**AMERICAN INTERNATIONAL UNIVERSITY BANGLADESH (AIUB)**

**FACULTY OF SCIENCE & TECHNOLOGY**

A picture containing calendar

Description automatically generated

Course Title

**INTRODUCTION TO DATABASE (2108)**

**Semester: Spring 23-24  
Section: [O]**

**TITLE**

**Student Information Management System**

**Supervised By**

**Md. Sajid Bin Faisal**

**Submitted By: Group no: 06**

|  |  |
| --- | --- |
| **Name** | **ID** |
| FAHAD HASSAN FAHIM | 23-50940-1 |
| MD. TARIQUL ISLAM SESIR | 23-51017-1 |
| MD SHAHRIAR LASKHER | 23-51207-1 |
| MEHRAB IBNE KHALED | 22-49524-3 |

**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** | **6-14** |
| **5.** | **Finalization** | **15** |
| **6.** | **Table Creation** | **16-22** |
| **7.** | **Data Insertion** | **23-29** |
| **8.** | **Query Test** | **30-34** |
| **9.** | **DB connection** | **35** |
|  |  |  |
|  |  |  |

# **Introduction**

Our student management system project aims to revolutionize administrative processes in educational institutions by providing a centralized platform for storing and managing student data efficiently. Leveraging SQL for database management ensures robust data organization and retrieval.

**Tools:**

We utilize SQL for database management, ensuring efficient organization and querying of student data.

**Our Vision:**

Our goal is to simplify administrative tasks, enhance data accuracy, and improve communication within educational institutions. We aim for widespread adoption globally, leading to more efficient operations and better educational outcomes.

**Contribution to the Tech Sector:**

Through this project, we showcase the power of SQL in educational technology, providing a practical solution to administrative challenges and contributing valuable resources to the tech community.

**Beneficiaries and Target Users:**

Our project benefits administrators, teachers, students, and parents by streamlining administrative processes and improving communication channels within educational institutions of all levels.

# **Case Study / Scenario**

|  |  |  |
| --- | --- | --- |
| StudentID1: 23-50940-1  Name: FAHAD HASSAN FAHIM | StudentID3: 23-50940-1 Name: MD SHAHRIAR LASKHER | |
| StudentID2: 23-51017-1  Name: MD. TARIQUL ISLAM SESIR | StudentID4:22-49524-3 Name: MEHRAB IBNE KHALED | |
| **CO2**: Understand the fundamental concepts underlying database systems and gain hands-on experience with ER diagram Case study | | |
| **PO-c2:** Develop process for complex computer science and engineering problems considering cultural and societal factors. | | Marks |

In a Student Information Management System, a student may take multiple courses. A course can be taken by many students. A student is identified by Student ID. The system also stores information about the student's name, phone number, email, date of birth and address. Address contains city and country. A department is identified by department id and department name. A faculty is identified by faculty id. Faculty name and email are also stored. A department contains many faculties, but a faculty is from only one department. A student is in only one department, but A department has a large number of students. Students enroll in courses and faculties take courses. A student can enroll in many courses and a course can be enrolled by many students. A faculty can take multiple courses and a course can be taken by many faculties. A course is identified by course id and the system stores information about the course name and credit hours. A student has semester information. Semester Information is identified by Information Serial. The system also stores information about results and semester name. Result contains CGPA and grade. A student can have many semesters information and semester information can be of many students.

**ER Diagram**

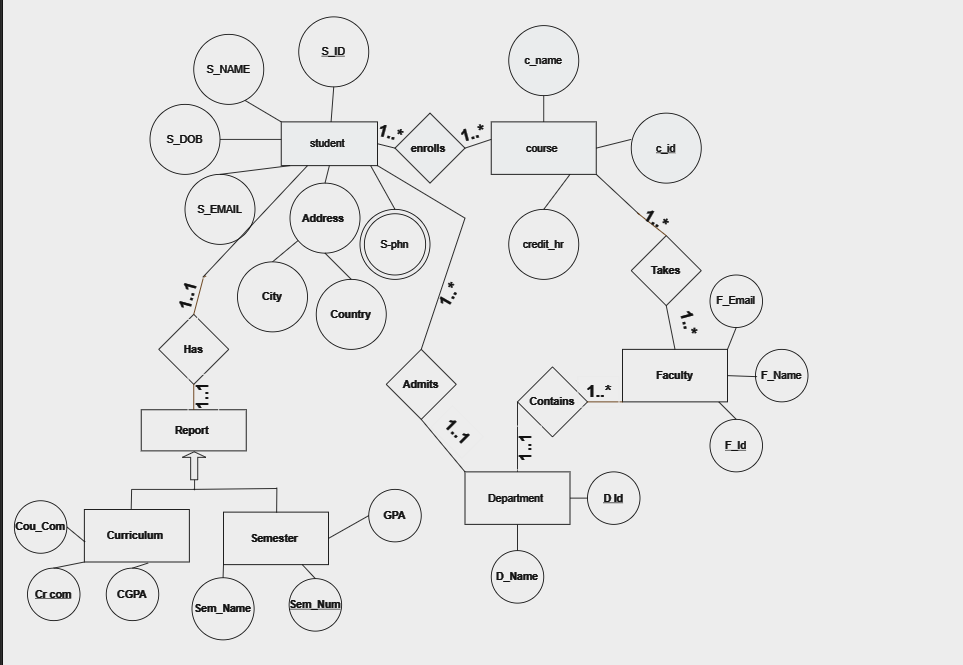


Fig: ER Diagram

# **Normalization**

## Enroll:

UNF:

credit\_hour, c\_id, c\_name, s\_id, s\_name, s\_DOB, s\_email, address , country, city, s\_phone

1NF:

credit\_hour, c\_id, c\_name, s\_id, s\_name, s\_DOB, s\_email, country, city, s\_phone

2NF:

1. credit\_hour, c\_id, c\_name
2. s\_id, s\_name, s\_DOB, country, city, s\_phone, s\_email
3. s\_id (P.K), c\_id (F.K)

3NF:

1. credit\_hour, c\_id, c\_name
2. s\_id, s\_name, s\_DOB, city, s\_phone, s\_email
3. s\_id(P.K), c\_id (F.K)
4. country, city

## Takes:

UNF:

credit\_hour, c\_id, c\_name, f\_email, f\_name, f\_id

1NF:

credit\_hour, c\_id, c\_name, f\_email, f\_name, f\_id

2NF:

1. credit\_hour, c\_id, c\_name

2. f\_email, f\_name, f\_id

3. f\_id (P.K), c\_id (F.K)

3NF:

3NF will be the same as 2NF.

## Contains:

UNF:

d\_name, d\_id, f\_name, f\_id, f\_email

1NF:

d\_name, d\_id, f\_name, f\_id, f\_email

2NF:

1. d\_name, d\_id
2. f\_name, f\_id, f\_email, d\_id (F.K)

3NF:

3NF will be the same as 2NF.

## Admits:

UNF:

d\_name, d\_id, s\_name, s\_DOB, s\_phone, s\_id, s\_email, address, country, city

1NF:

d\_name, d\_id, s\_name, s\_DOB, s\_phone, s\_id, s\_email, country, city

2NF:

1. d\_name, d\_id
2. s\_name, s\_DOB, s\_phone, s\_id, s\_email, country, city, d\_id (F.K)

3NF:

1. d\_name, d\_id
2. s\_name, s\_DOB, s\_phone, s\_id, s\_email, city, d\_id (F.K)
3. country, city

## Has:

UNF:

s\_id, s\_name, s\_DOB, s\_phone, s\_email, address, country, city, course\_com, cr\_com, CGPA

1NF:

s\_id, s\_name, s\_DOB, s\_phone, s\_email, country, city, course\_com, cr\_com, CGPA

2NF:

1. s\_id, s\_name, s\_DOB, s\_phone, s\_email, country, city

2. course\_com, cr\_com , CGPA

3. S\_id (P.K), cr\_com (F.K)

3NF:

1. s\_id, s\_name, s\_DOB, s\_phone, s\_email, city

2. course\_com, cr\_com , CGPA

3. S\_id (P.K), cr\_com (F.K)

4. country, city

Hold:

UNF:

s\_id, s\_name, s\_DOB, s\_phone, s\_email, address, country, city, sem\_no, sem\_name, GPA

1NF:

s\_id, s\_name, s\_DOB, s\_phone, s\_email, country, city, sem\_no, sem\_name, GPA

2NF:

1. s\_id, s\_name, s\_DOB, s\_phone, s\_email, country, city
2. sem\_no, sem\_name, GPA
3. S\_id (P.K), sem\_no (F.K)

3NF:

1. s\_id, s\_name, s\_DOB, s\_phone, s\_email, city
2. sem\_no, sem\_name, GPA
3. S\_id (P.K), sem\_no (F.K)
4. Country, city

# **Finalization**

Enroll:

1. credit\_hour, c\_id, c\_name [Course]
2. s\_id, s\_name, s\_DOB, city, s\_phone, s\_email [Student]
3. s\_id(P.K), c\_id (F.K) [Enroll]
4. country, city [Address]

Takes:

1. ~~Credit\_hour, c\_id, c\_name~~
2. f\_email, f\_name, f\_id [faculty]
3. f\_id (P.K), c\_id (F.K) [Take]

Contains:

1. d\_name, d\_id [Department]
2. f\_name, f\_id, f\_email, d\_id (F.K) [Contain]

Admit:

1. ~~d\_name, d\_id~~
2. s\_name, s\_DOB, s\_phone, s\_id, s\_email, d\_id (F.K)[Admit]
3. ~~country, city~~

Has:

~~1. s\_id, s\_name, s\_DOB, s\_phone, s\_email, city~~

2. course\_com, cr\_com , CGPA [Curriculum]

3. s\_id (P.K), cr\_com (F.K) [Has]

~~4. country, city~~

Hold:

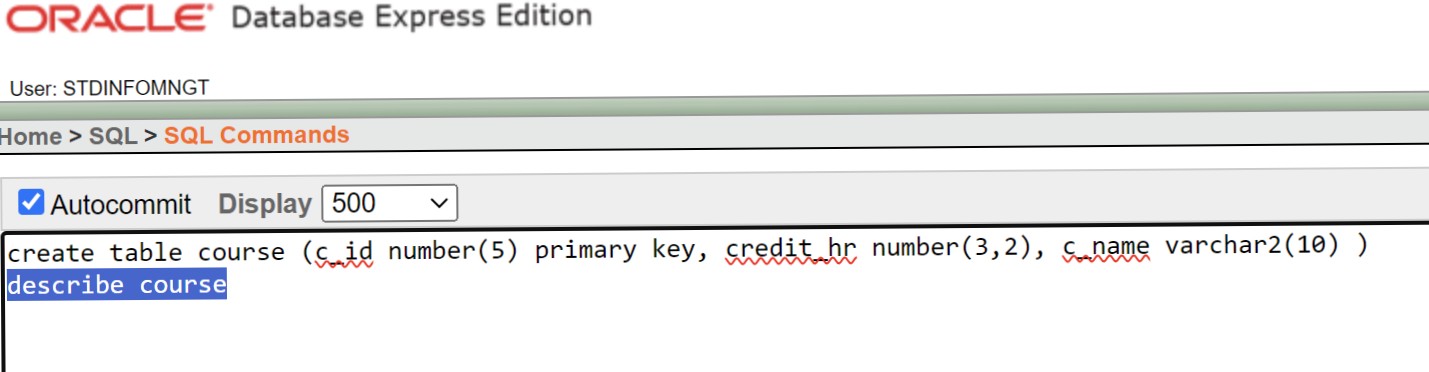
1. ~~s\_id, s\_name, s\_DOB, s\_phone, s\_email, city~~
2. sem\_no, sem\_name, GPA [Semester]
3. s\_id (P.K), sem\_no (F.K) [Hold]
4. ~~Country, city~~

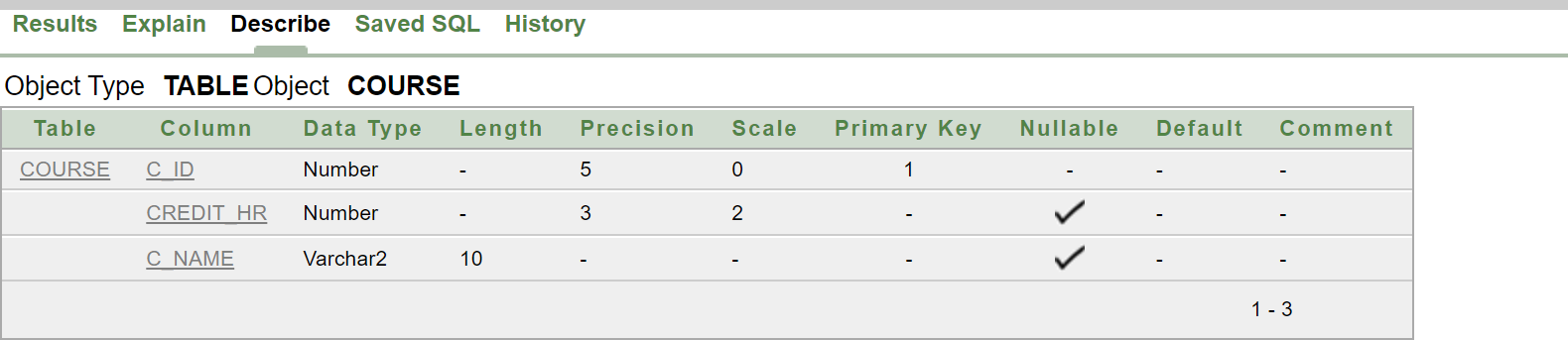
**Final Table**

|  |
| --- |
| 1. credit\_hour, c\_id, c\_name [Course] |
| 1. s\_id, s\_name, s\_DOB, city, s\_phone, s\_email [Student] |
| 1. s\_id(P.K), c\_id (F.K) [Enroll] |
| 1. country, city [Address] |
| 1. f\_email, f\_name, f\_id [faculty] |
| 1. f\_id (P.K), c\_id (F.K) |
| 1. d\_name, d\_id [Department] |
| 1. f\_name, f\_id, f\_email, d\_id (F.K) [Contain] |
| 1. s\_name, s\_DOB, s\_phone, s\_id, s\_email, d\_id (F.K)[Admit] |
| 10. course\_com, cr\_com , CGPA [Curriculum] |
| 11. s\_id (P.K), cr\_com (F.K) [Has] |
| 12. sem\_no, sem\_name, GPA [Semester] |
| 1. s\_id (P.K), sem\_no (F.K) [Hold] |

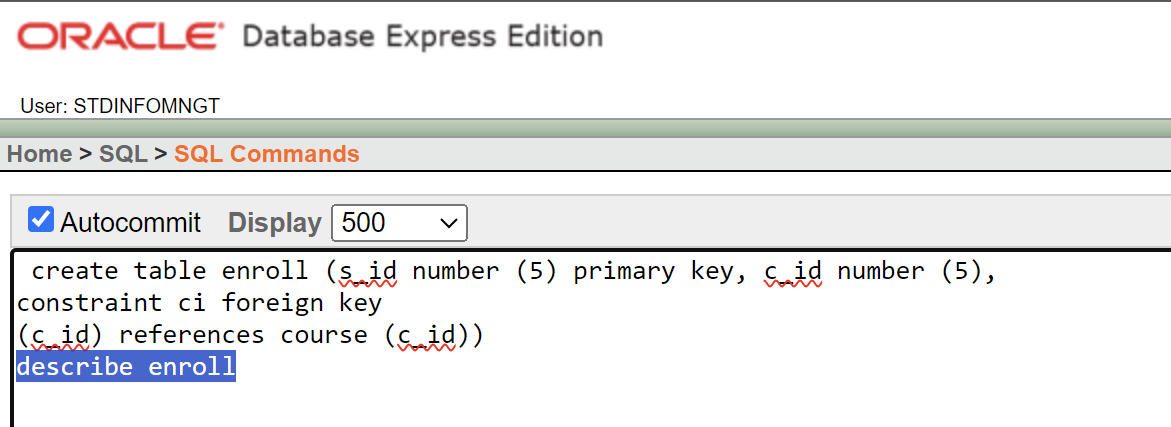
# **Table Creation (DDL Operations)**

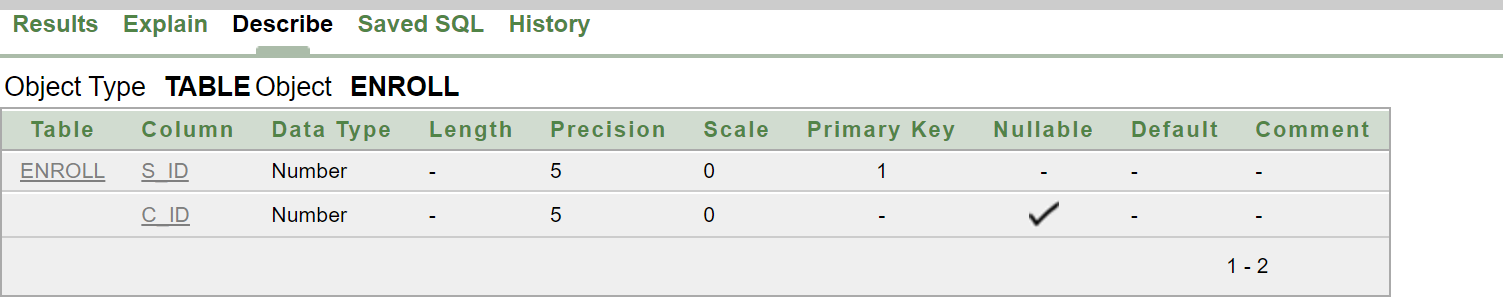
|  |  |  |
| --- | --- | --- |
| StudentID1: 23-50940-1 Name: Fahad Hassan Fahim | StudentID3: Name: | |
| StudentID2: Name: | StudentID4: Name: | |
| **CO4**: Creating DML, DDL using Oracle and connection with ODBC/JDBC for existing JAVA application | | |
| **PO-e-2:** Use modern engineering and IT tools for prediction and modeling of complex computer science and engineering problem | | Marks |

Course:

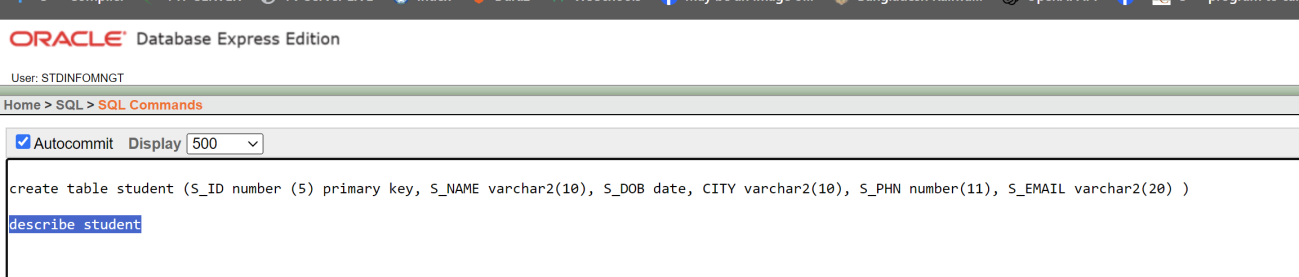


**Fig: Course Table Creation**

Enroll:



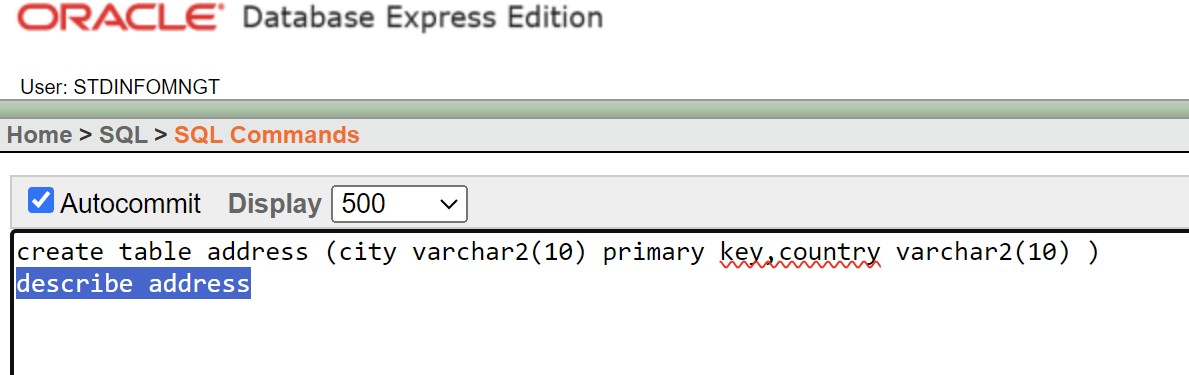
**Fig: Enroll Table Creation**

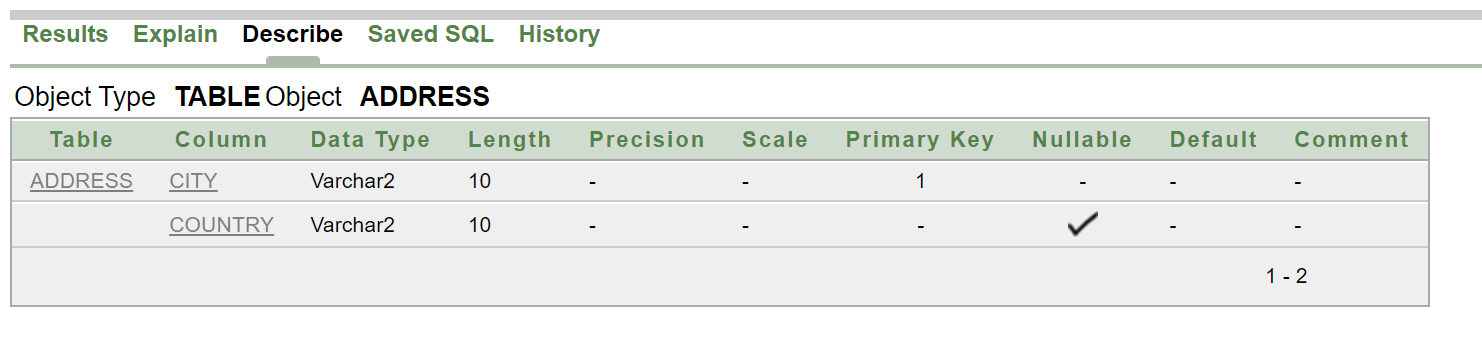
Student:



**Fig: Student Table Creation**

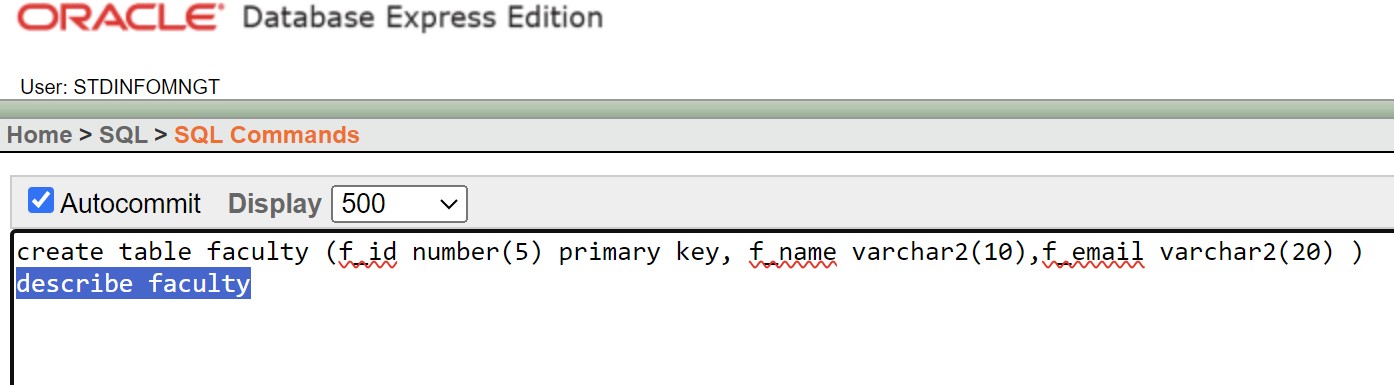
Address:

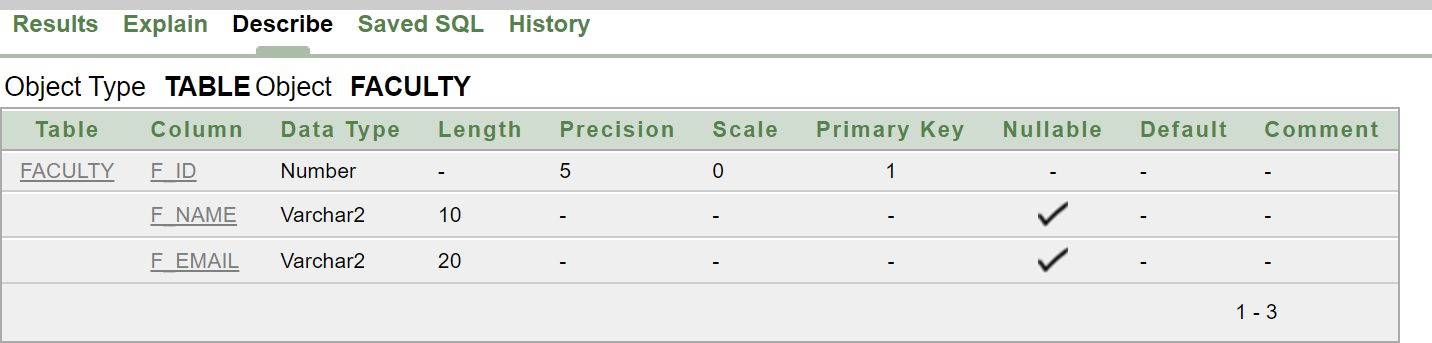


****

**Fig: Adress Table Creation**

Faculty:



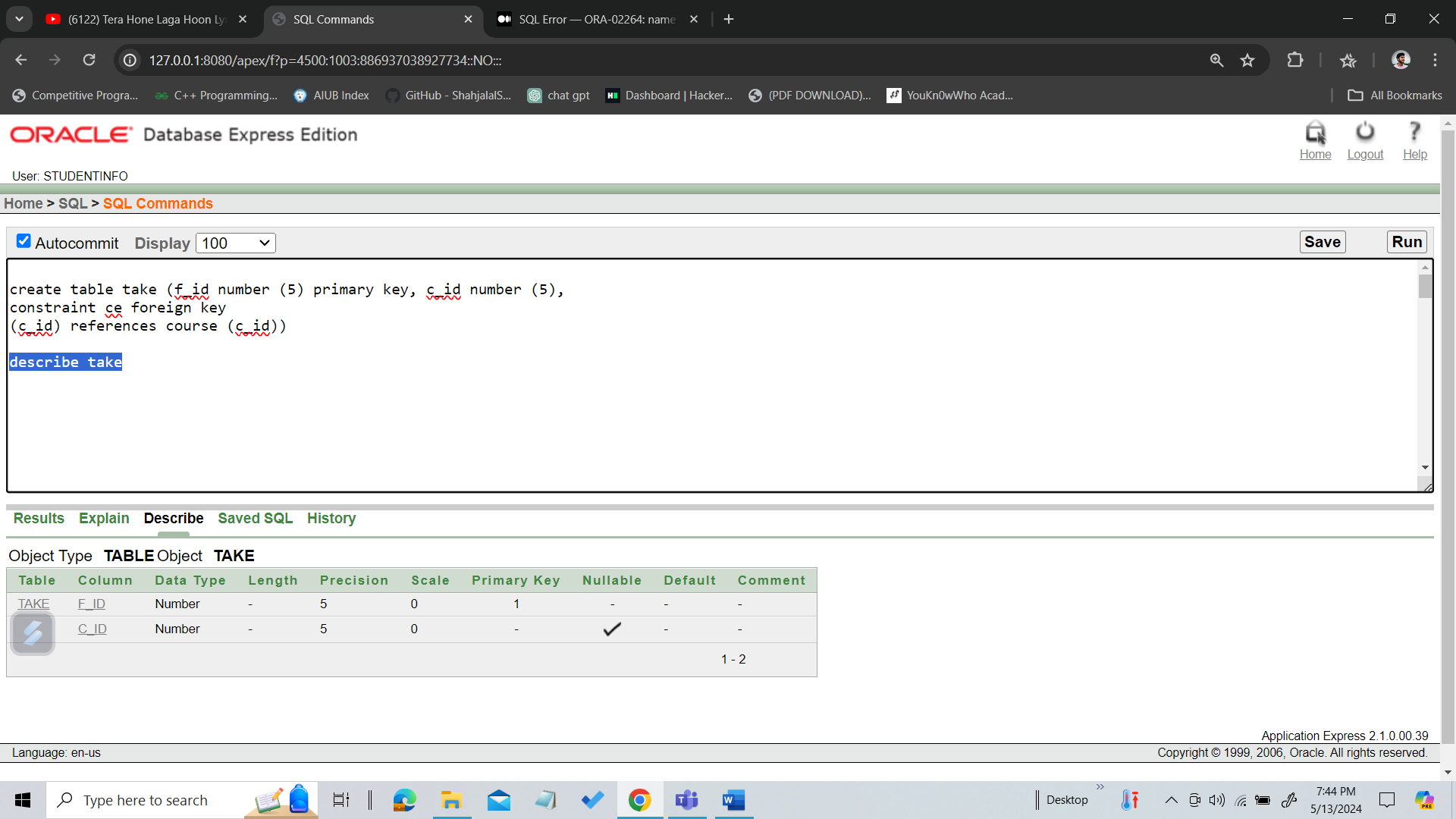


# **Inserted Values in the tables**

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

**Fig: Faculty Table Creation**

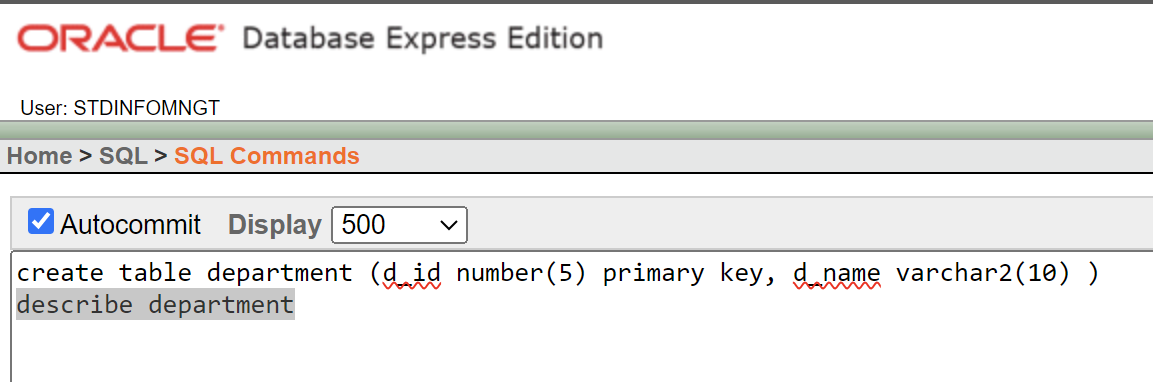
Take:

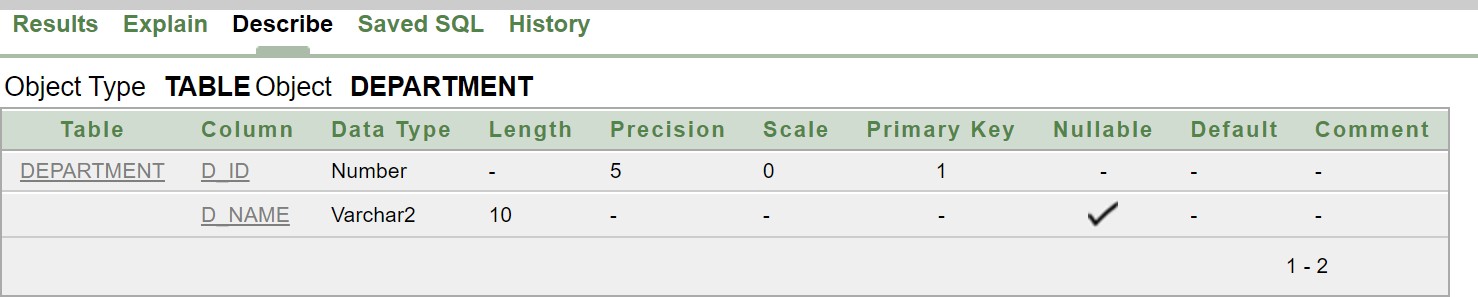


A screenshot of a computer

Description automatically generated

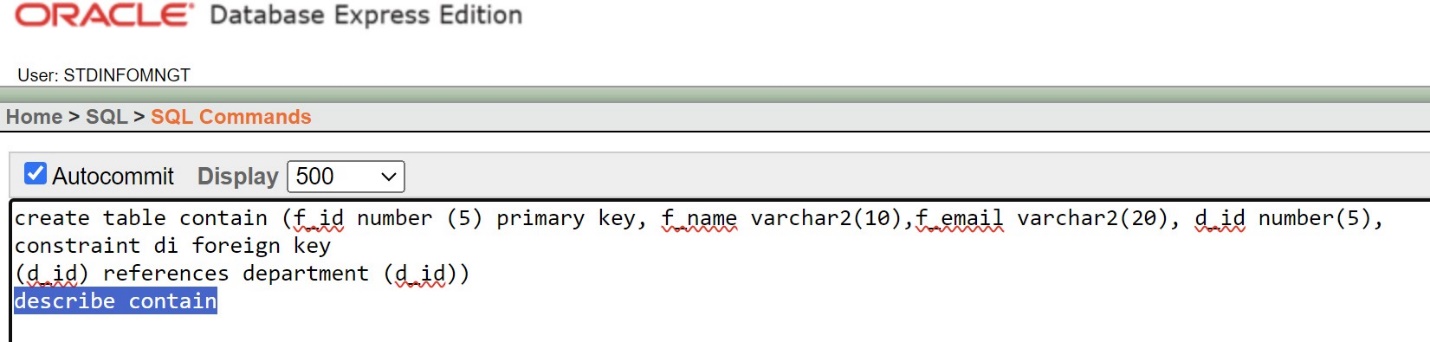
**Fig: Take Table Creation**

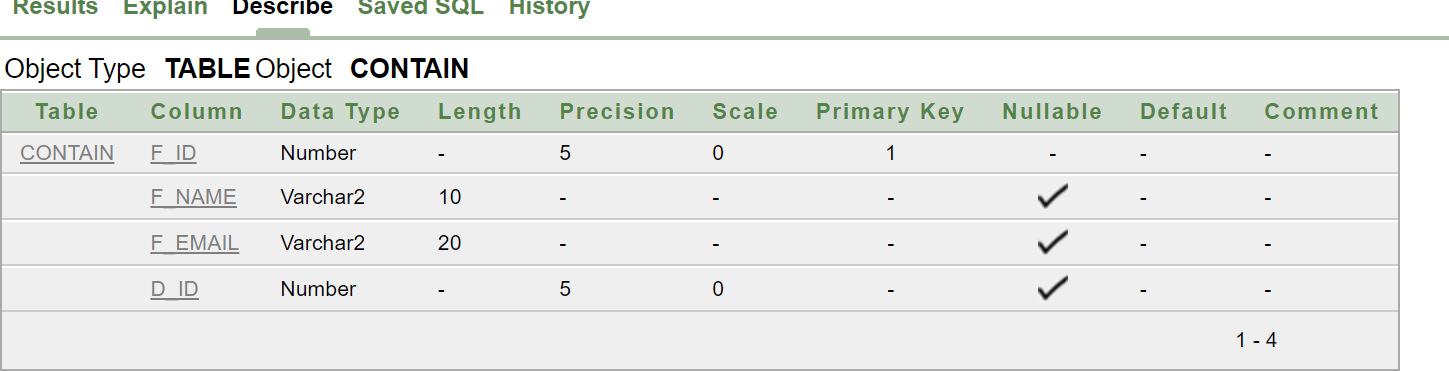
Department:



**Fig: Department Table Creation**

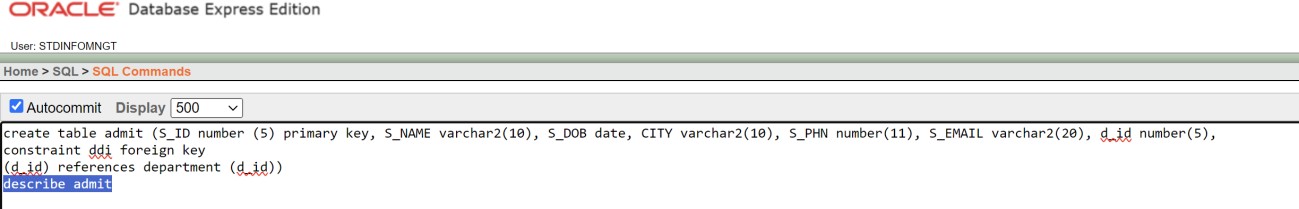
Contain:

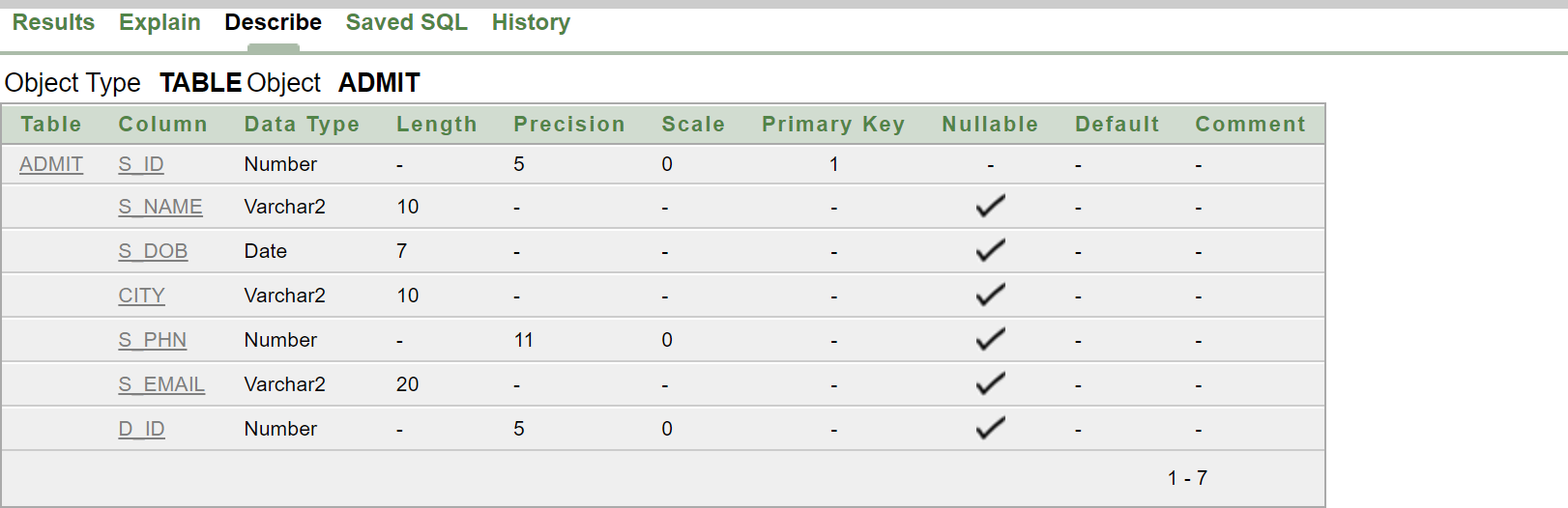




**Fig: Contain Table Creation**

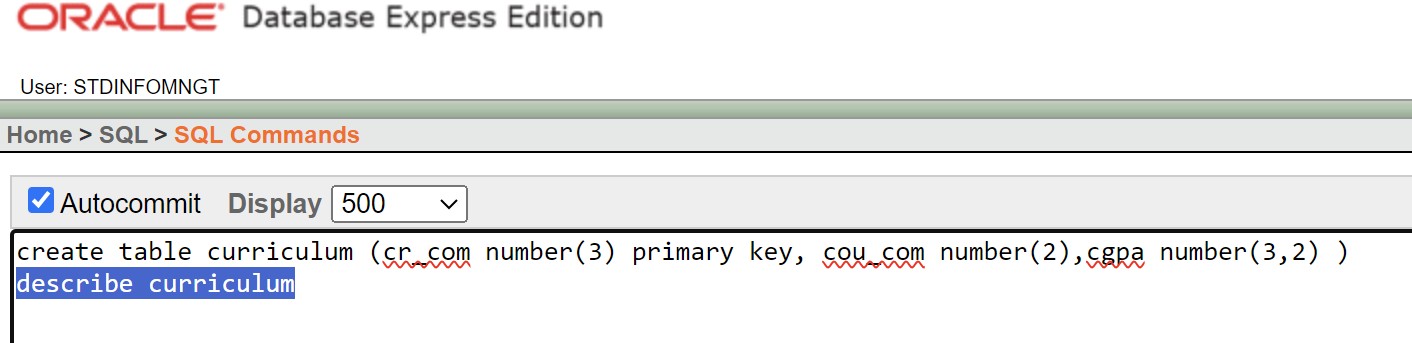
Admit:

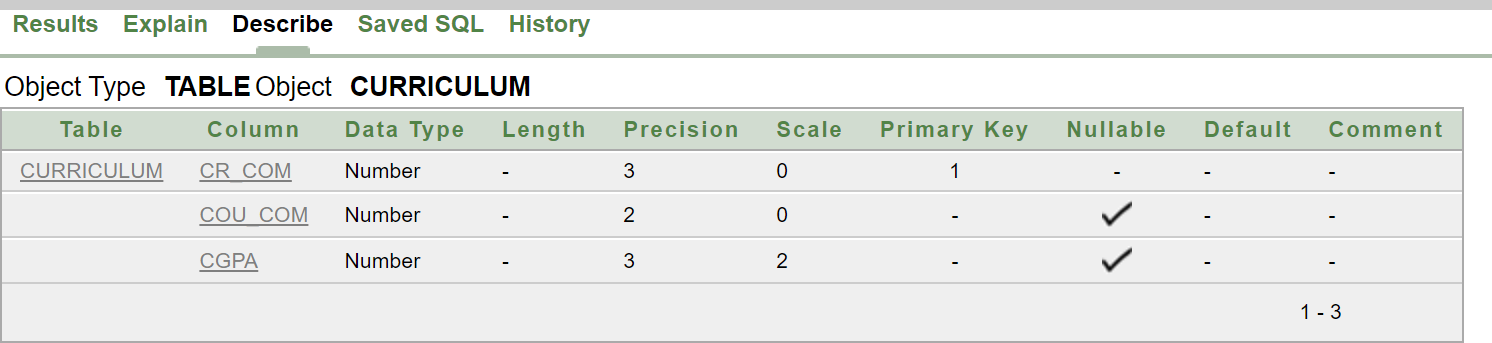




**Fig: Admit Table Creation**

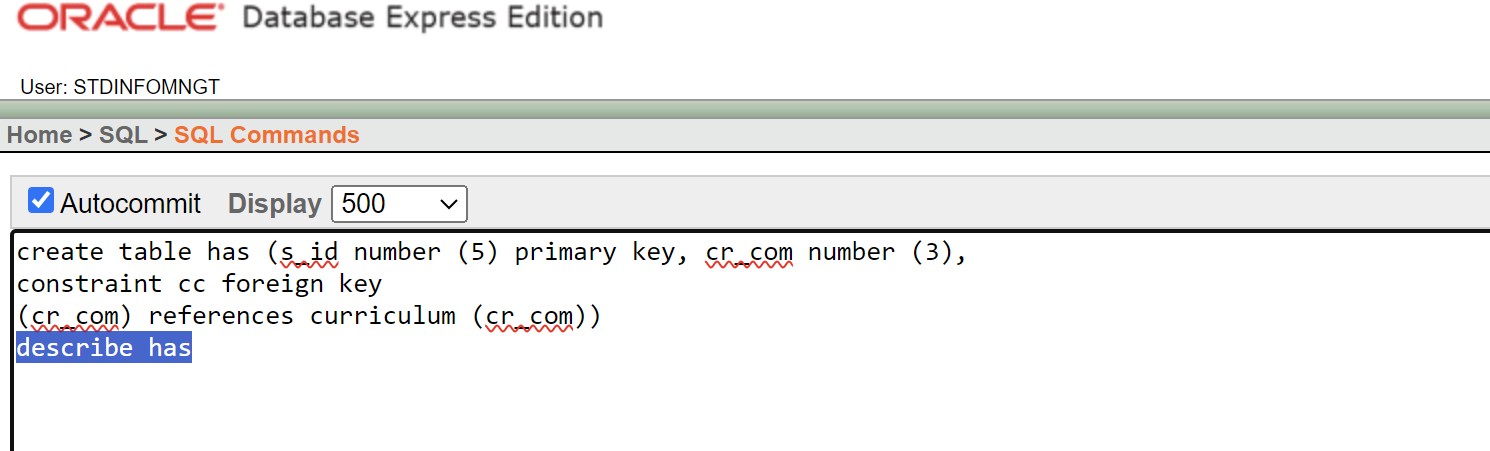
Curriculum:





**Fig: Curriculum Table Creation**

Has:



A screenshot of a computer

Description automatically generated

**Fig: Has Table Creation**

Semester:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Fig: Semester Table Creation**

A screenshot of a computer

Description automatically generatedHold:

A screenshot of a computer

Description automatically generated

**Fig: Hold Table Creation**

# **Inserted Values in the tables**

Course:

A screenshot of a computer

Description automatically generated

**Fig: Course Table Insertion**

Student:

A screenshot of a computer

Description automatically generated

**Fig: Student Table Insertion**

Enroll:

**A screenshot of a computer

Description automatically generated**

**Fig: Enroll Table Insertion**

Address:

A screenshot of a computer

Description automatically generated

**Fig: Address Table Insertion**

Faculty:

A screenshot of a computer

Description automatically generated

**Fig: Faculty Table Insertion**

Take:

A screenshot of a computer

Description automatically generated

**Fig: Take Table Insertion**

Department:

A screenshot of a computer

Description automatically generated

**Fig: Department Table Insertion**

Contain:

A screenshot of a computer

Description automatically generated

**Fig: Contain Table Insertion**

Admit:

**A screenshot of a data

Description automatically generated**

**Fig: Admit Table Insertion**

Curriculum:

A screenshot of a computer

Description automatically generated

**Fig: Curriculum Table Insertion**

Has:

A screenshot of a computer

Description automatically generated

**Fig: Has Table Insertion**

Semester:

A screenshot of a computer

Description automatically generated

**Fig: Semester Table Insertion**

Hold:

A screenshot of a computer

Description automatically generated

**Fig: Hold Table Insertion**

# **Query Test in DB**

**1. Simple Query:**

Question: Show the student name, address and date of birth from the student table.

A screenshot of a computer

Description automatically generated

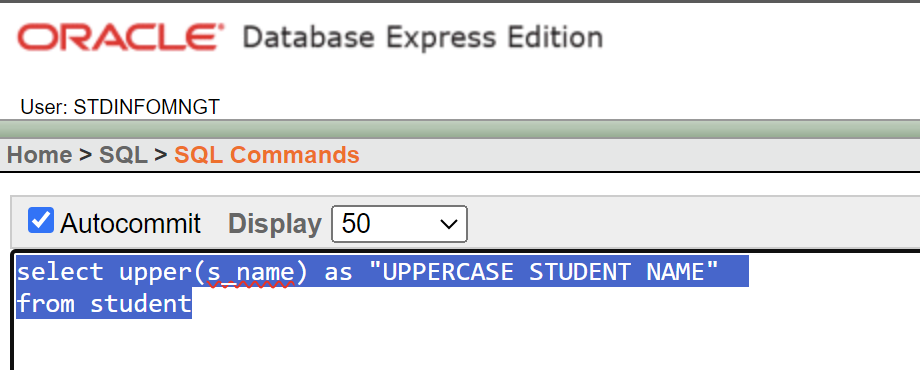
A screenshot of a computer

Description automatically generated

**Fig: Simple Query**

**2.Single Row Function:**

**Question:** Show the name of students (in upper case) from the student table**.**



A screenshot of a computer

Description automatically generated

**Question:** Show the faculty ID, faculty E-mail where faculty name is sajid.

A screenshot of a computer

Description automatically generated

A screenshot of a email

Description automatically generated

**Fig: Single Row Function**

**3.Group Function:**

**Question:** Show the average salary of GPA from the semester table.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Fig: Group Function**

**4. Single Row Subquery:**

**Question:** Show name, D-O-B and city for the student whose has completed 43 credit

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Fig: Single Row Subquery**

**5.Multiple Row Subquery:**

**Question:** show name, Phone number and email for the student whose has completed more than 4 semesters

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Fig: Multiple Row Subquery**

**5.Simple View:**

Create a view from faculty table where faculty id is 12

**Permission:** grant create view to stdinfomngt

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a email

Description automatically generated

**Fig: Simple View**

**6.Complex View:**

**Question:** Create a view from the curriculum table where it shows average CGPA.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Fig: Complex View**

**7. Equi-Join:**

Question: Write a SQL query to show student ID and their completed credit.

A screenshot of a computer

Description automatically generated

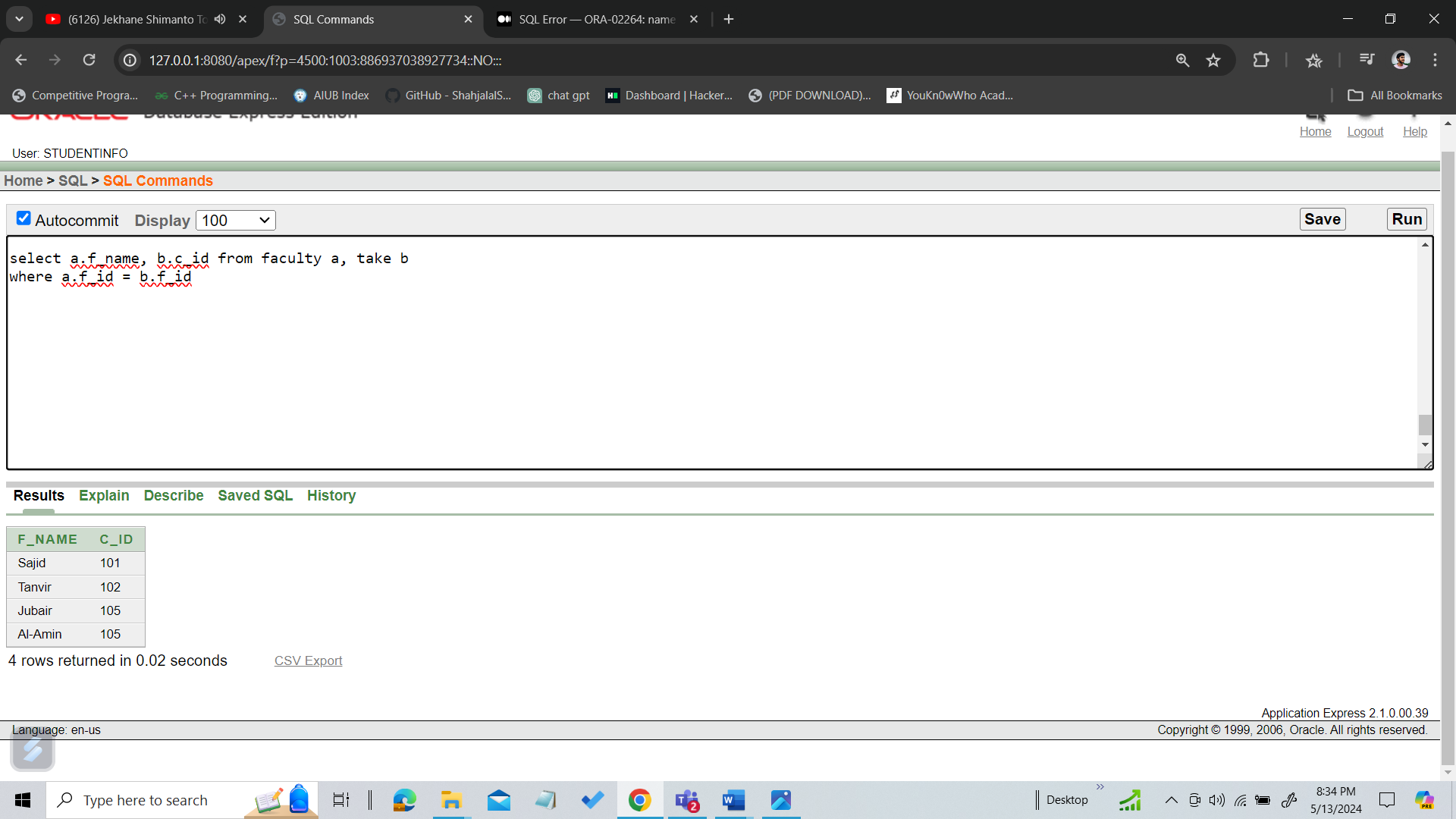
A screenshot of a computer

Description automatically generated

**Fig: Equi-Join**

**8.Non-Equijoin:**

Question: Write a SQL query to show faculty name and their course id.



A screenshot of a computer

Description automatically generated

**Fig: Non Equi-Join**

# **Description of a Successful DB connection**

**DataBase Connection  by  (Sesir, Md Tariqul Islam….23-51017-1)**

If we don't already have it installed, we must first install a few items on our device in order for the database connection to function.

The first thing I did was download and install MySQL-connector-java 8.0.28 together with **Xampp** server.

I launched the MySQL module, Apache, and phpMyAdmin after opening the **Xampp** server from the Xampp control panel. I made a table called **"Faculty"** with three columns there, as well as a database named "**stdinfomngt."** The values that follow are then added to the table, which has three rows total.

A screenshot of a computer

Description automatically generated

**Fig 1:   table create in MySQL server**

A screenshot of a computer

Description automatically generated

**Fig 2:  Output of the table from the MySQL server**

**3) Writing the java code**

Chosen an Integrated Development Environment (IDE) of my choice. I used **visual studio** code for Java development. In my Java code, loaded the MySQL Java Connector driver by importing the JAR file into my project and using the Class.forName() method to register it. Used the DriverManager.getConnection() method to establish a connection to my MySQL database by providing the appropriate URL, username, and password. My username was root and there was no password set. After establishing the connection, created a Statement or Created Statement object to execute SQL queries. Used the created statement to execute SQL queries like SELECT, INSERT, UPDATE, or DELETE. Captured the necessary results. Used the ResultSet object to retrieve and process the data. Performed necessary operations on the data retrieved from the database.

A screenshot of a computer program

Description automatically generated

**Fig 3: code in vs code for java and sql connection**

A screen shot of a computer

Description automatically generated

**Fig 4:  output from vs code after connect with my sql connector java**

**(Md. Shahriar Laskher , 23-51207-1)**

To make database connection work we first need to have few things install on our device if we do not have it already.

* First I installed Xampp server and downloaded mtsql-connector-java-8.0.28.

* I opened Xampp server from Xampp control panel then started Apache and MySQL module then went to phpMyAdmin. There I created a Database by name “stdinfomngt” and created a table that name “Student” which contains six column. The subsequent values are then inserted into the table comprising three rows in total.

A screenshot of a computer

Description automatically generated

**Fig: 1 Output of the table from the mysql server**

Writing the java code:

1. Selected Visual Studio Code as the IDE for Java development.
2. Imported the MySQL Java Connector JAR file into the project.
3. Used Class.forName() method to register the MySQL Java Connector driver.
4. Established a connection to the MySQL database using DriverManager.getConnection() method.
5. Provided the appropriate URL, username (root), and no password for the database connection.
6. Created a Statement object to execute SQL queries (SELECT, INSERT, UPDATE, DELETE).
7. Executed SQL queries using the created Statement object.
8. Captured the results of the queries using ResultSet object.
9. Retrieved and processed the data from the ResultSet as needed.
10. Performed necessary operations on the retrieved data from the database.



Fig 2:code in vs code for java and sql connection

A computer screen shot of a program code

Description automatically generated

Fig 4:  output from vs code after connect with my sql connector java

**(Fahad Hassan Fahim, 23-50940-1)**

**1.**The mysql connector (jar file) is installed at first. For that, mysql java connector maven containing jar file of version 8.0.28 is downloaded.

**2.**The xampp apache perl php mysql xampp server id downloaded which contain many servers. Among that apache and mysql server is started and admin panel of mysql server is opened.

**3.**After that in that panel a database system is created namely “stdinfomngt”, and a table name “course” which contains three columns. The subsequent values are then inserted into the table comprising three rows in total.

The table in mysql :

A screenshot of a computer

Description automatically generated

                                   Fig 9.1.3: Output of the table from the mysql server

**(Mehrab Ibne Khaled, 22-49524-3)**

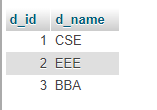
For Database Connection, We have to install some programs first.

Installing the required programs:

* I installed the MySql Connector from “mvnrepository” (8.0.33) first.
* For database connection as IDE I used VSCode.
* Then I installed the XAMPP server and after installing the XAMPP server I opened the XAMPP control panel and started “Apache” and “MySql” together.

Creating a Table:

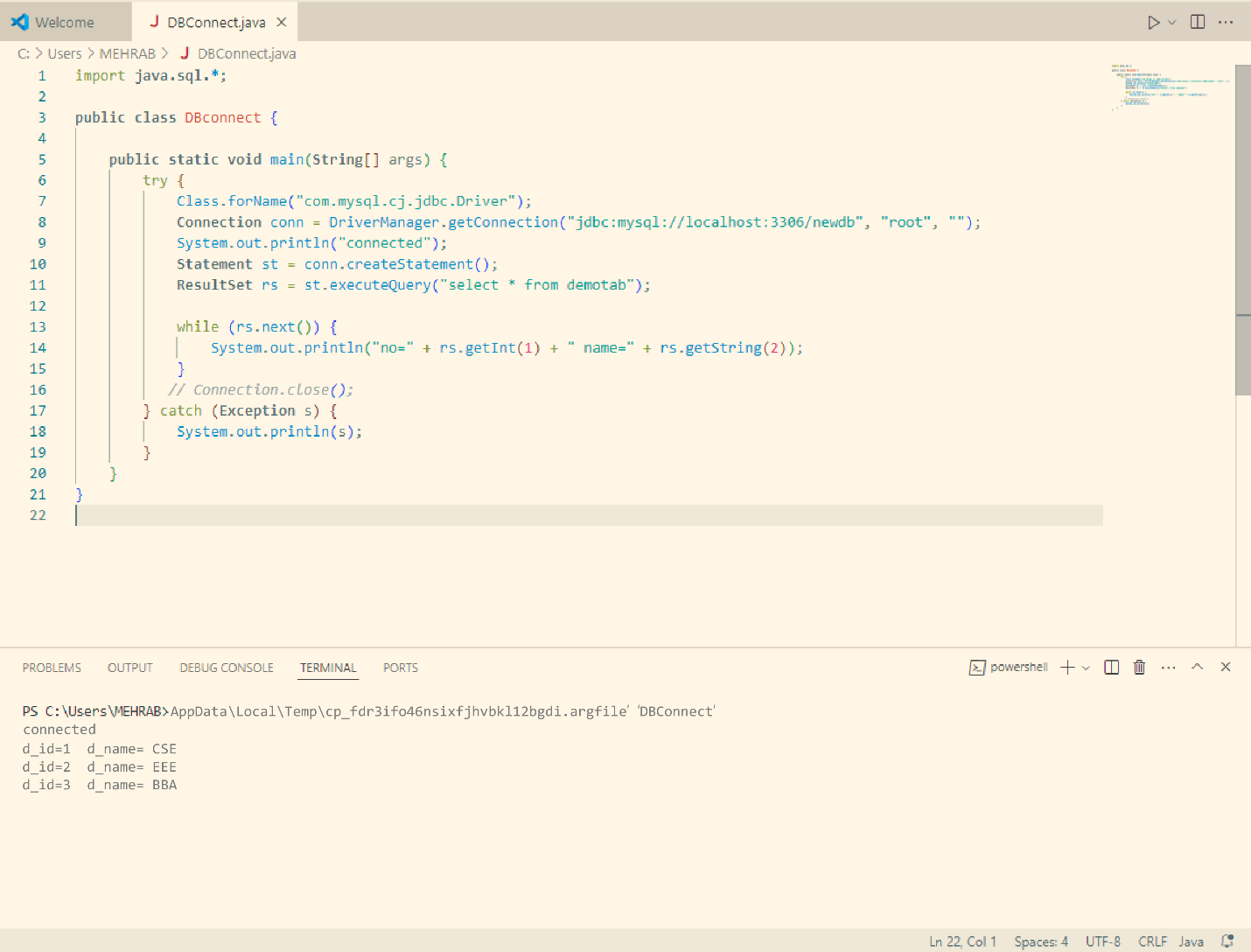
* After opening the control panel of XAMPP server, I went to the admin panel of “MySql”
* Then I created a database named “stdinfomngt”
* I created a new table then which is named “Dept” (department table) and which contains three columns and two rows.
* Then I inserted the value in the table.



**Fig: 1 Output of the table from the mysql server**

Connecting Java code:

* First I used VSCode as the IDE to write the Java code
* Then I imported MySql JAR file which I instaleed before
* For registration, I used ClassforName() to register Java Connection Driver
* Then I used DriverManager.getConnection() method to establish a connection to the MySql database
* I provided the required URL, username for the database connection (no password were needed)
* I created a Statement object ( select, insert, update, delete) for executing the queries
* Used the ResultSet object to retrieve and process the data
* Lastly performed necessary operations on the data retrieved from the database.



**Fig: 2 code in vs code for java and sql connection**

# 

**Fig: 3 output from vs code after connect with my sql connector java**

# **Conclusion**

**Future Aspects:**

In the future, we envision expanding the capabilities of our student management system to include features such as automated reporting, integration with learning management systems, and predictive analytics to identify at-risk students. Additionally, we aim to enhance the user experience by developing mobile applications for convenient access to student information on the go.

**Summary:**

Our student management system project introduces a centralized platform powered by SQL for efficient storage and management of student data in educational institutions. By streamlining administrative processes and improving communication channels, we aim to enhance operational efficiency and educational outcomes. With future enhancements planned, including automated reporting and predictive analytics, our system is poised to become an indispensable tool for educational institutions worldwide.