

**Hospital Data**

Hospital Code : **5**

Hospital Name : **Shrutesh**

Normal Beds Available :**10**

H.I.C.U Beds Available: **5**

I.C.U Beds Available :**8**

Ventilators Beds Available : **1**

## Before and After triggers:

18

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 13 | 5 | 4 | 2 | UPDATED | 2022-11-19 |
| 1 | 12 | 5 | 4 | 2 | UPDATED | 2022-11-19 |
| 2 | 6 | 4 | 8 | 3 | INSERTED | 2022-11-19 |
| 2 | 6 | 8 | 4 | 3 | UPDATED | 2022-11-19 |
| 2 | 6 | 7 | 4 | 3 | UPDATED | 2022-11-19 |

●

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 9 | 5 | 4 | 2 | UPDATED | 2022-11-21 |
| 1 | 8 | 5 | 4 | 2 | UPDATED | 2022-11-21 |
| 1 | 2 | 6 | 1 | 5 | UPDATED | 2022-11-21 |
| 5 | 7 | 4 | 6 | 10 | INSERTED | 2022-11-21 |
| 5 | 10 | 5 | 8 | 1 | UPDATED | 2022-11-21 |

●