**Name: Vaishnavi Dinesh Thavai**

**College name:BVIMIT**

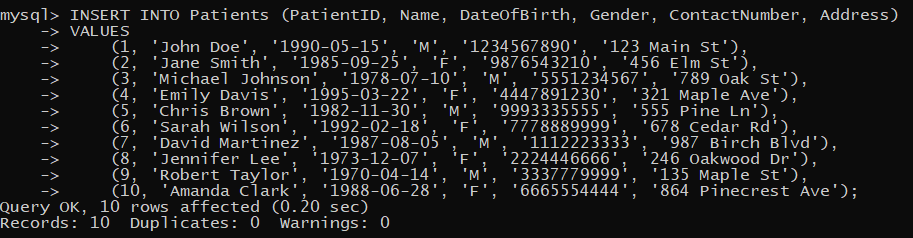
**Project 1**

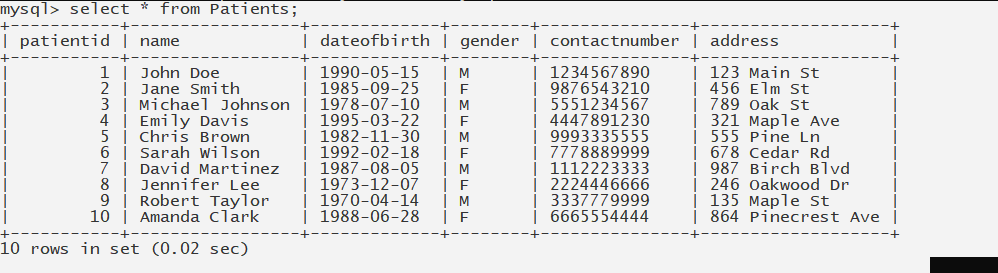
Q.1 Create a DB Schema for Hospital Management System.

**Table: Patients**

| **Column Name** | **Data Type** | **Constraints** |
| --- | --- | --- |
| PatientID | INT | Primary Key |
| Name | VARCHAR |  |
| DateOfBirth | DATE |  |
| Gender | CHAR(1) |  |
| ContactNumber | VARCHAR |  |
| Address | VARCHAR |  |
|  |  |  |

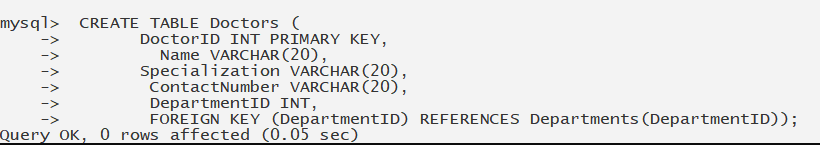
****

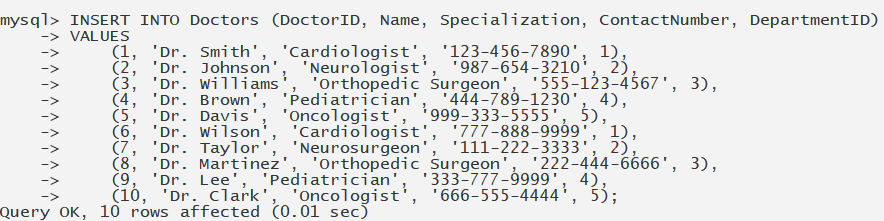
****

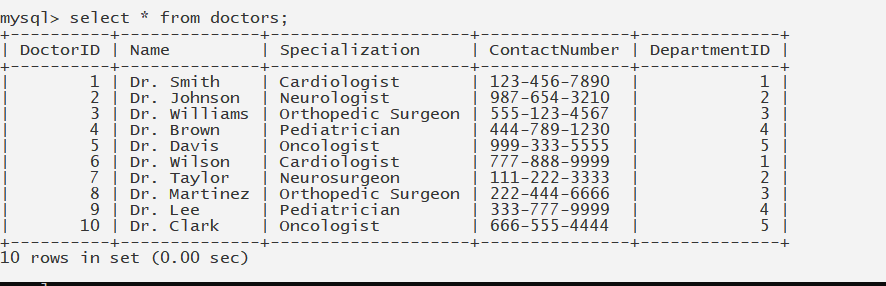
****

**Table: Doctors**

| **Column Name** | **Data Type** | **Constraints** |
| --- | --- | --- |
| DoctorID | INT | Primary Key |
| Name | VARCHAR |  |
| Specialization | VARCHAR |  |
| ContactNumber | VARCHAR |  |
| DepartmentID | INT | Foreign Key referencing Departments table |

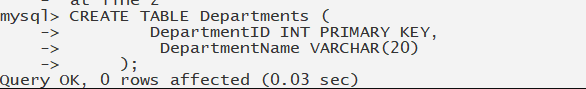
****

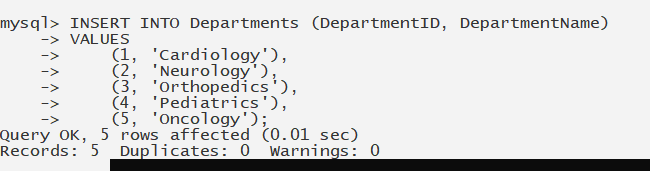
****

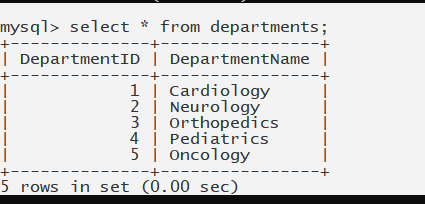
****

**Table: Departments**

| **Column Name** | **Data Type** | **Constraints** |
| --- | --- | --- |
| DepartmentID | INT | Primary Key |
| DepartmentName | VARCHAR |  |

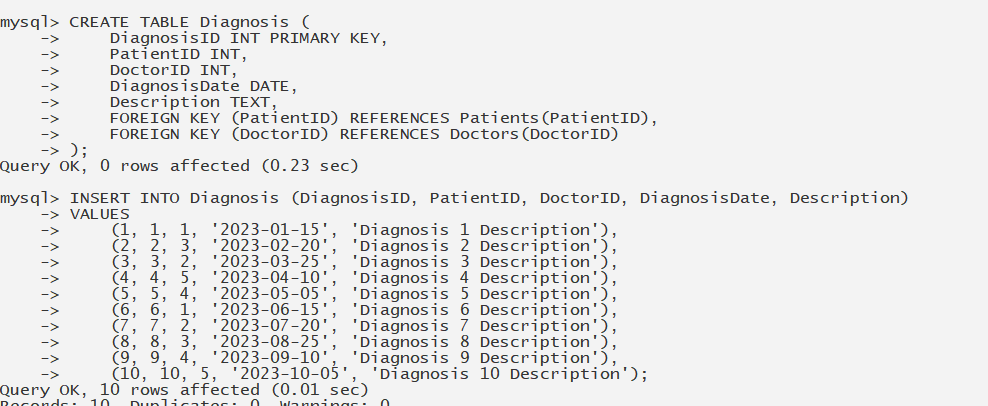
****

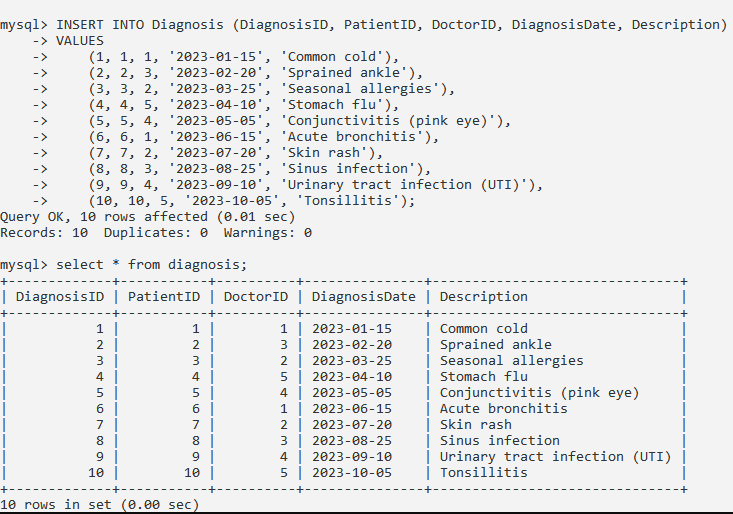
****

****

**Table: Diagnosis**

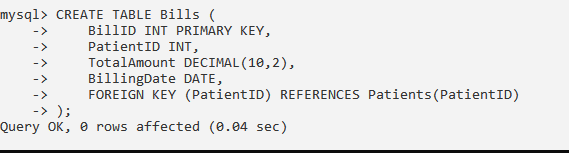
| **Column Name** | **Data Type** | **Constraints** |
| --- | --- | --- |
| DiagnosisID | INT | Primary Key |
| PatientID | INT | Foreign Key referencing Patients table |
| DoctorID | INT | Foreign Key referencing Doctors table |
| DiagnosisDate | DATE |  |
| Description | TEXT |  |

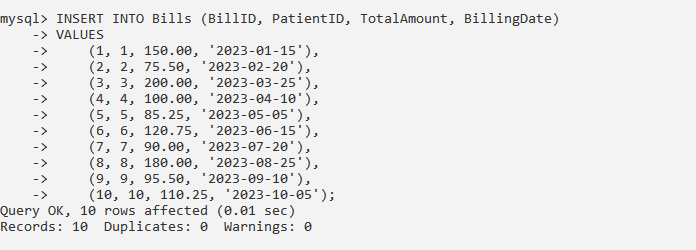
****

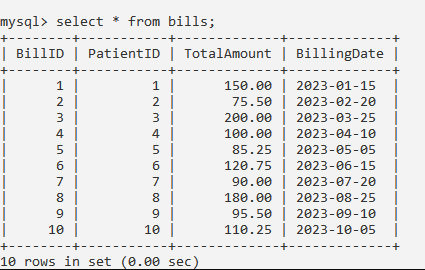
****

**Table: Bills**

| **Column Name** | **Data Type** | **Constraints** |
| --- | --- | --- |
| BillID | INT | Primary Key |
| PatientID | INT | Foreign Key referencing Patients table |
| TotalAmount | DECIMAL(10,2) |  |
| BillingDate | DATE |  |

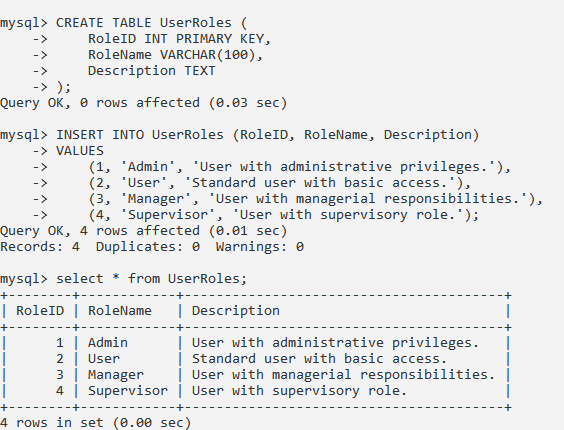
****

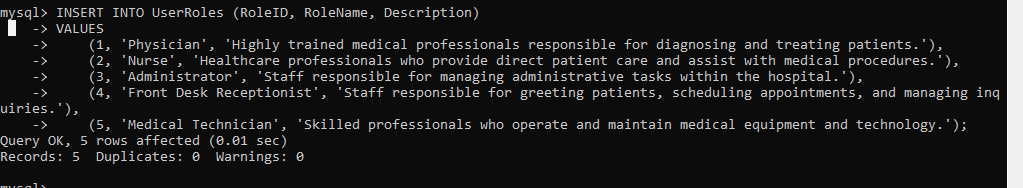
****

****

**Table: UserRoles**

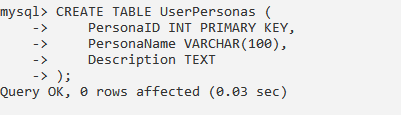
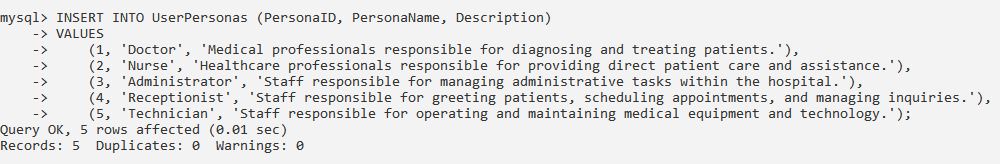
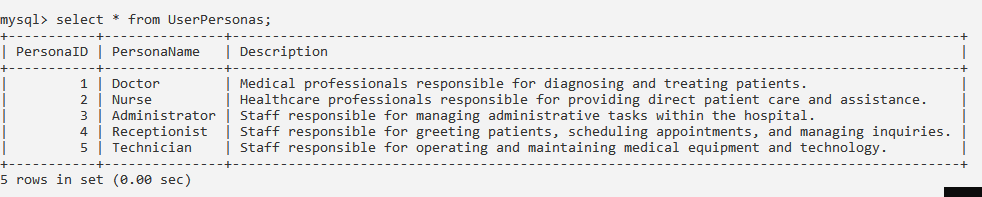
| **Column Name** | **Data Type** | **Constraints** |
| --- | --- | --- |
| RoleID | INT | Primary Key |
| RoleName | VARCHAR |  |
| Description | TEXT |  |

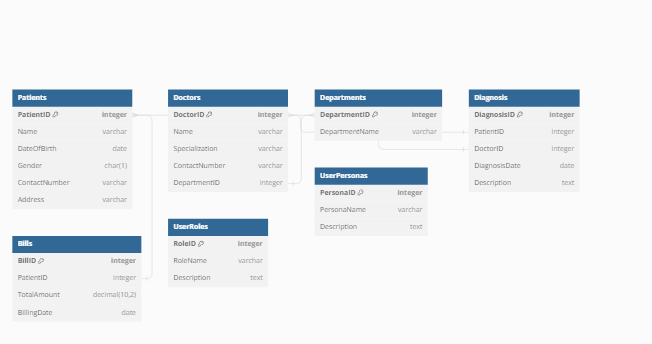
****

****

**Table: UserPersonas**

| **Column Name** | **Data Type** | **Constraints** |
| --- | --- | --- |
| PersonaID | INT | Primary Key |
| PersonaName | VARCHAR |  |
| Description | TEXT |  |

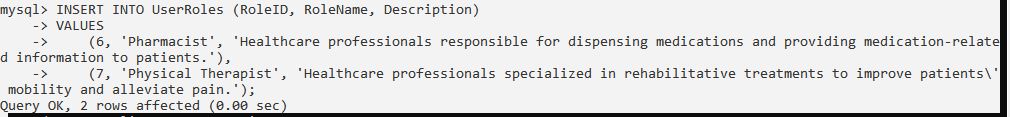
  



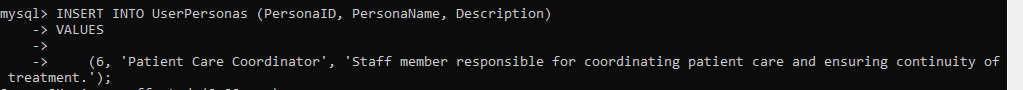
**HMS should be capable to recognize already registered patients and user roles.**

**- Write necessary queries to register new user roles and personas**

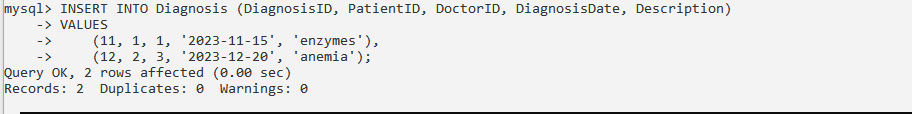
1)new user roles



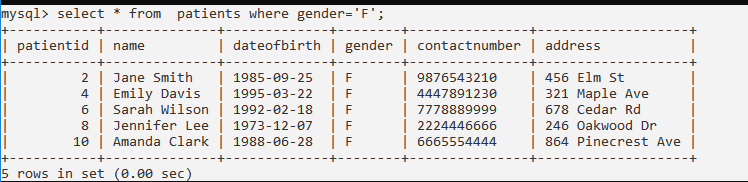
2)new user persona



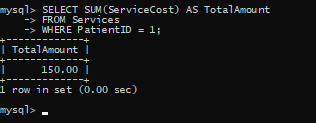
- **Write necessary queries to add to the list of diagnosis of the patient tagged by date.**



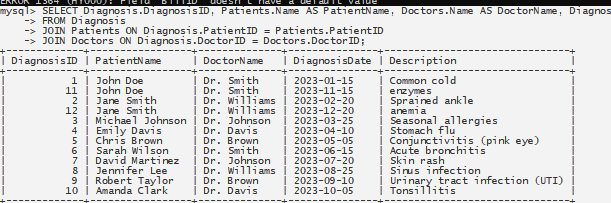
**- Write necessary queries to fetch required details of a particular patient.**



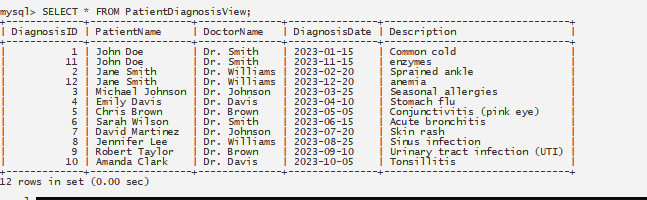
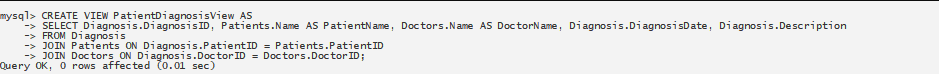
**- Write necessary queries to prepare bill for the patient at the end of checkout.**



**- Write necessary queries to fetch and show data from various related tables (Joins)**



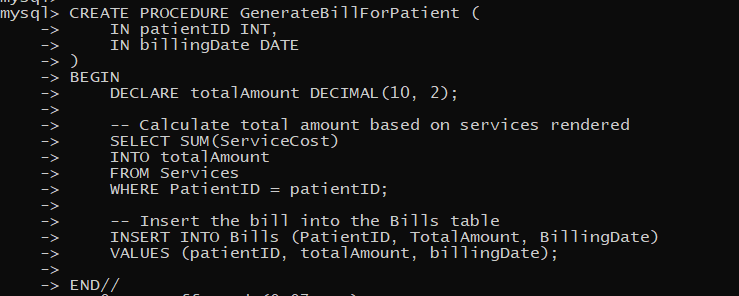
**- Optimize repeated read operations using views/materialized views.**



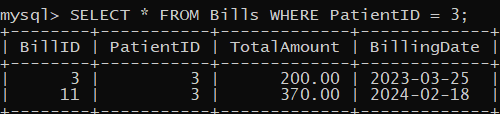
**- Optimize read operations using indexing wherever required. (Create index on at least 1 table)**



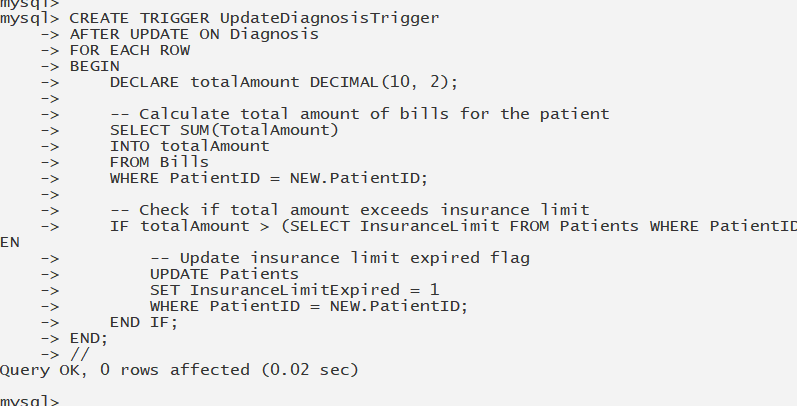
**- Try optimizing bill generation using stored procedures.**







- **Add necessary triggers to indicate when patients medical insurance limit has expired.**



**Q.2 Write a report on your understanding of Rendering and Design Patterns. Mention and elaborate where a particular Rendering pattern is applicable and is well suited for which use case.**

In web development, a rendering pattern refers to the way in which the HTML, CSS, and JavaScript code is all processed and rendered in a web application or website. Different rendering patterns are used to achieve different performance and user experience goals.

Types

SSG

CSR

SSR

|  |  |  |  |
| --- | --- | --- | --- |
| Rendering pattern | description | Use case | example |
| Server-side rendering (SSR) | In SSR, the web server generates the HTML content of a web page on the server-side and sends it to the client's browser. | One common use case for SSR is improving search engine optimization (SEO)  By implementing SSR, you can generate the HTML content on the server and send it to the search engine crawler, ensuring that all your product listings and relevant information are included in the initial response. This helps improve the SEO of your website and ensures that your products are more likely to appear in search engine results, ultimately driving more organic traffic to your site |  |
| Client-side rendering (CSR) | In CSR, the client's browser generates the HTML content of a web page on the client-side using JavaScript. This approach can provide a fast and interactive user experience. | One common use case for client-side rendering is in single-page applications (SPAs). In SPAs, the initial HTML page is loaded from the server, but subsequent interactions and page changes are handled by JavaScript, which dynamically updates the content of the page without requiring a full page reload. | For example, consider a social media application like Facebook or Twitter. When a user navigates to their feed, instead of fetching a new HTML page from the server every time they scroll or interact with a post, the client-side JavaScript code requests data from the server in the background (often in JSON format), and then uses that data to dynamically update the content of the page, such as adding new posts, comments, or notifications. This results in a smoother and more responsive user experience, as the browser only needs to update the parts of the page that have changed, rather than reloading the entire page. |
| Static site generation (SSG) | In SSG, the HTML content of a web page is generated at build time and served to the client as a static file. This approach can provide excellent performance and security but can be less flexible for dynamic content. | One use case for SSG is blog posting r content posting where static file are rendered | Imagine you're a blogger who wants to share your thoughts, experiences, and knowledge with the world. You want your blog to load quickly, be easily navigable, and have a professional appearance. |