

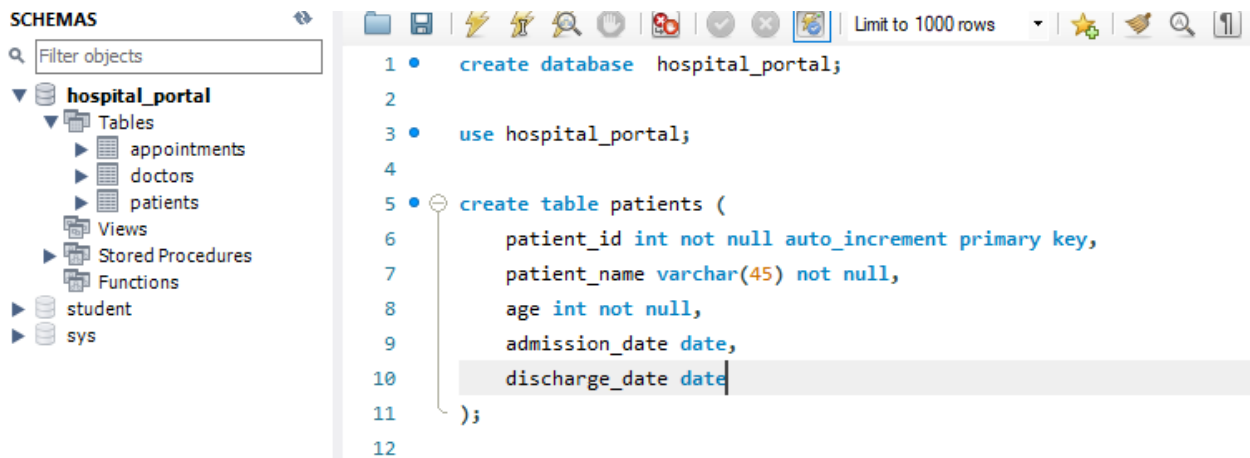
This report outlines the Hospital Portal database using MySQLWorkBench and the linked SQL procedures, along with the functionality of Python portal Database and portal server.

MySQLWorkbench Structure

Tables

1. Patients

- patient_id (Primary Key)
- patient_name
- age
- admission_date
- discharge_date



The screenshot displays the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows a tree view with 'hospital_portal' expanded, containing 'Tables' (appointments, doctors, patients), 'Views', 'Stored Procedures', and 'Functions'. The main editor window shows the following SQL code:

```
1 • create database hospital_portal;
2
3 • use hospital_portal;
4
5 • create table patients (
6     patient_id int not null auto_increment primary key,
7     patient_name varchar(45) not null,
8     age int not null,
9     admission_date date,
10    discharge_date date
11 );
12
```

2. Appointments

- appointment_id (Primary Key)
- patient_id (Foreign Key referencing Patients)
- doctor_id (Foreign Key referencing Doctors)
- appointment_date
- appointment_time

```

12
13 • create table appointments (
14     appointment_id int not null auto_increment primary key,
15     patient_id int not null,
16     doctor_id int not null,
17     appointment_date date not null,
18     appointment_time decimal not null,
19     FOREIGN KEY (patient_id) REFERENCES patients(patient_id),
20     FOREIGN KEY (doctor_id) REFERENCES doctors(doctor_id)
21 );
22

```

3. Doctors

- doctor_id (Primary Key)
- doctor_name

```

55 • create table doctors (
56     doctor_id int not null auto_increment primary key,
57     doctor_name varchar(45) not null
58 );
59
60 • INSERT INTO doctors (doctor_name) VALUES
61     ('Dr. Jose'),
62     ('Dr. Alex'),
63     ('Dr. Chris');

```

Procedures

1. **ScheduleAppointment**

- Constraints: p_patient_id, p_doctor_id, p_appointment_date, p_appointment_time
- This Inserts a new appointment into the Appointments table.

```

31     delimiter //
32
33 • create procedure ScheduleAppointment(
34     IN p_patient_id INT,
35     IN p_doctor_id INT,
36     IN p_appointment_date DATE,
37     IN p_appointment_time DECIMAL
38 )
39 begin
40     insert into appointments (patient_id, doctor_id, appointment_date, appointment_time)
41     VALUES (p_patient_id, p_doctor_id, p_appointment_date, p_appointment_time);
42 end //
43
44 delimiter ;
45

```

2. DischargePatient

- Constraints: p_patient_id
- Updates the discharge date for a patient in the Patients table.

```

43
44     delimiter ;
45
46     delimiter //
47
48 • create procedure DischargePatient(IN p_patient_id INT)
49 begin
50     update patients SET discharge_date = current_date where patient_id = p_patient_id;
51 end //
52
53     delimiter ;
54

```

ViewAppoinments

appointment_view

- Rows: appointment_id, patient_name, age, appointment_date, appointment_time, doctor_name
- A view combining data from Appointments, Patients, and Doctors tables for easy appointment information recover.

Python Structure

hospital Database

- Defines the Database for interacting with the MySQL database.

portal Database

- Includes ways for adding patients, scheduling appointments, viewing appointments, and discharging patients.

hospital Portal Server

- It handles requests and interacts with the database.

Functionality

1. Adding Patients

- Patients can be added using the do_POST method.

2. Scheduling Appointments

- Appointments can be scheduled using the do_POST method.

3. Viewing Appointments

- All appointments can be viewed using the do_GET method.

4. Discharging Patients

- Patients can be discharged using the do_POST method.