**TABLE OF CONTENTS**

**CONTENTS**  Page

TABLE OF CONTENTS

**CHAPTER 1: COURSE MANAGEMENT SYSTEM**

1.1 Introduction 1

1.2 Objectives 1

1.3 Tables1

**CHAPTER 2: DATA DICTIONARY OF SYSTEM**

2.1 Data Dictionary of System 2

**CHAPTER 3: ENTITY RELATIONSHIPS(E-R) DIAGRAM**

3.1 ER Diagram 3

**CHAPTER 4: TABLE SECTIONS**

4.1 Department Table 4

4.2 Instructor Table 5

4.3 Section Table 6

4.4 Student Table 6

4.5 Course Table 7

**CHAPTER 5: SQL STATEMENTS**

5.1 SQL Statements 8

**CHAPTER 6: CONCLUSION**

6.1 Conclusion 14

6.2 Future Work 14

**CHAPTER 1**

**INTRODUCTION**

* 1. **Introduction**

A course management system is a software designed to help course manage their academic and administrative operations more efficiently. It provides a centralized platform for managing student records, course schedules, faculty information and other important aspects of course operations.

With a course management system, administrators and faculty members can easily access and update student records, track course schedules and enrollment, and generate reports on academic performance. This system helps to streamline operations and reduce administrative burden, allowing faculty members to focus on teaching and research and students to focus on their studies.

* 1. **Objectives**

The objective of having a University Management System (UMS) is to efficiently and effectively manage the operations and resources of a university. Here as the objectives of implementing university DBMS:

The objectives of the system are-

* To reduce paperwork
* To Reduce operational time
* To Increase accuracy and reliability
* To Increase operational efficiency
* To secure data
* To maintain daily update on placement head
  1. **Tables**
* Department Table include department information.
* Instructor Table include instructor information.
* Student Table include student information.
* Course Table include course information.
* Section Table include section information.

**CHAPTER 2**

**DATA DICTIONARY OF SYSTEM**

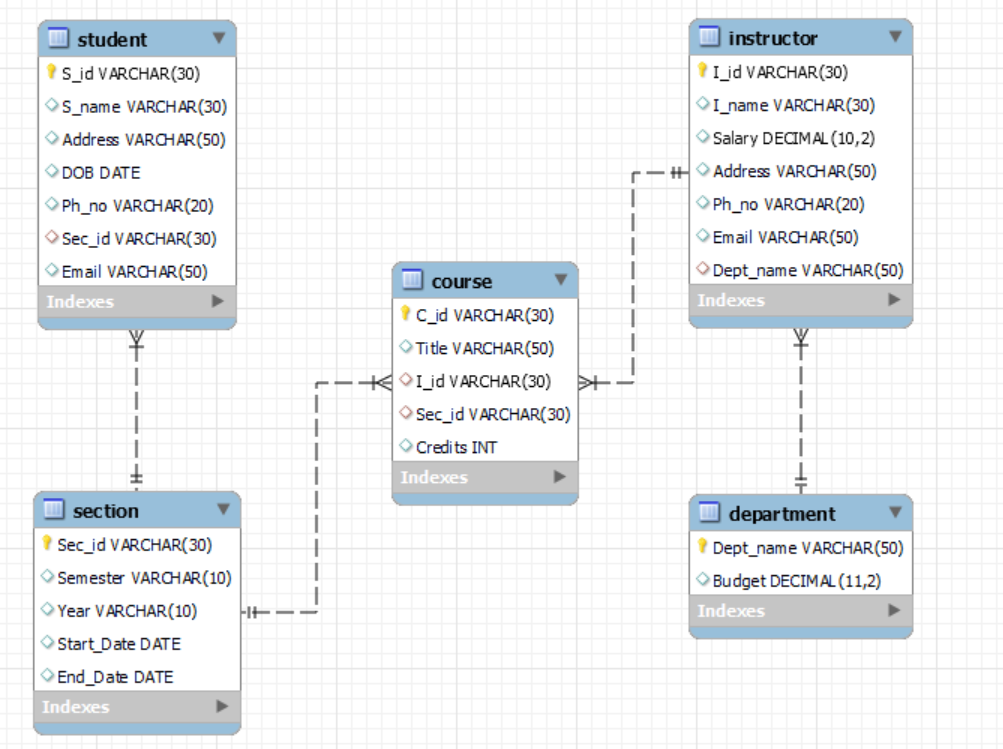
**2.1 Table Design**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table | Attributes | Data type and length | Require PK/FK | PK references tables |
| Department | Dept\_name | varchar(50) | PK |  |
| Budget | decimal(11,2) |  |  |
| Instructor | I\_id | varchar(30) | PK |  |
| I\_name | varchar(30) |  |  |
| Salary | decimal(10,2) |  |  |
| Addresss | varchar(50) |  |  |
| Ph\_no | varchar(20) |  |  |
| Email | varchar(30) |  |  |
| Dept\_name | varchar(50) | PK | Department |
| Course | C\_id | varchar(30) | PK |  |
| Title | varchar(50) |  |  |
| Sec\_id | varchar(30) | FK | Section |
| Credits | int |  |  |
| I\_id | varchar(30) | FK | Instructor |
| Section | Sec\_id | varchar(30) | PK |  |
| Semester | varchar(10) |  |  |
| Year | varchar(10) |  |  |
| State\_Date | date |  |  |
| End\_Date | date |  |  |
| Student | S\_id | varchar(30) | PK |  |
| S\_name | varchar(30) |  |  |
| Address | varchar(50) |  |  |
| DOB | date |  |  |
| Ph\_no | varchar(20) |  |  |
| Sec\_id | varchar(30) | FK | Section |

**CHAPTER 3**

**ENTITY-RELATIONSHIP DIAGRAM**

**3.1 ER Diagram**

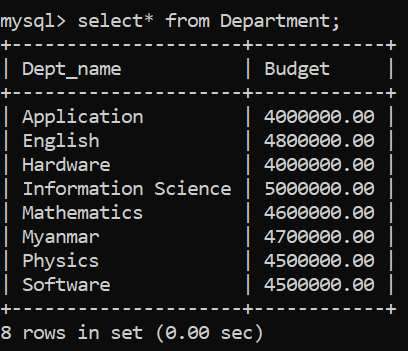
****

**CHAPTER 4**

**TABLE SECTION**

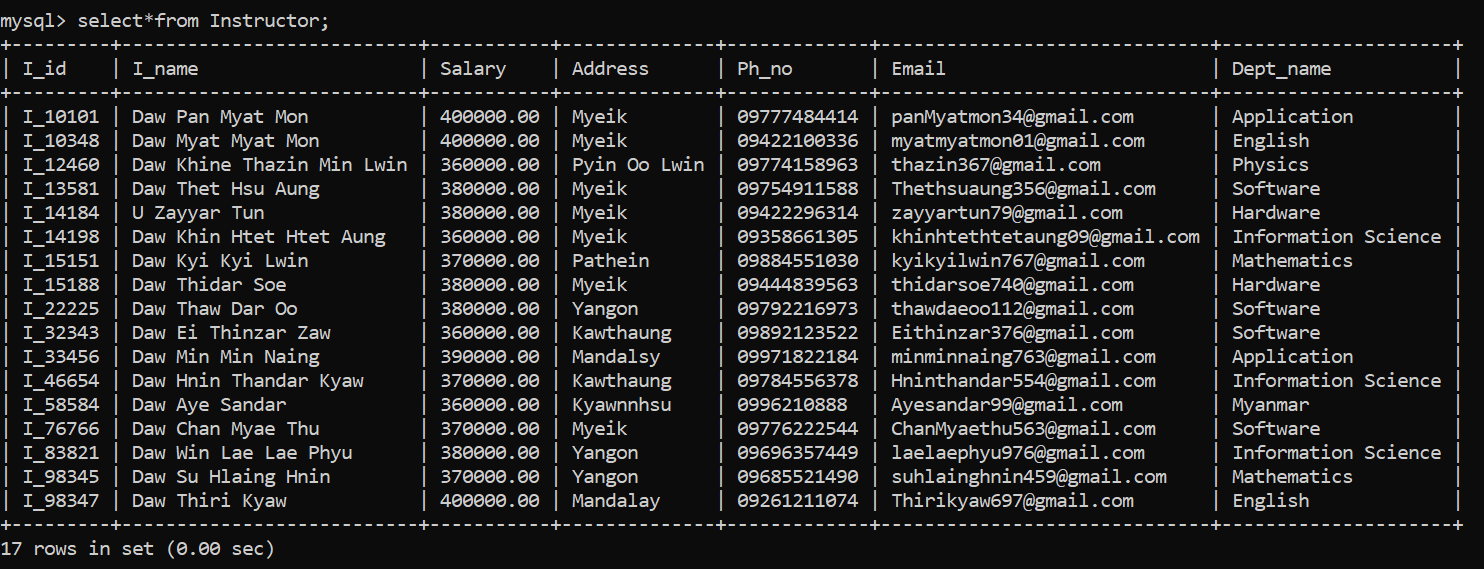
**4.1 Department Table**

create table Department(Dept\_name varchar(50),Budget numeric(11,2),primary key(Dept\_name));

****

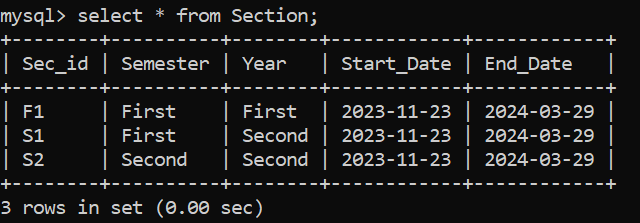
**4.2 Instructor Table**

create table Instructor(I\_id varchar(30),I\_name varchar(30),Salary numeric(10,2),Address varchar(50),Ph\_no varchar(20),Email varchar(50),Dept\_name varchar(50),primary key(I\_id),foreign key(Dept\_name) references Department(Dept\_name));



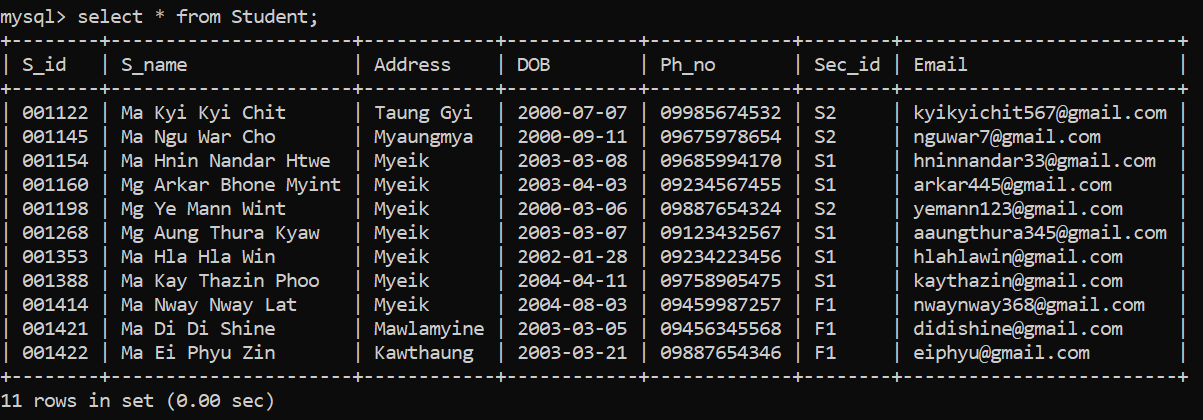
**4.3 Section Table**

create table Section(Sec\_id varchar(30),Semester varchar(10),Year varchar(10),Start\_Date date,End\_Date date,primary key(Sec\_id));



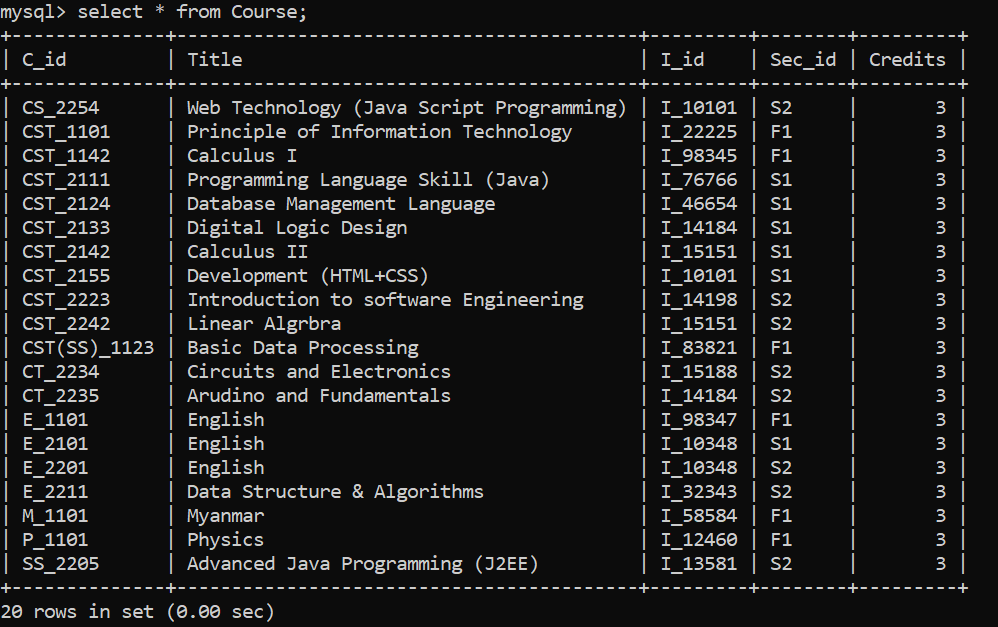
**4.4 Student Table**

create table Student(S\_id varchar(30),S\_name varchar(30),Address varchar(50),DOB date,Ph\_no varchar(20),Sec\_id varchar(30),primary key(S\_id),foreign key(Sec\_id) references Section(Sec\_id));



**4.5 Course Table**

create table Course(C\_id varchar(30),Title varchar(50),I\_id varchar(30),Sec\_id varchar(30),Credits int,primary key(C\_id),foreign key(I\_id) references Instructor(I\_id),foreign key(Sec\_id) references Section(Sec\_id));

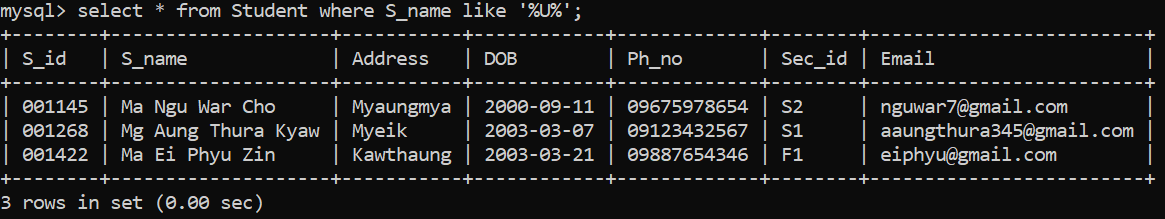


**CHAPTER 5**

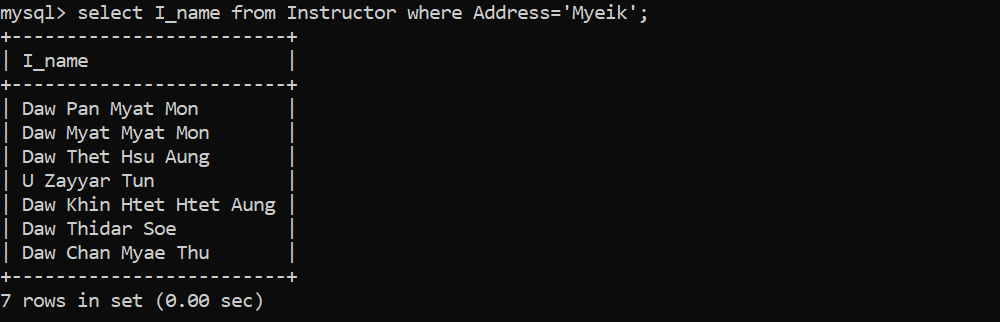
**SQL STATEMENT**

**5.1 SQL Statement**

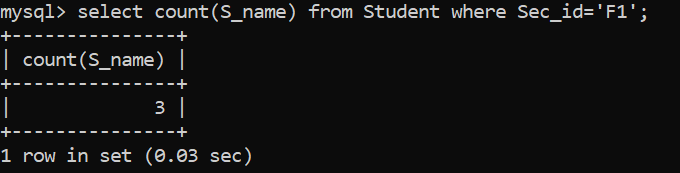
1.Retrieve all student’s information whose name with contain ‘U’.

****

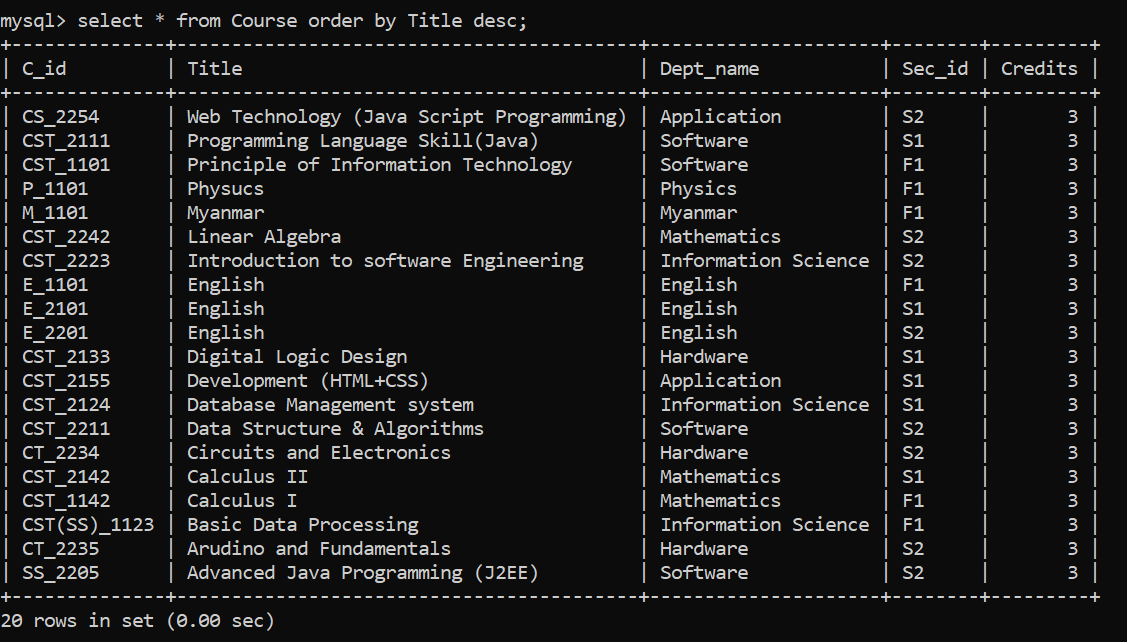
2. Retrieve instructor names that instructor’s address ‘Myeik’.

****

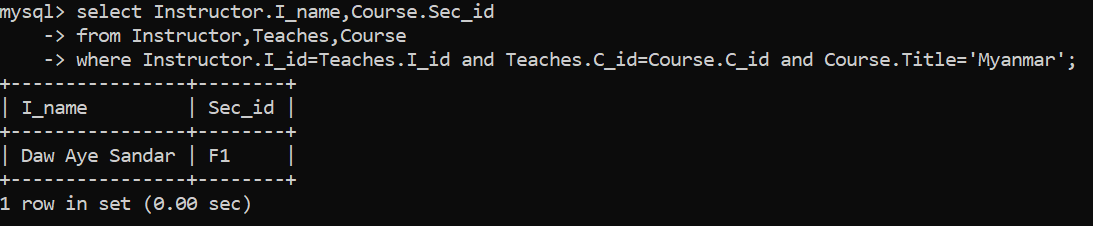
3. Get the total number of student in the section ‘F1’.

****

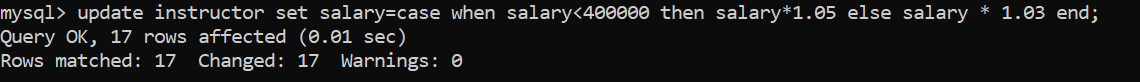
4. Retrieve title of course ordered by descending.

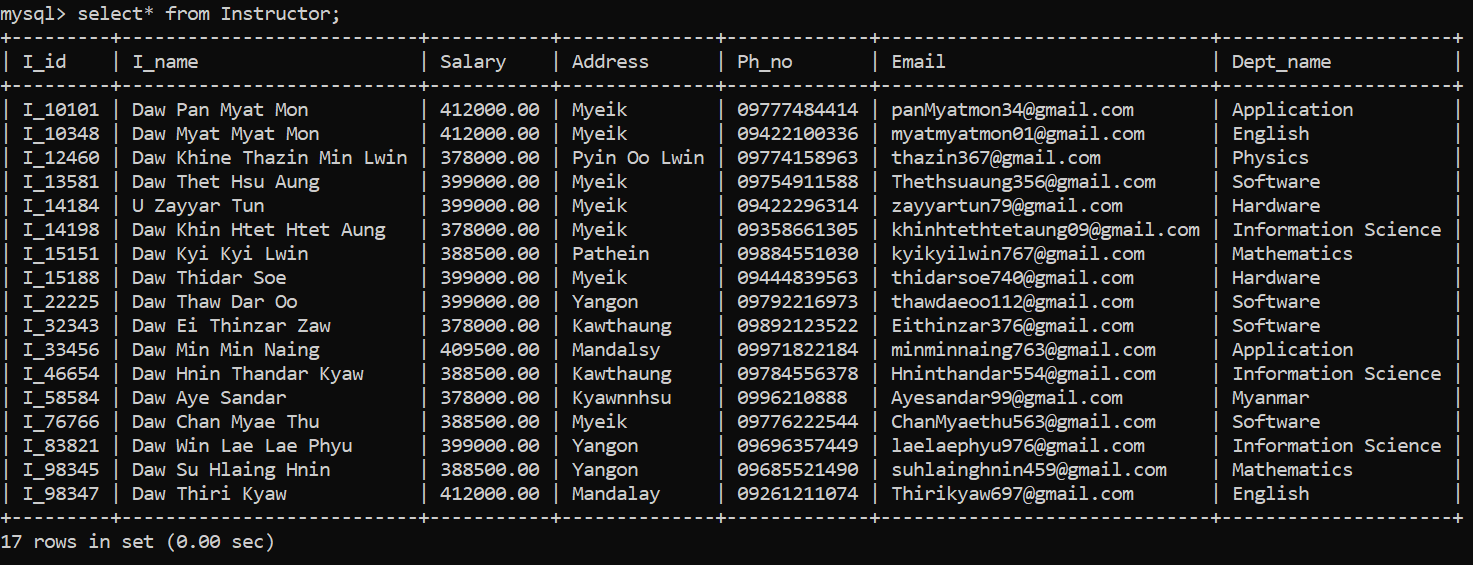
****

5. Find the section\_id and name of instructor who teaches course title Myanmar.

****

6. Increasing salary of instructor that salary is below 400000 by 5% and all other by 3%.

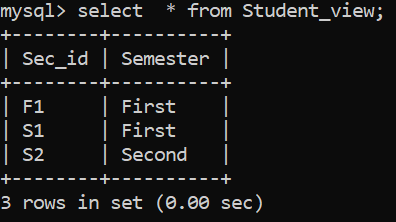
****

****

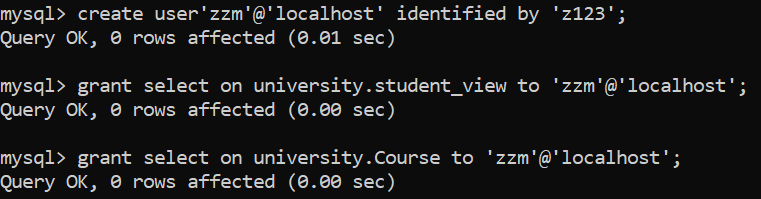
7. Get query to create view as Student\_view table as get the section’s id and semester.

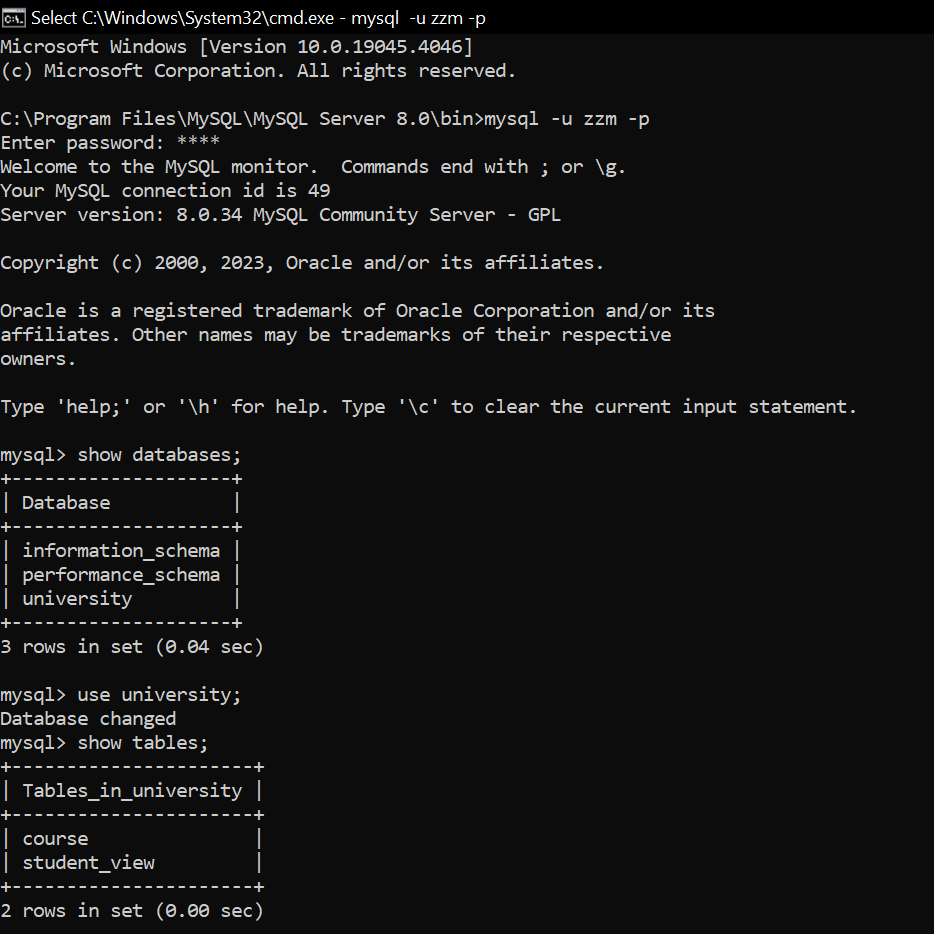
****

8. Get all detail of the Student\_view.

****

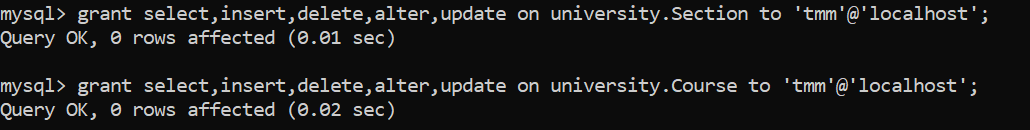
9. This is the student section, student can see only Student\_view and Course tables, so student allowed to select.

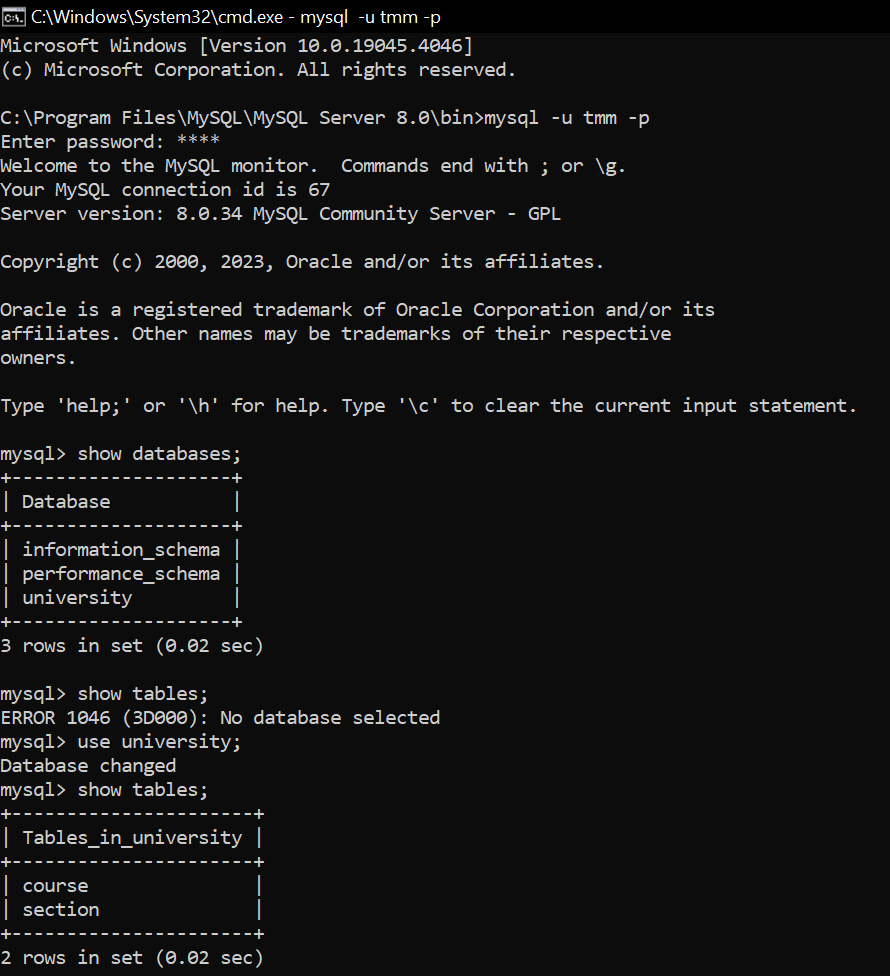
****

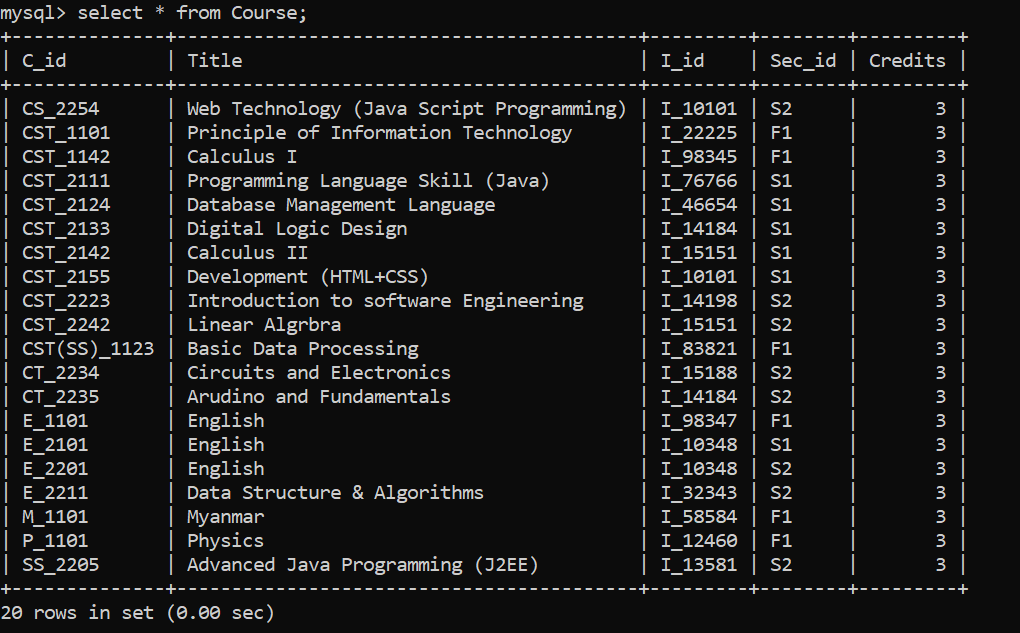
****

10.This is instructor section, instructor can see tables, so instructor is allowed to do all privileges.

****







11. Revoke permission to select the course table that is allowed to select , insert, delete, update, and alter.



**CHAPTER 6**

**CONCLUSION**

**6.1 Conclusion**

University Management System (UMS) is an essential tool for academic institutions to manage their administrative tasks effectively. It allows for streamlined processes, improved data management, and increased efficiency in handling student records, faculty data, course scheduling, and more.

Furthermore, a UMS can be tailored to meet the specific needs of different institutions, ensuring that it aligns with their unique requirements. It can also be integrated with other systems and applications to improve functionality and usability.

Overall, a Course Management System is a crucial investment for academic institutions seeking to improve their operations and provide better services to their stakeholders.

**6.2 Future Work**

* Updates systems
* Better facilities
* More efficient
* Add algorithm system
* Add design
* Reduce time and complexity
* Add more monitoring

**APPENDIX**

**MySQL Server 8.0**

MySQL is a purely relation database, whereas PostgreSQL is an object-relational database. PostgreSQL offers more sophisticated data types, and lets objects inherit properties. On the flip side, it also makes it more complex to work with PostgreSQL.

**MySQL Server 8.0**

The MySQL server provides a database management system with querying and connectivity capabilities, as well as the ability to have excellent data structure and integration with many different platforms. It can handle large database reliably and quickly in high-demanding production environments.

**How to install MySQL Server 8.0**

Generally, the MySQL Command Line Client is installed together with the MySQL server, and you don't need to download and install the MySQL client separately. To check whether you have the client program on your machine, go to search, switch to the Apps tab, and type "In case you don't have the client on your PC and don't want to install the entire MySQL Server package, go to the MYSQL download page (http://dev.mysql.com/download/mysql/), select the Microsoft Windows(x86, 64-bit), ZIP Archive. You will find MySQL Command Line Client binaries in the bin folder.

Step 1: Download and install MySQL

Step 2: Start the Server

Step 3: Start the Client

Step 4: Start the password for the Superuser "root"

Step 5: Create a New User

Step 6: Create a new Database, a new Table in the Database, Insert Records, Query and Update.

<http://www>.mysql.com/

REFERENCES

* <https://www.mysql.com/>
* <https://www.w3schools.com/MySQL/default.asp>
* <https://db-book.com>