**DBMS - Mini Project**

**COVID BED SLOT MANAGEMENT**

**Name: Aryan Puranik**

**SRN: PES1UG20CS081**

**Section: 5th B**

**Short Description and Scope of the Project**

* Main aim of the project is for covid patients to be able book beds in hospitals.
* The project consists of 5 tables. User login table, Staff Login table, User table, Booking patient table and the Hospital Data Table.
* Each table has Operations, such as Add, View, Update and Delete.

**Scope:**

A Covid bed allotment database management system is a computerized system that helps manage the allocation of hospital beds for patients with Covid-19.

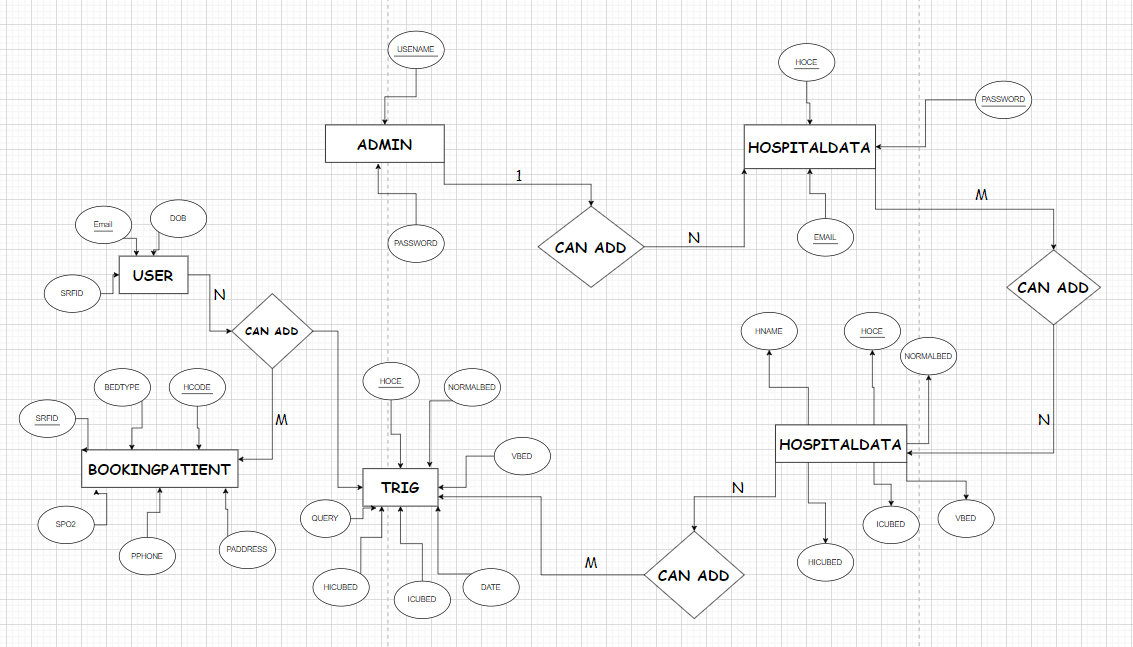
It is designed to ensure that hospitals have enough beds available to accommodate all incoming patients, as well as provide an efficient way to track and monitor patient occupancy rates.

The system can also be used to identify potential areas of overcrowding or underutilization in order to better allocate resources.

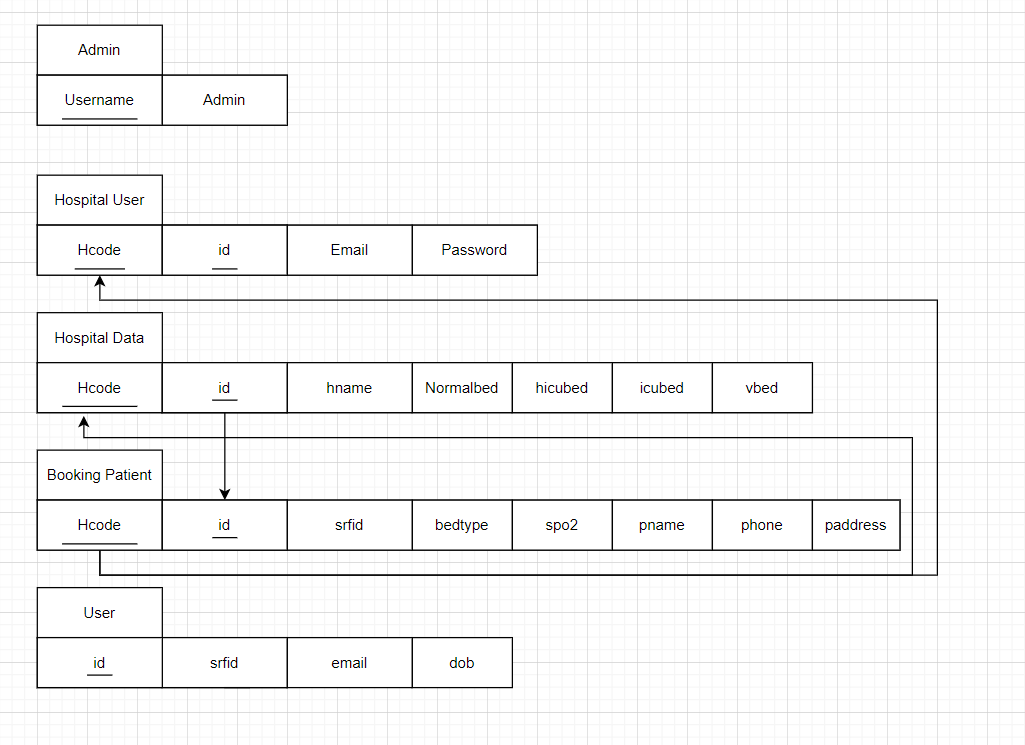
The main features of this type of system include an automated bed allocation algorithm which takes into account the availability of beds, patient history and current medical condition when assigning a bed.

Real-time tracking capabilities so administrators can view up-to-date information on who has been admitted and which beds are currently occupied.

**ER Diagram:**

****

**Relational Schema:**

****

**DLL Statements – Building the Database**

def create\_bookingpatient\_table():

c.execute('CREATE TABLE IF NOT EXISTS `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);')

def create\_hospitaldata\_table():

c.execute('CREATE TABLE IF NOT EXISTS `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);')

def create\_hospitaluser\_table():

c.execute('CREATE TABLE IF NOT EXISTS `hospitaluser` (`id` int(11) NOT NULL,`hcode` varchar(20) NOT NULL,`email` varchar(100) NOT NULL,`password` varchar(1000) NOT NULL);')

def create\_test\_table():

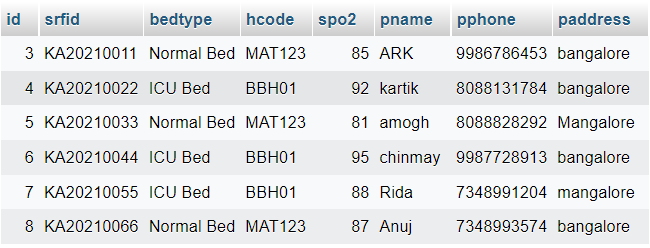
c.execute('CREATE TABLE IF NOT EXISTS `test` (`id` int(11) NOT NULL,`name` varchar(50) NOT NULL);')

def create\_user\_table():

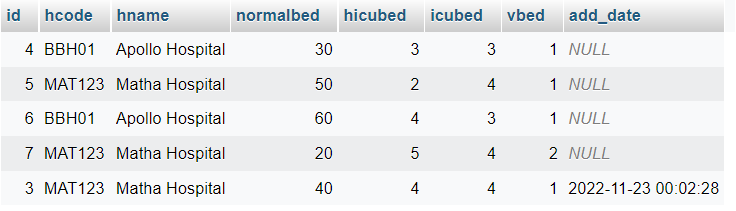
c.execute('CREATE TABLE IF NOT EXISTS `user` (`id` int(11) NOT NULL,`srfid` varchar(20) NOT NULL,`email` varchar(100) NOT NULL,`dob` varchar(1000) NOT NULL);')

**Populating the Database**

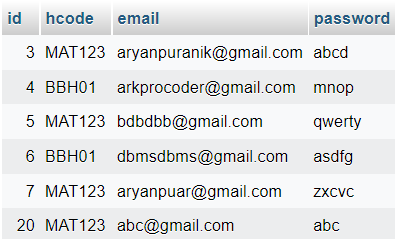
**Booking Patient:**

****

**Hospital data:**

****

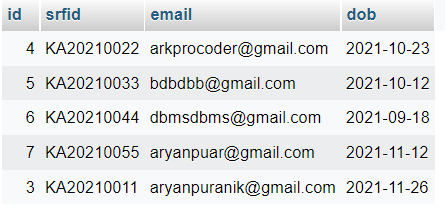
**Hospital User:**

****

**Test:**

****

**User:**

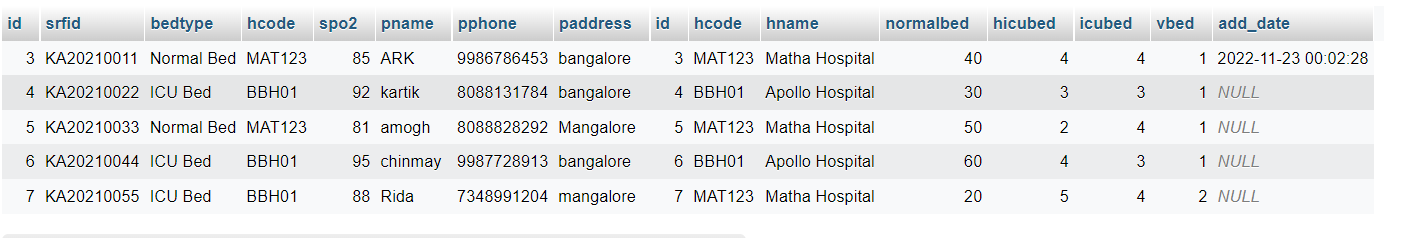
****

**Join Queries:**

Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results

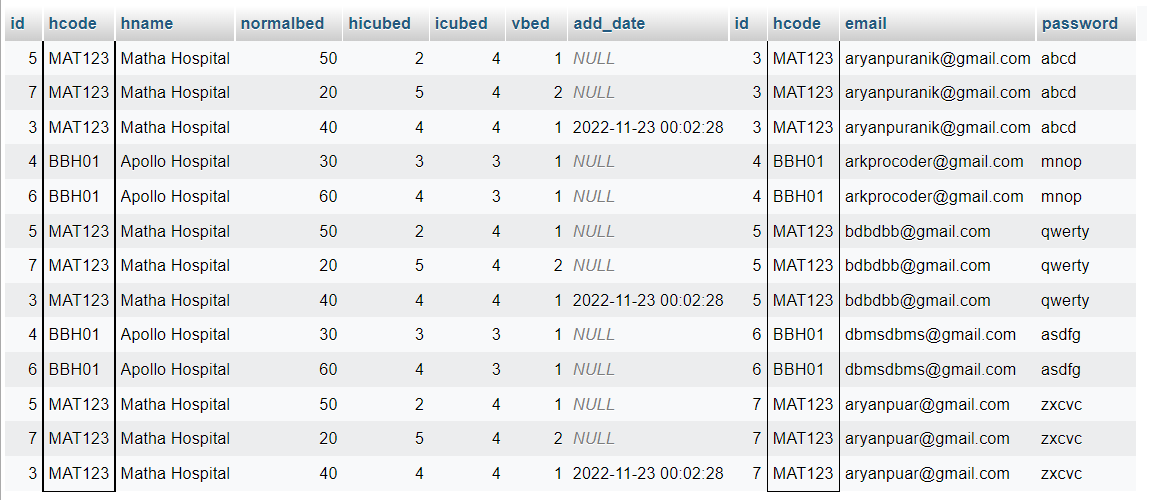
1. To display booking patient details with hospital data details

select \* from bookingpatient inner join hospitaldata on bookingpatient.id=hospitaldata.id;

****

1. Display hospitaldata and hospitaluser data based on hospital code

SELECT \* from hospitaldata join hospitaluser WHERE hospitaldata.hcode=hospitaluser.hcode;

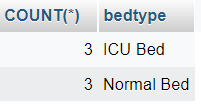


**Aggregate Functions**

Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results

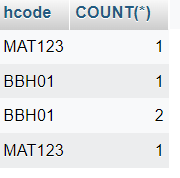
1. To count number of beds in each hospital :

SELECT COUNT(\*),bedtype from bookingpatient GROUP BY bedtype;



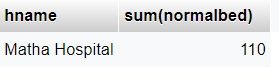
2) To count number hicubeds in each hospital based on hospitalcode

SELECT hcode,COUNT(\*) FROM hospitaldata GROUP BY hicubed;



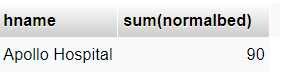
3)To display the total number of NORMAL beds in hospital code MAT123

select hname,sum(normalbed) from hospitaldata where hcode = 'MAT123' group by hcode;



4)To display the total number of NORMAL beds in hospital code BBH01

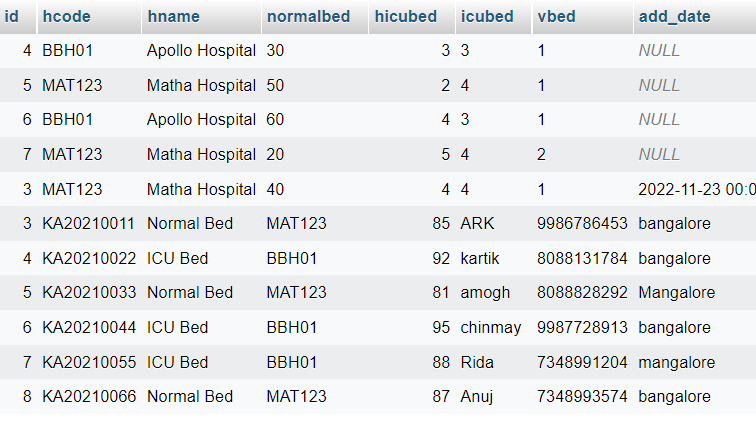
select hname,sum(normalbed) from hospitaldata where hcode = 'BBH01' group by hcode;



**Set Operations**

1. Union – Doing union operation of bookingpatient and hospitaldata

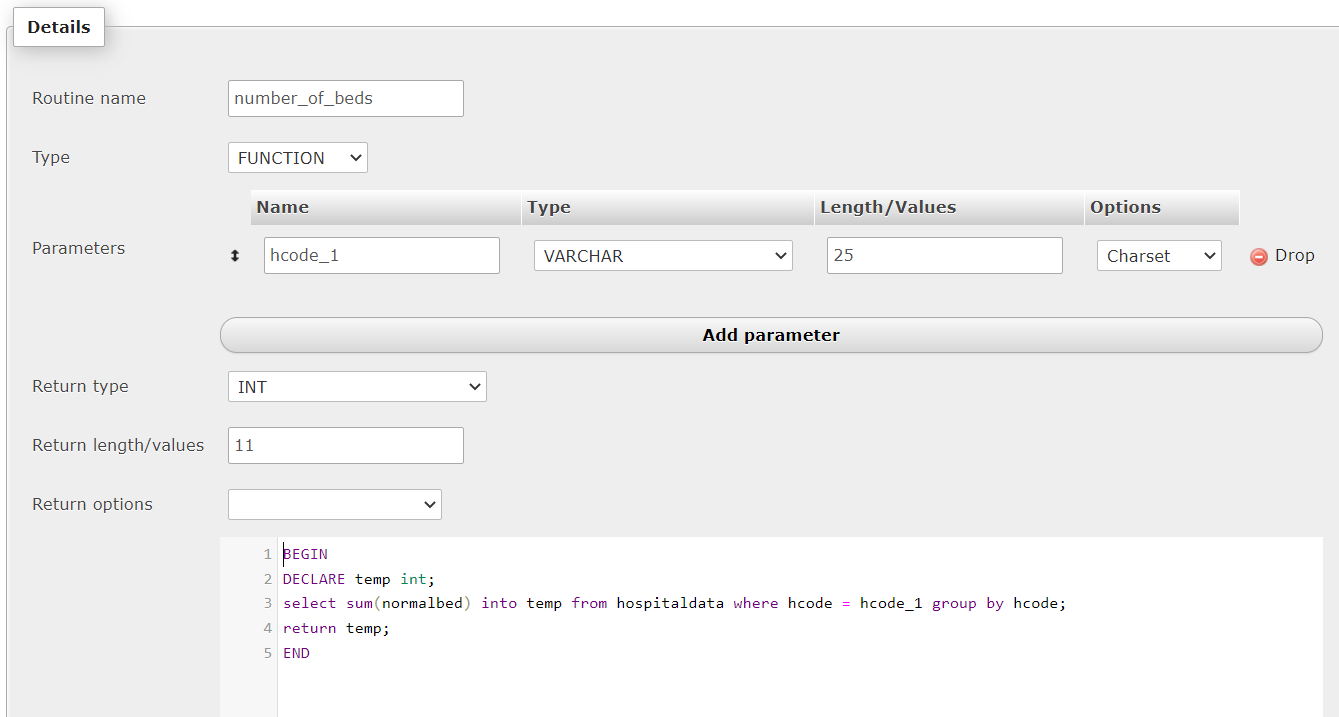
SELECT \* FROM hospitaldata UNION SELECT \* FROM bookingpatient;



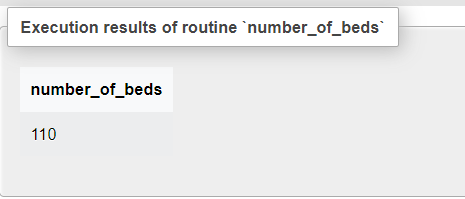
**Functions and Procedures**

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

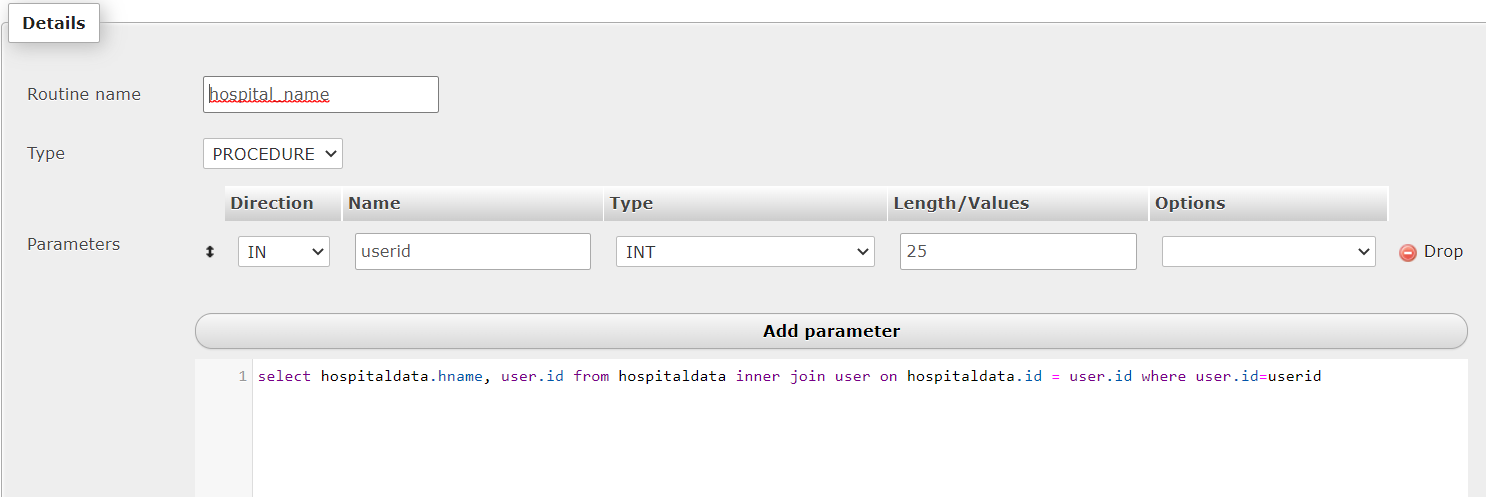
1. **Creating a function which will takein the Hospital code and display the Number of normalbeds in that hospital.**

****

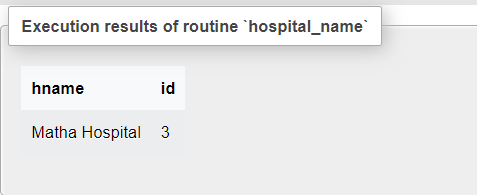
**Result:**

****

1. **Creating a Procedure which will take in the Hospital ID and give the name of the hospital.**

****

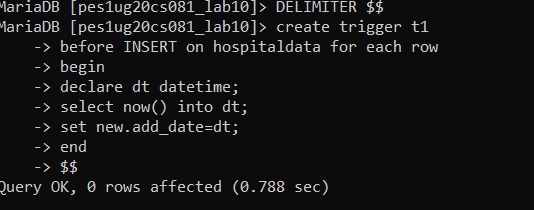
**RESULT**

****

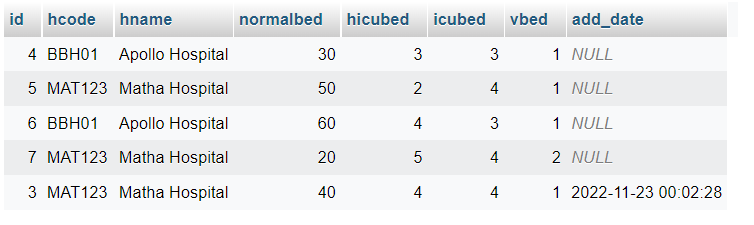
**Triggers and Cursors**

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

**A trigger t1 was created on the entry table in which as soon as the details are entered there will be no manual entry of datetime but a automatic one due to this trigger**

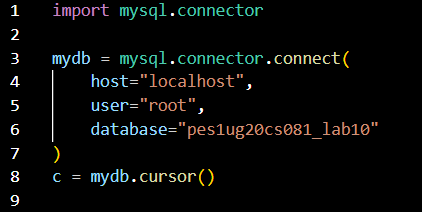


Result:



The curent date and time is automatically added in the column add\_date.

**A cursor is normally used in the python scripting which is needed to handle all the CRUD operations:**

****

**Developing a Frontend**

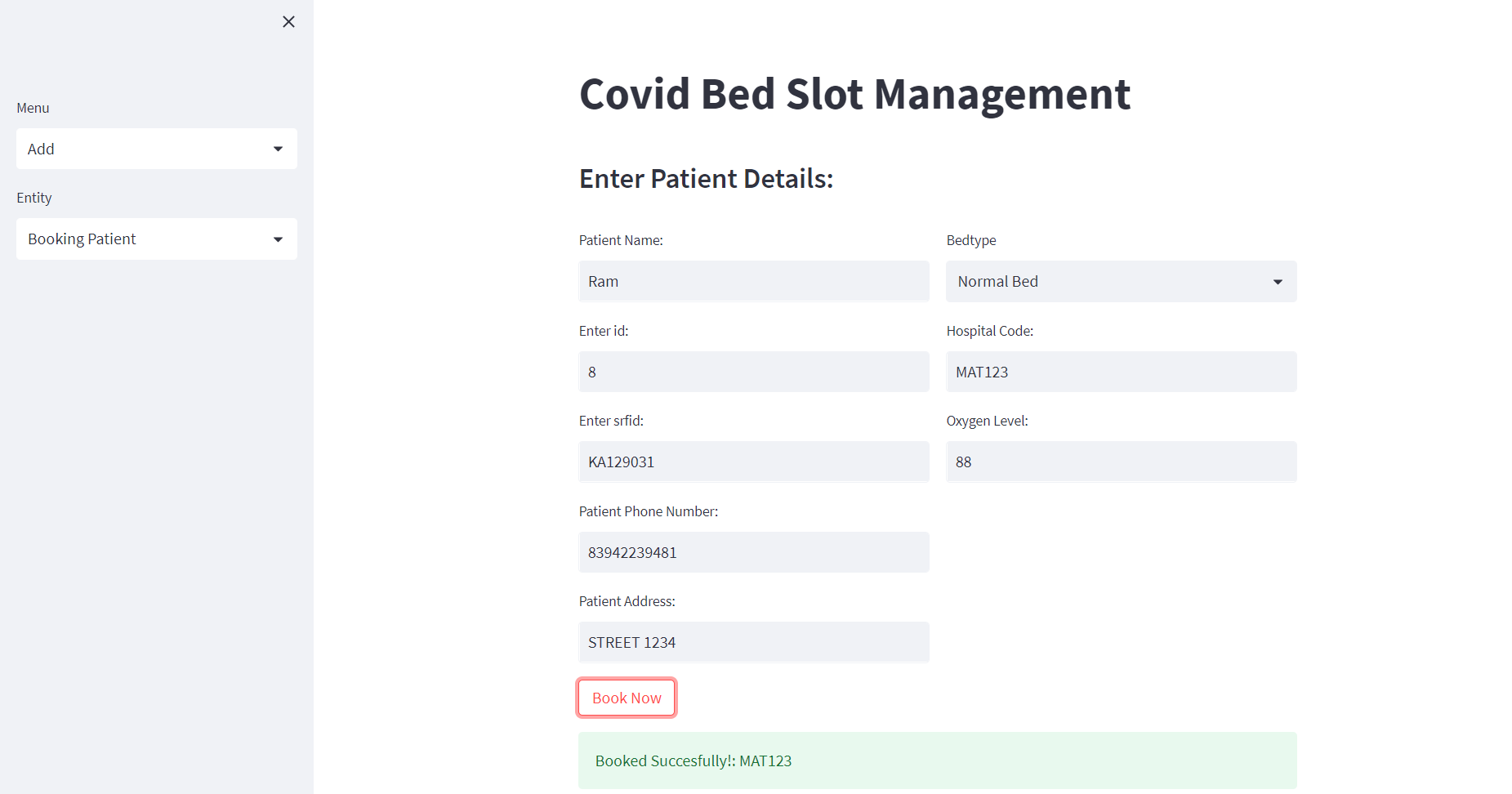
The frontend should support

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

***CRUD Operations shown are done on the BookingPatient table***

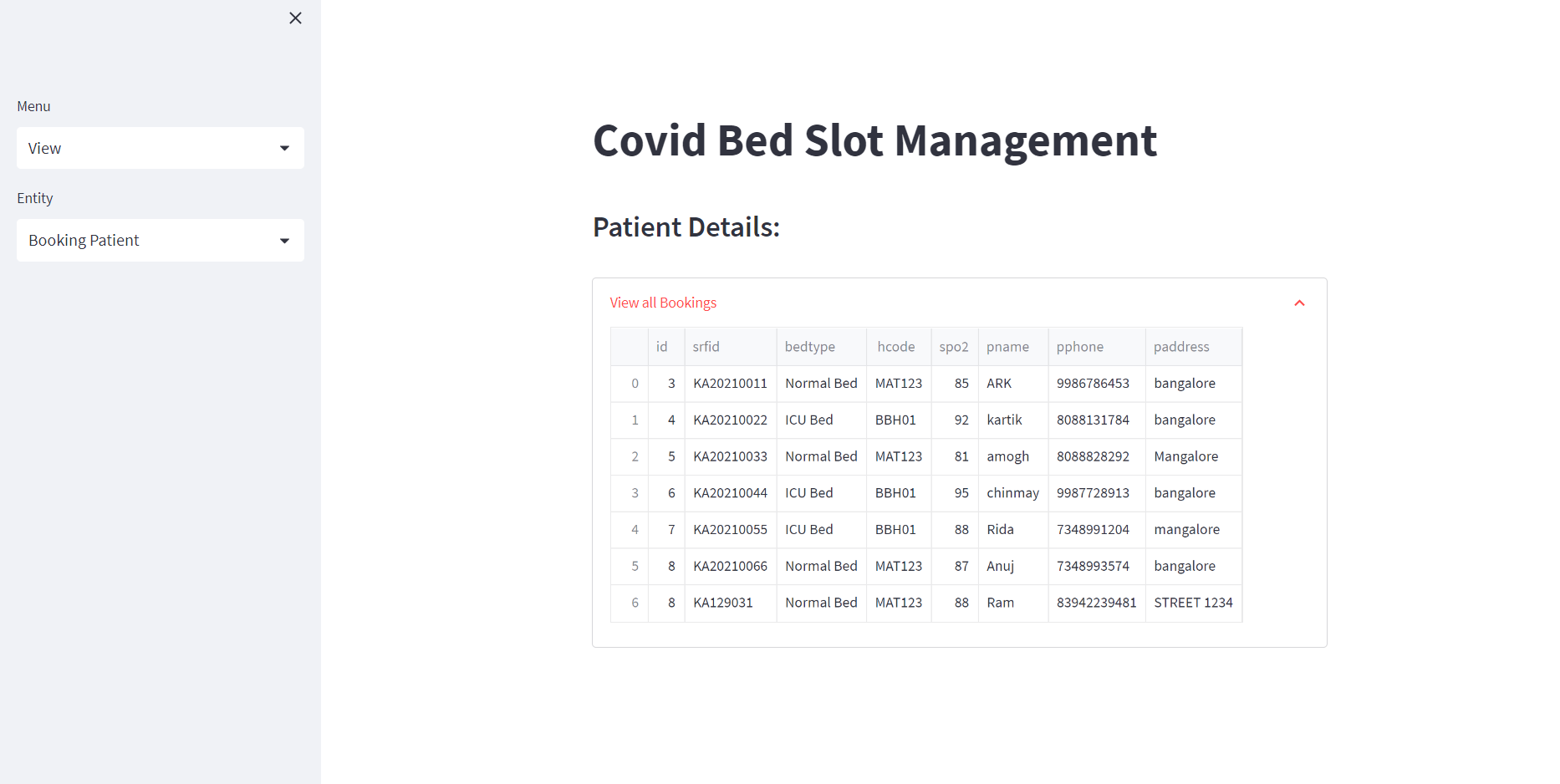
Add:

******

Read:

Shown 3 tables for example

Booking Patient:



Hospital Data:

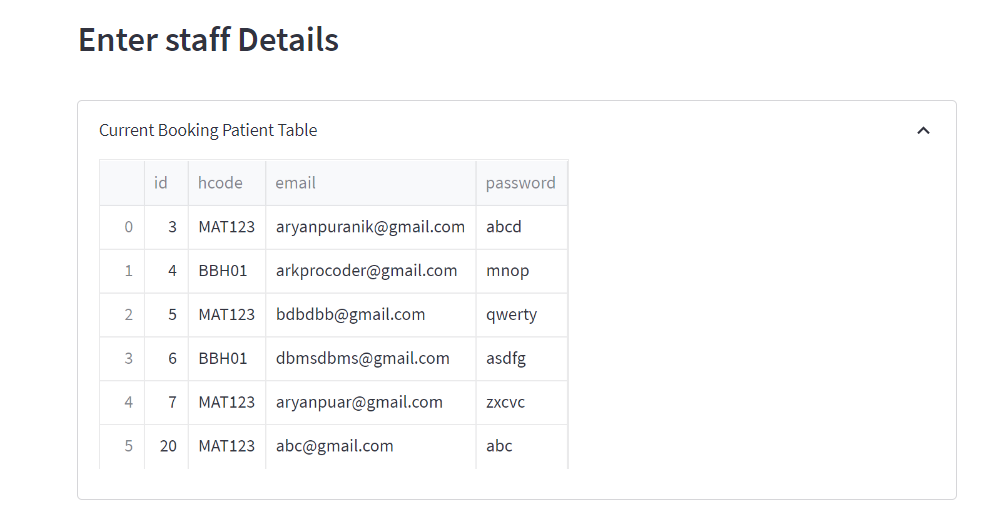


Hospital User / Staff Login:

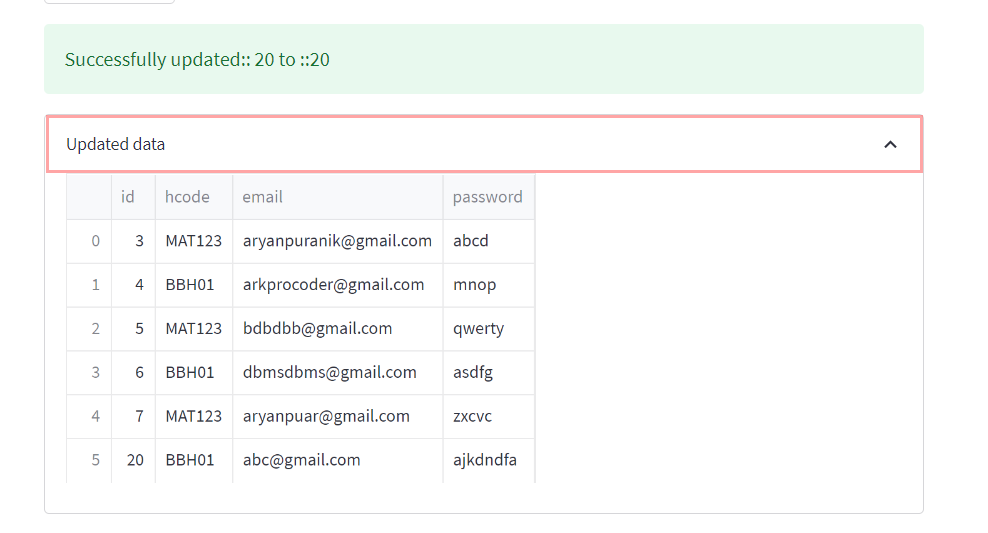


Edit:

Before Updating:

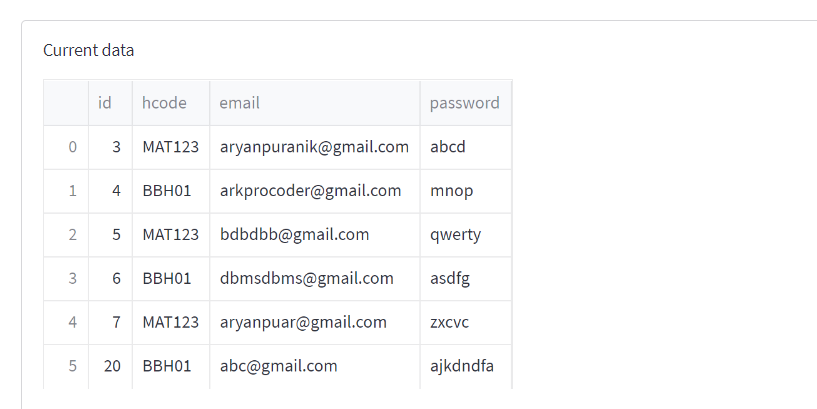


After Updating:

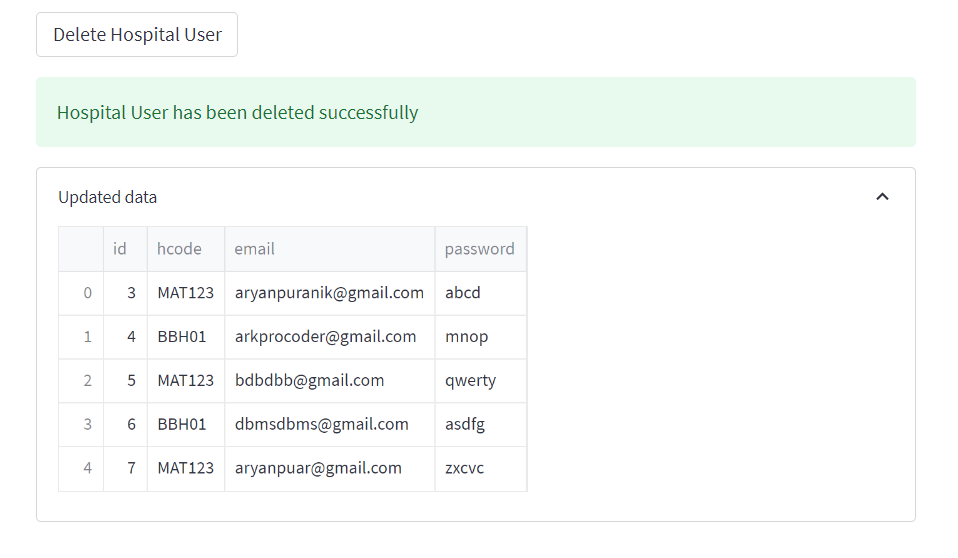


Delete:

Before Deleting:

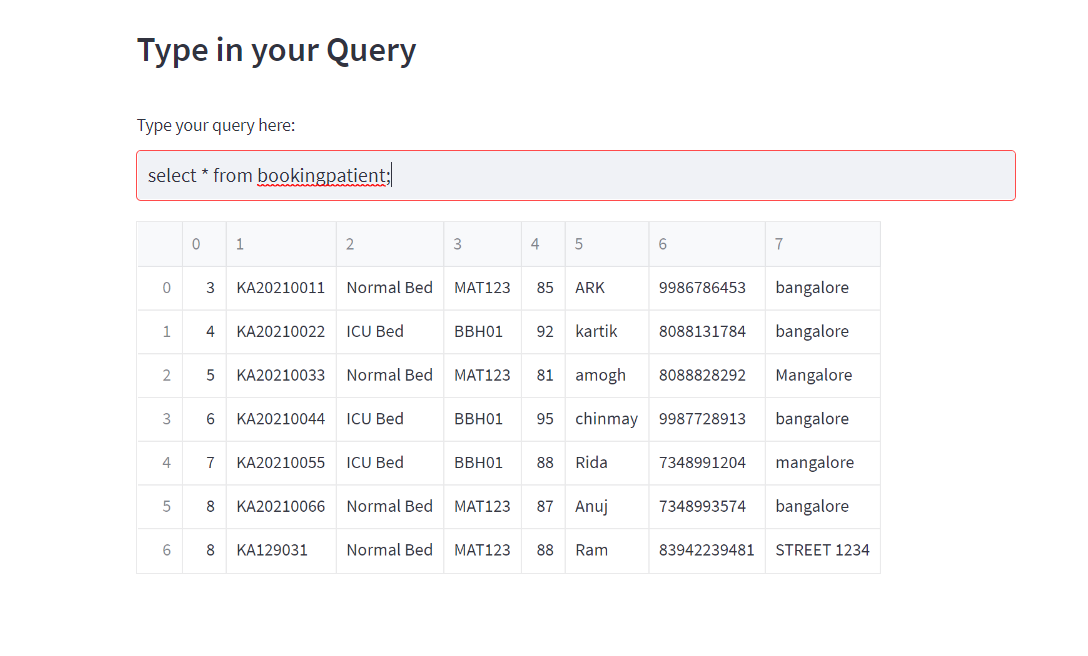


After Deleting:



Showing SQL Queries via Front-end:

1)



2)

