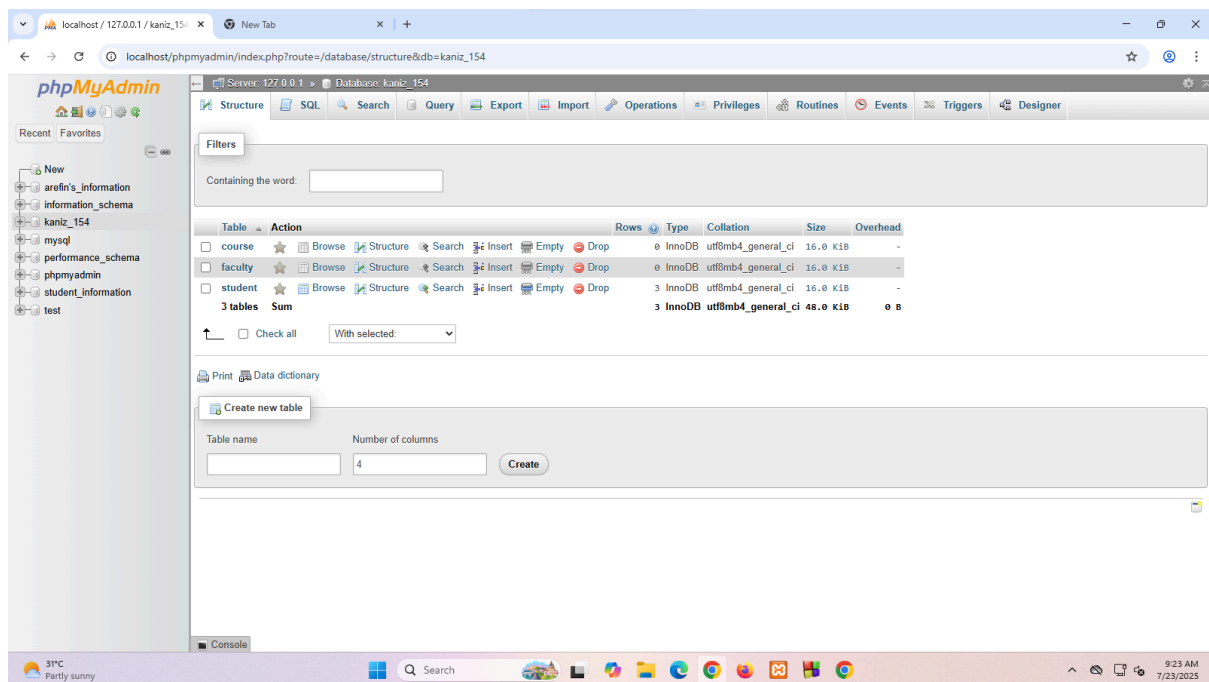
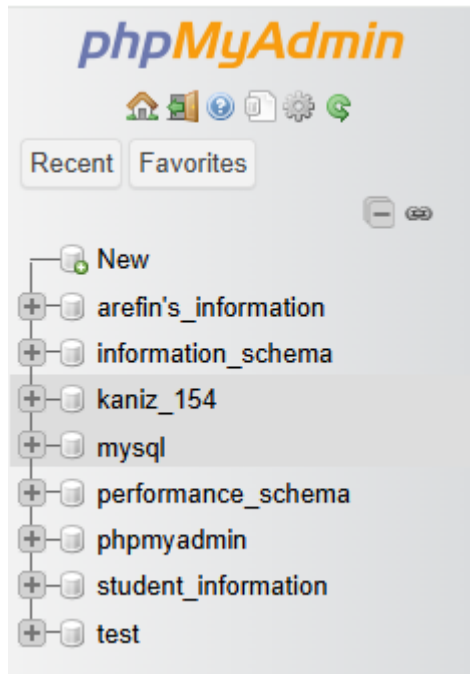


Lab1_23 July 2025

Q1. Create a DB Name “kaniz_154”

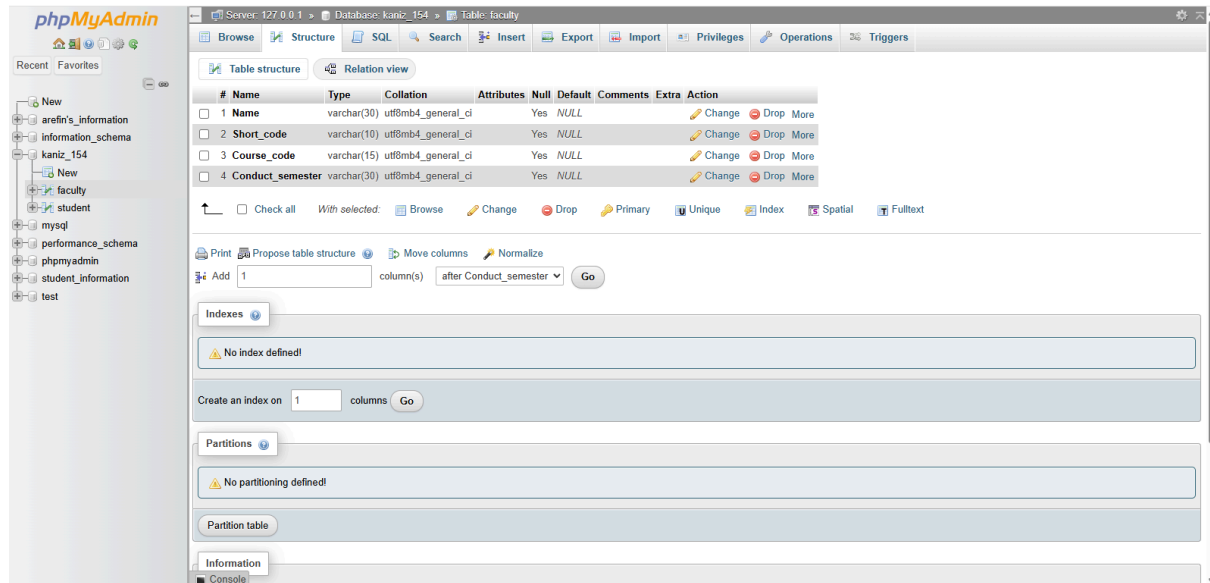
CREATE DATABASE kaniz_154



Q2. Create 3 tables in this DB named "Student", "Faculty",
"Course"

Faculty

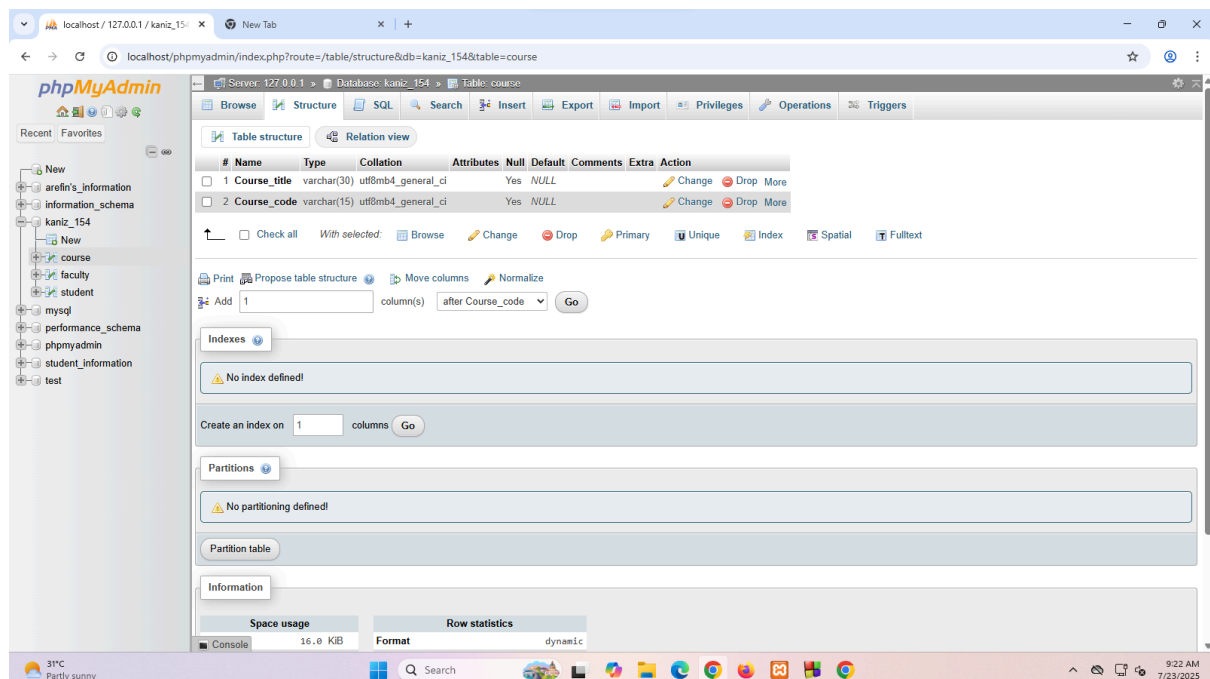
```
CREATE TABLE Faculty(Name varchar(30), Short_code varchar(10), Course_code  
varchar(15), Conduct_semester varchar(30))
```



The screenshot shows the phpMyAdmin interface for a database named 'kaniz_154'. The 'Table: faculty' tab is selected, displaying the 'Table structure' view. The table has four columns: 'Name' (varchar(30)), 'Short_code' (varchar(10)), 'Course_code' (varchar(15)), and 'Conduct_semester' (varchar(30)). All columns are of type 'utf8mb4_general_ci' and have 'Yes' for 'Attributes' and 'NULL' for 'Default'. The 'Extra' column is empty for all. The 'Action' column contains links for 'Change', 'Drop', and 'More' for each column. Below the table structure, there are sections for 'Indexes' (No index defined), 'Partitions' (No partitioning defined), and 'Information'.

Course

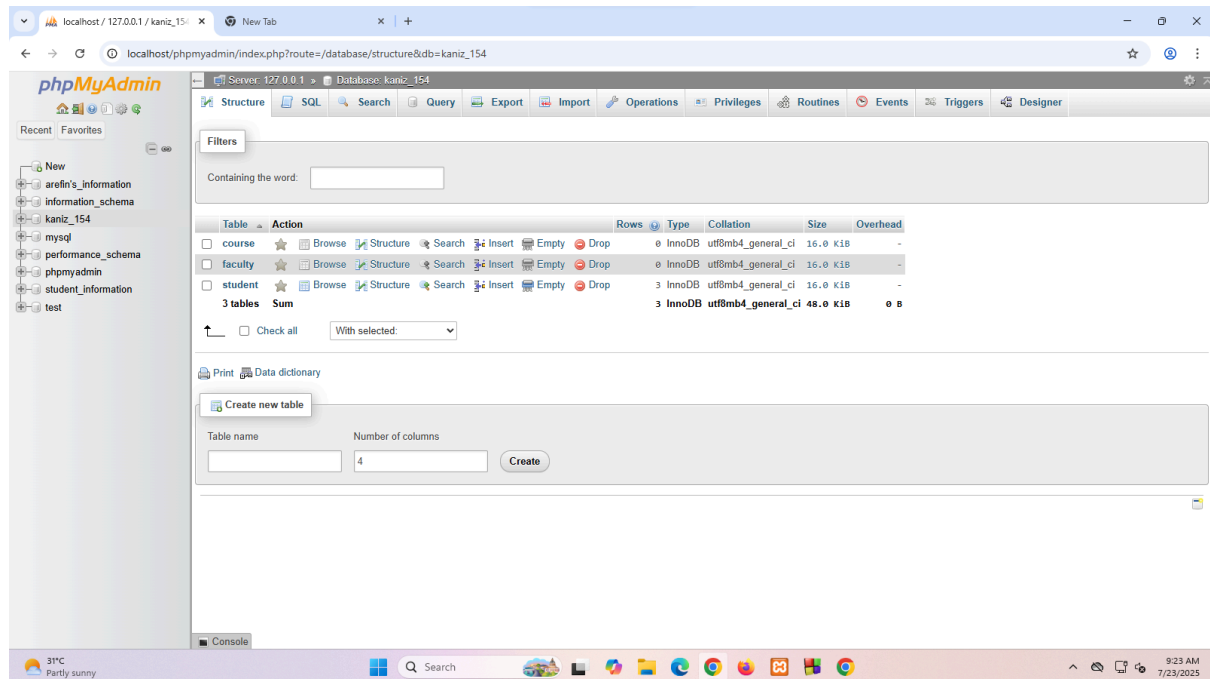
```
CREATE TABLE Course(Course_title varchar(30), Course_code varchar(15))
```



The screenshot shows the phpMyAdmin interface for a database named 'kaniz_154'. The 'Table: course' tab is selected, displaying the 'Table structure' view. The table has two columns: 'Course_title' (varchar(30)) and 'Course_code' (varchar(15)). Both columns are of type 'utf8mb4_general_ci' and have 'Yes' for 'Attributes' and 'NULL' for 'Default'. The 'Extra' column is empty for both. The 'Action' column contains links for 'Change', 'Drop', and 'More' for each column. Below the table structure, there are sections for 'Indexes' (No index defined), 'Partitions' (No partitioning defined), and 'Information'. The 'Space usage' section shows 16.0 KB and the 'Row statistics' section shows 'Format' as 'dynamic'.

Student

CREATE TABLE student(Name varchar(30), student_ID int(7), NID int(3), Intake int(2), section int(2), CGPA float(3.2))



Q2.1 Insert values on “Student”, “Course”, “Faculty”

Faculty

```
INSERT INTO `faculty`(`Name`, `Short_code`, `Course_code`, `Conduct_semester`)  
VALUES ('Meena','MN','CSE_111','Spring_2025')
```

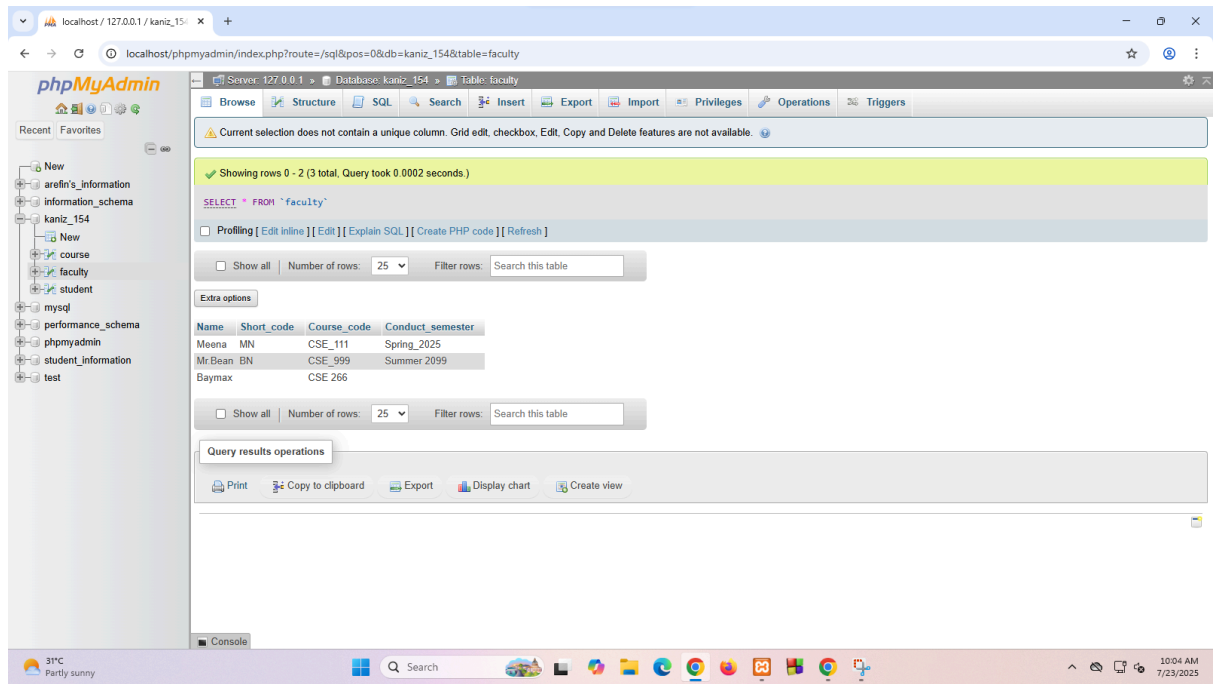
Insert Null values

```
INSERT INTO `faculty`(`Name`, `Short_code`, `Course_code`, `Conduct_semester`)  
VALUES ('Spiderman','SM','','')
```

Or,

```
INSERT into faculty values("Mr.Bean", "BN","CSE_999","Summer 2099")
```

```
INSERT into faculty values("Baymax", " ","CSE_111","Summer 2030")
```

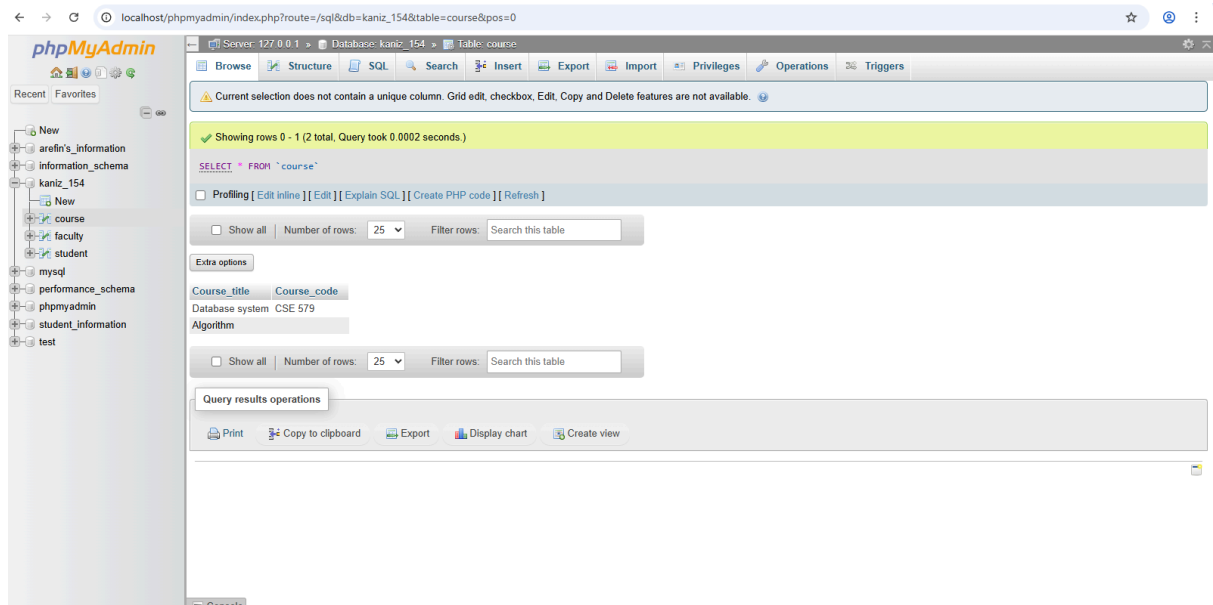


Course

```
INSERT INTO `course`(`Course_title`, `Course_code`) VALUES ('Database system','CSE 579')
```

Insert Null Value

```
INSERT INTO `course`(`Course_title`, `Course_code`) VALUES ('Algorithm','')
```



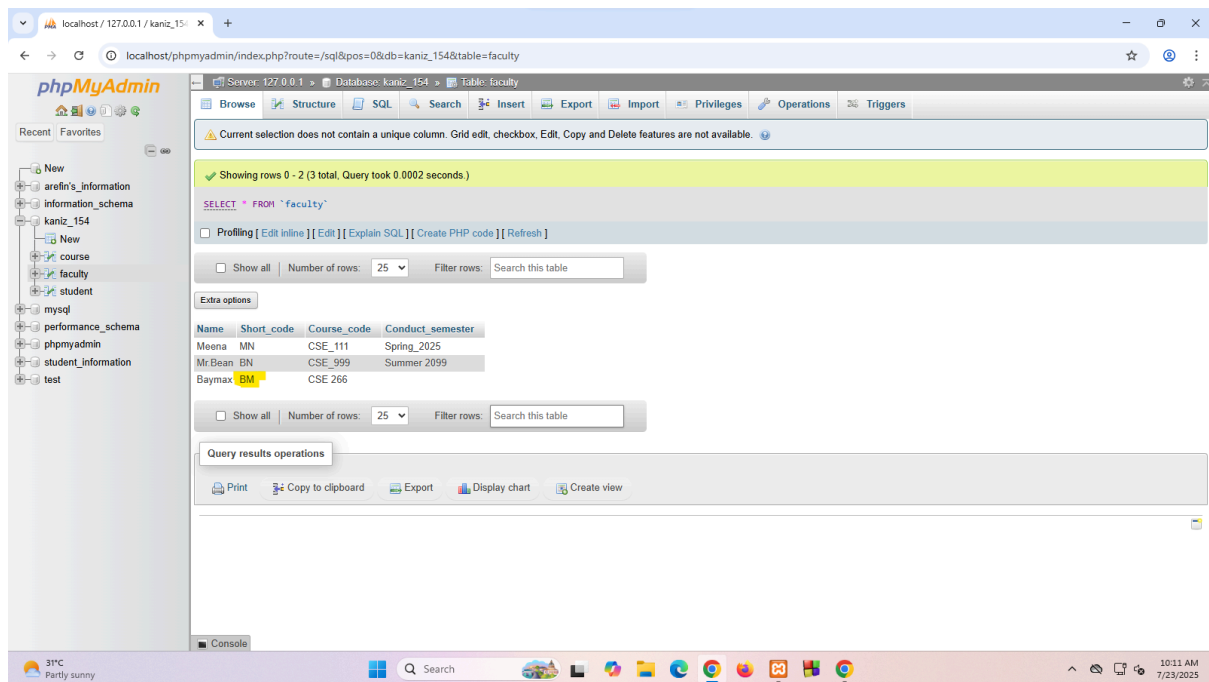
Student

```
// CREATE TABLE student(Name varchar(30), student_ID int(7), NID int(3), Intake int(2), section int(2), CGPA float(3.2))
```

INSERT into student values("Kaniz", 154,001,53, 1, 5.9)
INSERT into student values("Fatema", 023,001,53, 1, 4.9)

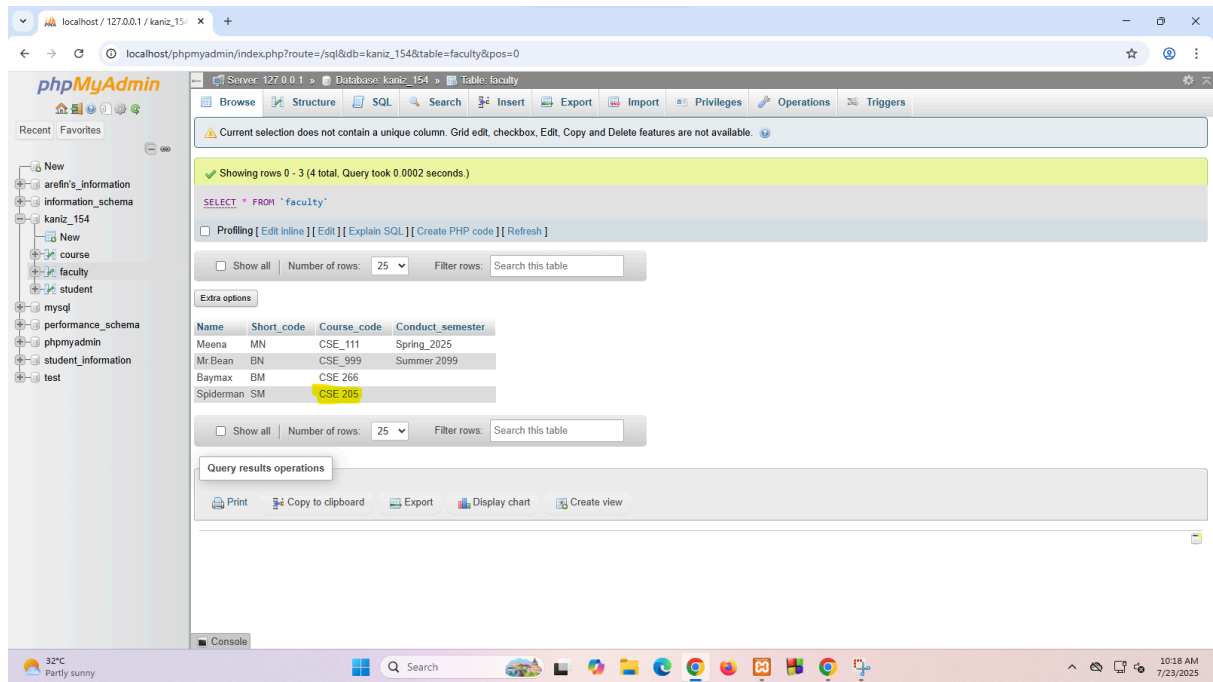
Q3. a) Update the short code whose short code is NULL.

UPDATE faculty SET Short_code = "BM" WHERE Name= "Baymax"



Q3. b) Update the course code that is NULL.

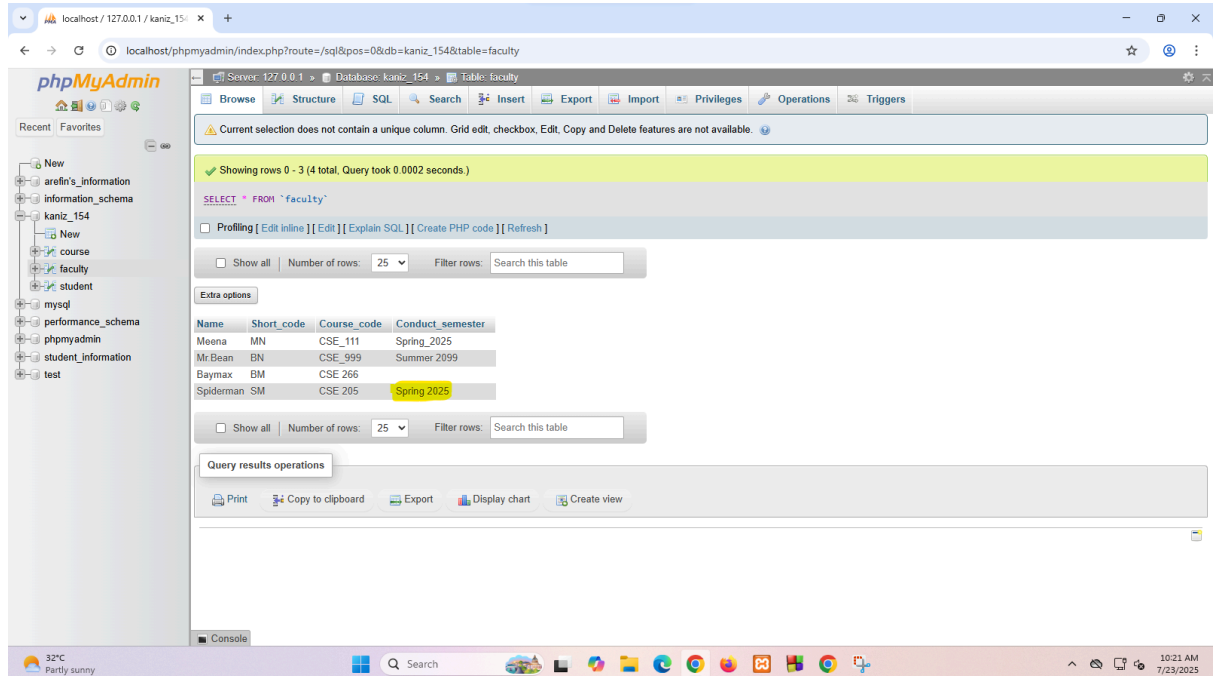
UPDATE faculty SET Course_code ="CSE 205" WHERE Name= "Spiderman"



Q3. c) Update the conduct semester that is NULL.

UPDATE faculty SET Conduct_semester = "Spring 2025" WHERE Name = "Spiderman"

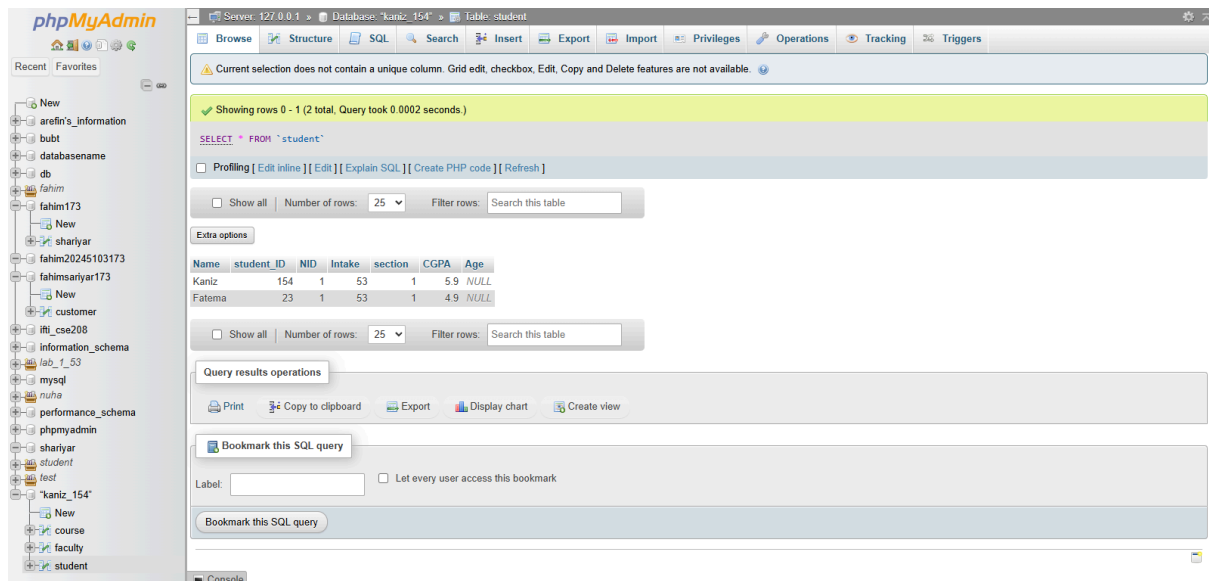
UPDATE faculty SET Conduct_semester = "Spring 2025" WHERE Name = "Meena"



Lab2_30 July 2025

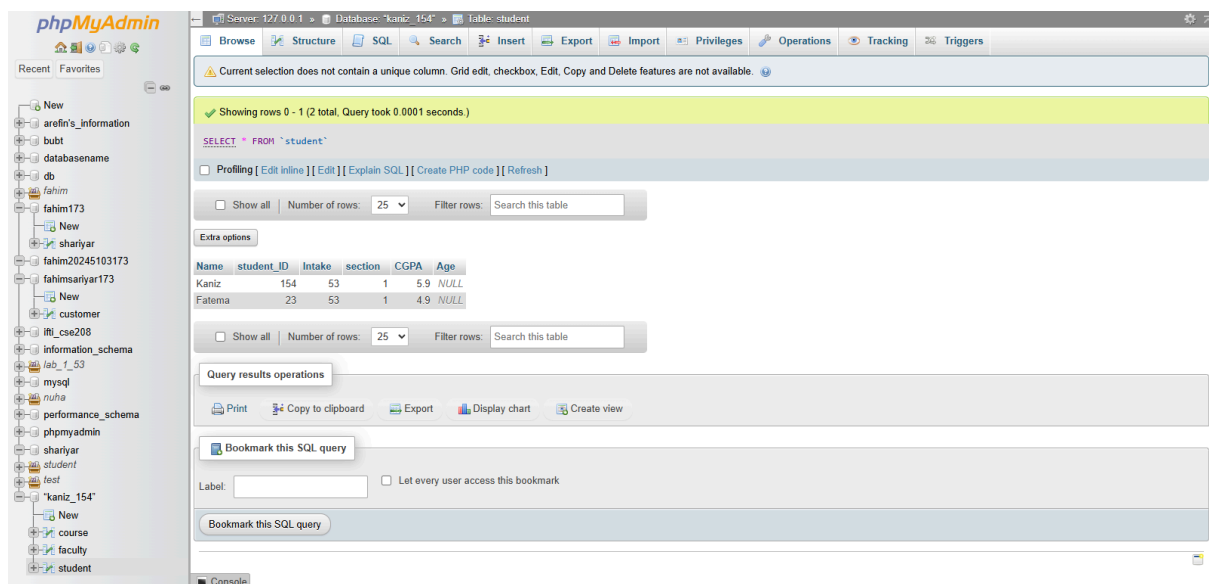
Q3. d) Add a new column named "Age" in student relation.

Alter table Student ADD Column Age int(2)



Q3. d.1) Delete column named “NID” from Student database

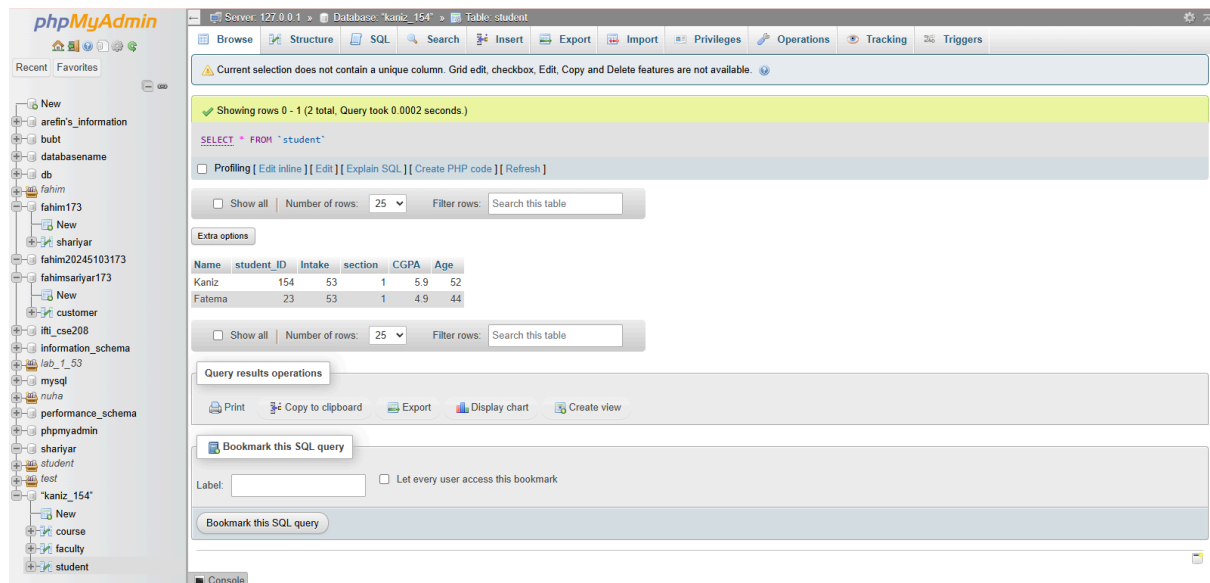
Alter table Student DROP Column NID



Q3. e) Update the age of each student.

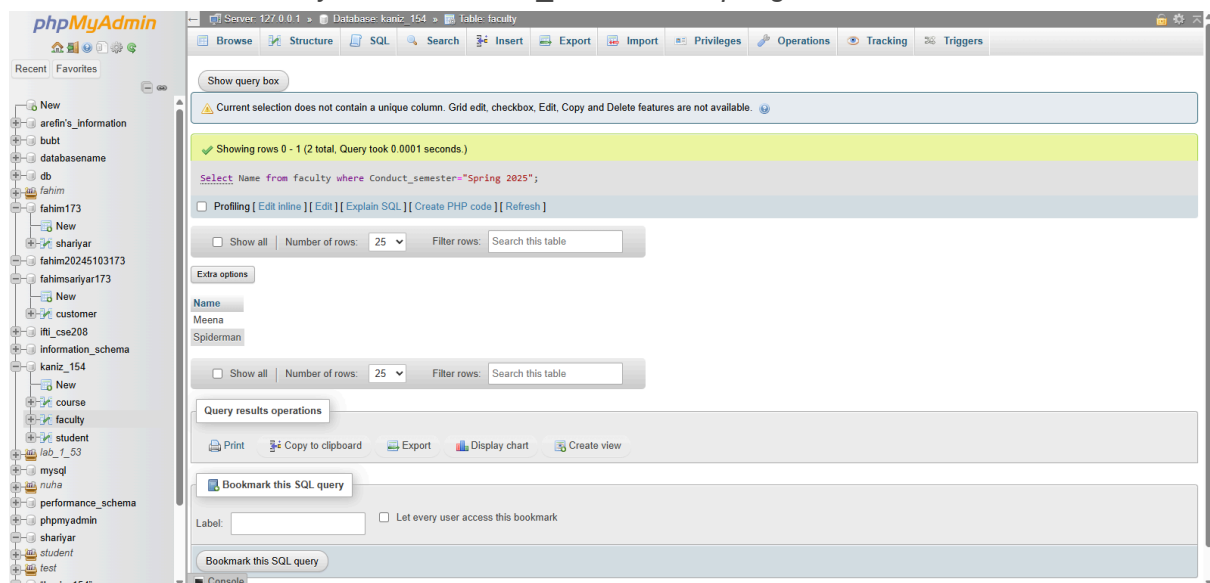
update Student set Age=52 where student_ID=154

update Student set Age=44 where student_ID=23



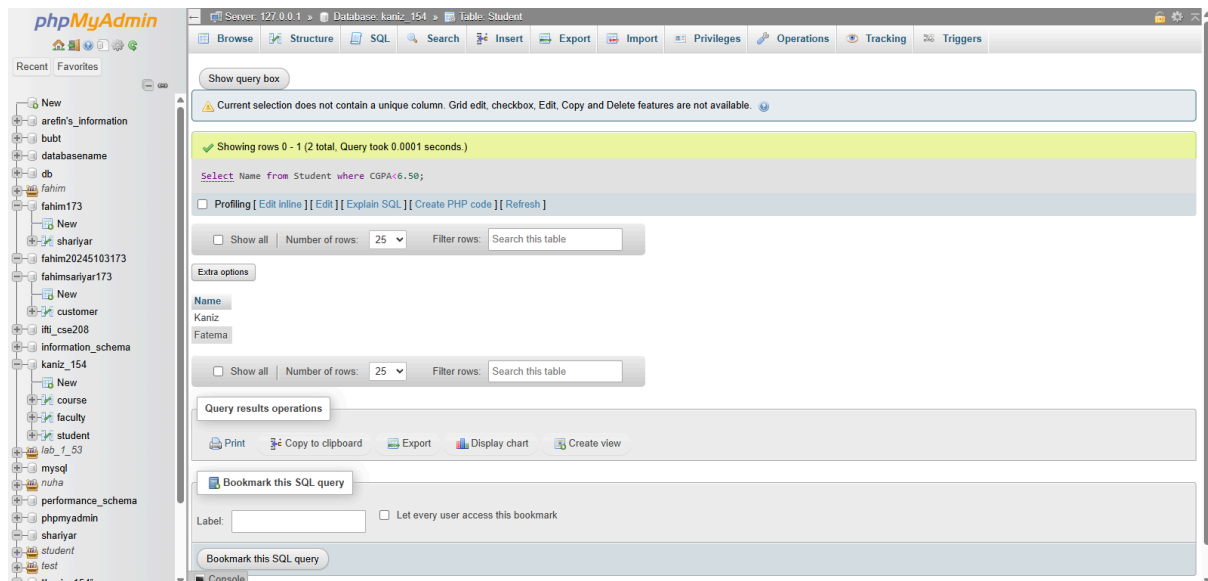
Q3. g) Find the names of all teachers who has conducted in Spring 2025.

Select Name from faculty where Conduct_semester="Spring 2025"



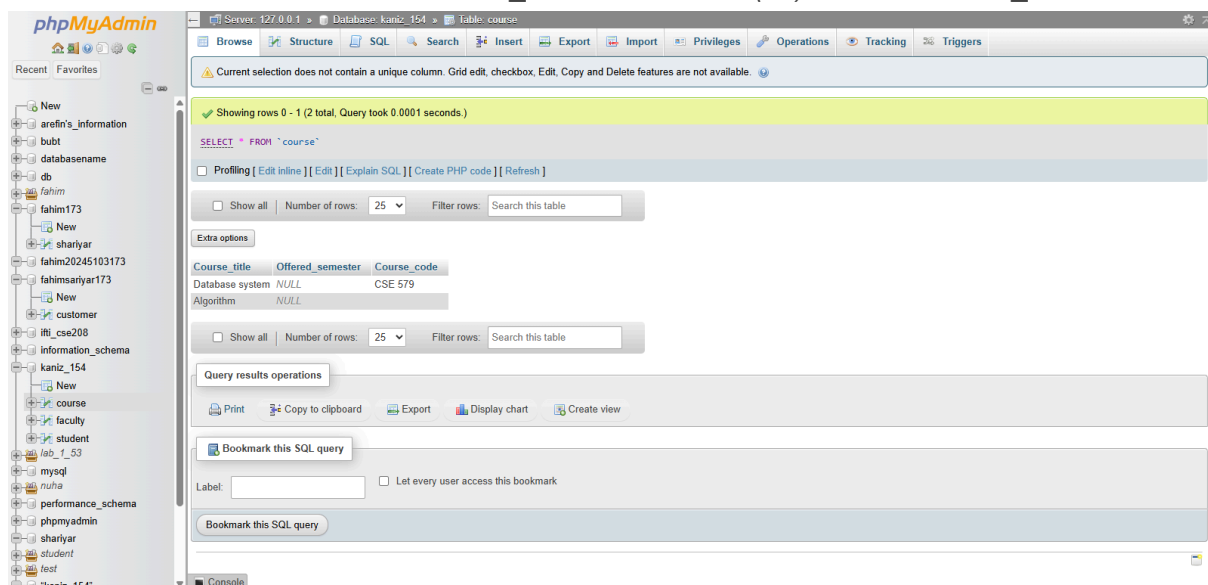
Q3. h) Find the names of all students whose CGPA is greater than 6.50.

Select Name from Student where CGPA<6.50



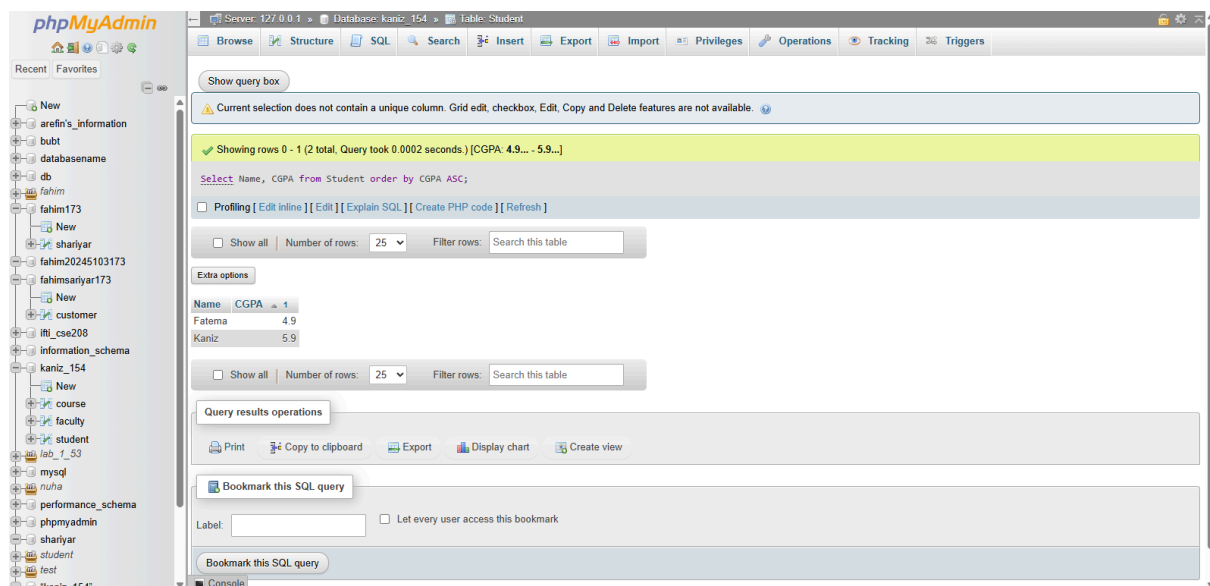
Q3. i) Add a new column “Offered semester” after the column “Course Title”.

Alter table course ADD column Offered_semester varchar(20) AFTER Course_title



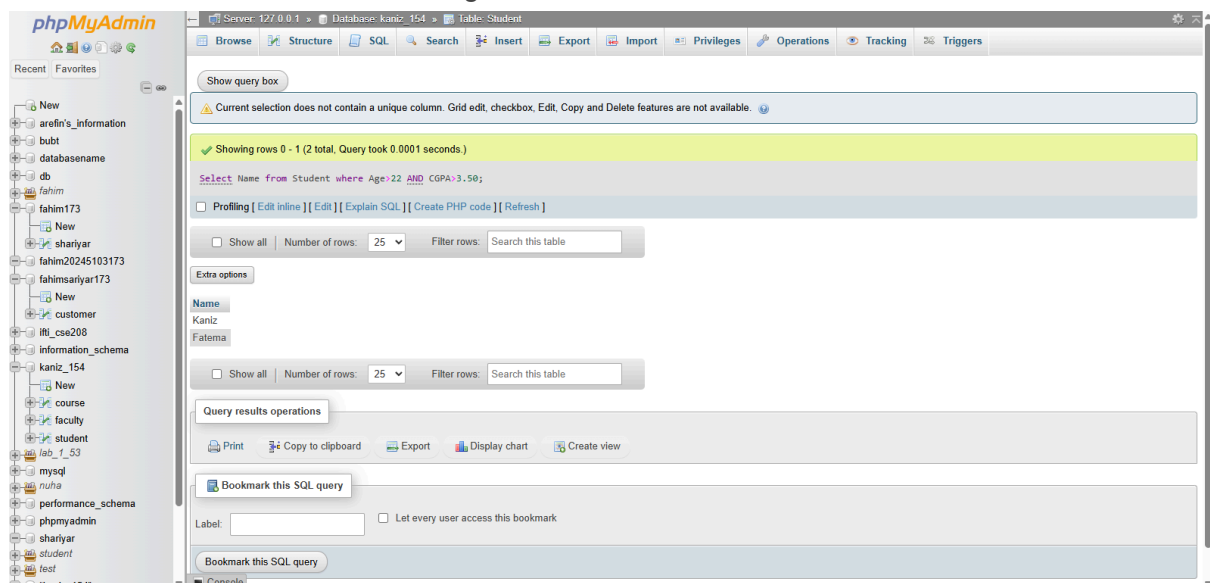
Q3. j) Find the name and CGPA of each student order by their CGPA (Ascending order.).

Select Name, CGPA from Student order by CGPA ASC



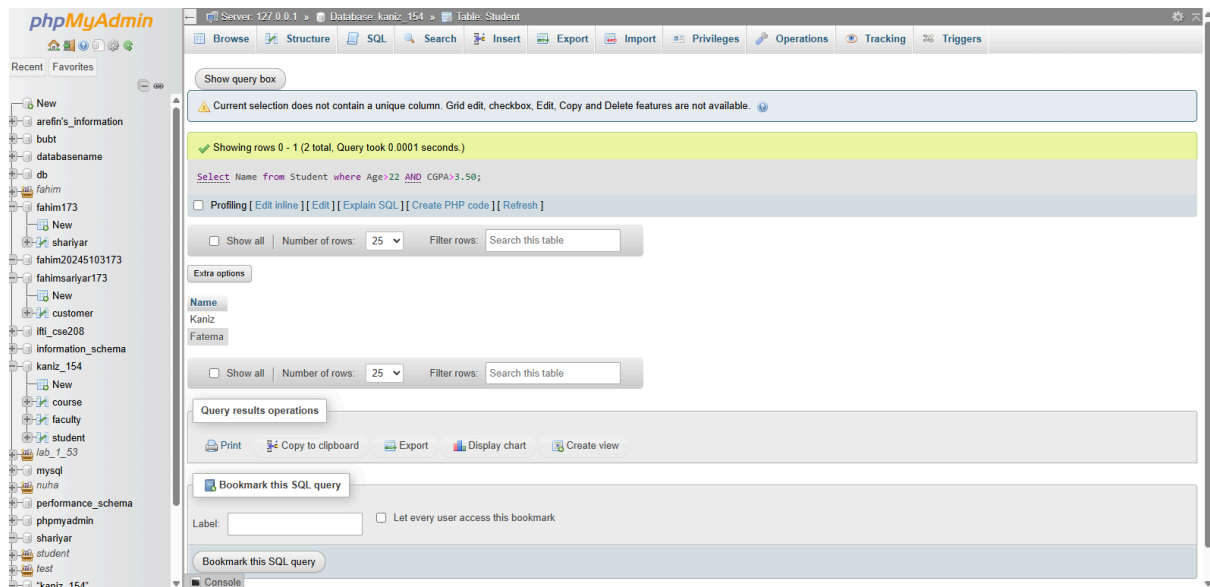
Q3. k) Find the name of all students whose age is greater than 22 and CGPA is greater than 3.50.

Select Name from Student where Age>22 AND CGPA>3.50



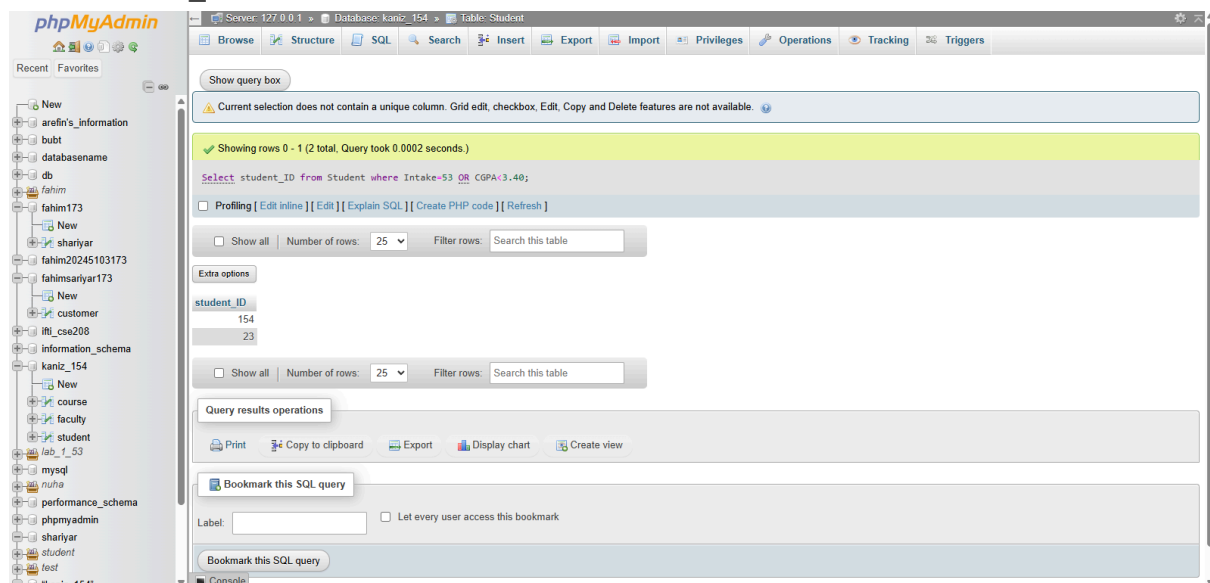
Q3. k) Find the name of all students whose age is greater than 22 and CGPA is greater than 3.50.

Select Name from Student where Age>22 AND CGPA>3.50



Q3. l) Find the Id of all those students who are from 53 intake or CGPA is less than 3.40.

Select student_ID from Student where Intake=53 OR CGPA<3.40



Q3. m) Find the Id of all those students whose CGPA is 5.9 or 4.9 or 3.96.

Select student_ID from student where CGPA In(5.9, 4.9, 3.96)

phpMyAdmin

Recent Favorites

New

arefin's_information

bubt

database_name

db

fahim

fahim173

New

shariyar

fahim20245103173

fahimsariyar173

New

customer

ifit_cse208

information_schema

kaniz_154

New

course

faculty

student

lab_1_53

mysql

nutha

performance_schema

phpmyadmin

shariyar

student

test

Server: 127.0.0.1 » Database: kaniz_154 » Table: student

Browse

Structure

SQL

Search

Insert

Export

Import

Privileges

Operations

Tracking

Triggers

Show query box

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0001 seconds)

Select student_ID from student where CGPA In(5.9, 4.9, 3.96);

Profiling

Edit inline

Edit

Explain SQL

Create PHP code

Refresh

student_ID

Query results operations

Create view

Bookmark this SQL query

Label: ☐ Let every user access this bookmark

Bookmark this SQL query

Console