Java Bootcamp – Day 30 – Lab 9 – Hospital Database System

Submitted by: Nawaf AlMutairi
Date: 28/11/2024

Table of Contents

Description

The database is designed to manage and organize a hospital's critical information, including patient care, doctor assignments, departmental structure, and financial records. Each table is interrelated to ensure consistency and accuracy across the system. The **Hospitals Table** serves as the core, linking other tables for seamless data integration.

Diagram of The Database Schema

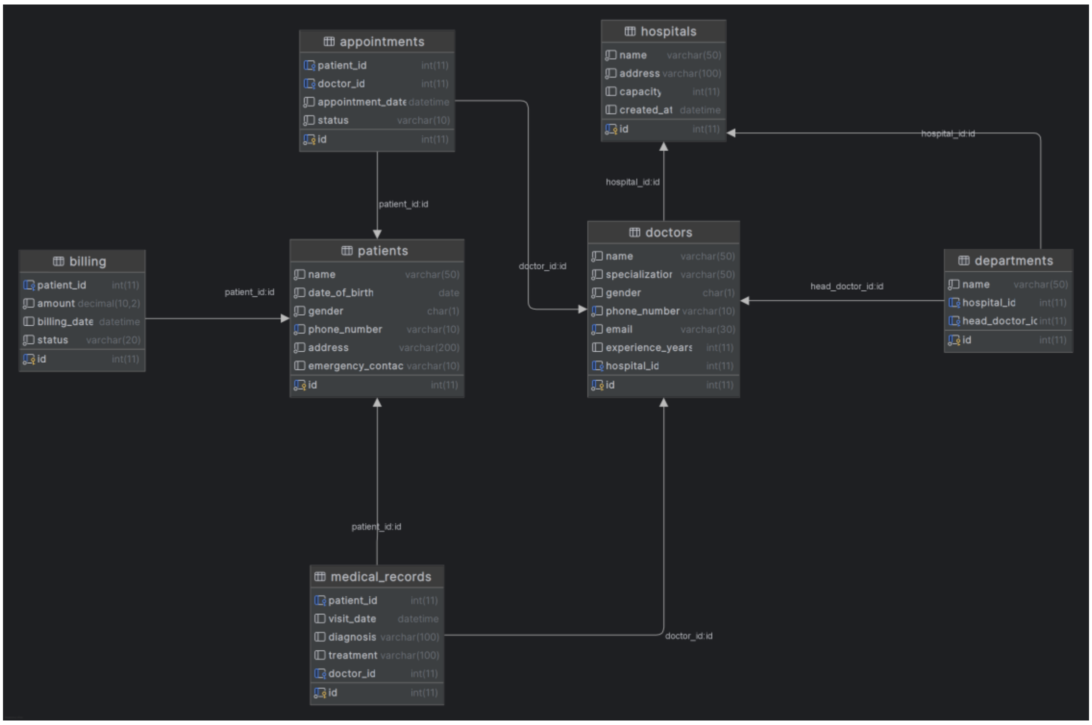


Table 1: Hospitals

- **Description**: Stores information about hospitals.
- **Columns and Constraints**:
  - id: Primary Key (Unique, Non-NULL).
  - name: Non-NULL.
  - address: Non-NULL.
  - capacity: Must be non-negative (CHECK (capacity >= 0)).
  - created_at: Default value is the current timestamp.

Table 2: Doctors

- **Description**: Contains information about doctors.
- **Columns and Constraints**:
  - id: Primary Key (Unique, Non-NULL).
  - name: Non-NULL.
  - specialization: Non-NULL.
  - gender: Must be M or F (CHECK (gender = 'M' OR gender = 'F')).
  - phone_number: Unique, Non-NULL.
  - email: Unique, Non-NULL.
  - experience_years: Must be non-negative (CHECK (experience_years >= 0)).
  - hospital_id: Foreign Key referencing hospitals(id).

Table 3: Departments

- **Description**: Manages department details.
- **Columns and Constraints**:
  - id: Primary Key (Unique, Non-NULL).
  - name: Non-NULL.
  - hospital_id: Foreign Key referencing hospitals(id) (Non-NULL).
  - head_doctor_id: Foreign Key referencing doctors(id).

Table 4: Patients

- **Description**: Stores details of patients.
- **Columns and Constraints**:
  - id: Primary Key (Unique, Non-NULL).
  - name: Non-NULL.
  - date_of_birth: Non-NULL.
  - gender: Must be M or F (CHECK (gender = 'M' OR gender = 'F')).
  - phone_number: Unique, Non-NULL.
  - address: Non-NULL.
  - emergency_contact: No specific constraints (optional).

Table 5: Appointments

- **Description**: Tracks appointments between doctors and patients.
- **Columns and Constraints**:
  - id: Primary Key (Unique, Non-NULL).
  - patient_id: Foreign Key referencing patients(id) (Non-NULL).
  - doctor_id: Foreign Key referencing doctors(id) (Non-NULL).
  - appointment_date: Non-NULL.
  - status: Must be one of Pending, Canceled, or Completed (CHECK (status IN ('Pending', 'Canceled', 'Completed'))).

Table 6: Medical Records

- **Description**: Stores patient medical history.
- **Columns and Constraints**:
  - id: Primary Key (Unique, Non-NULL).
  - patient_id: Foreign Key referencing patients(id) (Non-NULL).
  - visit_date: Default value is the current timestamp.
  - diagnosis: No specific constraints.
  - treatment: No specific constraints.
  - doctor_id: Foreign Key referencing doctors(id).

Table 7: Billing

- **Description**: Tracks billing and payment details.
- **Columns and Constraints**:
  - id: Primary Key (Unique, Non-NULL).
  - patient_id: Foreign Key referencing patients(id) (Non-NULL).
  - amount: Non-NULL, Must be a positive number.
  - billing_date: Default value is the current timestamp.
  - status: Must be one of Pending or Paid (CHECK (status IN ('Pending', 'Paid'))).

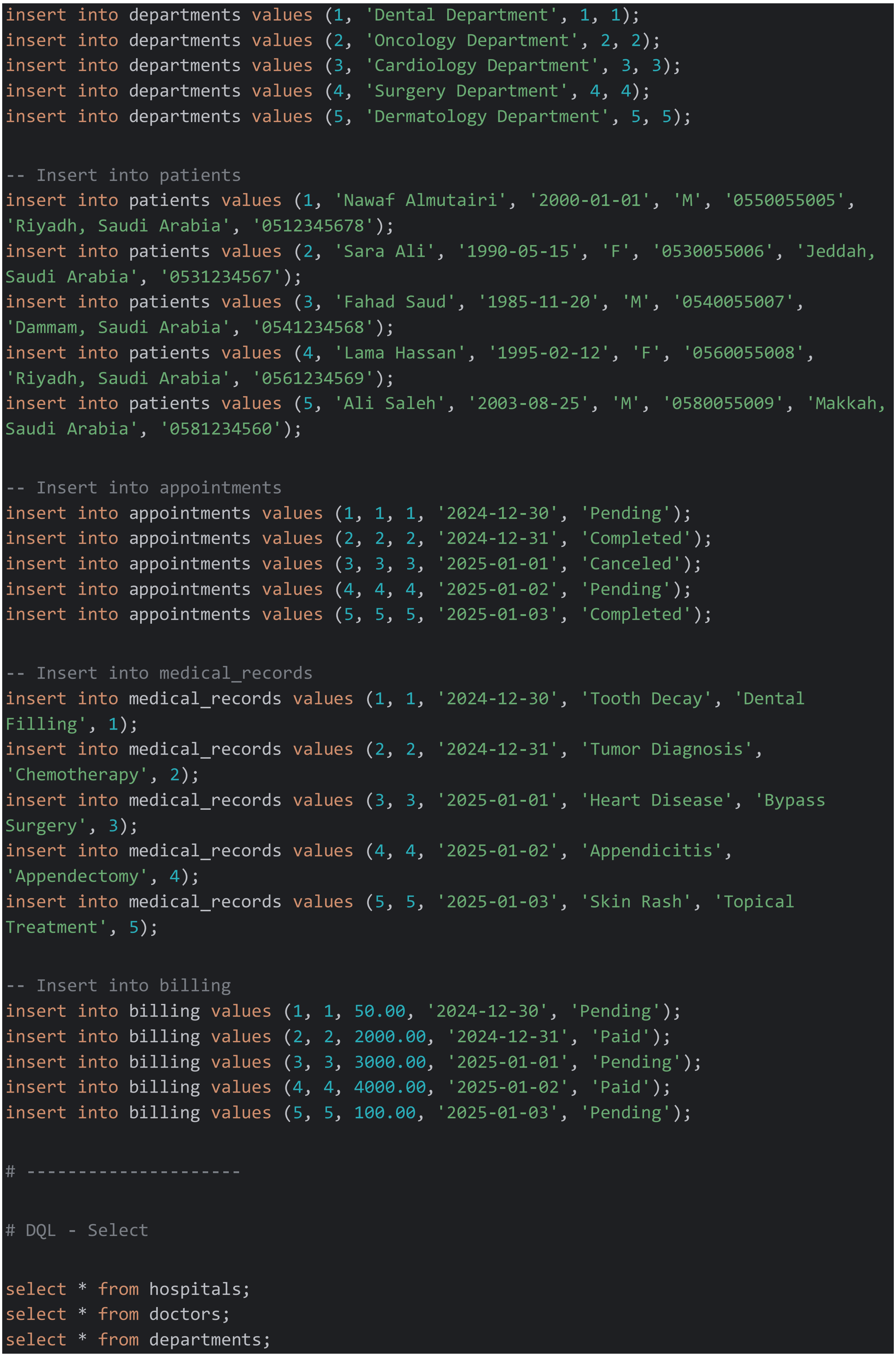
Relationships Between Tables

- A **hospital** can have multiple **doctors**.
- A **department** belongs to a single **hospital** and is managed by a specific **doctor**.
- A **patient** can have multiple **appointments** with different **doctors**.
- Each **appointment** generates a corresponding entry in the **medical records** table.
- **Billing** is linked directly to the **patients** and tracks their financial obligations.

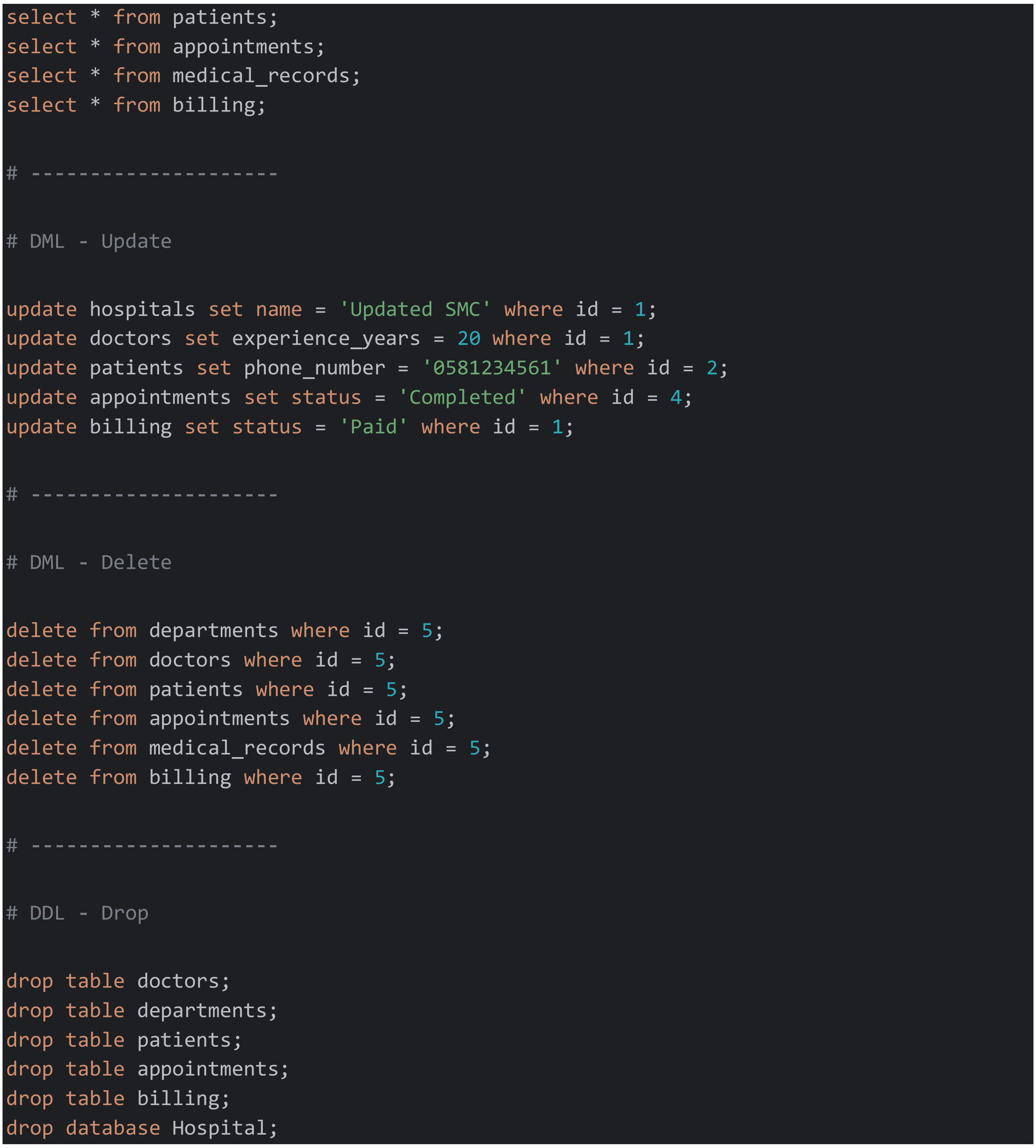
Implementation

```
# DDL - Create

create database Hospital;

use hospital;

create table hospitals (
id int primary key ,
    name varchar(50) not null ,
    address varchar(100) not null ,
    capacity int check (capacity >= 0) ,
    created_at datetime default current_timestamp
);

create table doctors (
    id int primary key ,
    name varchar(50) not null ,
    specialization varchar(50) not null ,
    gender char(1) not null check ( gender = 'M' or gender = 'F' ) ,
    phone_number varchar(10) not null unique ,
    email varchar(30) not null unique ,
    experience_years int check (experience_years >= 0) ,
    hospital_id int ,
    foreign key (hospital_id) references hospitals(id)
);

create table departments (
    id int primary key ,
    name varchar(50) not null ,
    hospital_id int ,
    head_doctor_id int ,
    foreign key (hospital_id) references Hospitals(id) ,
    foreign key (head_doctor_id) references doctors(id)
);

create table patients (
    id int primary key ,
    name varchar(50) not null ,
    date_of_birth date not null ,
    gender char(1) not null check ( gender = 'M' or gender = 'F' ) ,
    phone_number varchar(10) not null unique ,
    address varchar(200) not null ,
    emergency_contact varchar(10)
);

create table appointments (
    id int primary key ,
    patient_id int ,
    doctor_id int ,
    appointment_date datetime not null ,
    status varchar(10) not null
```

```
        check ( status = 'Pending' or status = 'Canceled' or status = 'Completed' ) ,
    foreign key (patient_id) references patients(id) ,
    foreign key (doctor_id) references doctors(id)
);

create table medical_records (
    id int primary key ,
    patient_id int ,
    visit_date datetime default current_timestamp ,
    diagnosis varchar(100) ,
    treatment varchar(100) ,
    doctor_id int ,
    foreign key (patient_id) references patients(id) ,
    foreign key (doctor_id) references doctors(id)
);

create table billing (
    id int primary key ,
    patient_id int ,
    amount decimal(10,2) not null ,
    billing_date datetime default current_timestamp ,
    status varchar(20) not null check ( status = 'Pending' or status = 'Paid' ) ,
    foreign key (patient_id) references patients(id)
);

# --------------------

# DML - Insert

-- Insert into hospitals
insert into hospitals values (1, 'SMC', 'Riyadh, King Abdullah Road', 100, default);
insert into hospitals values (2, 'Habeeb Hospital', 'Riyadh, Saudi Arabia', 200,
default);
insert into hospitals values (3, 'SFH', 'Riyadh, Saudi Arabia', 500, default);
insert into hospitals values (4, 'HWH', 'Riyadh, Saudi Arabia', 500, default);
insert into hospitals values (5, 'Faqeeh Hospital', 'Jeddah, Saudi Arabia', 400,
default);

-- Insert into doctors
insert into doctors values (1, 'Abdullah Alqahtani', 'Dentist', 'M', '0521234567',
'abdullah@gmail.com', 15, 1);
insert into doctors values (2, 'Mohammed Bahrawe', 'Oncologist', 'M', '0541234567',
'mohammed@hotmail.com', 9, 2);
insert into doctors values (3, 'Omar Alessa', 'Cardiologist', 'M', '0551234567',
'omar@example.com', 6, 3);
insert into doctors values (4, 'Faisal Alhabeeb', 'Surgery', 'M', '0561234567',
'faisal@yahoo.com', 7, 4);
insert into doctors values (5, 'Noura Alsheikh', 'Dermatologist', 'F', '0571234567',
'noura@gmail.com', 4, 5);

-- Insert into departments
```



```
insert into departments values (1, 'Dental Department', 1, 1);
insert into departments values (2, 'Oncology Department', 2, 2);
insert into departments values (3, 'Cardiology Department', 3, 3);
insert into departments values (4, 'Surgery Department', 4, 4);
insert into departments values (5, 'Dermatology Department', 5, 5);

-- Insert into patients
insert into patients values (1, 'Nawaf Almutairi', '2000-01-01', 'M', '0550055005',
'Riyadh, Saudi Arabia', '0512345678');
insert into patients values (2, 'Sara Ali', '1990-05-15', 'F', '0530055006', 'Jeddah,
Saudi Arabia', '0531234567');
insert into patients values (3, 'Fahad Saud', '1985-11-20', 'M', '0540055007',
'Dammam, Saudi Arabia', '0541234568');
insert into patients values (4, 'Lama Hassan', '1995-02-12', 'F', '0560055008',
'Riyadh, Saudi Arabia', '0561234569');
insert into patients values (5, 'Ali Saleh', '2003-08-25', 'M', '0580055009', 'Makkah,
Saudi Arabia', '0581234560');

-- Insert into appointments
insert into appointments values (1, 1, 1, '2024-12-30', 'Pending');
insert into appointments values (2, 2, 2, '2024-12-31', 'Completed');
insert into appointments values (3, 3, 3, '2025-01-01', 'Canceled');
insert into appointments values (4, 4, 4, '2025-01-02', 'Pending');
insert into appointments values (5, 5, 5, '2025-01-03', 'Completed');

-- Insert into medical_records
insert into medical_records values (1, 1, '2024-12-30', 'Tooth Decay', 'Dental
Filling', 1);
insert into medical_records values (2, 2, '2024-12-31', 'Tumor Diagnosis',
'Chemotherapy', 2);
insert into medical_records values (3, 3, '2025-01-01', 'Heart Disease', 'Bypass
Surgery', 3);
insert into medical_records values (4, 4, '2025-01-02', 'Appendicitis',
'Appendectomy', 4);
insert into medical_records values (5, 5, '2025-01-03', 'Skin Rash', 'Topical
Treatment', 5);

-- Insert into billing
insert into billing values (1, 1, 50.00, '2024-12-30', 'Pending');
insert into billing values (2, 2, 2000.00, '2024-12-31', 'Paid');
insert into billing values (3, 3, 3000.00, '2025-01-01', 'Pending');
insert into billing values (4, 4, 4000.00, '2025-01-02', 'Paid');
insert into billing values (5, 5, 100.00, '2025-01-03', 'Pending');

# --------------------

# DQL - Select

select * from hospitals;
select * from doctors;
select * from departments;
```



```
select * from patients;
select * from appointments;
select * from medical_records;
select * from billing;

# --------------------

# DML - Update

update hospitals set name = 'Updated SMC' where id = 1;
update doctors set experience_years = 20 where id = 1;
update patients set phone_number = '0581234561' where id = 2;
update appointments set status = 'Completed' where id = 4;
update billing set status = 'Paid' where id = 1;

# --------------------

# DML - Delete

delete from departments where id = 5;
delete from doctors where id = 5;
delete from patients where id = 5;
delete from appointments where id = 5;
delete from medical_records where id = 5;
delete from billing where id = 5;

# --------------------

# DDL - Drop

drop table doctors;
drop table departments;
drop table patients;
drop table appointments;
drop table billing;
drop database Hospital;
```