Muhammad Kaleem

56614

BS – DS 4-1

Lab Task 10 :

Trigger Implementation

Hospital Database

Step 1 :

Creating and using Hospital Database

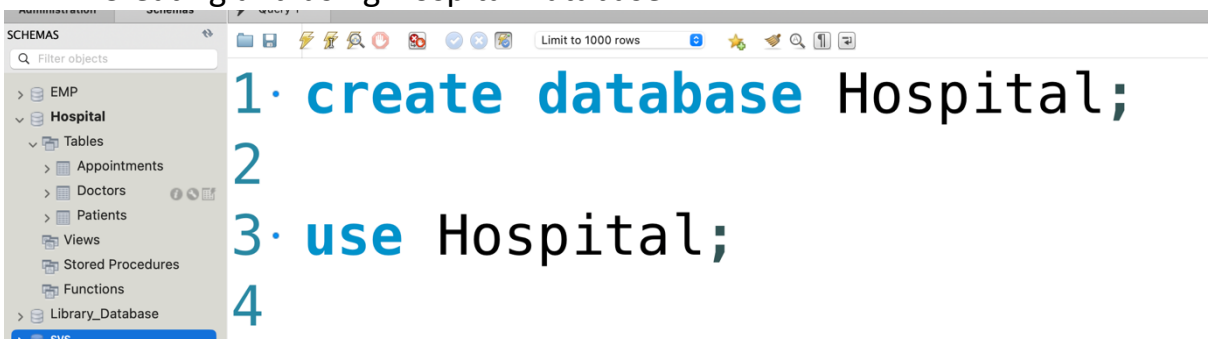

Figure 1.creating and using hospital

Step 2:

Creating table of Doctors to store their info

```
create table Doctors(
doctor_id int primary key,
doctor_name varchar(50),
specialization varchar(50)
);
```

Step 3:

Creating table of Patients to keep their record

```
create table Patients (
patient_id int primary key,
patient_name varchar(50),
date_of_birth date,
phone_no varchar(30)
);
```

Step 4:

Creating the table for appointments

```
create table Appointments (
appointment_id int primary key,
patient_id int,
doctor_id int,
appointment_date date,
visit_reason varchar(200),
consulation_fee float,
foreign key (patient_id) references Patients(patient_id),
foreign key (doctor_id) references Doctors(doctor_id)
);
```

Step 5:

Creating the table for the records of bill

```
create table Billing (
bill_id INT PRIMARY KEY AUTO_INCREMENT,
patient_id int,
bill_amount float,
bill_date date,
bill_month int,
bill_year int,
foreign key (patient_id) references Patients(patient_id)
);
```

Step 6:

Creating trigger to set the new year and month of the bill

```
delimiter \\

create trigger trg_set_bill_month_year
before insert on Billing
for each row
begin
set new.bill_month = month(new.bill_date);
set new.bill_year = year(new.bill_date);
end;
//
delimiter;
```

Step 7:

Creating trigger for the billing depending upon the appointments

```
delimiter \\

create trigger trg_appointment_to_billing
before insert on Appointments
for each row
begin
    declare v_count int;


    select count(*) into v_count
    from Billing
    where patient_id = new.patient_id
      and bill_month = month(new.appointment_date)
      and bill_year = year(new.appointment_date);


    if v_count > 0 then
        update Billing
        set bill_amount = bill_amount + new.consulation_fee,
            bill_date = new.appointment_date
        where patient_id = new.patient_id
          and bill_month = month(new.appointment_date)
          and bill_year = year(new.appointment_date);
    else
        insert into Billing (patient_id, bill_amount, bill_date)
        values (new.patient_id, new.consulation_fee, new.appointment_date);
    end if;
end;
//

delimiter;
```

Step 8:

Inserting values into Doctors table

```
insert into Doctors (doctor_id, doctor_name, specialization)
values (1, 'Dr. Nazrah', 'General Surgeon'),
       (2, 'Dr. Rashida ', 'Medical Speciaist');
```

Step 9:

Inserting values into Patients table

```
insert into Patients (patient_id, patient_name, date_of_birth, phone_no)
values (1, 'John Doe', '1990-01-15', '1234567890'),
       (2, 'Jane Roe', '1985-06-20', '2345678901');
```

Step 10:

Inserting values into Appointments table

```
insert into Appointments (appointment_id, patient_id, doctor_id, appointment_date, visit_reason, consulation_fee)
values (1, 1, 1, '2025-05-01', 'Routine Checkup', 100),
       (2, 1, 2, '2025-05-15', 'Hormonal disease Chheckup', 150),
       (3, 2, 1, '2025-05-20', 'Operation', 120);
```

Step 11:

Data in the Doctors table

```
107
108     select * from Doctors;
```

23:108

Result Grid Filter Rows: Search

doctor_id	doctor_name	specialization
1	Dr. Nazrah	General Surgeon
2	Dr. Rashida	Medical Speciaist
NULL	NULL	NULL

Step 12:

Data in the Patients table

```
107
108    select * from Patients;
```

24:108

Result Grid Filter Rows: Search Ed

patient_id	patient_name	date_of_bir...	phone_no
1	John Doe	1990-01-15	1234567890
2	Jane Roe	1985-06-20	2345678901
NULL	NULL	NULL	NULL

Step 13:

Data in the Appointments table

```
108    select * from Appointments;
```

28:108

Result Grid Filter Rows: Search Edit: Export/Import:

appointment...	patient_id	doctor_id	appointment_d...	visit_reason	consulation_f...
1	1	1	2025-05-01	Routine Checkup	100
2	1	2	2025-05-15	Hormonal disease Chheckup	150
3	2	1	2025-05-20	Operation	120
NULL	NULL	NULL	NULL	NULL	NULL

Conclusion :

We designed and implemented a hospital database in MySQL Workbench, creating multiple interconnected tables using primary and foreign keys to ensure data integrity. We further enhanced the system by adding triggers, which automatically manage billing records when new appointments are made, ensuring real-time updates without manual intervention. This setup demonstrates how relational databases, combined with automation through triggers, can streamline hospital operations like appointments and billing, improve accuracy, and reduce administrative workload.