

AMERICAN INTERNATIONAL UNIVERSITY BANGLADESH
(AIUB)

FACULTY OF SCIENCE & TECHNOLOGY



Course Title
INTRODUCTION TO DATABASE (CSC2108)

Semester: Fall 2024-25

Section: [Y]

TITLE

University Student Management System

Supervised By

Tamanna Zaman Bristy

Submitted By: Group no: 5

Name	ID
Maruf Hossain Chowdhury Durlov	23-54369-3
MD. Zahin Daiyan	23-53680-3
Sadia Afrin	23-54097-3
Arpita Tarafder Arna	23-54080-3

TABLE OF CONTENTS

TOPICS	Page no.
Title Page	1
Table of Content	2
1. Introduction	3
2. Case Study	4
3. ER Diagram	5
4. Normalization	6-10
5. Finalization	11
6. Table Creation	12-14
7. Data Insertion	15-17
8. Query Test	18-25
9. DB connection	26-44

Introduction

This project aims to develop a **The University Student Management System** designed to streamline the management of students, faculty, departments, courses, clubs, and faculty contacts. The system will use relational database tables, which include: student, faculty, course, department, major, and faculty_contact.

Tools Used

- **Database:** Oracle SQL for creating and managing relational tables.
- **Programming Languages:** SQL for querying and managing data in the database.

Vision/Goal

The goal of this project is to build a comprehensive system that can handle various aspects of a student's academic life, from course enrollment to faculty and club memberships. By the end of the project, the system will ensure an efficient, accessible way to track and manage student and faculty data.

Contribution

This project offers a comprehensive **University Student Management System** that can be used by universities to efficiently manage student data, courses, faculty, and extracurricular activities. It simplifies access to academic information for students, allowing them to track courses, and faculty details. For universities, it streamlines data management, improves decision-making, and ensures scalability to accommodate growing student populations. This system provides a centralized platform, benefiting both students and university staff in managing and organizing educational data efficiently.

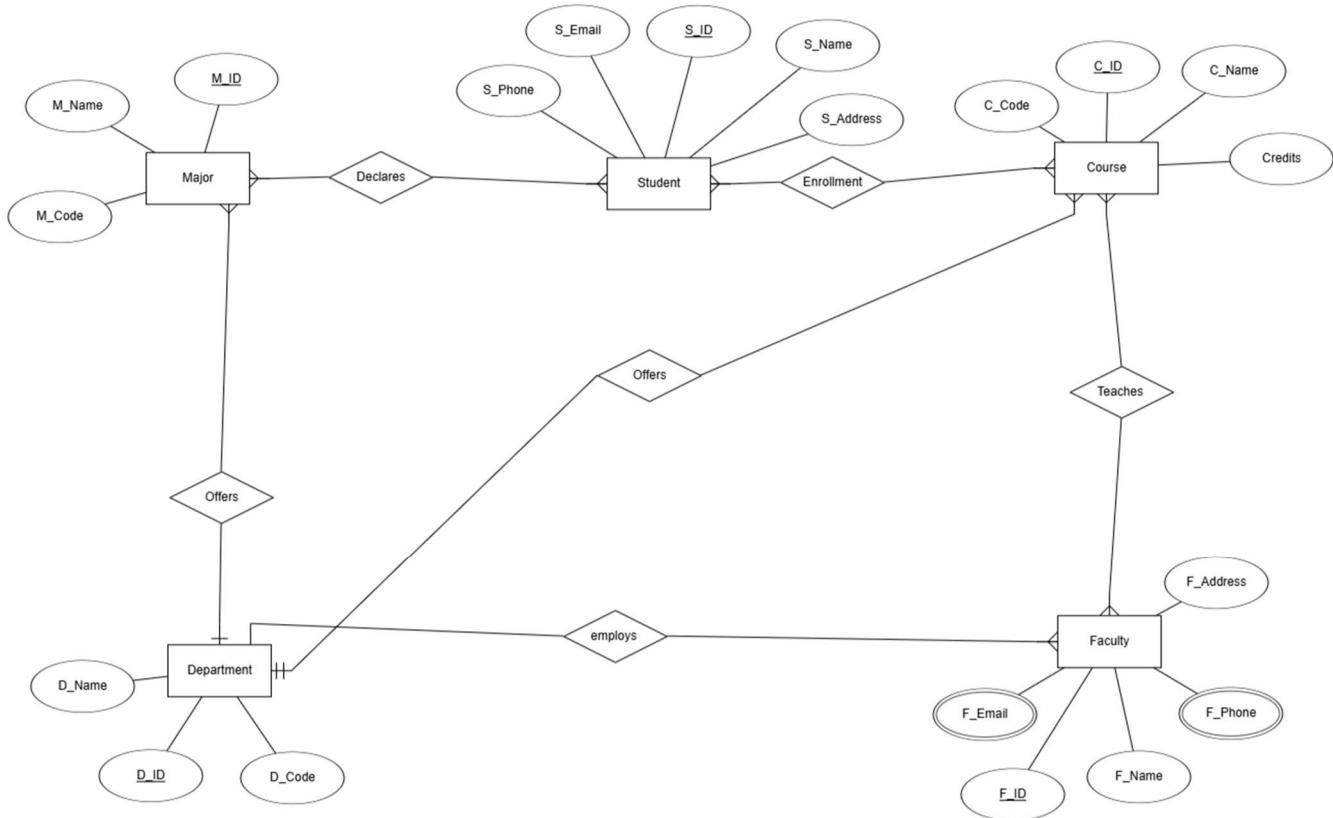
Targeted Users

- **Targeted Users:** Students, faculty members, administrative staff, and academic departments.

Case Study / Scenario

In a university management system, students enroll in various courses as part of their academic journey. Each student is uniquely identified by a student ID and has attributes such as name, email, and address. A student can enroll in multiple courses, and each course can have many students, establishing a many-to-many relationship. The enrollment process is managed through an Enrollment relationship that connects students and courses. Each course is uniquely defined by a course ID and has attributes like course name, course code, and course credits. Additionally, the university is organized into departments, each identified by a department ID and department name. Faculty members, who are responsible for teaching courses, are identified by a faculty ID, along with details like name, address, and email. Faculty members may teach multiple courses, forming another many-to-many relationship between faculty and courses. This structured database design ensures efficient management of student records within the university.

ER Diagram



Normalization

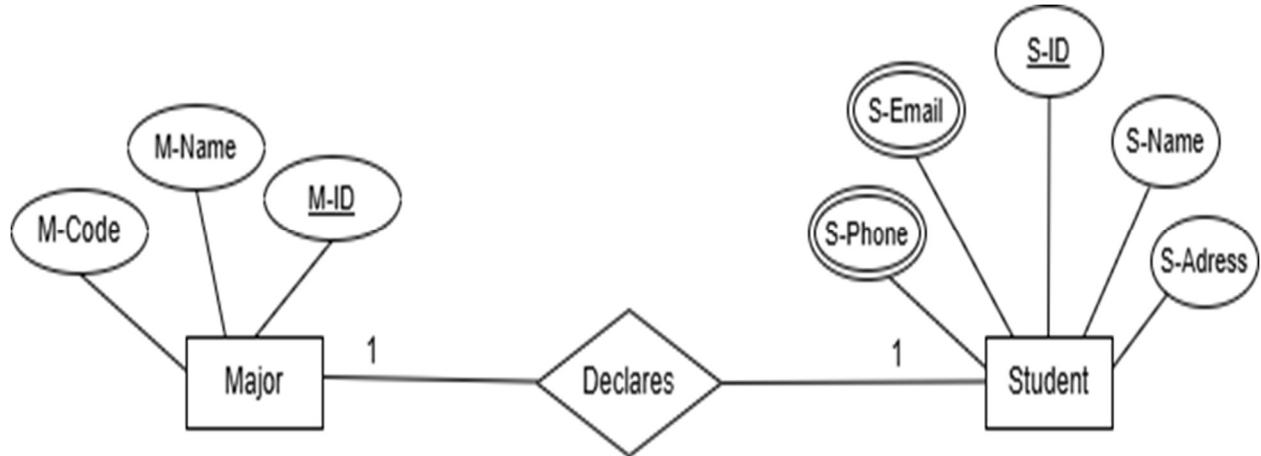


Fig-1: ER-Diagram for Student – Major

Student - Declares - Major

UNF : M_ID, M_Name, M_code, S_ID, S_Name, S_Email, S_Phone, S_Address

1NF: Multivalued attribute

S_Phone, S_Email

M_ID, M_Name, M_Code, S_ID, S_Name, S_Address

2NF: 1. M_ID, M_Name, M_Code
2. S_ID, S_Name, S_Address, M_ID
3. S_ID, S_Email, S_Phone

3NF: Same as 2NF

Student-Enrollment-Course

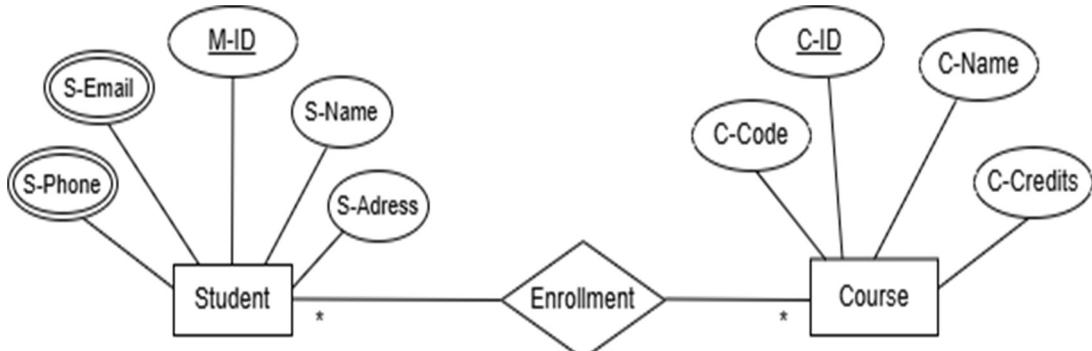


Fig-3: ER-Diagram for Student – Course

UNF : S_ID, S_Name, S_Address, S_Email, S_Phone, C_ID, C_Code, C_Name, C_Credits

1NF: Multivalued attribute

S_Phone, S_Email

S_ID, S_Name, S_Address, C_ID, C_Name, C_Code, C_Credits

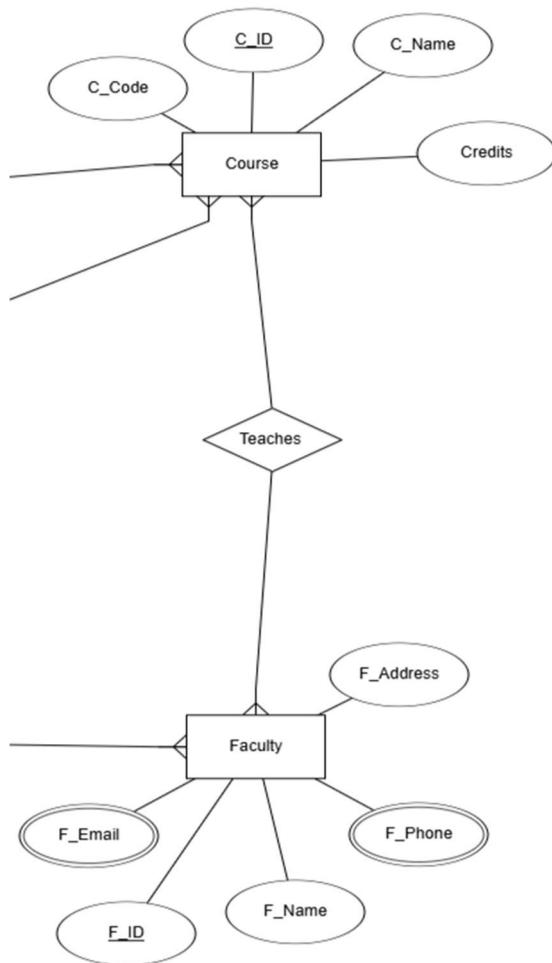
2NF: 1. S_ID, S_Name, S_Address

2. C_ID, C_Name, C_Code, C_Credits, S_ID

3. S_ID, S_Phone, S_Email,

3NF: Same as 2NF

Faculty-Teaches-Course



UNF: C_ID, C_Code, C_Name, Credits, F_ID, F_Email, F_Name, F_Phone, F_Adress

1NF: Multivalued attribute

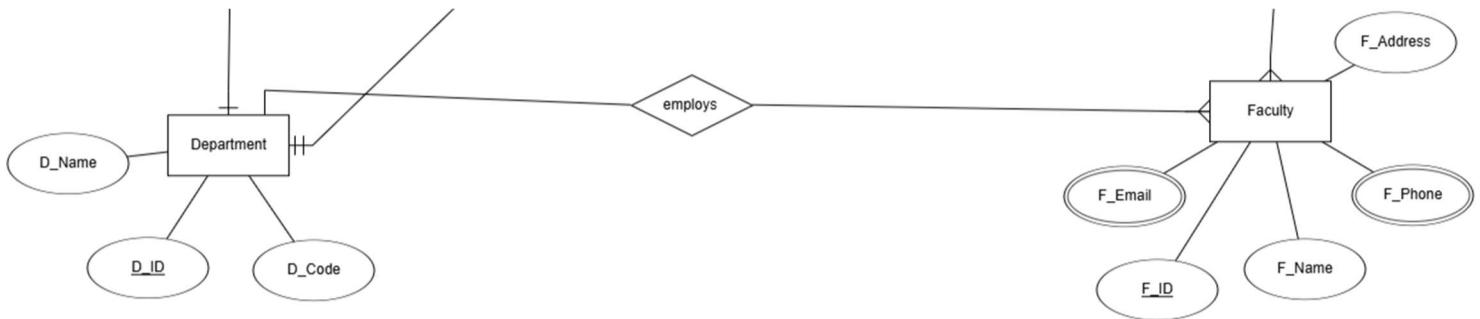
F_Phone, F_Email

C_ID, C_Name, C_Code, Credits, F_ID, F_Name, F_Address

2NF: 1. C_ID, C_Name, Credits, C_Code
2. F_ID, F_Name, F_Address, C_ID
3. F_ID, F_Email, F_Phone

3NF: Same as 2NF

Department-Employees-Faculty



UNF : D_ID,D_Name,D_Code,F_ID,F_Email,F_Name,F_Phone,F_Address

1NF: Multivalued attribute

F_Email,F_Phone

D_ID,D_Name,D_Code ,F_ID,F_Name,F_Address

2NF: 1.D_ID,D_Name,D_Code,
2.F_ID,F_Name,F_Address,D_ID
3.F_ID,F_Email,F_Phone

3NF: Same as 2NF

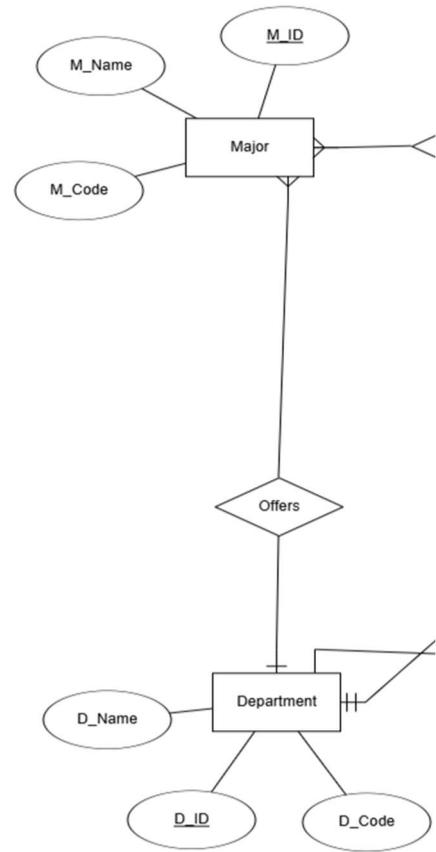
Department-Offers-Major

UNF : M_ID, M_Name, M_code,D_ID,D_Name,D_Code

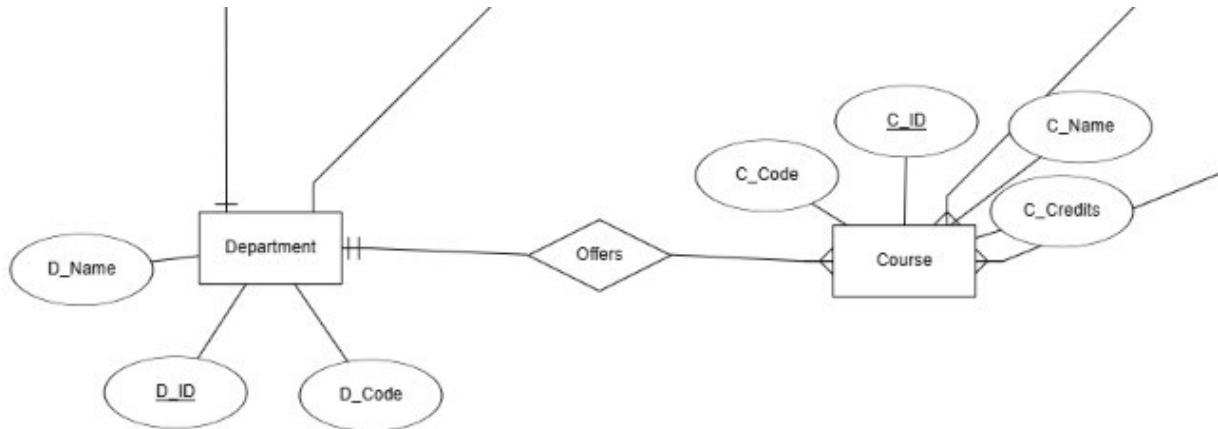
1NF: M_ID,M_Name,M_Code,D_ID,D_Name,D_Code

2NF: 1. M_ID,M_Name,M_Code,D_ID
2. D_ID,D_Name,D_Code

3NF: Same as 2NF



Department-Offers-Course



UNF: D_ID, D_Name, D_Code, C_ID, C_Name, C_Code, C_Credits

1NF: D_ID, D_Name, D_Code, C_ID, C_Name, C_Code, C_Credits

2NF: 1. D_ID, D_Name, D_Code
2. C_ID, C_Code, C_Name, C_Credits, D_ID

3NF: Same as 2NF

Finalization

1. S_ID,S_Name,S_Address,M_ID
2. S_ID,S_Email,S_Phone
3. C_ID,C_Name,C_Code, Credits,S_ID,D_ID
4. D_ID,D_Name,D_Code
5. F_ID,F_Name,F_Address,D_ID,C_ID
6. M_ID,M_Name,M_Code,D_ID
7. F_ID,F_Email,F_Phone

Table Creation (DDL Operations)

StudentID1: 23-53680-3 Name: MD. Zahin Daiyan	StudentID3: 23-54097-3 Name: Sadia Afrin
StudentID2: 23-54369-3 Name: Maruf Hossain Chowdhury Durlov	StudentID4: 23-54080-3 Name: Arpita Tarafder Arna
CO4: Creating DML, DDL using Oracle and connection with ODBC/JDBC for existing JAVA application	
PO-e-2: Use modern engineering and IT tools for prediction and modeling of complex computer science and engineering problem	Marks

1. Student Table Creation

Autocommit Display 10 Save Run

```
CREATE TABLE student (s_id INT PRIMARY KEY, s_name VARCHAR(100), s_address VARCHAR2(250), m_id INT,
CONSTRAINT fk_major FOREIGN KEY (m_id) REFERENCES major(m_id));
Describe student
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **STUDENT**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
STUDENT	S_ID	Number	-	-	0	1	-	-	-
	S_NAME	Varchar2	100	-	-	-	✓	-	-
	S_ADDRESS	Varchar2	250	-	-	-	✓	-	-
	M_ID	Number	-	-	0	-	✓	-	-
1 - 4									

2. Student Contacts table creation

Autocommit Display 10 Save Run

```
CREATE TABLE student_contact (s_email VARCHAR2(250) UNIQUE, s_phone VARCHAR2(250), s_id INT,
CONSTRAINT fk_student FOREIGN KEY (s_id) REFERENCES student(s_id));
describe student_contact
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **STUDENT_CONTACT**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
STUDENT_CONTACT	S_EMAIL	Varchar2	250	-	-	-	✓	-	-
	S_PHONE	Varchar2	250	-	-	-	✓	-	-
	S_ID	Number	-	-	0	-	✓	-	-
1 - 3									

3. Department table creation

Autocommit Display 10 Save 

```
CREATE TABLE department (d_id INT PRIMARY KEY, d_name VARCHAR2(100), d_code VARCHAR2(8));
describe department
```

Results Explain Describe Saved SQL History

Object Type TABLE Object DEPARTMENT

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DEPARTMENT	D_ID	Number	-	-	0	1	-	-	
	D_NAME	Varchar2	100	-	-	-	✓	-	
	D_CODE	Varchar2	8	-	-	-	✓	-	
									1 - 3

4. Course table creation

Autocommit Display 10 Save 

```
CREATE TABLE course (c_id INT PRIMARY KEY, c_name VARCHAR(100), c_code VARCHAR(8), credits INT,
s_id INT, d_id INT, CONSTRAINT fk_student_course FOREIGN KEY (s_id) REFERENCES student(s_id),
CONSTRAINT fk_department_course FOREIGN KEY (d_id) REFERENCES department(d_id));
```

Results Explain Describe Saved SQL History

Object Type TABLE Object COURSE

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
COURSE	C_ID	Number	-	-	0	1	-	-	
	C_NAME	Varchar2	100	-	-	-	✓	-	
	C_CODE	Varchar2	8	-	-	-	✓	-	
	CREDITS	Number	-	-	0	-	✓	-	
	S_ID	Number	-	-	0	-	✓	-	
	D_ID	Number	-	-	0	-	✓	-	
									1 - 6

5. Faculty table creation

Autocommit Display 10 Save 

```
CREATE TABLE faculty (f_id INT PRIMARY KEY, f_name VARCHAR2(100), f_address VARCHAR2(300), d_id
INT, c_id INT, CONSTRAINT fk_department_faculty FOREIGN KEY (d_id) REFERENCES department(d_id),
CONSTRAINT fk_course_faculty FOREIGN KEY (c_id) REFERENCES course(c_id));
```

Results Explain Describe Saved SQL History

Object Type TABLE Object FACULTY

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
FACULTY	F_ID	Number	-	-	0	1	-	-	
	F_NAME	Varchar2	100	-	-	-	✓	-	
	F_ADDRESS	Varchar2	300	-	-	-	✓	-	
	D_ID	Number	-	-	0	-	✓	-	
	C_ID	Number	-	-	0	-	✓	-	
									1 - 5

6. Faculty contacts table creation

Autocommit Display 10 Save Run

```
CREATE TABLE faculty_contact (f_email VARCHAR2(100) UNIQUE, f_phone VARCHAR2(100), f_id INT,
CONSTRAINT fk_faculty FOREIGN KEY (f_id) REFERENCES faculty(f_id));
describe faculty_contact
```

Results Explain Describe Saved SQL History

Object Type TABLE Object FACULTY_CONTACT

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
FACULTY_CONTACT	F_EMAIL	Varchar2	100	-	-	-	✓	-	-
	F_PHONE	Varchar2	100	-	-	-	✓	-	-
	F_ID	Number	-	-	0	-	✓	-	-

1 - 3

7. Major table creation

Autocommit Display 10 Save

```
CREATE TABLE major (m_id INT PRIMARY KEY, m_name VARCHAR2(100), m_code VARCHAR2(10), d_id INT,
CONSTRAINT fk_department_major FOREIGN KEY (d_id) REFERENCES department(d_id));
```

Results Explain Describe Saved SQL History

Object Type TABLE Object MAJOR

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MAJOR	M_ID	Number	-	-	0	1	-	-	-
	M_NAME	Varchar2	100	-	-	-	✓	-	-
	M_CODE	Varchar2	10	-	-	-	✓	-	-
	D_ID	Number	-	-	0	-	✓	-	-

1 - 4

Inserted Values in the tables

Give the screenshot of all the created tables in Oracle. All the tables should be attached one by one.

S_ID	S_NAME	S_ADDRESS	M_ID
2	Bill Gates	Rajshahi	2
3	Mark Zuckerberg	Dhaka	3
4	Zahin Daiyan	Chittagong	4
5	Maruf Chowdhury	Dhaka	5
6	Arpita Tarafdar	Sylhet	6
7	Sadia Afrin	Mirpur	7
8	Nayeem Ahmed	Uttara	8
9	Sujoy Kumar Mondol	Rangpur	1
10	Ovi Chowdhury	Khulna	5
1	Elon Musk	Chittagong	1

Figure: 'Student' table

S_EMAIL	S_PHONE	S_ID
zahin.daiyan@gmail.com	1234567890	4
maruf.chowdhury@gmail.com	2345678901	5
arpita.tarafdar@gmail.com	3456789012	6
sadia.afrin@gmail.com	4567890123	7
nayeem.ahmed@gmail.com	5678901234	8
sujoy.mondol@gmail.com	6789012345	9
ovi.chowdhury@gmail.com	7890123456	10
elon.musk@gmail.com	8901234567	1
bill.gates@gmail.com	9012345678	2
mark.zuckerberg@gmail.com	0123456789	3

Figure: 'Student_contact' table

D_ID	D_NAME	D_CODE
1	Computer Science and Engineering	CSE101
2	Computer Network and Cyber Security	CNCS102
3	Data Science	DS103
4	Electrical and Electronics Engineering	EEE104
5	Mechanical Engineering	ME105

Figure: Department table

C_ID	C_NAME	C_CODE	CREDITS	S_ID	D_ID
1	Data Structures	CS1011	3	1	1
2	Database Management	CS1012	3	2	1
3	Algorithms	CS1021	3	3	1
4	Software Engineering	CS1031	4	4	1
5	Machine Learning	AI1011	3	5	3
6	Artificial Intelligence	AI1021	4	6	3
7	Data Analytics	DA1011	3	7	2
8	Cybersecurity	CS1041	3	8	1
9	Networking	CS1051	3	9	1
10	Cloud Computing	CS1061	4	10	1
11	Database Systems	DB1011	3	2	2
12	Big Data Analysis	DA1021	4	3	2
13	Robotics	RE1011	4	4	4
14	Control Systems	RE1021	3	5	4
15	Mechanical Engineering Design	ME1011	4	6	5
16	Thermal Systems	ME1021	3	7	5

Figure: Course table

F_ID	F_NAME	F_ADDRESS	D_ID	C_ID
1	Sharukh Khan	456 Al Ave, Bashundhara	1	-
2	Salman Khan	789 Innovation St, Uttara	2	-
3	Amir Khan	123 Science Blvd, Mirpur	3	-
4	Batman	Gotham City, Batcave	4	-
5	Superman	Krypton, Fortress of Solitude	5	-
6	Ironman	Stark Tower, New York	1	-
7	Spiderman	Queens, New York	2	-

Figure: Faculty Table

F_EMAIL	F_PHONE	F_ID
spiderman@webslinger.com	7778889999	7
ironman@starkindustries.com	6667778888	6
superman@krypton.com	5556667777	5
batman@gotham.com	4445556666	4
amir.khan@aiub.edu	3334445555	3
salman.khan@aiub.edu	2223334444	2
sharuk.khan@aiub.edu	1112223333	1

Figure: Faculty Contact Table

M_ID	M_NAME	M_CODE	D_ID
1	Software Engineering	SE01	1
2	Data Analytics	DA02	2
3	Artificial Intelligence	AI03	3
4	Robotics Engineering	RE04	4
5	Thermal Engineering	TE05	5
6	Network Security	NS06	1
7	Cybersecurity	CS07	2
8	Mechatronics Engineering	ME08	4

Figure: Major Table

Query Test in DB

a) Simple query:

Autocommit Display 50 ▾

```
select * from student
```

Results Explain Describe Saved SQL History

S_ID	S_NAME	S_ADDRESS	M_ID
2	Bill Gates	Rajshahi	2
3	Mark Zuckerberg	Dhaka	3
4	Zahin Daiyan	Chittagong	4
5	Maruf Chowdhury	Dhaka	5
6	Arpita Tarafdar	Sylhet	6
7	Sadia Afrin	Mirpur	7
8	Nayeem Ahmed	Uttara	8
9	Sujoy Kumar Mondol	Rangpur	1
10	Ovi Chowdhury	Khulna	5
1	Elon Musk	Chittagong	1

b) Query with a single row function:

Autocommit Display 50 ▾

```
SELECT UPPER(s_name) AS Student_Name FROM student;
```

Results Explain Describe Saved SQL History

STUDENT_NAME
BILL GATES
MARK ZUCKERBERG
ZAHIN DAIYAN
MARUF CHOWDHURY
ARPITA TARAFDAR
SADIA AFRIN
NAYEEM AHMED
SUJOY KUMAR MONDOL
OVI CHOWDHURY
ELON MUSK

c) Query with a Multiple row function/ aggregate function:

The screenshot shows a MySQL query editor interface. At the top, there is a toolbar with a checked 'Autocommit' checkbox, a 'Display' dropdown set to 50, and a 'Save' button. Below the toolbar is the SQL query:

```
SELECT s.s_name, SUM(c.credits) AS total_credits FROM student s, course c WHERE s.s_id = c.s_id GROUP BY s.s_name
```

Below the query, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is selected, displaying the following table:

S_NAME	TOTAL_CREDITS
Sadia Afrin	6
Maruf Chowdhury	6
Nayeem Ahmed	3
Sujoy Kumar Mondol	3
Elon Musk	3
Bill Gates	6
Mark Zuckerberg	7
Zahin Daiyan	8
Arpita Tarafdar	8
Ovi Chowdhury	4

d) 2 Single row subquery:

The screenshot shows a MySQL query editor interface. At the top, there is a toolbar with a checked 'Autocommit' checkbox, a 'Display' dropdown set to 50, and a 'Save' button. Below the toolbar is the first query:

```
SELECT s_name FROM student WHERE m_id = (SELECT m_id FROM student WHERE s_name = 'Elon Musk')
```

Below the first query, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is selected, displaying the following table:

S_NAME
Sujoy Kumar Mondol
Elon Musk

At the bottom of the editor, there is another toolbar with a checked 'Autocommit' checkbox, a 'Display' dropdown set to 50, and a 'Save' button. Below this toolbar is the second query:

```
SELECT s_name FROM student WHERE m_id = (SELECT m_id FROM student WHERE s_name = 'Maruf Chowdhury')
```

Below the second query, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is selected, displaying the following table:

S_NAME
Maruf Chowdhury
Ovi Chowdhury

2 multiple row subquery:

Autocommit Display 50 Save

```
SELECT s.s_name , m.m_name FROM student s, major m WHERE s.m_id IN (SELECT m_id FROM major WHERE d_id = 2)
```

Results Explain Describe Saved SQL History

S_NAME	M_NAME
Bill Gates	Mechatronics Engineering
Bill Gates	Cybersecurity
Bill Gates	Network Security
Bill Gates	Thermal Engineering
Bill Gates	Robotics Engineering
Bill Gates	Artificial Intelligence
Bill Gates	Data Analytics
Bill Gates	Software Engineering
Sadia Afrin	Mechatronics Engineering
Sadia Afrin	Cybersecurity
Sadia Afrin	Network Security
Sadia Afrin	Thermal Engineering
Sadia Afrin	Robotics Engineering
Sadia Afrin	Artificial Intelligence
Sadia Afrin	Data Analytics
Sadia Afrin	Software Engineering

Autocommit Display 50

```
SELECT s.s_name FROM student s WHERE s.m_id IN (SELECT m_id FROM major WHERE d_id = 1)
```

Results Explain Describe Saved SQL History

S_NAME
Arpita Tarafdar
Sujoy Kumar Mondol
Elon Musk

e) 4 kinds of joining :

Equijoin:

The screenshot shows a MySQL query interface with the following details:

- Autocommit is checked.
- Display dropdown is set to 50.
- Save button is visible.
- SQL query: `select student.s_name , student.m_id , major.m_id , major.m_name from student , major where student.m_id = major.m_id`
- Results tab is selected.
- Table output:

S_NAME	M_ID	M_ID	M_NAME
Bill Gates	2	2	Data Analytics
Mark Zuckerberg	3	3	Artificial Intelligence
Zahin Daiyan	4	4	Robotics Engineering
Maruf Chowdhury	5	5	Thermal Engineering
Arpita Tarafdar	6	6	Network Security
Sadia Afrin	7	7	Cybersecurity
Nayeem Ahmed	8	8	Mechatronics Engineering
Sujoy Kumar Mondol	1	1	Software Engineering
Ovi Chowdhury	5	5	Thermal Engineering
Elon Musk	1	1	Software Engineering

Non-equijoin:

Autocommit Display 50

```
SELECT s.s_name, c.c_name, c.credits FROM student s, course c WHERE c.credits BETWEEN 4 AND 9
```

Results Explain Describe Saved SQL History

S_NAME	C_NAME	CREDITS
Bill Gates	Software Engineering	4
Mark Zuckerberg	Software Engineering	4
Zahin Daiyan	Software Engineering	4
Maruf Chowdhury	Software Engineering	4
Arpita Tarafdar	Software Engineering	4
Sadia Afrin	Software Engineering	4
Nayeem Ahmed	Software Engineering	4
Sujoy Kumar Mondol	Software Engineering	4
Ovi Chowdhury	Software Engineering	4
Elon Musk	Software Engineering	4
Bill Gates	Artificial Intelligence	4
Mark Zuckerberg	Artificial Intelligence	4
Zahin Daiyan	Artificial Intelligence	4
Maruf Chowdhury	Artificial Intelligence	4
Arpita Tarafdar	Artificial Intelligence	4
Sadia Afrin	Artificial Intelligence	4
Nayeem Ahmed	Artificial Intelligence	4
Sujoy Kumar Mondol	Artificial Intelligence	4
Ovi Chowdhury	Artificial Intelligence	4
Elon Musk	Artificial Intelligence	4
Bill Gates	Cloud Computing	4
Mark Zuckerberg	Cloud Computing	4
Zahin Daiyan	Cloud Computing	4
Maruf Chowdhury	Cloud Computing	4
Arpita Tarafdar	Cloud Computing	4

Outer join:

Autocommit Display 10

```
select s.s_name, m.m_name from student s, major m where s.m_id(+) = m.m_id
```

Results Explain Describe Saved SQL History

S_NAME	M_NAME
Bill Gates	Data Analytics
Mark Zuckerberg	Artificial Intelligence
Zahin Daiyan	Robotics Engineering
Maruf Chowdhury	Thermal Engineering
Arpita Tarafdar	Network Security
Sadia Afrin	Cybersecurity
Nayeem Ahmed	Mechatronics Engineering
Sujoy Kumar Mondol	Software Engineering
Ovi Chowdhury	Thermal Engineering
Elon Musk	Software Engineering

Self Join:

Autocommit Display 50 Save

```
SELECT f1.f_name || ' and ' || f2.f_name || ' both work in the ' || d.d_name || ' department'
FROM faculty f1, faculty f2, department d
WHERE f1.d_id = f2.d_id
AND f1.f_id <> f2.f_id
AND f1.d_id = d.d_id;
```

Results Explain Describe Saved SQL History

F1.F_NAME 'AND' F2.F_NAME 'BOTHWORKINTHE' D.D_NAME 'DEPARTMENT'
Ironman and Sharukh Khan both work in the Computer Science and Engineering department
Spiderman and Salman Khan both work in the Computer Network and Cyber Security department
Sharukh Khan and Ironman both work in the Computer Science and Engineering department
Salman Khan and Spiderman both work in the Computer Network and Cyber Security department

1 simple view
1 complex view

SIMPLE VIEW:

```
✓ Autocommit Display 50 ▾  
create view major1 as select s_name , m_id from student where m_id = 1
```

Figure: Simple view creation command

Object Type **VIEW** Object **MAJOR1**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MAJOR1	S_NAME	Varchar2	100	-	-	-	✓	-	-
	M_ID	Number	-	-	0	-	✓	-	-
1 - 2									

Figure: Description of simple view

S_NAME	M_ID
Sujoy Kumar Mondol	1
Elon Musk	1

Figure: Result of the simple view as a whole table

Complex View:

```
✓ Autocommit Display 50 ▾  
  
CREATE VIEW TotalCredits AS  
SELECT s.s_name, SUM(c.credits) AS total_credits, m.m_name, d.d_name  
FROM student s, course c, major m, department d  
WHERE s.s_id = c.s_id  
AND s.m_id = m.m_id  
AND m.d_id = d.d_id  
GROUP BY s.s_name, m.m_name, d.d_name;
```

Figure: Complex view creation command

Object Type VIEW Object TOTALCREDITS										
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment	
TOTALCREDITS	S_NAME	Varchar2	100	-	-	-	✓	-	-	
	TOTAL_CREDITS	Number	-	-	-	-	✓	-	-	
	M_NAME	Varchar2	100	-	-	-	✓	-	-	
	D_NAME	Varchar2	100	-	-	-	✓	-	-	
1 - 4										

Figure: Description of Complex view

S_NAME	TOTAL_CREDITS	M_NAME	D_NAME
Maruf Chowdhury	6	Thermal Engineering	Mechanical Engineering
Sadia Afrin	6	Cybersecurity	Computer Network and Cyber Security
Ovi Chowdhury	4	Thermal Engineering	Mechanical Engineering
Mark Zuckerberg	7	Artificial Intelligence	Data Science
Zahin Daiyan	8	Robotics Engineering	Electrical and Electronics Engineering
Nayeem Ahmed	3	Mechatronics Engineering	Electrical and Electronics Engineering
Sujoy Kumar Mondol	3	Software Engineering	Computer Science and Engineering
Bill Gates	6	Data Analytics	Computer Network and Cyber Security
Arpita Tarafdar	8	Network Security	Computer Science and Engineering
Elon Musk	3	Software Engineering	Computer Science and Engineering

Figure: Result of the complex view as a whole table

Description of a Successful DB connection

Maruf Hossain Chowdhury Durlov [23-54369-3]

1) Acquiring tools: MySQL Java Connector: Obtained the MySQL Java Connector JAR file from the official MySQL website to facilitate Java and MySQL interaction. Additionally, installed XAMPP to access myphp Admin

2) Environment Setup:

- Opened XAMPP, activating both Apache and MySQL services via the control panel.
- Accessed the MySQL admin panel to administer databases.
- Created a database named "The University Management System" and formulated a table named "Student" within it, specifying relevant columns and data types. Filled the table with various values for operations.

ID	Name	Address
54369	Maruf	C Block, Basundhara R/A
54370	Sadia	Mirpur DoHS
54371	Arpita	Badda
54372	Zahin	Uttara

The screenshot shows the phpMyAdmin interface with the following details:

- Database:** the university management system
- Table:** student
- SQL Query:** SELECT * FROM `student`
- Results:** Showing rows 0 - 3 (4 total, Query took 0.0002 seconds.)
- Data:** The table contains 4 rows with the following data:

ID	Name	Address
54369	Maruf	C Block, Basundhara R/A
54370	Sadia	Mirpur DoHS
54371	Arpita	Badda
54372	Zahin	Uttara
- Operations:** Print, Copy to clipboard, Export, Display chart, Create view, Bookmark this SQL query, Console.

Figure: Screenshot of phpMyAdmin when database and tables are created

3) Java Code :

- Opted for Notepad++ and Java development kit (JDK) as the Integrated Development Environment (IDE) for Java development.
- Then downloaded the DBCODE file from portal and opened the Object.java file.
- Typed the database name after //localhost::3306/ prompt. (at line 9)
- Typed the table name in the st.executeQuery("select * from student) (at line 14)

```

C:\Users\durlo\OneDrive\Documents\Study Materials>IDB\DBCODE\Object.java - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
Object.java
1 import java.sql.*;
2
3 public class Object {
4
5     public static void main(String[] args) {
6         try {
7             Class.forName("com.mysql.cj.jdbc.Driver"); // register jdbc driver of mysql.Driver Registration
8             //This line ensures that the MySQL JDBC driver is loaded and registered with the JDBC API. It allows your application to conne
9             Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/the university management system", "root", ""); //E
10            //This line establishes a connection to the database named newdb running on localhost at the default MySQL port 3306.
11            //DriverManager.getConnection() returns a Connection object which represents the session with the database.
12            System.out.println("connected");//printing that database is connected
13            Statement st = conn.createStatement(); //Creating a Statement which allows you to send SQL queries to the database.
14            ResultSet rs = st.executeQuery("select * from Student");//Executing the SQL Query
15            while (rs.next()) {//moves the cursor to the next row of the result set. If there are no more rows, it returns false, terminat
16                System.out.println("ID=" + rs.getString(1));
17                System.out.println("Name=" + rs.getString(2));
18                System.out.println("Address=" + rs.getString(3));
19            }
20        } // Connection.close(); //This line is commented out, which means the connection is not being closed after use. This could lead
21        // Ideally, you should close the connection after completing your operations. You can do this either by explicitly calling conn.
22    } catch (Exception s) {
23        System.out.println(s);
24    }
25
26 }
27
28
29
30 //javac -cp Driver.jar;. Object.java
31 //java -cp Driver.jar;. Object
32
33
Java source file
18°C Haze
Windows (CR LF) UTF-8 INS
ENG UK 04:35 AM 04/02/2025

```

- After that opened the containing folder in cmd, compile and execute the following command
- //javac -cp Driver.jar;. Object.java
- //java -cp Driver.jar;. Object

```

C:\Users\durlo\OneDrive\Documents\Study Materials>IDB\DBCODE\Object.java - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
Object.java
1 import java.sql.*;
2
3 public class Object {
4     public static void main(String[] args) {
5         try {
6             Class.forName("com.mysql.cj.jdbc.Driver");
7             // Microsoft Windows [Version 10.0.26100.3037]
8             // (c) Microsoft Corporation. All rights reserved.
9             Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/the university management system", "root", "");
10            Statement st = conn.createStatement();
11            ResultSet rs = st.executeQuery("select * from Student");
12            while (rs.next()) {
13                System.out.println("ID=" + rs.getString(1));
14                System.out.println("Name=" + rs.getString(2));
15                System.out.println("Address=" + rs.getString(3));
16            }
17        } catch (Exception s) {
18            System.out.println(s);
19        }
20    }
21
22 }
23
24
25
26
27
28
29
30 //javac -cp Driver.jar;. Object.java
31 //java -cp Driver.jar;. Object
32
33
Java source file
18°C Haze
Windows (CR LF) UTF-8 INS
ENG UK 04:36 AM 04/02/2025

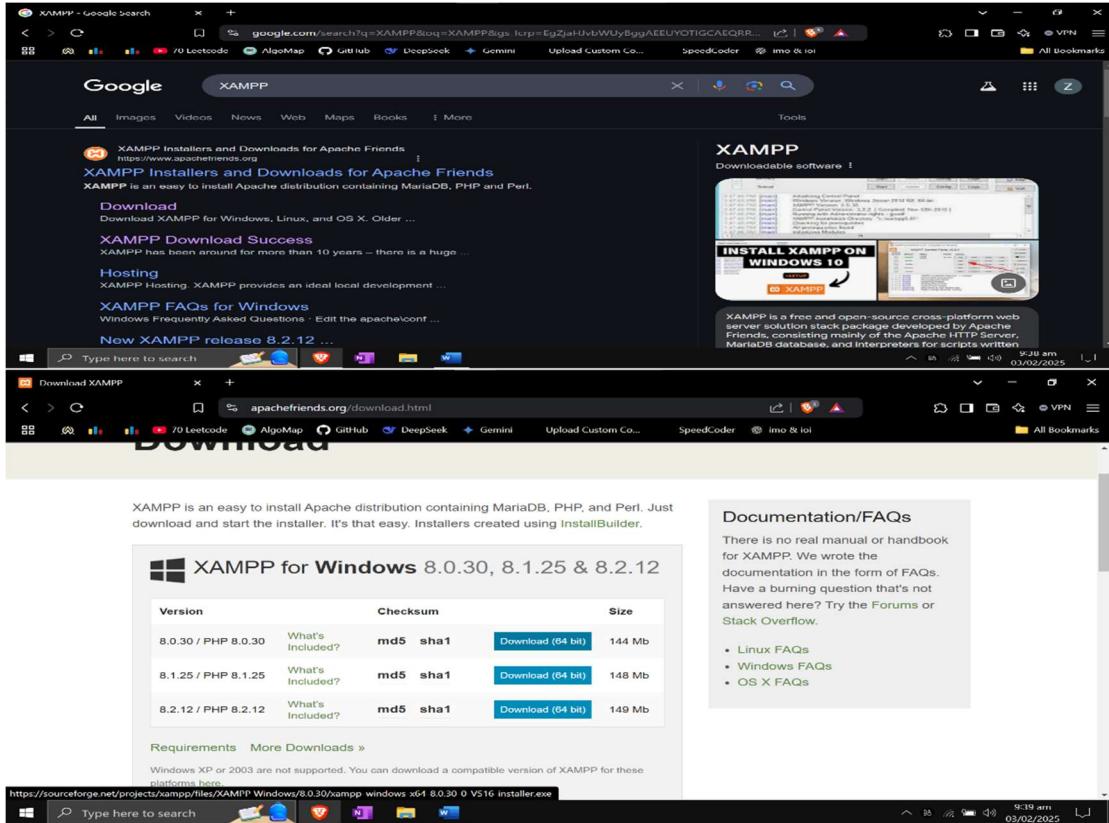
```

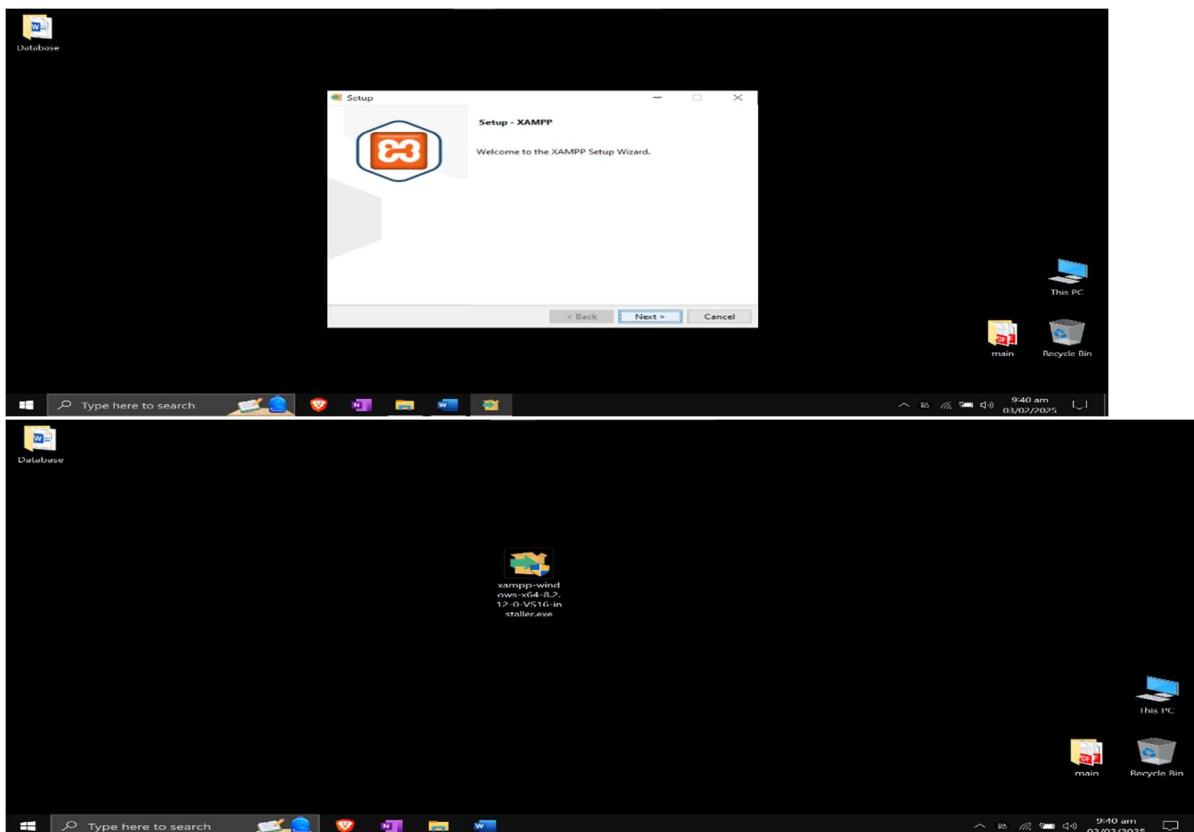
```
C:\Users\durlo\OneDrive\Documents\  
connected  
ID=54369  
Name=Maruf  
Address=C Block, Basundhara R/A  
ID=54370  
Name=Sadia  
Address=Mirpur DoHS  
ID=54371  
Name=Arpita  
Address=Badda  
ID=54372  
Name=Zahin  
Address=Uttara
```

MD Zahin Daiyan [23-53680-3]

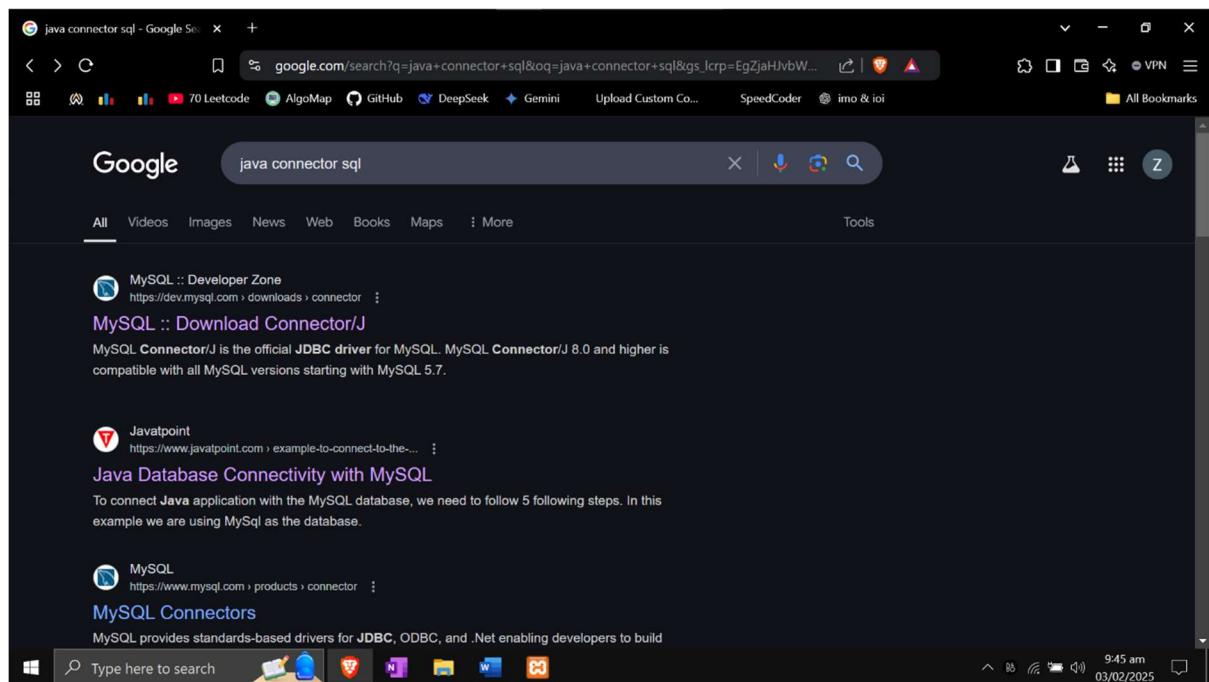
Steps to Establish Java Connection:

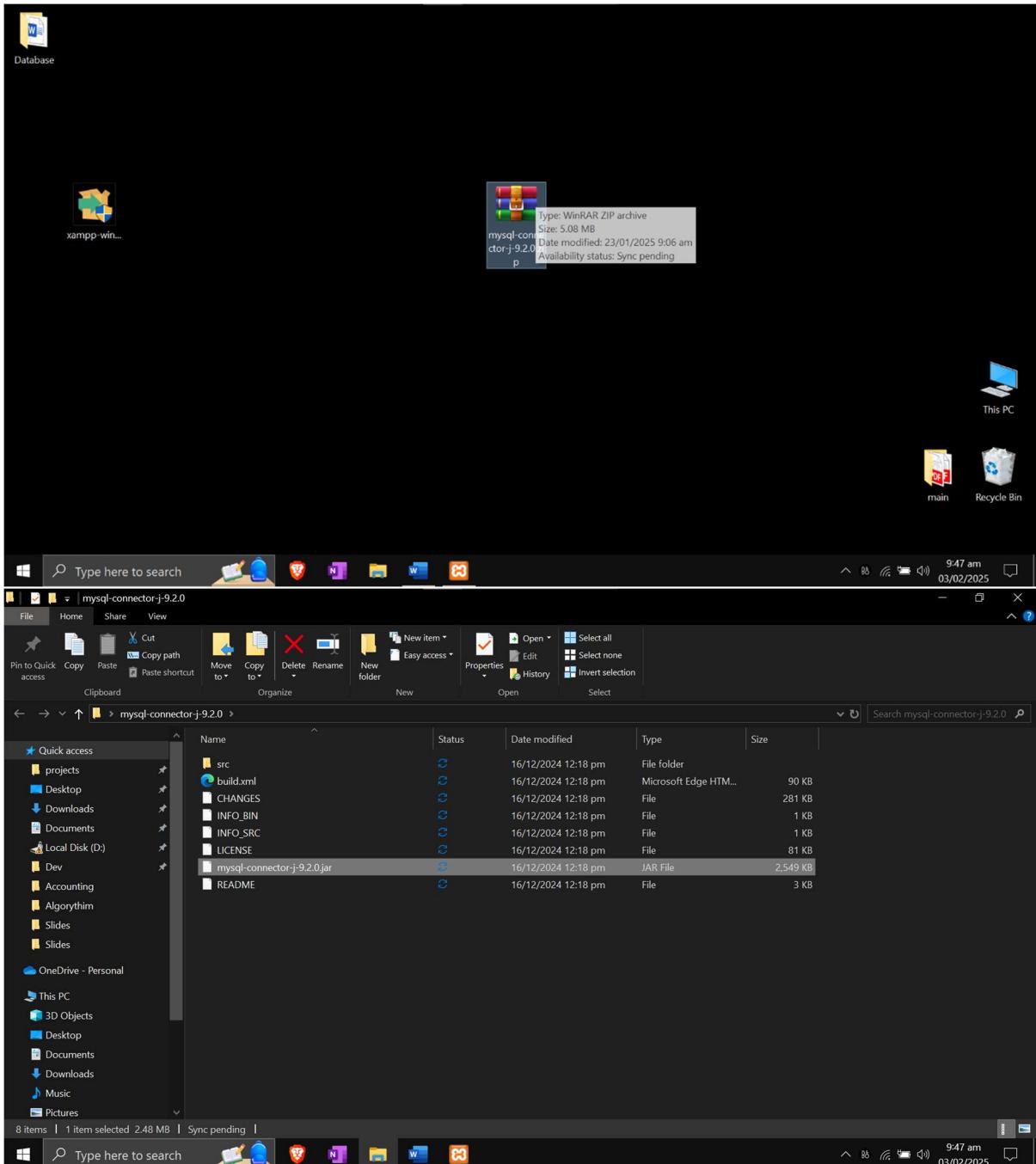
1. At first, I've downloaded and installed XAMPP for windows from the official website.



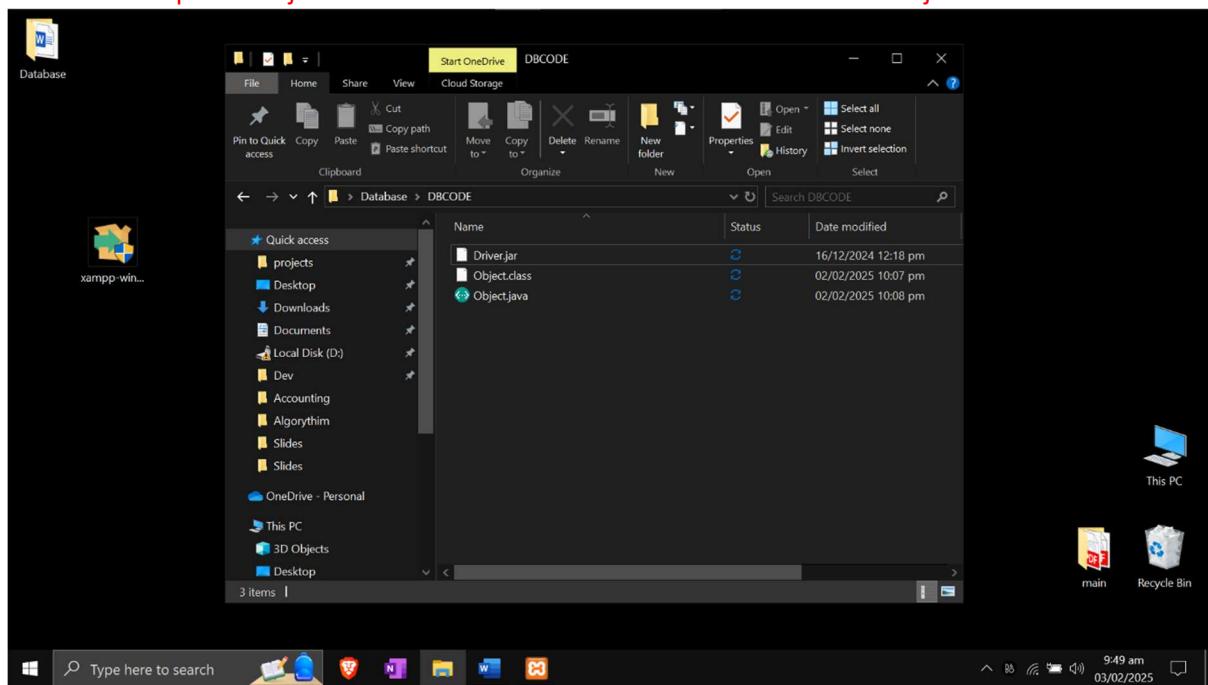


2. Then I downloaded Java Connector file

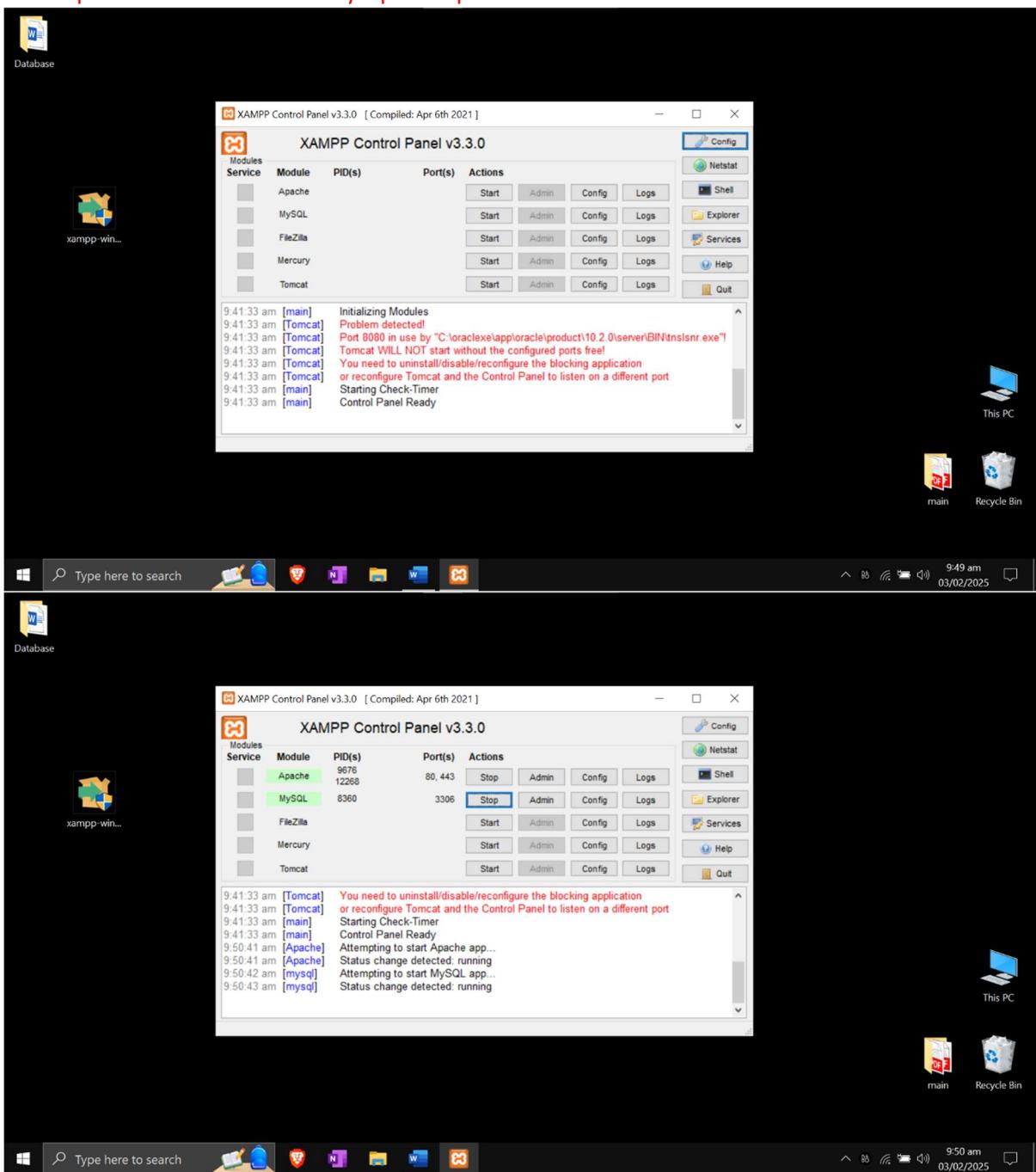




3. After that I Copied the .jar file to the driver folder and rename the file Driver.jar



4. And I opened XAMPP and start my sql and Apache



5. Clicked admin in my sql to go to php's admin landing page

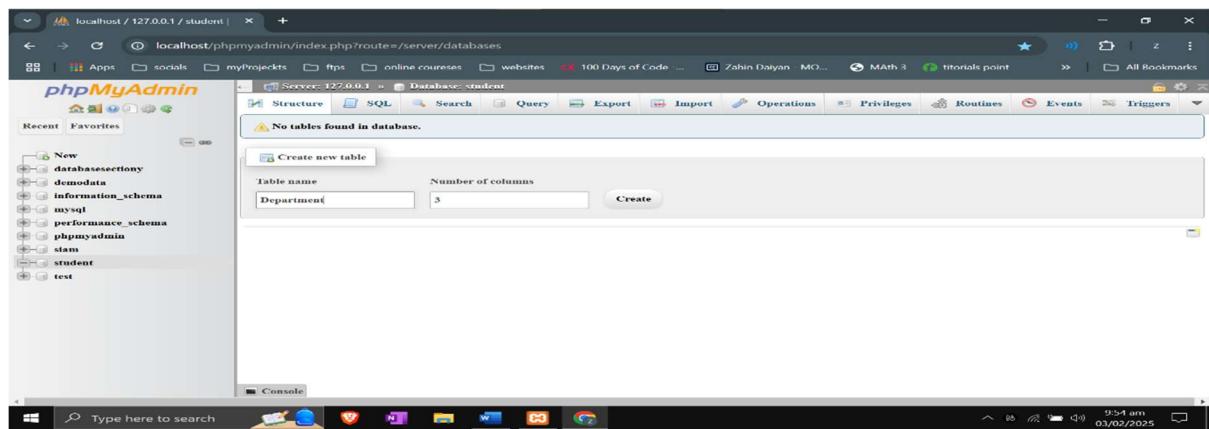
The screenshot shows the phpMyAdmin interface on a Windows desktop. The title bar says "localhost / 127.0.0.1 | phpMyAdmin". The main menu includes Databases, SQL, Status, User accounts, Export, Import, Settings, Replication, Variables, and More. On the left, a sidebar lists databases: New, databasesectiony, demodata, information_schema, mysql, performance_schema, phpmyadmin, siam, and test. The central area has sections for General settings (Server connection collation: utf8mb4_unicode_ci), Appearance settings (Language: English, Theme: pmahomme), Database server (Server: 127.0.0.1 via TCP/IP, Server type: MariaDB, Server connection: SSL is not being used, Server version: 10.4.32-MariaDB - mariadb.org binary distribution, Protocol version: 10, User: root@localhost, Server charset: UTF-8 Unicode (utf8mb4)), and Web server (Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12, Database client version: libmysql - mysqld 8.2.12, PHP extension: mysqli curl mbstring, PHP version: 8.2.12). At the bottom, there's a search bar and a taskbar with various icons.

6. Made a New Database

The screenshot shows the Databases section of phpMyAdmin. The title bar is "localhost / 127.0.0.1 | phpMyAdmin". The main menu is the same as the previous screenshot. The left sidebar shows the same database list. The central area has a "Create database" form with "Student" entered in the name field and "utf8mb4_general_ci" selected as the collation. Below it is a table of existing databases:

Database	Collation	Action
databasesectiony	utf8mb4_general_ci	Check privileges
demodata	utf8mb4_general_ci	Check privileges
information_schema	utf8_general_ci	Check privileges
mysql	utf8mb4_general_ci	Check privileges
performance_schema	utf8_general_ci	Check privileges
phpmyadmin	utf8_bin	Check privileges
siam	utf8mb4_general_ci	Check privileges
Console	latin1_swedish_ci	Check privileges

7. Entered tables and inserted rows and inserted relevant values



The screenshot shows two consecutive screenshots of the phpMyAdmin interface.

Screenshot 1: The user is creating a new table named 'Department'. The table structure is defined with three columns: 'D_Id' (INT), 'D_Name' (TEXT), and 'D_Code' (TEXT). The storage engine is set to InnoDB. The table comments field is empty.

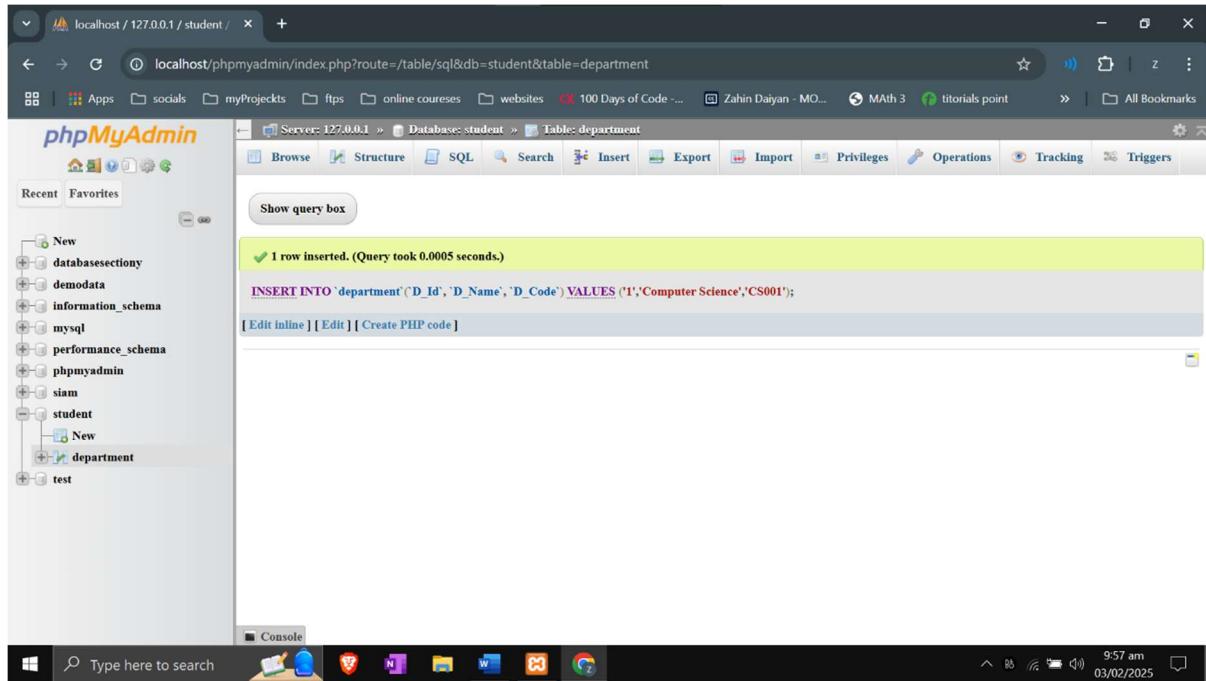
Name	Type	Length/Values	Default	Collation	Attributes	Null	Index
D_Id	INT		None				
D_Name	TEXT	100	None				
D_Code	TEXT	100	None				

Screenshot 2: The user has run an SQL query to insert a row into the 'Department' table. The query is: `INSERT INTO 'department' ('D_Id', 'D_Name', 'D_Code') VALUES ('1','Computer Science','CS001')`. The result of the query is shown in the results pane, displaying the inserted row.

```

1 INSERT INTO `department`(`D_Id`, `D_Name`, `D_Code`) VALUES ('1','Computer Science','CS001')

```



8. Open the folder where we kept our Driver.jar file and open the object.java file with a text editor
9. Change the Database name to the Same name we created the database with, enter the table name

```

C:\Users\zahii\OneDrive\Desktop\Database\DBCODE\Object.java - CP Editor
File Edit Actions View Options Help
Object.java - main\database\code\DBCODE *
1 import java.sql.*;
2
3 public class Object {
4
5     public static void main(String[] args) {
6         try {
7             Class.forName("com.mysql.cj.jdbc.Driver"); // register jdbc driver of mysql,Driver Registration
8             //This line ensures that the MySQL JDBC driver is loaded and registered with the JDBC API. It allows your application
9             Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/student", "root", ""); //Establishing a Connec
10            //This line establishes a connection to the database named newdb running on localhost at the default MySQL port 3306.
11            //DriverManager.getConnection() returns a Connection object which represents the session with the database.
12            System.out.println("connected");//printing that database is connected
13            Statement st = conn.createStatement(); //Creating a Statement which allows you to send SQL queries to the database.
14            ResultSet rs = st.executeQuery("select * from department"); //Executing the SQL Query
15            while (rs.next()) { //moves the cursor to the next row of the result set. If there are no more rows, it returns false,
16                System.out.println("D_ID = " + rs.getInt(1));
17                System.out.println("D_Name = " + rs.getString(2));
18                System.out.println("D_Code = " + rs.getString(3));
19            }
20            // Connection.close(); //This line is commented out, which means the connection is not being closed after use. This cc
21            // Ideally, you should close the connection after completing your operations. You can do this either by explicitly call
22            } catch (Exception s) {
23                System.out.println(s);
24            }
25        }
26    }
27 }
28

```

Line 1, Column 19 Compile Run Compile and Run

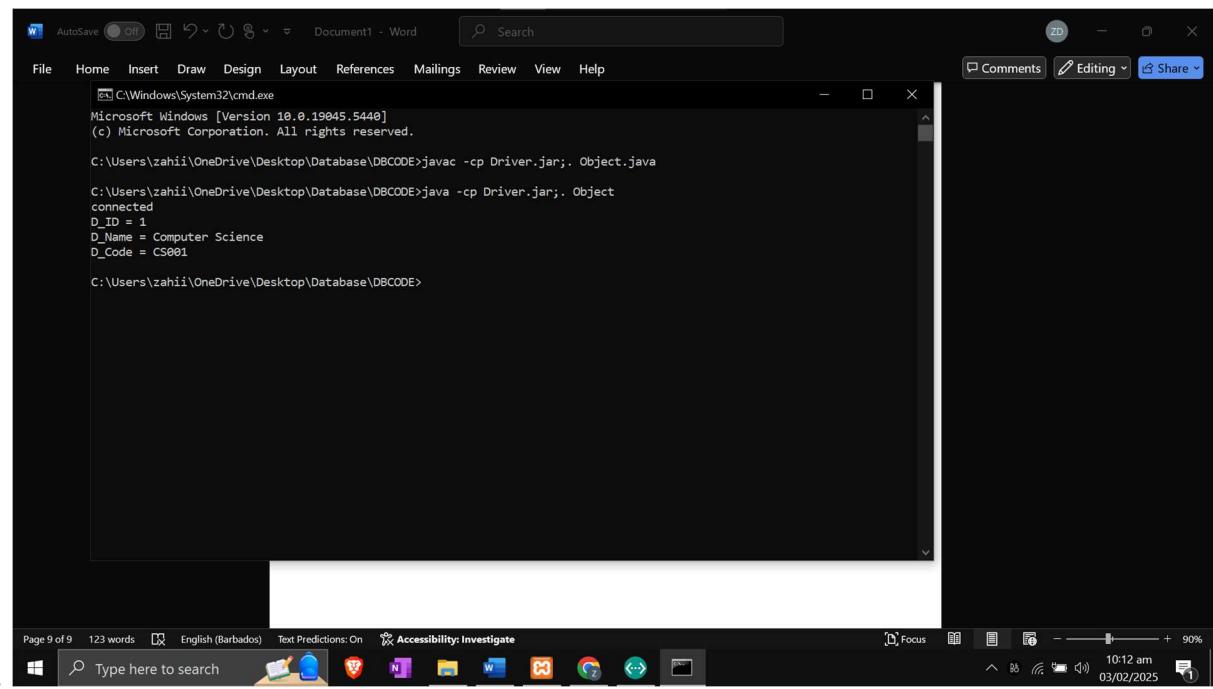
10. Open containing folders cmd

The screenshot shows a Windows Command Prompt window titled 'C:\Windows\System32\cmd.exe'. The command entered is 'javac -cp Driver.jar;. Object.java'. The output shows the Java code being compiled. A tooltip from the Java documentation for the 'Statement' class is visible on the right side of the screen, explaining its purpose and usage.

11. Complie the java file and execute it

The screenshot shows a Windows Command Prompt window titled 'C:\Windows\System32\cmd.exe'. The command entered is 'java -cp Driver.jar;. Object'. The output shows the Java code running and connecting to a MySQL database. A tooltip from the Java documentation for the 'Statement' class is visible on the right side of the screen, explaining its purpose and usage.

12. Connection is Established



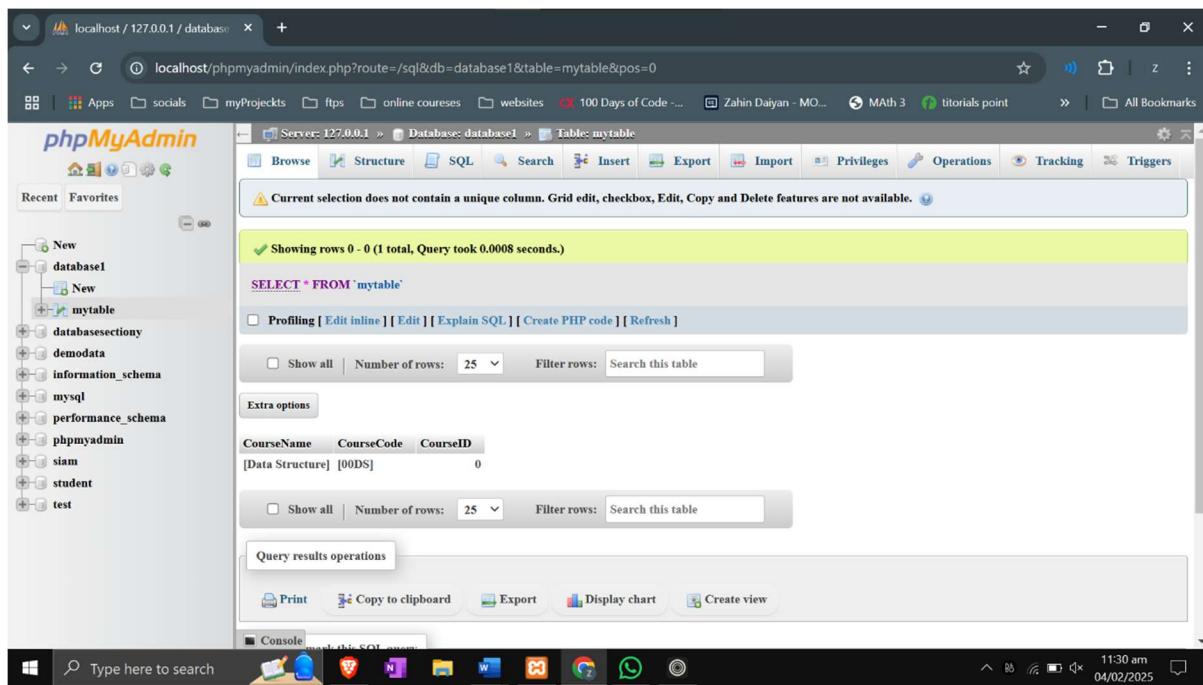
14.

Sadia Afrin [23-54097-3]

1. First I installed these software and jar file

- Install any IDE
- Download mysql-connector-java-8.0.28.jar
- Install xampp Server

2. I opened xampp server from xampp control panel then start Apache and MySQL module then open phpMyAdmin by clicking MySQL's Admin button. There I have created a Database named 'Database1' and created a table named 'myTable'. After that I had inserted some values in it.



The screenshot shows the phpMyAdmin interface. The left sidebar lists databases: New, database1 (selected), New, mytable, databasesectiony, demodata, information_schema, mysql, performance_schema, phpmyadmin, siam, student, and test. The main area shows the 'mytable' table under 'database1'. The table structure is defined by columns: CourseName, CourseCode, and CourseID. A single row is displayed: [Data Structure] [00DS] 0. Below the table, there are buttons for Print, Copy to clipboard, Export, Display chart, and Create view. At the bottom, there is a console window and a taskbar with various application icons.

After creating Database and Table, I opened myID(VS Code) and created a project .In the project I added the mysql-connector-java-8.0.28.jar file in library. After that I have written java code to connect my Database that I created through xampp. In the IDE have printed the table

C:/Users/zahii/OneDrive/Desktop/Database/DBCODE/Object.java - CP Editor

File Edit Actions View Options Help

Object.java * Insert Draw Design Layout References Mailings Review View Help

```

1 import java.sql.*;
2 public class Object {
3
4     public static void main(String[] args) {
5         try {
6             Class.forName("com.mysql.cj.jdbc.Driver");
7             Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/database1",
8             System.out.println("connected");
9             Statement st = conn.createStatement();
10            ResultSet rs = st.executeQuery("select * from myTable");
11            After creating Database and connecting to it in VS Code and created a
12            while (rs.next()) {
13                System.out.println("course Name = " + rs.getString(1));
14                System.out.println("course code = " + rs.getString(2));
15                System.out.println("Course ID = " + rs.getInt(3));
16            }
17        } catch (Exception s) {
18            System.out.println(s);
19        }
20    }

```

Line 1, Column 19

Type here to search

File C:\Windows\System32\cmd.exe

Microsoft Windows [Version 10.0.19045.5440]
(c) Microsoft Corporation. All rights reserved.

```

C:\Users\zahii\OneDrive\Desktop\Database\DBCODE>javac -cp Driver.jar;. Object.java
C:\Users\zahii\OneDrive\Desktop\Database\DBCODE>java -cp Driver.jar;. Object
connected
course Name = [Data Structure]
course code = [000DS]
Course ID = 0

```

C:\Users\zahii\OneDrive\Desktop\Database\DBCODE>

21 // / * System.out.println("SID=" + rs.getInt(1) + " First Name=" + rs.getString(2) + " Last Name=" + rs.getString(3));
22 // */
23 // */
24 // */
25 // */
26 // */
27 /* System.out.println("SID=" + rs.getInt(1) + " First Name=" + rs.getString(2) + " Last Name=" + rs.getString(3));
28 */

28 characters selected

Type here to search

Arpita Tarafder Arna [23-54080-3]

1. First of all, I downloaded the XAMPP software and installed it for creating a local host. After opening XAMPP control panel I started apache and MySQL module and clicked on MySQL admin for opening the server in a browser and created a database.

The screenshot shows the phpMyAdmin interface on a Windows desktop. The left sidebar lists databases: New, arpita, arpita tarafder, database, demo, New, demutable, information_schema, mysql, newdata, New, demutable, performance_schema, phpmyadmin, and test. The 'demutable' database is selected. The main area shows a table named 'demutable' with one row of data: name (Arpita) and id (23540803). Below the table, there are sections for 'Query results operations' (Print, Copy to clipboard, Export, Display chart, Create view) and 'Bookmark this SQL query' (Label input field, Let every user access this bookmark checkbox). At the bottom, there is a 'Console' section and a taskbar with various icons.

2. After creating a database I installed an IDE. I choose to use notepad++ because I seems easy for me. After installing notepad++ I created a project name university management system and wrote a code for connecting Database with java .

The screenshot shows a Notepad++ window with the title bar "C:\Users\hp\Desktop\DBCODE.zip\Object.java - Notepad++ [Administrator]". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, and Help. The toolbar contains icons for Open, Save, Find, Replace, Cut, Copy, Paste, Select All, Undo, Redo, and others. The status bar at the bottom shows "length: 770 lines: 29 Ln: 29 Col: 1 Pos: 771 Windows (CR LF) UTF-8 INS". The main code area displays the following Java code:

```
import java.sql.*;
public class Object {
    public static void main(String[] args) {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/newdata", "root", "");
            System.out.println("connected");
            Statement st = conn.createStatement();
            ResultSet rs = st.executeQuery("select * from demotable");
            while (rs.next()) {
                System.out.println("Names=" + rs.getString(1));
                System.out.println("ID=" + rs.getInt(2));
            }
        } catch (Exception s) {
            System.out.println(s);
        }
    }
}
```

3. Before executing the code I downloaded a java connector JAR file and attached it in the notepad++ referenced library. Then executed the program and it's connected to the database.

The screenshot shows a Notepad+ window with multiple tabs open, including 'DisplayResult.java', 'First.java', 'Firstpg.java', 'Object.java', and two instances of 'Object.java'. Below the tabs is a terminal window titled 'Administrator:C:\WINDOWS' showing Java command-line output. The terminal content includes:

```
Microsoft Windows [Version 10.0.26100.2894]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp\Desktop\DBCODE.zip>javac -cp Driver.jar;. Object.java

C:\Users\hp\Desktop\DBCODE.zip>java -cp Driver.jar;. Object
connected
Name = Arpita
ID = 23540803

C:\Users\hp\Desktop\DBCODE.zip>
```

At the bottom of the terminal window, line 26 of a Java source file is visible:

```
25
26     /* System.out.println("SID=" + rs.getInt(1) + " First Name=" + rs.getString(2) + " Last Name=" + r
```

Conclusion

Our University Student Management System helps universities manage student enrollment, courses, and faculty assignments more efficiently using a well-structured database. By applying SQL and normalization, we have ensured accurate data storage, reduced duplication, and automated important tasks. This system benefits students, teachers, and university staff by making academic management smoother.

In the future, we can improve the system by adding a user-friendly interface, cloud storage, and better reporting features. These upgrades will make it even more useful for universities. This project is a step toward modernizing education management, making it more organized and efficient.