Green University of Bangladesh

Department of Computer Science and Engineering(CSE)

Faculty of Sciences and Engineering

Semester: (Spring, Year:2024), B.Sc. in CSE (Day)

LAB REPORT NO #02

Course Title: Database Lab

Course Code: CSE 210 Section: 221_D9

Experiment Name: Implementation of Integrity Constraints in MySQL

Student Details

	Name	ID
1.	Jahidul Islam	221002504

Lab Date : 02 – 03 – 2024

Submission Date : 09 – 03 – 2024

Course Teacher's Name : Md. Nazmus Shakib

[For Teachers use only: Don't Write Anything inside this box]

Lab Report Status

Marks: …… **Signature:.....**

Comments:..... **Date:.....**

1. TITLE OF THE LAB EXPERIMENT

Implementation of Integrity Constraints in MySQL

2. OBJECTIVES

After complementing this lab experiment, we will gain practical knowledge and tthe outcomes of this experiment are

1. Database creation and Insert Data in each table using sql code.
2. Declare Primary Key
3. Assign primary key for each table.
4. Create Composite Key
5. Assign a unique in at least two tables.
6. Implement Unique Constraint
7. Implement Foreign Key Constraint
8. Browse data for each table.

3. PROCEDURE

To successfully complete the outcomes we have done the followings.

1. **Database Desgin: first of all, I need to come up with a idea of three table where I can implement all the outcomes like declaring primary and foregin key, and it's constraint I needed to think of a relation between three tables so that we can build a meaningful database.**
2. **Table Cration: courselist , advisor_list, students three tables.**
    a. **courselist has one primary key: course_code**
    b. **advisor list has 2 primary key**

Field	Type	Null	Key	Default	Extra
faculty_name	varchar(15)	NO	PRI	*NULL*	
faculty_id	int(11)	NO	PRI	*NULL*	
email	varchar(15)	NO		*NULL*	

Firgur 1: DESCRIBE advisor_list;

    c. **students table has 1 primary key, 1 MUL key and 1 unique key which is also foregin key.**

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	*NULL*	
NAME	varchar(15)	NO		*NULL*	
department	varchar(5)	NO		*NULL*	
course_code	int(11)	NO	UNI	*NULL*	
faculty_name	varchar(15)	NO	MUL	*NULL*	
phone	int(11)	YES		*NULL*	

Firgur 2: DESCRIBE students;

4. IMPLEMENTATION

Here's I have included all the code we need to obtain all the outcomes of this experiment.

Codes:

CREATE DATABASE exp3_lab_rpt2 ;
use exp3_lab_rpt2;

1. Table: courselist ,

a. Table creation:

CREATE TABLE courselist(course_code int NOT NULL, course_name varchar(30) NOT NULL, batch int NOT NULL, PRIMARY KEY(course_code), UNIQUE(course_code));

b. INSERTion in the table.

INSERT INTO courselist (course_code, course_name, batch) VALUES
(307, 'Data Communication', 221),
(308, 'Data Communication Lab', 221),
(209, 'Database System', 221),
(210, 'Database System Lab', 221),
(205, 'Electrical Drives and Instrumentation', 221),
(309, 'Operating System', 221),
(310, 'Operating System lab', 221),
(313, 'Software Engineering', 221);

2. Table : students

c. Table creation:

CREATE TABLE students(

id INT NOT NULL,
NAME VARCHAR(15) NOT NULL,
department VARCHAR(5) NOT NULL,
course_code INT NOT NULL,
faculty_name varchar(15) NOT NULL,
phone INT,
PRIMARY KEY(id),
UNIQUE(course_code),
FOREIGN KEY(faculty_name) REFERENCES advisor_list(faculty_name),
FOREIGN KEY(course_code) REFERENCES courselist(course_code)
);

d. INSERT in the table.

To insert data in this table we need to first input data in the courselist and advisor_list as students table has a foregin key relationship with these two table.

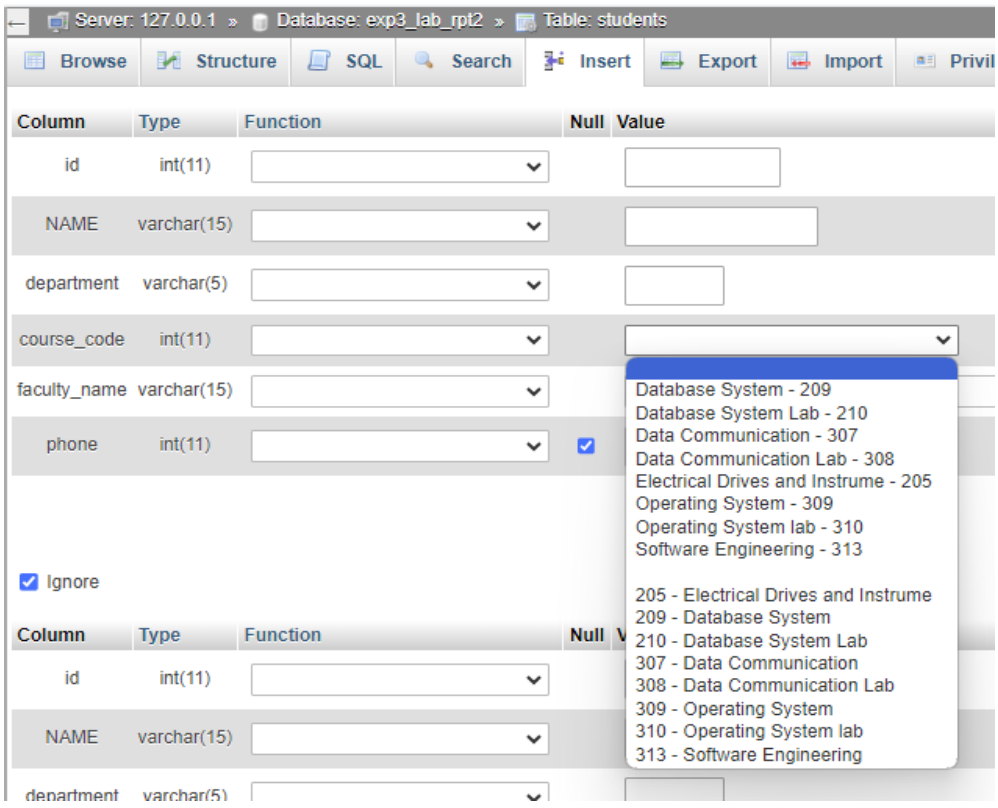

Figure 3: student table has a foregin key reference from courelist table (column:course_code).

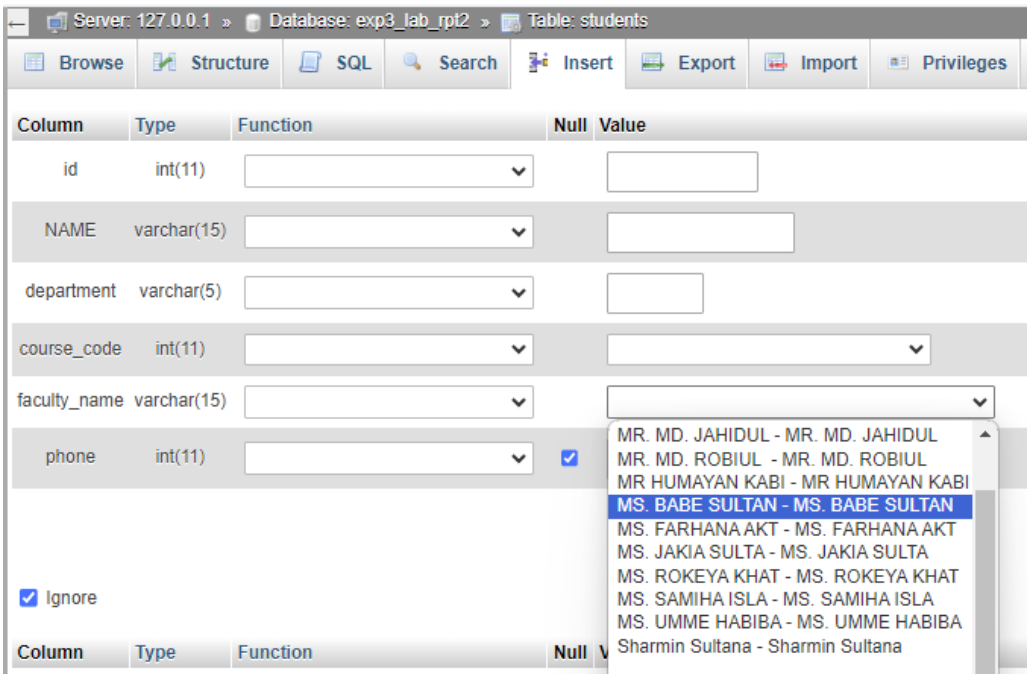

Figure 4: student table has a foregin key reference from advisor_list table(faculty_name).

3. Table: advisor_list:

e. Table creation:

CREATE TABLE advisor_list(faculty_name varchar(15) NOT NULL, faculty_id int NOT NULL, email varchar(15) NOT NULL, PRIMARY KEY(faculty_name(15), faculty_id));

f. INSERT in the table.

INSERT INTO advisor_list (faculty_name, faculty_id, email) VALUES
('MR. MD. JAHIDUL ISLAM', 100, 'jahid@cse.green.edu.bd'),
('MS. JAKIA SULTANA', 101, 'jakia@cse.green.edu.bd'),
('MS. FARHANA AKTER SUNNY', 102, 'farhana@cse.green.edu.bd'),
('Sharmin Sultana, 103, 'sharmin@cse.green.edu.bd'),
('MR. HUMAYAN KABIR RANA', 104, 'humayan@cse.green.edu.bd'),
('Monirul, 105, 'monirul@cse.green.edu.bd'),
('MR. JARGIS AHMED', 106, 'jargis@cse.green.edu.bd'),
('MS. SAMIHA ISLAM TANNI', 107, 'samiha@cse.green.edu.bd'),
('MR. MD. ROBIUL ISLAM', 109, 'robiul@cse.green.edu.bd'),
('MS. BABE SULTANA', 111, 'babe@cse.green.edu.bd'),
('MS. UMME HABIBA', 113, 'umme@cse.green.edu.bd'),
('MS. ROKEYA KHATUN', 115, 'rokeya@cse.green.edu.bd');

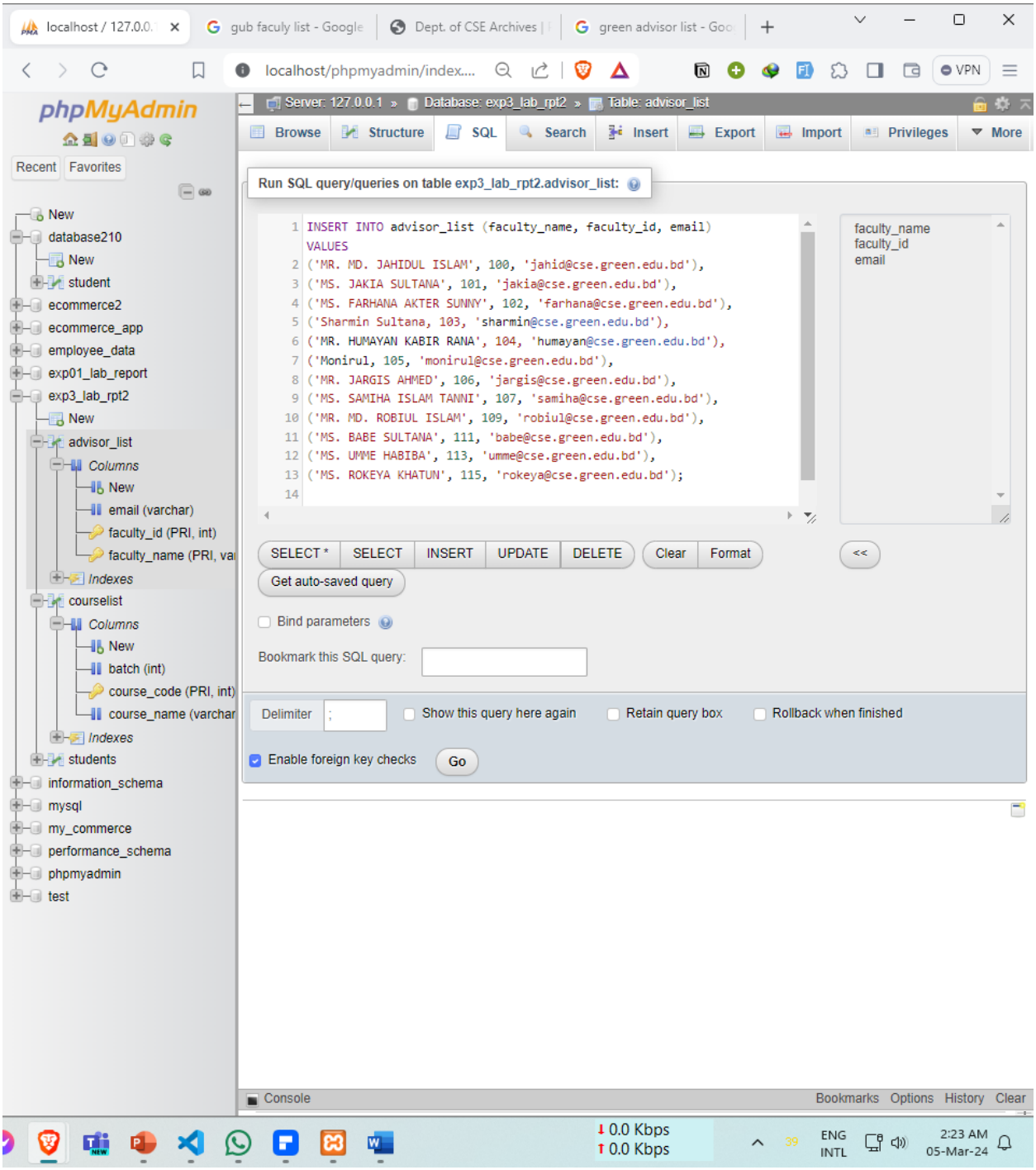

Figure 5: inseting data in advisor_list table.

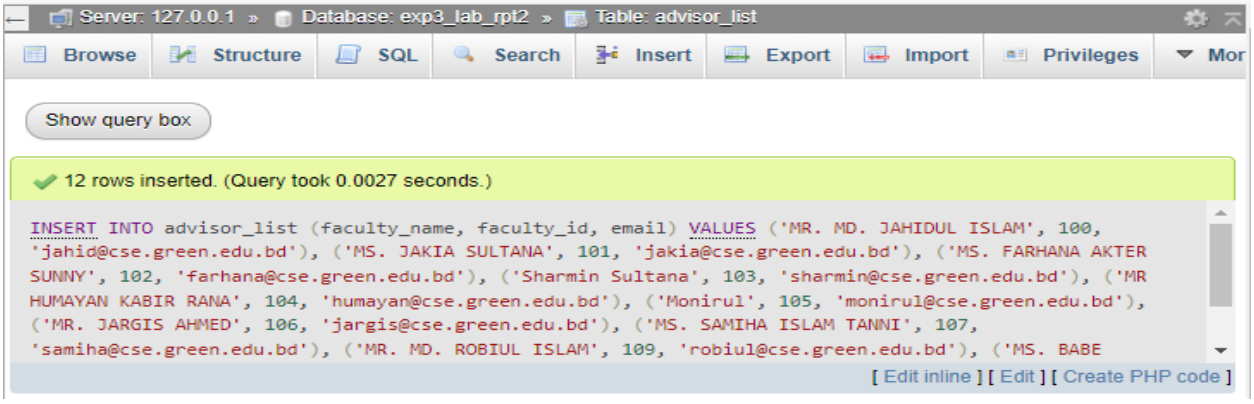

Figure 6: insetion successful in advisor_list table.

5. OUTPUT

We can see the data of our earlier created database uing the following command.

SELECT * FROM

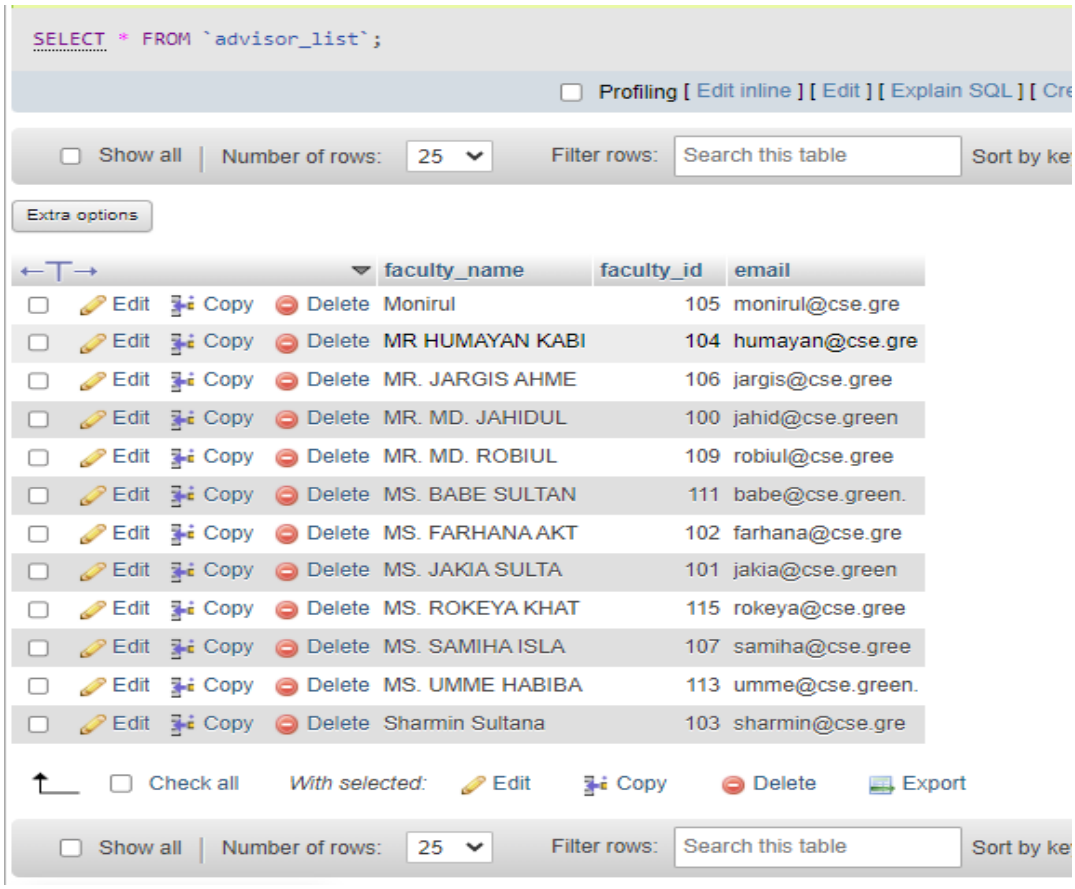

Figure 7: advisor_list table.

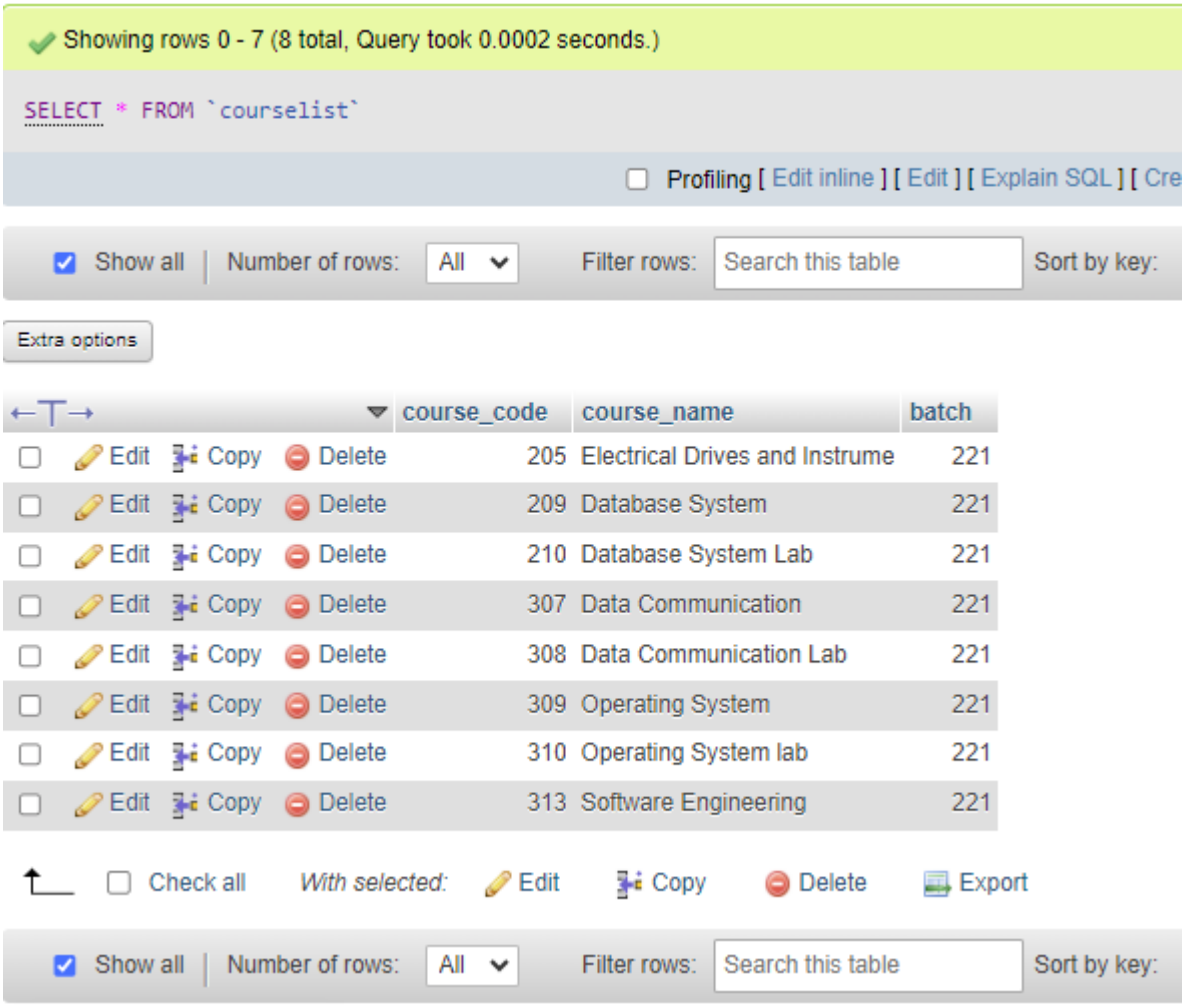

Figure 8: courselist table.

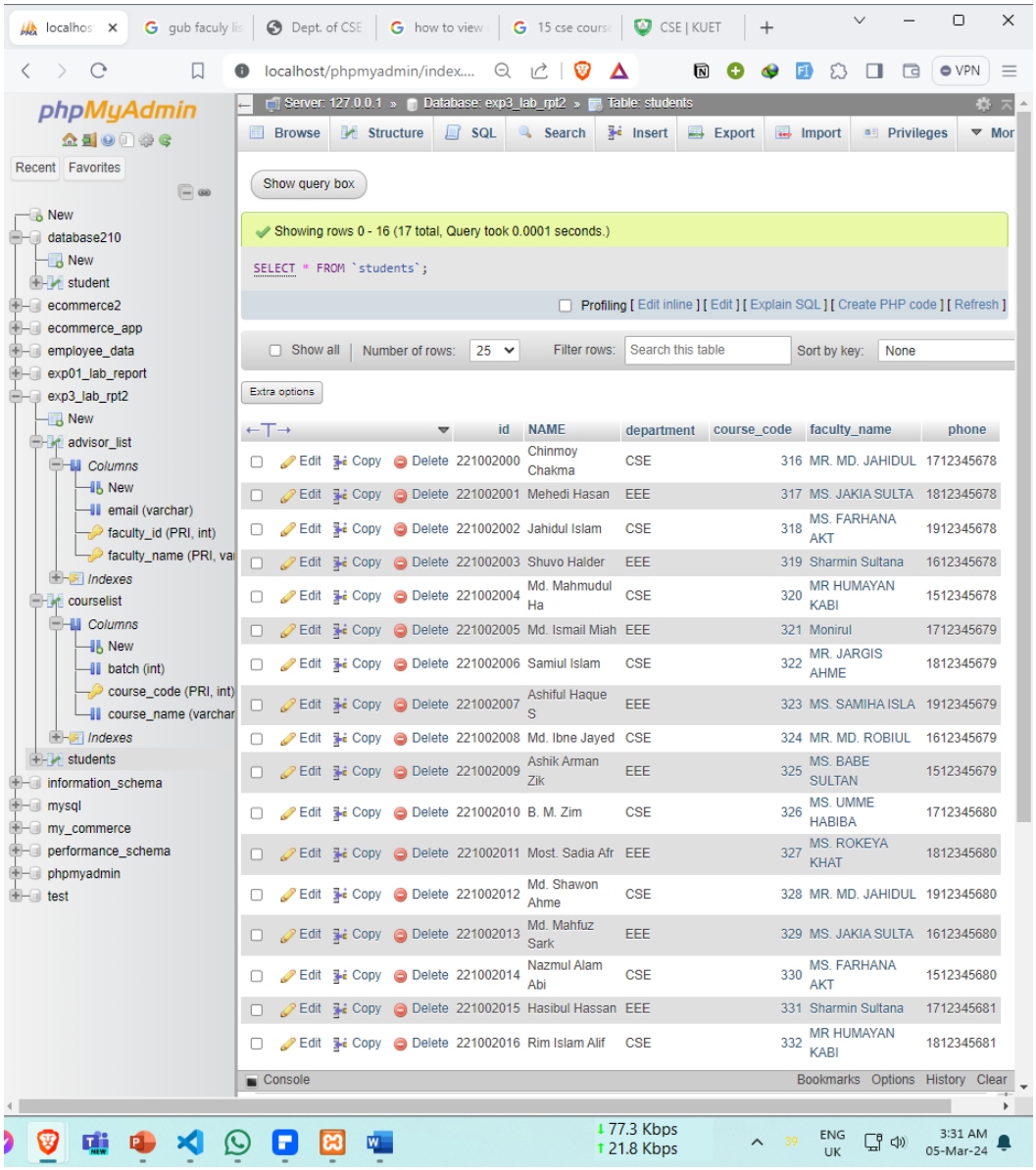

Figure 9: student table. By SELECT * FROM `students`;

6. ANALYSIS AND DISCUSSION:

I reflected on how integrity constraints maintained data accuracy. Explored relationships between tables, noting their role in data retrieval. Despite challenges, I found the implementation efficient, with scope for future enhancements.

7. SUMMARY:

The lab experiment is successfully completed on creating and inseting data in the database with three tables: "courselist," "advisor_list," and "students."
The data insertion has reference integrity by referencing existing course codes and faculty names from the respective tables. Overall, the lab aimed to demonstrate the implementation of integrity constraints and foreign key relationships in MySQL databases.