**CC-215**

**Database System**

*Fatima Tul Zahra*

**May 2,2025**

**Question:**

**Hospital System with Shifts and Medical History Logs**

Doctors, nurses, and administrative staff are assigned shifts based on availability and department load. Patient histories include lab reports, prescriptions, allergies, surgeries, and emergency visits. Each ward has capacity rules. The billing system integrates lab, pharmacy, and consultation charges and generates receipts. Insurance claims are digitally filed and tracked through the portal.

**Database Design and Implementation in MySQL**

**Entity Relationship Diagram**

**Relational schema\table schema**

**Normalization**

**Implementation in MySQL**

**Entity Relationship Diagram:**

**Notations used in ERD:**

Chen and Crow’s Foot are two different notations used in ERDs (Entity-Relationship Diagrams).

They both represent the same concepts (entities, attributes, relationships), but in different styles:

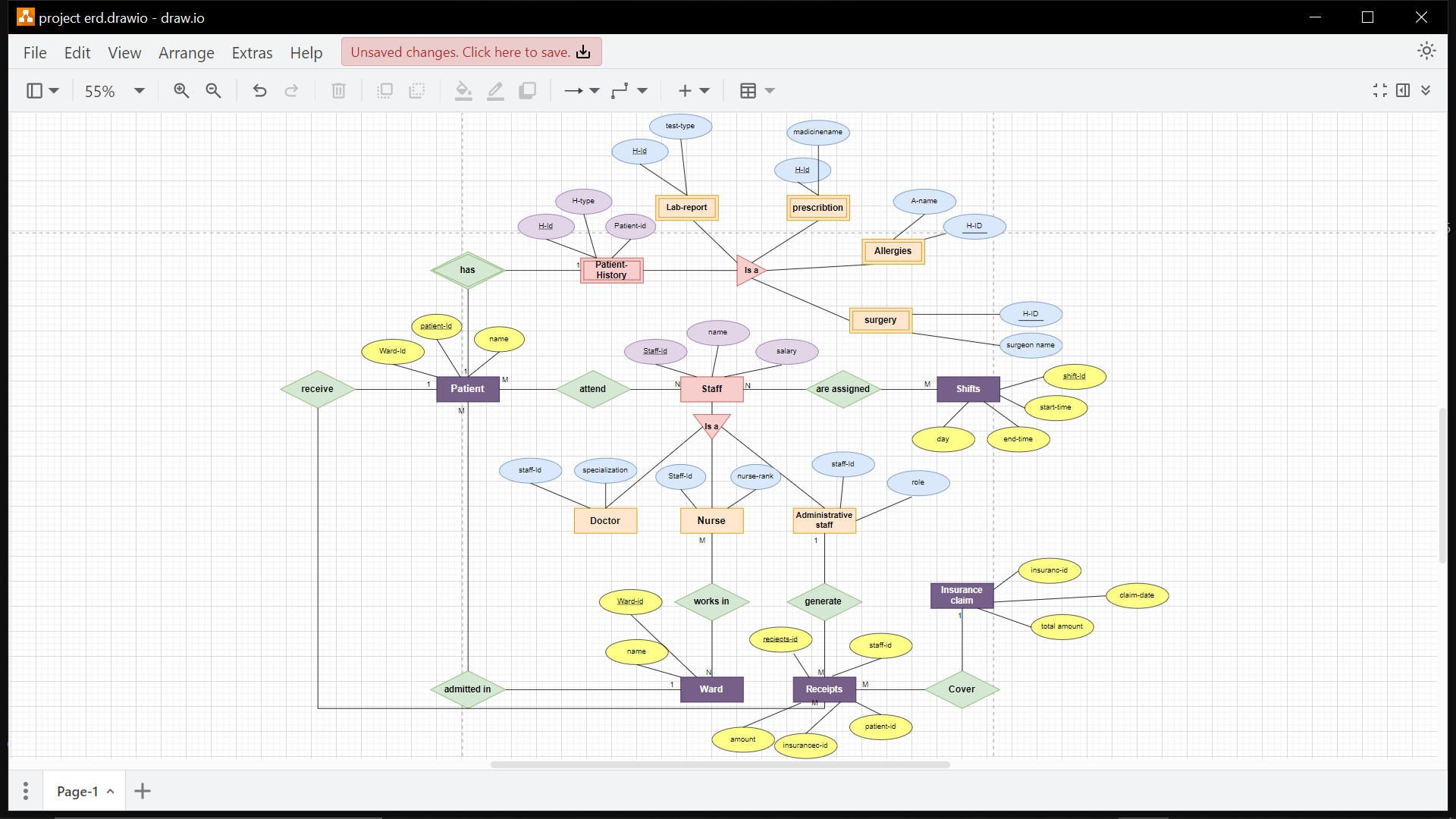
**Chen Model:**

Uses rectangles for entities, ellipses for attributes, and diamonds for relationships. It's simple and used in conceptual design.

**Crow’s Foot Model:**

Uses boxes for entities and lines with symbols (like a crow’s foot) for relationships. It’s more detailed and used in logical/physical design.

**ER Diagram using Chen Model**

****

Specialization is orange here

Attributes of specialization is blue

Superclass is red

Attributes of superclass is light purple

Relationships are green

Simple entities are dark purple

Attributes of Simple entities are yellow

Weak entity is represented by a double rectangle

**Relational schema\table schema from ERD**

**Staff** (Staff-Id, name, salary)

***Specialization:***

* **Doctor** (Staff-Id, specialization)
* **Nurse** (Staff-Id, nurse-rank)
* **Administrative staff** (Staff-Id, role)

**Shift** (Shift-Id, start-time, end-time, day)

**Staffshift** (Staff-Id, shift-Id) (3rd table of staff and shift)

**Patient** (Patient-Id, Ward-id, name)

**Patientstaff** (patient-Id, Staff-id) (3rd table of staff and patient)

**Ward** (Ward-id, name)

**wardNurse** (Ward-Id, staff-id) (3rd table of Ward and nurse)

**Receipts** (Receipts-id, staff-id, patient-id, insurance-id, amount)

**Insurance claim** (insurance-id, total-amount, claim-date)

**Patient history** (H-id, patient-id, H-type)

***Specialization:***

* **Lab report** (H-id, test-type)
* **Surgery** ( H-id, surgeon-name)
* **Prescription** ( H-id, medicine-name)

**Normalization**

**First Normal Form (1NF)**

**Definition:**

* No multivalued attributes (each attribute holds atomic values).
* Each field contains only one value.
* Each record is unique.

As all entities have atomic attributes, no multivalued attributes. Each record in every table is unique with a clear primary key. Therefore, the above table schema First Normal Form (1NF).

**Second Normal Form (2NF)**

**Definition:**

* Must be in 1NF.
* No partial dependency (non-prime attributes depend on the whole primary key, not part of it).
* If the table has a composite primary key, each non-key attribute must depend on the whole composite key, not just a part of it.
* **Tables with Primary Key and Attributes Dependent on It:** These tables have single primary keys and their attributes fully depend on the primary key, such as

**Staff, Doctor, Nurse, Administrative staff, Shift, Ward (satisfy the 2NF)**

In Doctor, Nurse, Administrative staff the staff-Id act as Primary key and foreign key at the same time

* **Tables with Composite Primary Key:** These tables use composite keys to handle many-to-many relationships such as **StaffShift, PatientStaff, WardNurse**

**(satisfy the 2NF)**

* **Tables with Attributes Dependent on Primary Key, with Foreign Keys:** These tables have single primary keys, and their attributes depend on the primary key, but they also include foreign keys referencing other tables such as **Patient, Receipt, Patient history. (satisfy the 2NF)**
* **Lab Report(Patient-Id, Test-Type)**

**Primary Key: ( H-Id) (Foreign key) at the same time.**

**Test-Type depends on H-Id . (satisfy the 2NF)**

* **Surgery( H-Id, Surgeon-Name)**

**Primary Key: ( H-Id) (Foreign key) at the same time.**

**Surgeon-Name depends on H-Id . (satisfy the 2NF)**

* **Prescription(H-Id, Medicine-Name)**

**Primary Key: (H-Id) (Foreign key) at the same time.**

**Medicine-Name depends on H-Id . (satisfy the 2NF)**

**So all the tables are in 2NF as there is no partial dependency**

**Third Normal Form (3NF)**

**Definition:**

* Must be in 2NF.
* No transitive dependency (non-prime attribute should not depend on another non-prime attribute).

**Staff table:** Name and Salary depend directly on Staff-Id.

**Doctor/Nurse/Admin:** specialization, nurse-rank, and role depend directly on Staff-Id.

**Receipts:** Amount depends directly on Receipts-id.

**Insurance claim**: total-amount, claim-date depend on Insurance-id.

PatientHistory specialization (Lab Report, Surgery, Prescription) are separated correctly.

**No non-key attribute depends on another non-key attribute.**

As there are no transitive dependencies, and non-prime attributes depend only on the primary key,

the database satisfies Third Normal Form (3NF).

***So the tables are in normal form as it satisfies all the conditions of normalization***

**Common commands**

1. **To see existing databases:**

show databases;

**2.To Create a database:**

CREATE DATABASE database\_name;

**3.To use the database:**

USE database\_name;

**4.To create table:**

CREATE TABLE table\_name (

column1\_name DATA\_TYPE CONSTRAINT,

column2\_name DATA\_TYPE CONSTRAINT,

…);

**5.To see tables in database:**

show tables;

**6.to insert data in tables:**

insert into table\_name values(‘value1’,’value2’,…);

**6.to describe table:**

desc table\_name;

**7.To see the table with values;**

SELECT \* FROM table\_name;

**8.To update the existing values:**

UPDATE table\_name SET column\_name = new\_value

WHERE condition;

**9.To add new column:**

alter table table\_name add new\_column\_name datatype(domain);

**10.To add foreign key to existing table:**

alter table table\_name add new\_column\_name datatype(domain);

alter table table\_name add constraint constraint\_name foreign key (column\_name) references referenced\_table (referenced\_column);

**11.To see specific column from table:**

select column\_name from table\_name;

**12.To see specific cell from table:**

select column\_name from table\_name where condition;

**13.To Rename column in output temporarily:**

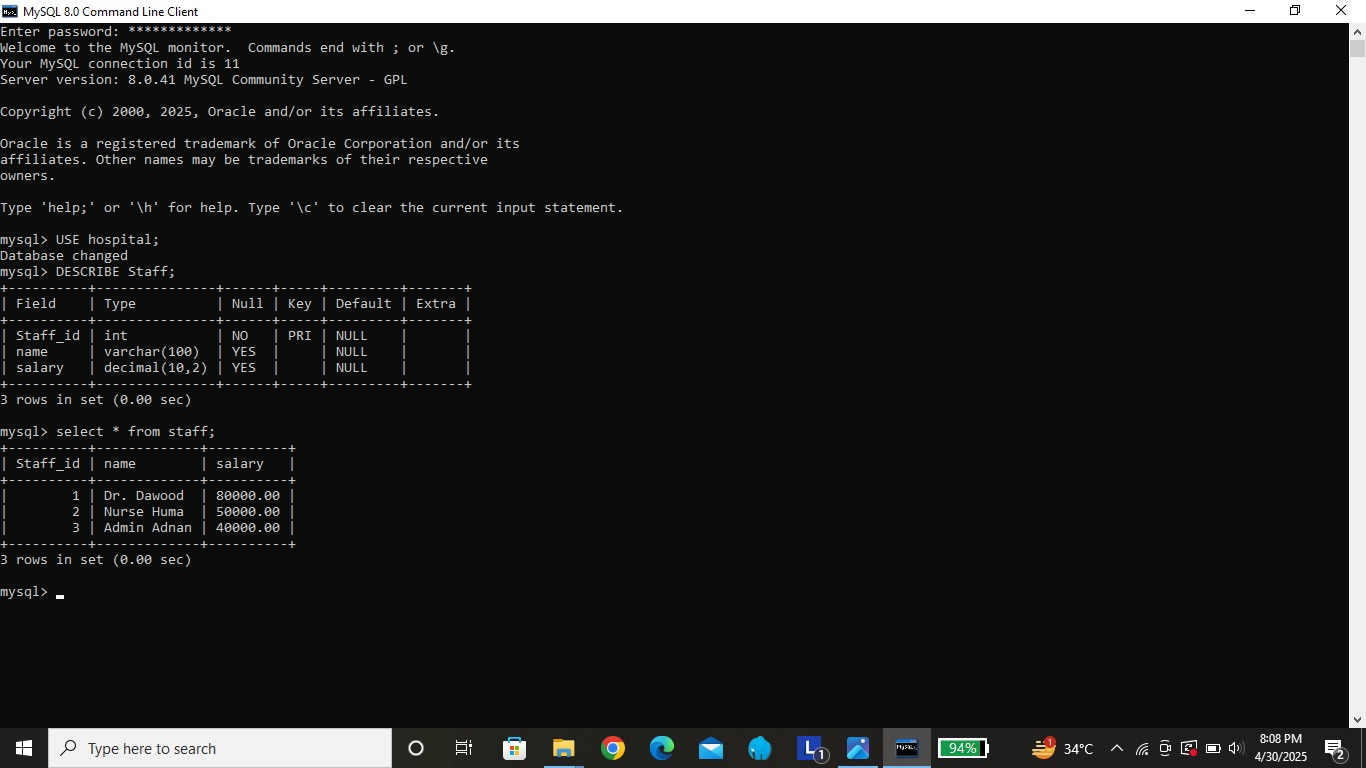
select column\_name as new\_column\_name from table\_name;

**14.To add foreign key at the time of table creation:**

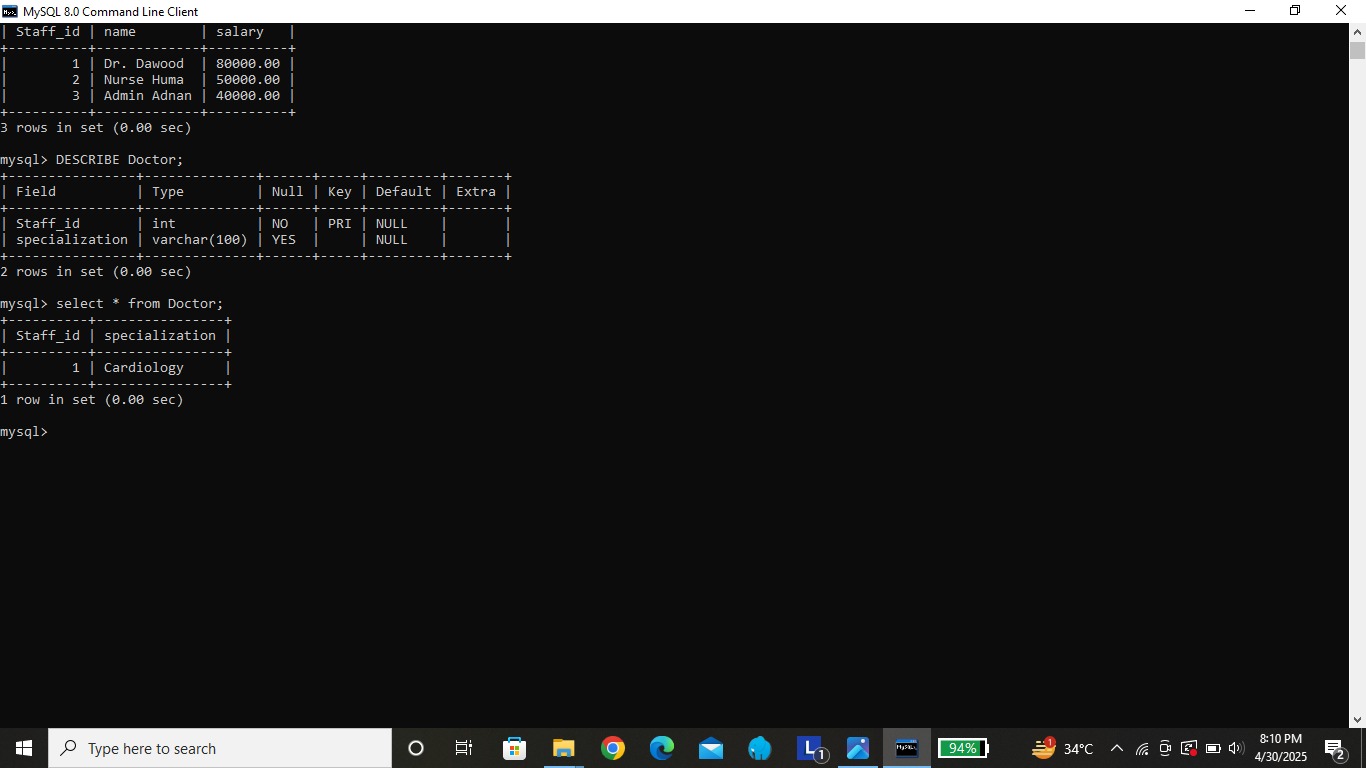
CREATE TABLE table\_name (Column\_name datatype constraint, FOREIGN KEY (column\_name) REFERENCES referenced\_table (referenced\_column);

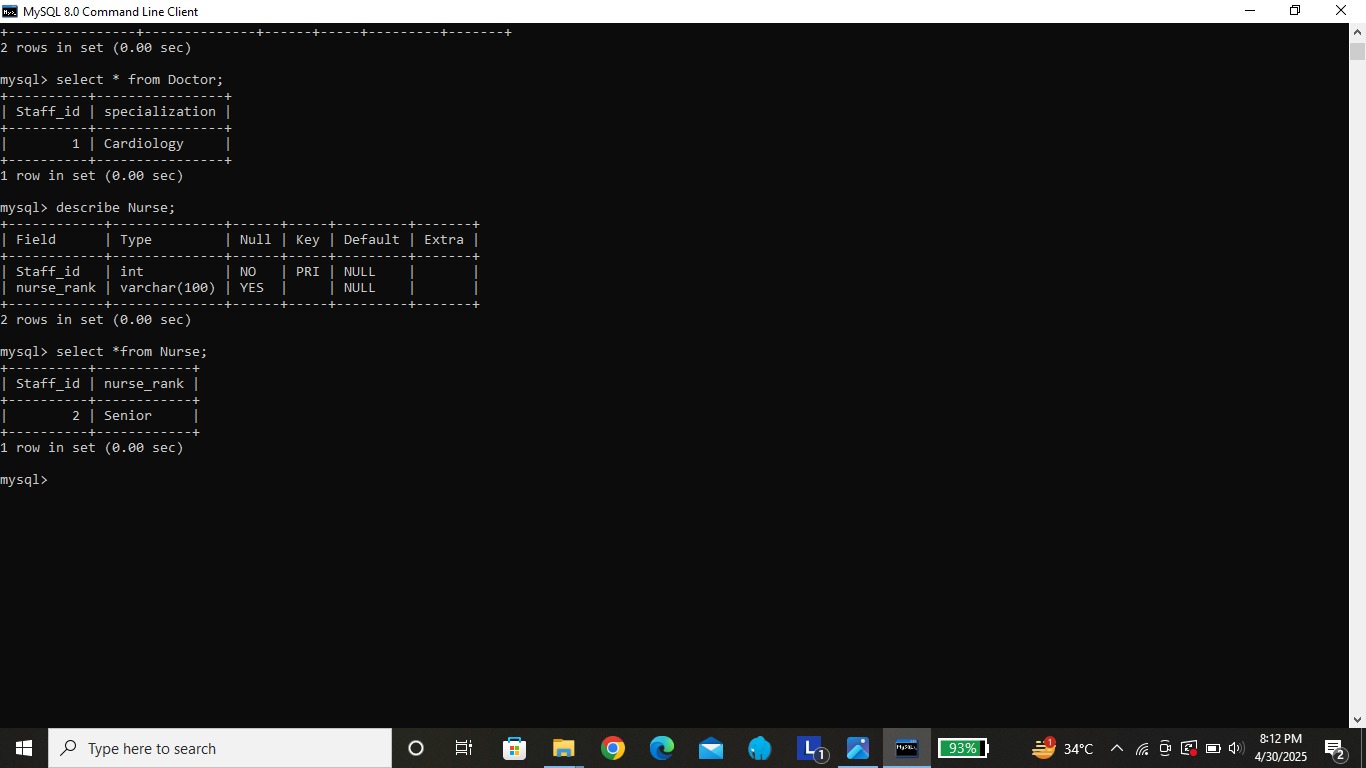
**Implementation in MySQL**

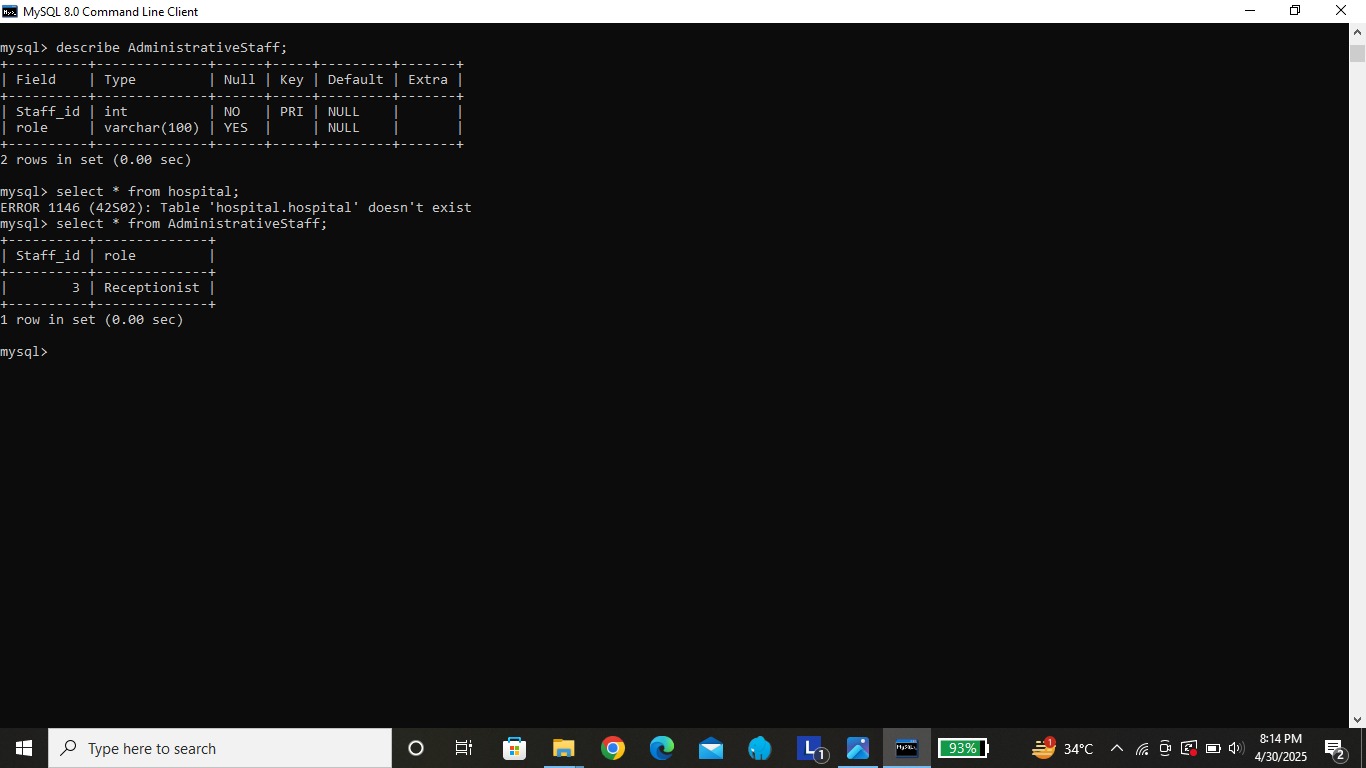
* show databases;
* create database hospital;
* use hospital;
* creatr table Staff(Staff\_id int(3) primary key,name varchar(100),salary decimal(10,2));
* desc staff;
* insert into Staff values('1','Dr.dawood','80000.00');
* insert into Staff values('2','nurse Huma','50000.00'),('3','Admin Adnan','40000.00');
* select \* from Staff;



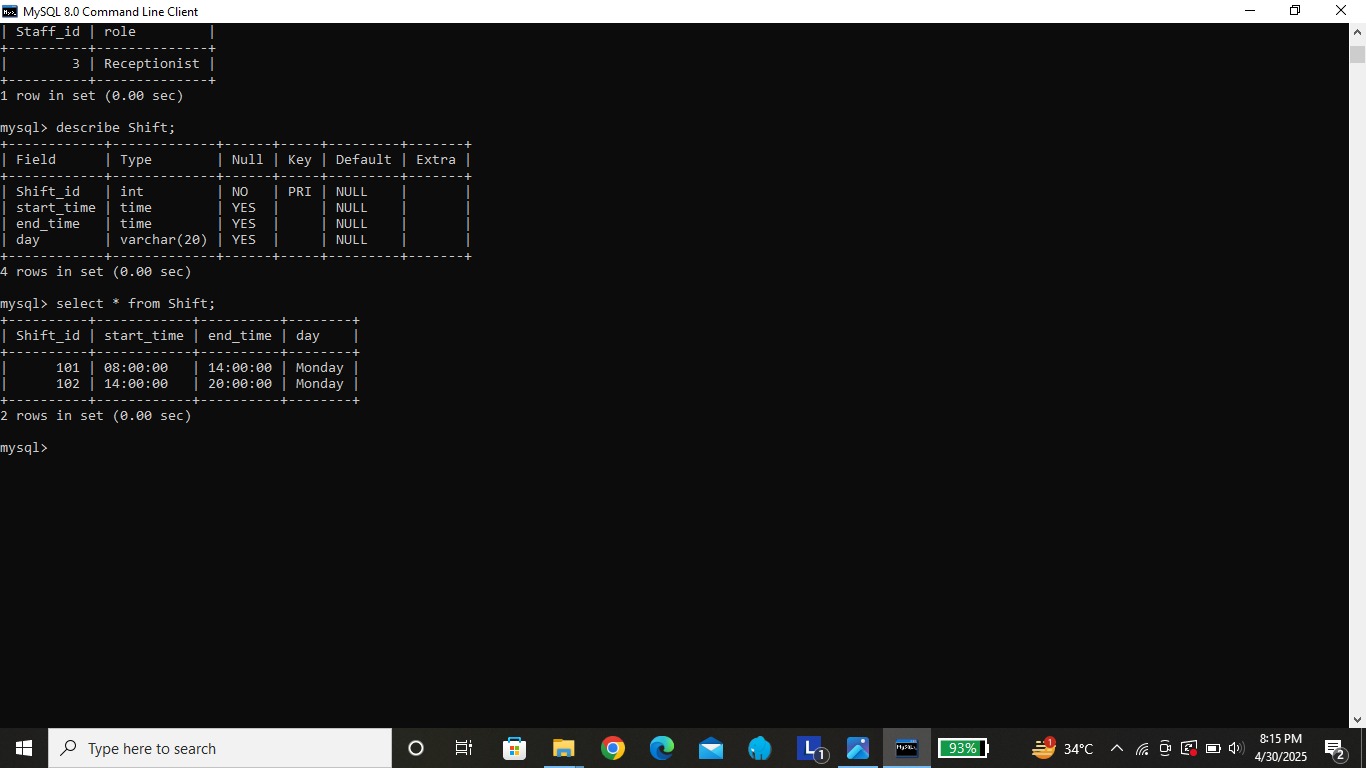
* CREATE TABLE Doctor ( Staff\_Id INT PRIMARY KEY, specialization VARCHAR(100), CONSTRAINT fk\_Staff FOREIGN KEY (Staff\_Id) REFERENCES Staff(Staff\_Id));
* CREATE TABLE Nurse (Staff\_Id INT PRIMARY KEY,nurse\_rank VARCHAR(100), CONSTRAINT fk\_Staff FOREIGN KEY (Staff\_Id) REFERENCES Staff(Staff\_Id));
* CREATE TABLE AdministrativeStaff ( Staff\_Id INT PRIMARY KEY, role VARCHAR(100), CONSTRAINT fk\_Staff FOREIGN KEY (Staff\_Id) REFERENCES Staff(Staff\_Id));
* INSERT INTO Doctor VALUES (1, 'Cardiology');
* INSERT INTO Nurse VALUES (2, 'Senior');
* INSERT INTO AdministrativeStaff VALUES (3, 'Receptionist');
* desc Doctor;
* desc Nurse;
* desc AdministrativeStaff;
* select \* from Doctor;
* select \* from Nurse;
* select \* from AdministrativeStaff;



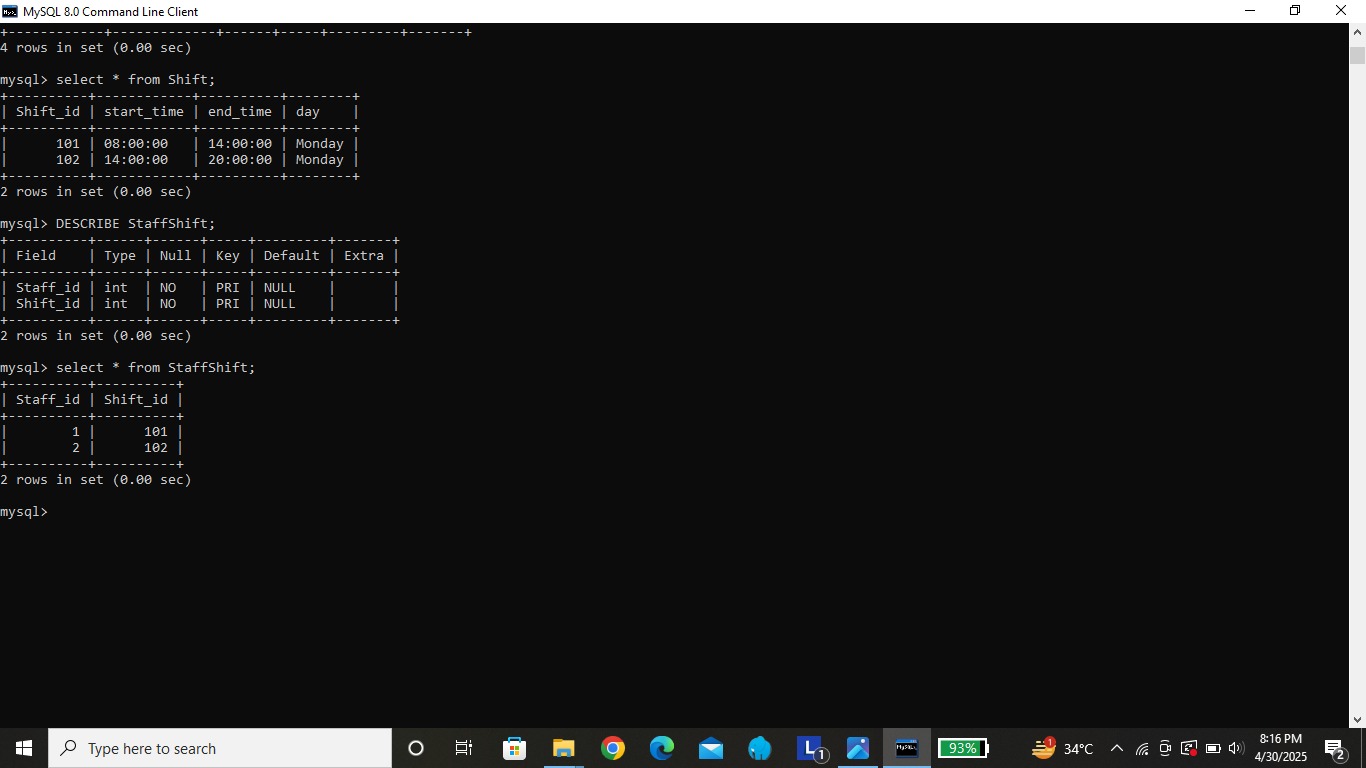




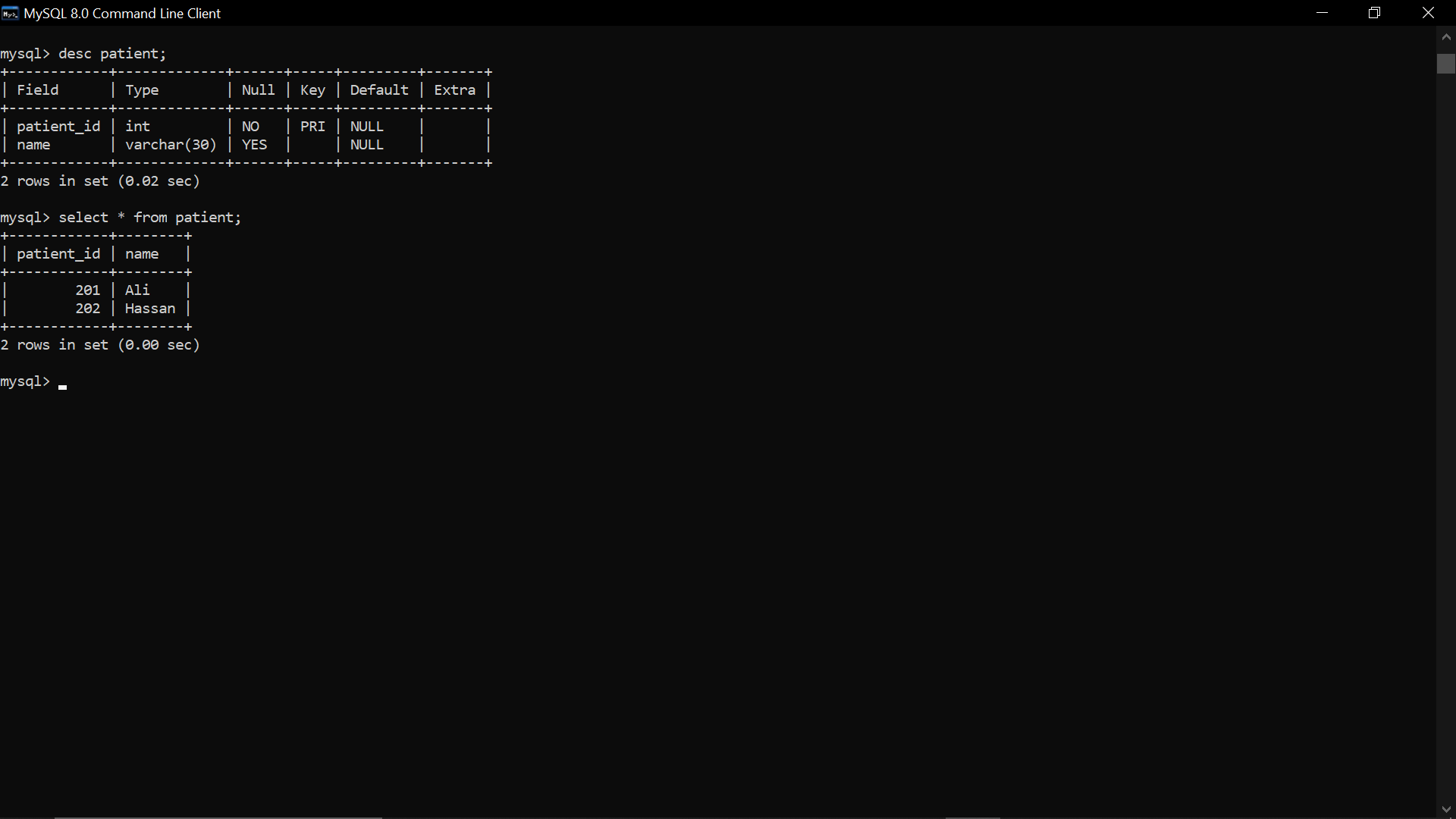
* CREATE TABLE Shift ( Shift\_Id INT PRIMARY KEY, start\_time TIME,end\_time TIME, day VARCHAR(20));
* INSERT INTO Shift VALUES (101, '08:00:00', '14:00:00', 'Monday');
* INSERT INTO Shift VALUES (102, '14:00:00', '20:00:00', 'Monday');



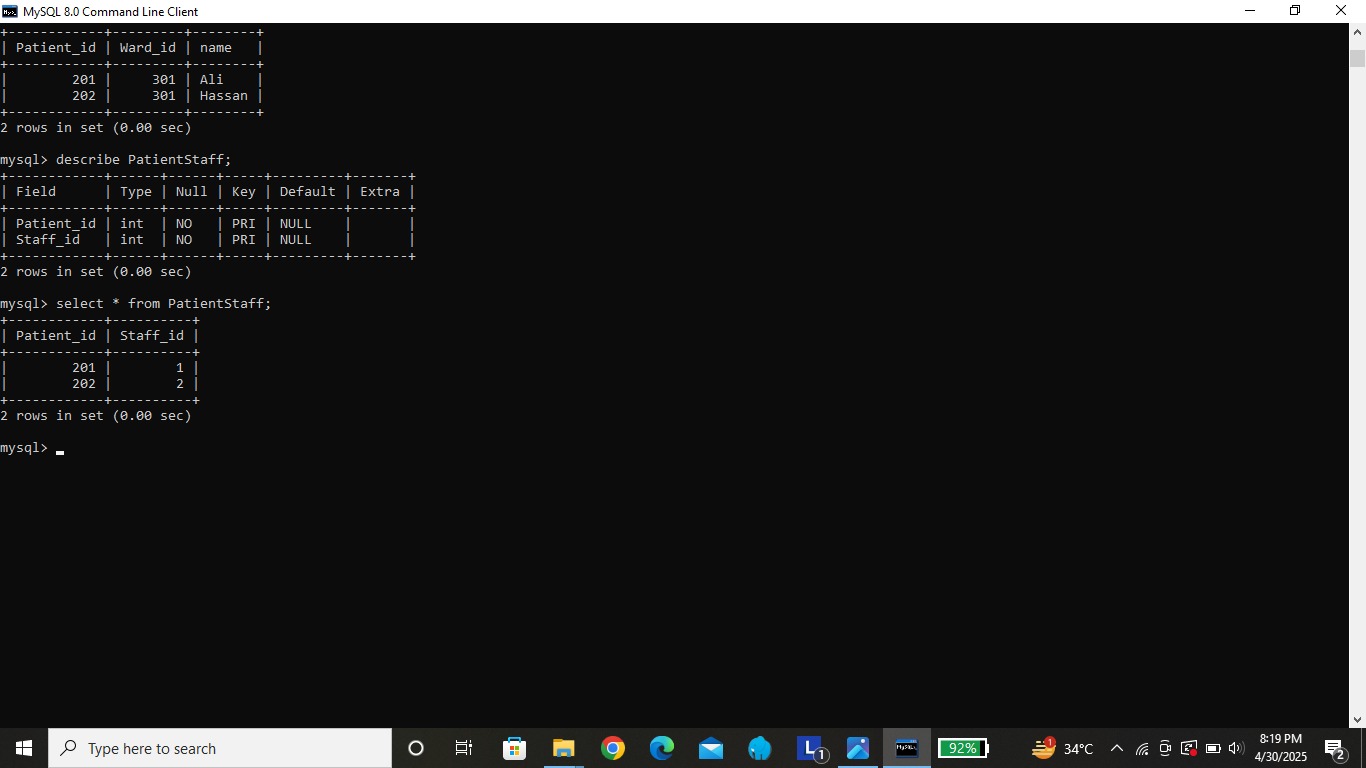
* CREATE TABLE StaffShift ( Staff\_Id INT,Shift\_Id INT, PRIMARY KEY (Staff\_Id, Shift\_Id), CONSTRAINT fk\_StaffShift\_Staff FOREIGN KEY (Staff\_Id) REFERENCES Staff(Staff\_Id), CONSTRAINT fk\_StaffShift\_Shift FOREIGN KEY (Shift\_Id) REFERENCES Shift(Shift\_Id));
* INSERT INTO StaffShift VALUES (1, 101);
* INSERT INTO StaffShift VALUES (2, 102);



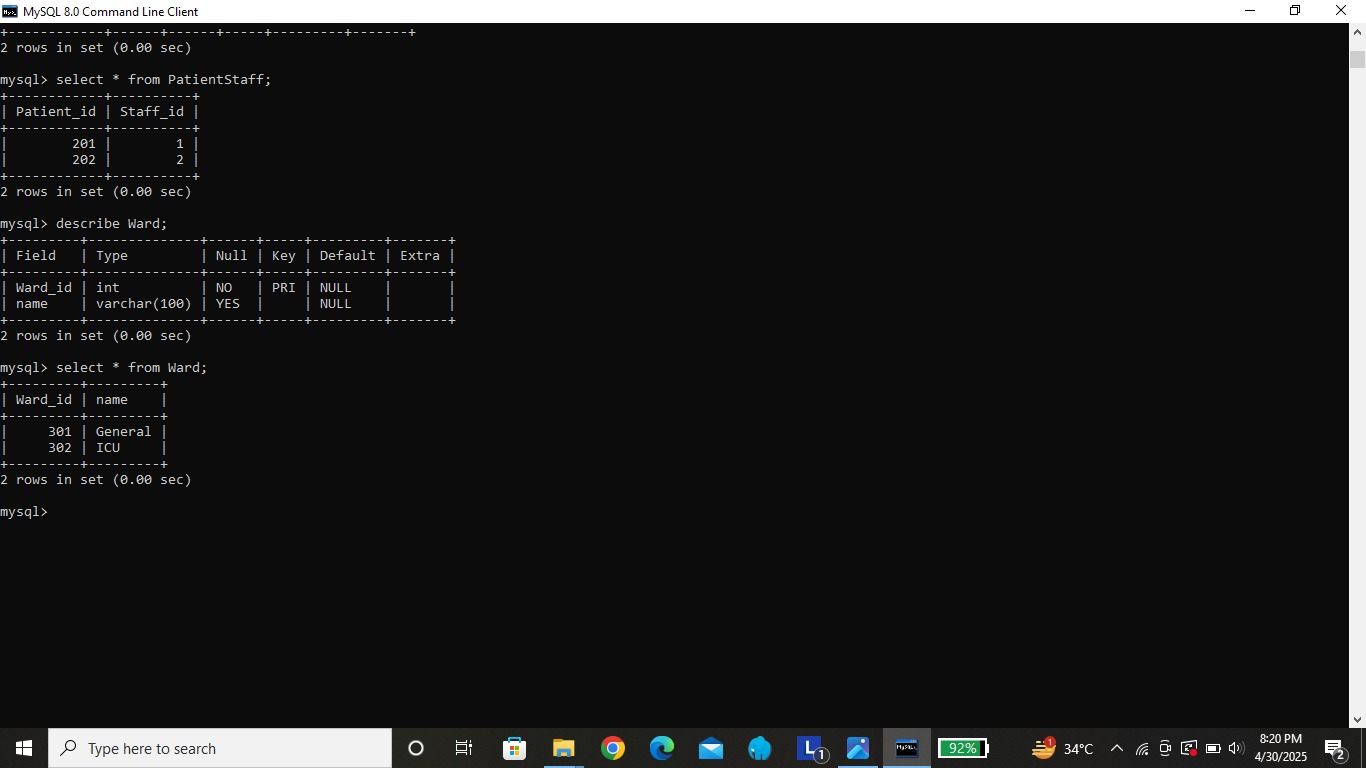
* create table patient(patient\_id int(3) primary key,name varchar(30));
* insert into patient values('201','Ali'),('202','Hassan');



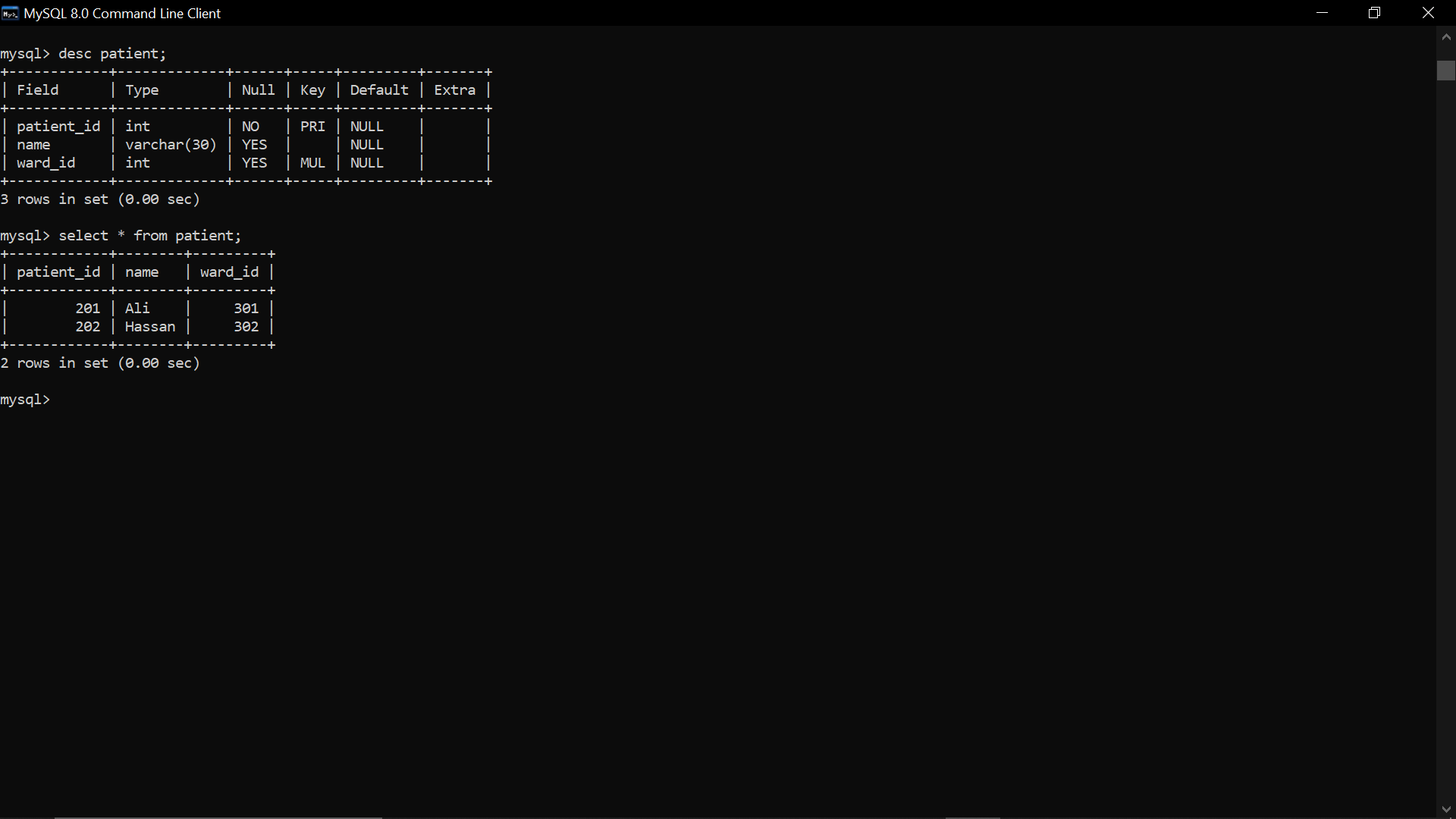
* CREATE TABLE PatientStaff (Patient\_Id INT,Staff\_Id INT, PRIMARY KEY (Patient\_Id, Staff\_Id), CONSTRAINT fk\_Patient FOREIGN KEY (Patient\_Id) REFERENCES Patient(Patient\_Id), CONSTRAINT fk\_PatientStaff\_Staff FOREIGN KEY (Staff\_Id) REFERENCES Staff(Staff\_Id));
* INSERT INTO PatientStaff VALUES (201, 1);
* INSERT INTO PatientStaff VALUES (202, 2);



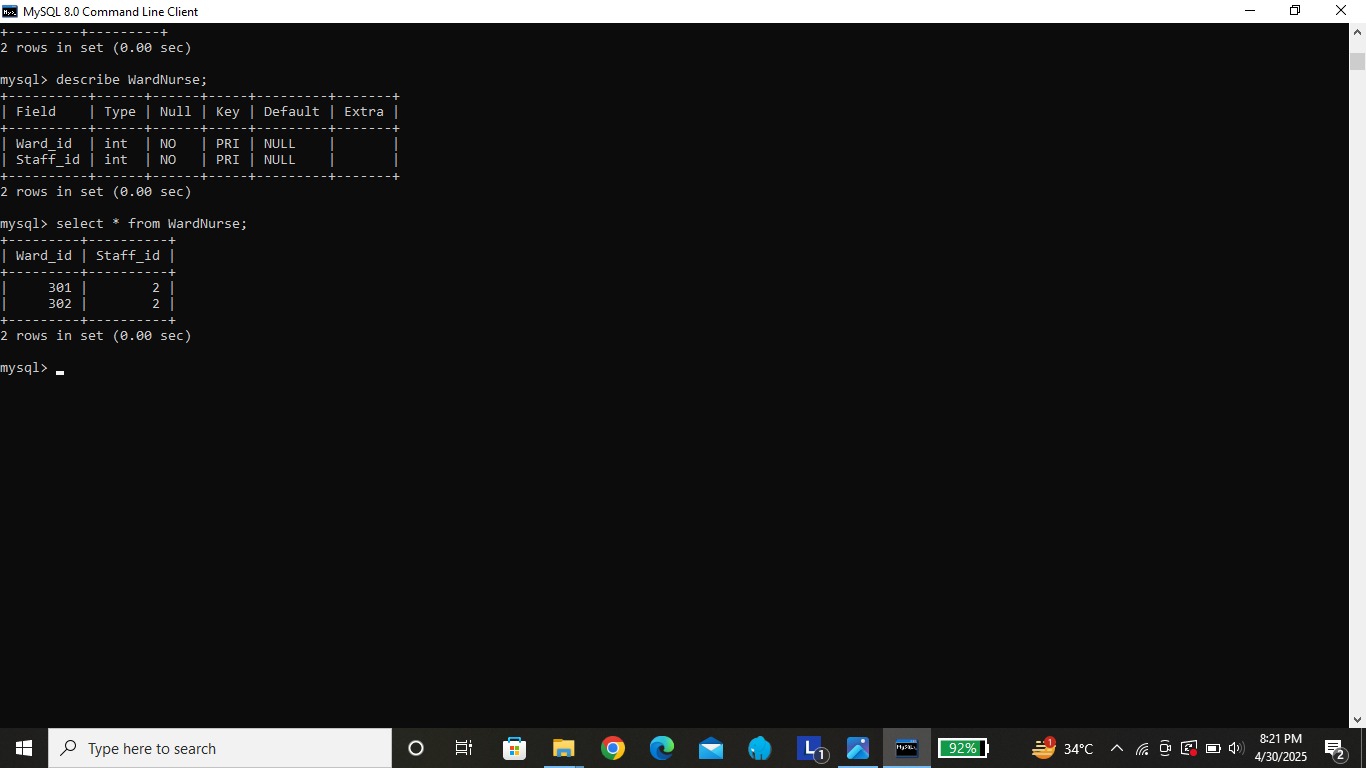
* CREATE TABLE Ward (Ward\_Id INT PRIMARY KEY, name VARCHAR(100));
* INSERT INTO Ward VALUES (301, 'General'), (302, 'ICU');



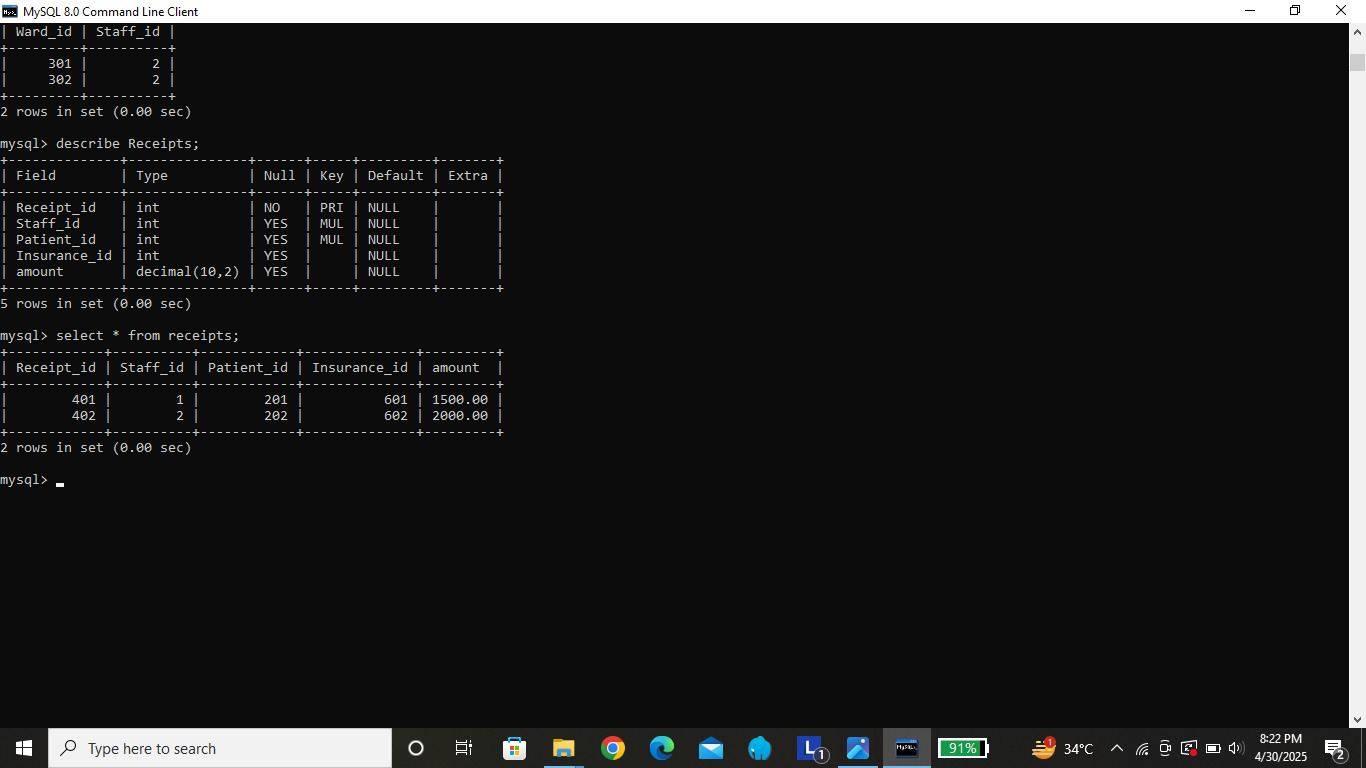
* ALTER TABLE patient ADD ward\_id INT,ADD CONSTRAINT fk\_ward FOREIGN KEY (ward\_id) REFERENCES ward(ward\_id);
* update patient set ward\_id=’301’ where patient\_id=201;

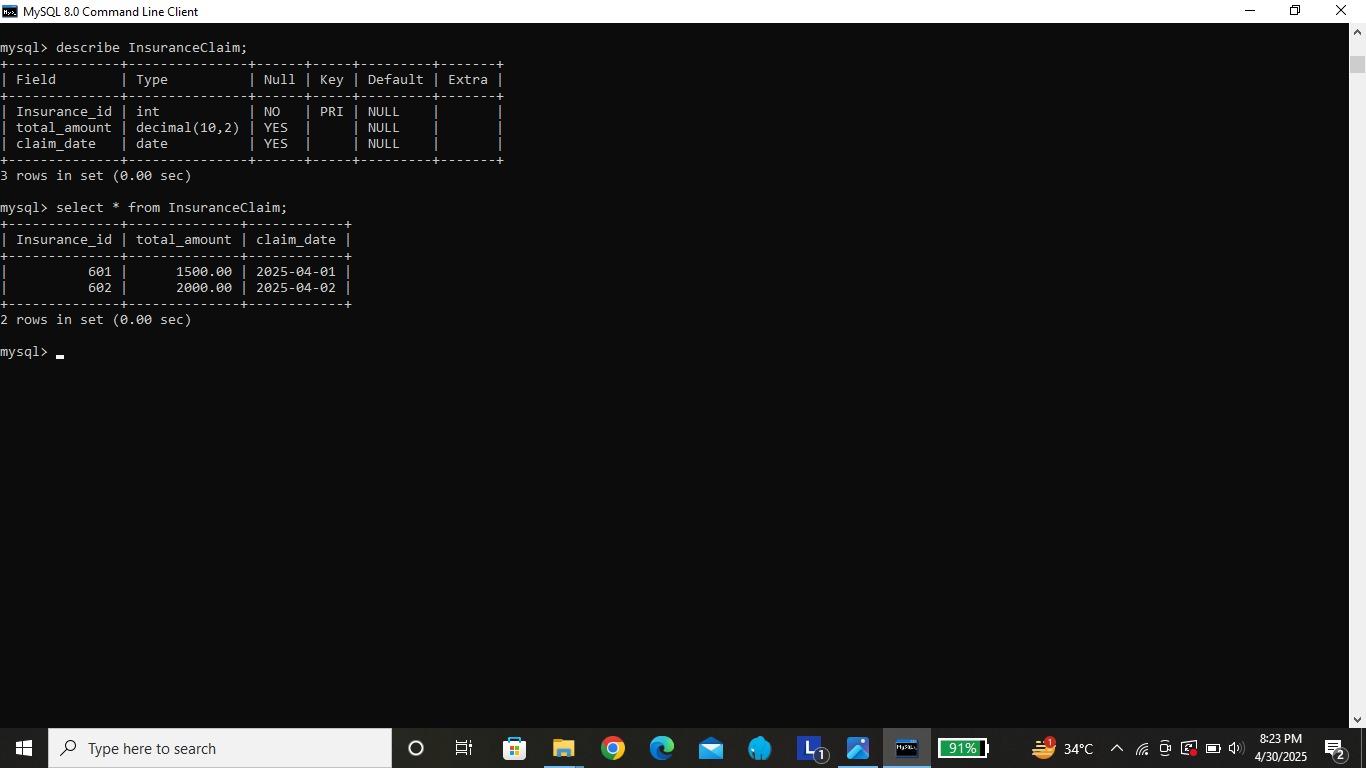


* CREATE TABLE WardNurse (Ward\_Id INT,Staff\_Id INT, PRIMARY KEY (Ward\_Id, Staff\_Id), CONSTRAINT fk\_Ward FOREIGN KEY (Ward\_Id) REFERENCES Ward(Ward\_Id), CONSTRAINT fk\_WardNurse\_Staff FOREIGN KEY (Staff\_Id) REFERENCES Staff(Staff\_Id));

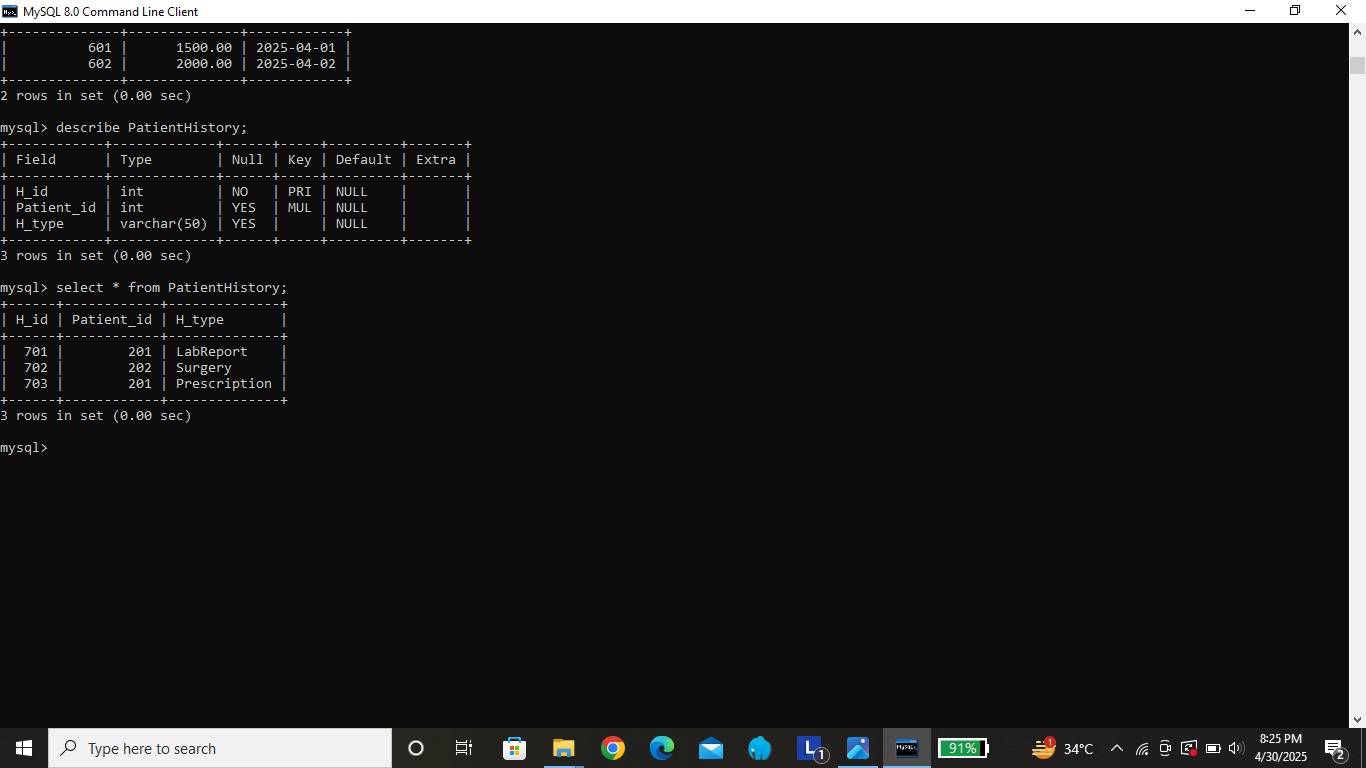


* CREATE TABLE Receipts ( Receipts\_Id INT PRIMARY KEY, Staff\_Id INT, Patient\_Id INT, Insurance\_Id INT, amount DECIMAL(10, 2), CONSTRAINT fk\_Receipts\_Staff FOREIGN KEY (Staff\_Id) REFERENCES Staff(Staff\_Id), CONSTRAINT fk\_Receipts\_Patient FOREIGN KEY (Patient\_Id) REFERENCES Patient(Patient\_Id));
* CREATE TABLE InsuranceClaim ( Insurance\_Id INT PRIMARY KEY, total\_amount DECIMAL(10, 2),claim\_date DATE);
* INSERT INTO Receipts VALUES (401, 1, 201, 601, 1500.00);
* INSERT INTO Receipts VALUES (402, 2, 202, 602, 2000.00);
* INSERT INTO InsuranceClaim VALUES (601, 1500.00, '2025-04-01');
* INSERT INTO InsuranceClaim VALUES (602, 2000.00, '2025-04-02');





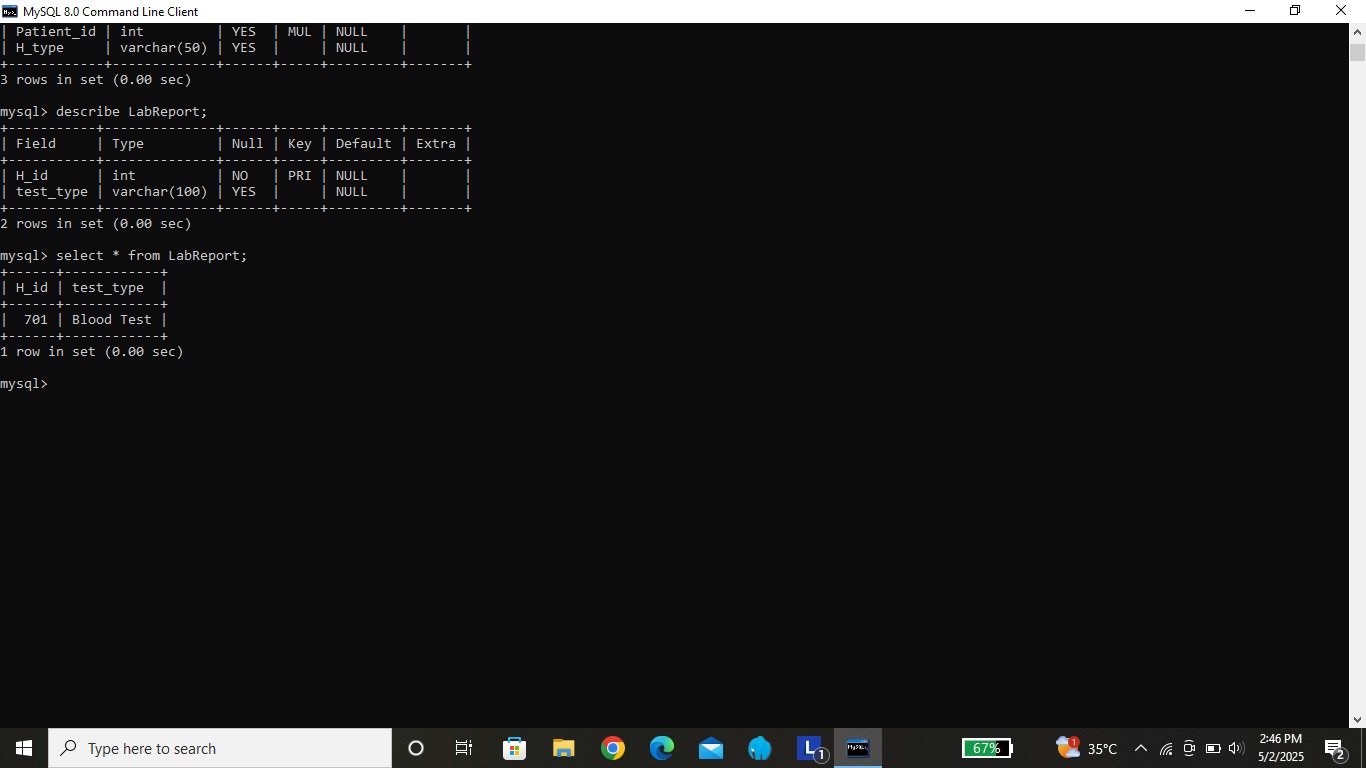
* CREATE TABLE PatientHistory (H\_Id INT PRIMARY KEY, Patient\_Id INT, H\_type VARCHAR(50), CONSTRAINT fk\_PatientHistory\_Patient FOREIGN KEY (Patient\_Id) REFERENCES Patient(Patient\_Id));
* INSERT INTO PatientHistory VALUES (701, 201, 'LabReport');

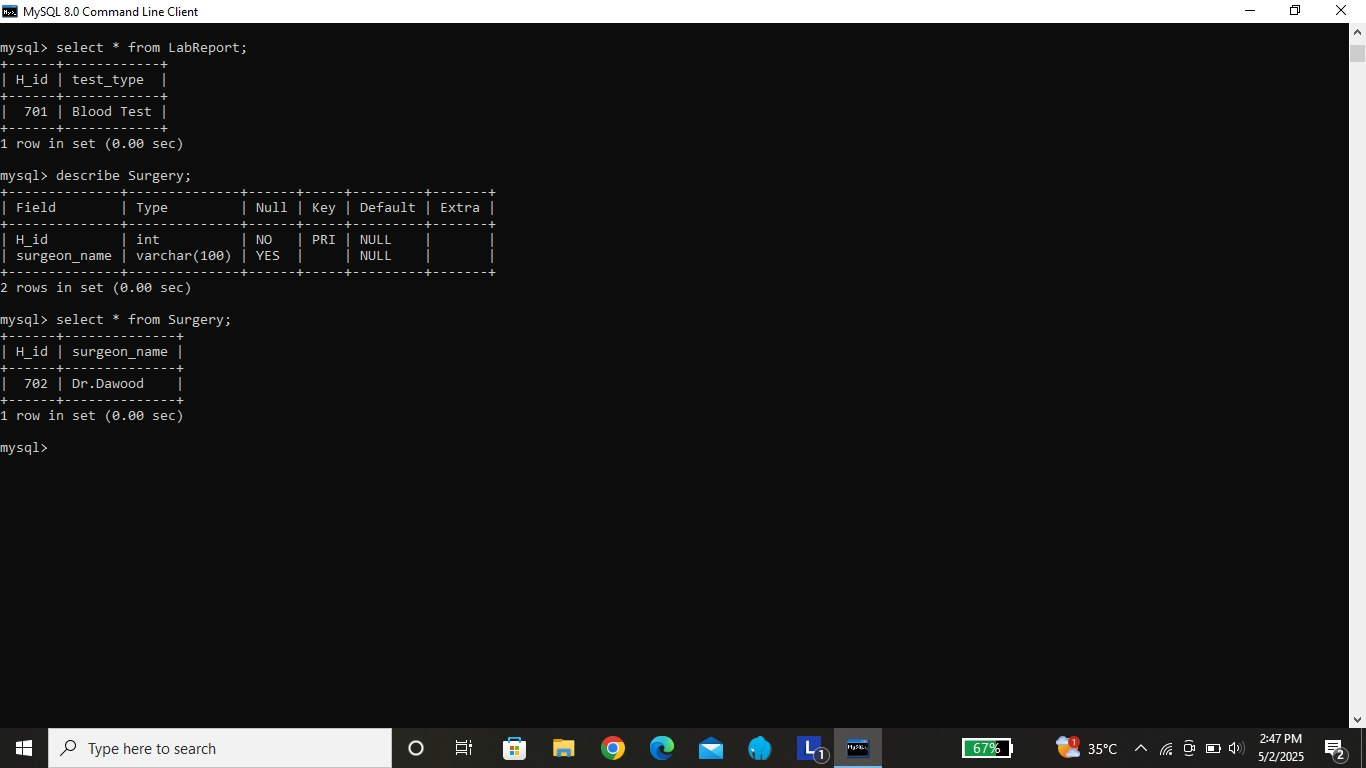


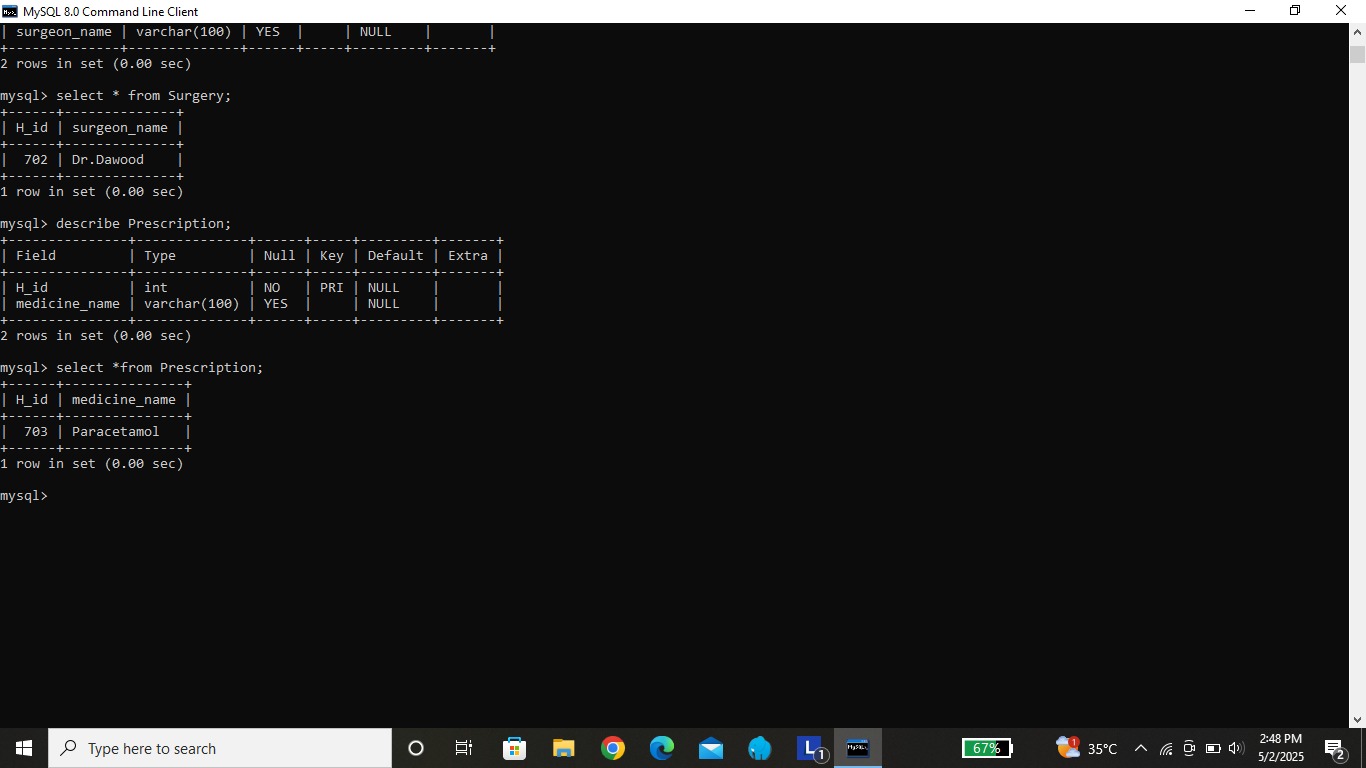
* CREATE TABLE LabReport ( H\_Id INT(3) primary key, test\_type VARCHAR(100), CONSTRAINT fk\_PatientHistory

FOREIGN KEY (H\_Id) REFERENCES PatientHistory(H\_Id));

* CREATE TABLE Surgery ( H\_Id INT(3) primary key, surgeon\_name VARCHAR(100), CONSTRAINT fk\_Surgery\_PatientHistory FOREIGN KEY (H\_Id) REFERENCES PatientHistory(H\_Id));
* CREATE TABLE Prescription ( H\_Id INT(3) primary key, medicine\_name VARCHAR(100),CONSTRAINT fk\_Prescription\_PatientHistory FOREIGN KEY (H\_Id) REFERENCES PatientHistory(H\_Id));







Some queries with implementation:

* Show tables;
* update Staff set salary= salary+500 where staff\_id=1;
* alter table Staff add contact\_num varchar(15);
* alter table Staff drop column contact\_num;
* DELETE FROM nurse WHERE staff\_id = 4;
* DELETE FROM staff WHERE staff\_id = 4;
* RENAME TABLE hospital\_ward TO ward;
* SELECT H\_Id, H\_type FROM patient\_history;
* SELECT test\_type FROM lab\_report;
* SELECT name AS staff\_name FROM staff;

