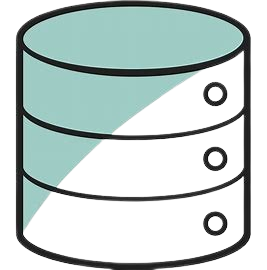


|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Programme | : | B. Tech (CSE) | Semester | : | Winter Semester 2023-2024 |
| Course Code &  Course Title | : | BCSE302L & Database Systems | Slot (s) | : | A2+TA2 |
| Faculty | : | Dr. M. Premalatha | Marks | : | 10 |

**DIGITAL ASSIGNMENT – PROJECT**

**Connection Of Front-End Scripting Languages with Databases**

**DATABASE SIMPLIFIER**



**By**

**Register Number: 22BCE5007**

**Name: Rohit Sinha**

* **Introduction**

In the realm of modern technology, data serves as the lifeblood of countless applications and systems, making efficient interaction with databases paramount. The seamless connection between front-end scripting languages and databases is pivotal in shaping user experiences and streamlining data management processes. This project endeavors to bridge the gap between front-end scripting languages and databases, enhancing user-friendliness and accessibility in data interaction.

The project addresses the challenge of facilitating smooth communication between front-end scripts and databases, which can often be complex and cumbersome, requiring intricate handling of queries, data retrieval, and presentation. By developing a robust solution that fosters intuitive and efficient data exchange, this project aims to mitigate these complexities.

At its core, the project recognizes that modern web applications rely heavily on dynamic data exchange, with users expecting real-time access to information, seamless updates, and intuitive interactions with underlying data repositories. However, traditional methods of interfacing front-end scripts with databases often introduce bottlenecks, security vulnerabilities, and scalability issues.

The proposed solution introduces a comprehensive framework that streamlines communication between front-end scripting languages and databases. By leveraging cutting-edge technologies and adhering to industry best practices, the project aims to create a unified, standardized approach that abstracts away complexities, enabling developers to focus on delivering rich user experiences.

Key objectives include enhancing security through robust authentication mechanisms, input validation, and encrypted communication channels, as well as addressing scalability and performance optimization through advanced caching, query optimization, and load balancing strategies.

Furthermore, the project emphasizes user-friendliness and accessibility, providing intuitive interfaces and comprehensive documentation to empower developers of varying skill levels to leverage the power of database interactions.

Ultimately, this project represents a significant step forward in bridging the gap between front-end scripting languages and databases, paving the way for more robust, efficient, and user-friendly applications that harness the full potential of data-driven experiences.

* **Project Overview:**

The project represents a comprehensive endeavor aimed at revolutionizing the interaction between front-end scripting languages and databases by seamlessly integrating Java Swing for the user interface (UI) alongside a Flask server and MySQL database. This multifaceted approach encompasses a range of functionalities, from query submission to result presentation, ensuring a cohesive and intuitive user experience throughout the data management process.

* **Key Components:**

1. **Flask Server:** Positioned as the central communication conduit, the Flask server orchestrates the flow of data between the Java Swing UI and the MySQL database. Leveraging Flask's lightweight and adaptable framework, the server adeptly handles incoming requests, processes queries, and facilitates interactions with the database layer. Its role extends beyond mere data transmission to encompass query optimization, error handling, and performance tuning, thereby ensuring optimal functionality and reliability.
2. **MySQL Database:** Serving as the repository for structured data, the MySQL database lies at the core of the system, storing and managing information accessed and manipulated by the front-end UI. The integration with the Flask server enables seamless execution of SQL queries, data retrieval, and manipulation, fostering a robust and scalable data management solution. With features such as transaction support, data indexing, and concurrency control, the MySQL database ensures data integrity and consistency, crucial for mission-critical applications across various domains.
3. **Java Swing UI:** The incorporation of Java Swing for the UI component elevates the project's user experience to new heights, offering a rich and interactive interface for users to engage with the system. From intuitive form inputs to dynamic data visualization, the Swing framework empowers developers to design elegant and user-friendly interfaces that resonate with modern design principles. Leveraging Swing's extensive widget library and event-driven architecture, the UI facilitates seamless navigation, query submission, and result presentation, enhancing usability and accessibility for users of all proficiency levels.

* **Project Workflow:**

The project represents a comprehensive endeavour aimed at revolutionizing the interaction between front-end scripting languages and databases by seamlessly integrating Java Swing for the user interface (UI) alongside a Flask server and MySQL database. This multifaceted approach encompasses a range of functionalities, from query submission to result presentation, ensuring a cohesive and intuitive user experience throughout the data management process.

* **User Authentication and Database Connection:**

1. The user opens the application and is prompted to login using their database credentials.
2. The Flask server receives the login request and validates the provided credentials.
3. Upon successful validation, the Flask server establishes a connection to the database using a dedicated module.
4. The connection status is relayed back to the Java client, indicating whether the connection was successful or if any errors occurred.
5. If successful, the user is directed to the query page; otherwise, an error message is displayed, prompting the user to retry the login process.

* **Query Submission and Validation:**

1. On the query page, the user inputs the desired SQL query into the designated field.
2. The Java client sends the query to the Flask server for validation.
3. The Flask server verifies the query syntax and semantics to ensure its compatibility with the database.
4. If the query is validated successfully, the Flask server returns a response code of 200 to the Java client, indicating readiness for execution.
5. In case of validation failure, a response code of 400 is returned, prompting the user to correct the query.

* **Query Execution and Result Handling:**

1. Upon receiving confirmation of query validity, the Java client sends the validated query to the Flask server for execution.
2. The Flask server executes the query against the connected database and retrieves the result set, if any.
3. If a result set is obtained, it is serialized into JSON format by the Flask server and transmitted back to the Java client.
4. The Java client parses the JSON result set using a custom parser and populates a JTable component with the retrieved data.
5. If the query does not return a result set, the Java client proceeds without displaying a results page.

* **Displaying Results and Exporting Data:**

1. If a result set is received, the Java client displays the results in a new page, termed the "Results Page," using the populated JTable component.
2. The Results Page allows users to visualize and interact with the queried data in a tabular format.
3. Additionally, the Results Page features an "Export to CSV" button, enabling users to save the current result set as a CSV file for further analysis or archival purposes.

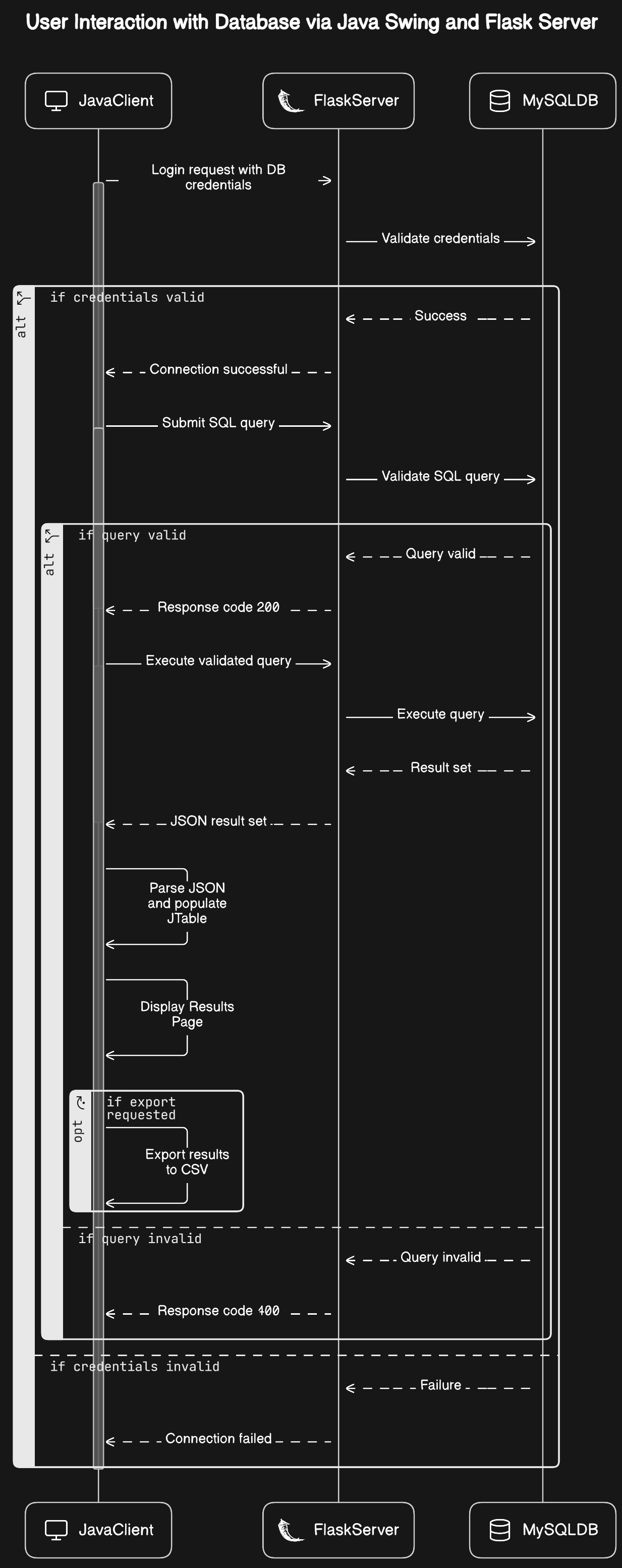
* **User Interaction and Navigation:**

1. Throughout the workflow, users interact with the Java Swing UI to input queries, navigate between pages, and interact with the displayed data.
2. Clear and intuitive navigation elements guide users through the application's functionalities, ensuring a seamless and user-friendly experience.

* **Error Handling and Feedback:**

1. The system provides informative error messages and feedback at each stage of the workflow, guiding users in resolving issues and completing tasks effectively.
2. Error handling mechanisms in both the Java client and Flask server ensure robustness and reliability, minimizing disruptions to the user experience and facilitating smooth operation under varying conditions.

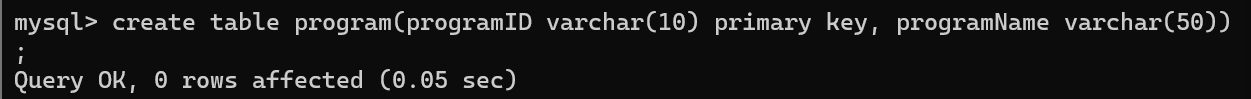
*PTO*→



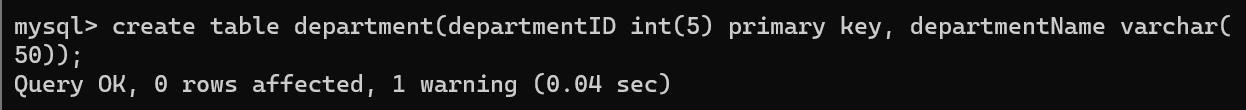
* **Generate 10 SQL queries:**

1. **DDL**: Create tables for Student, Course, Department, Professor, Grade, Advisor, etc.

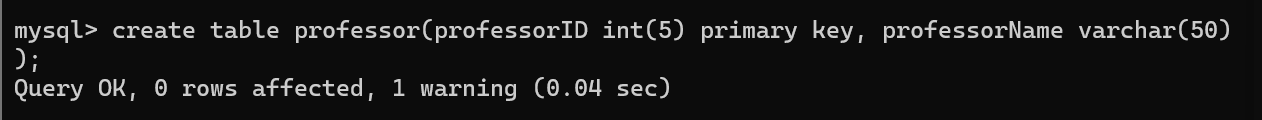
mysql> create table program(programID varchar(10) primary key, programName varchar(50));



mysql> create table department(departmentID int(5) primary key, departmentName varchar(50));

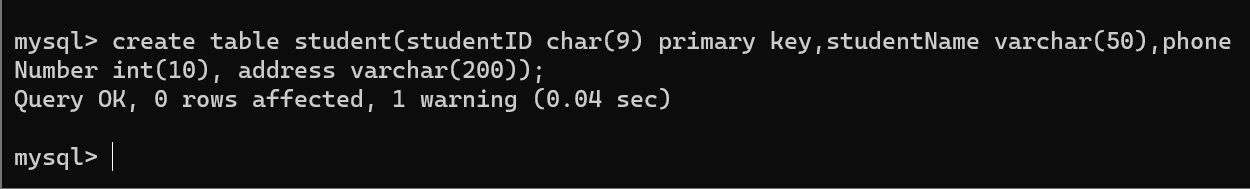


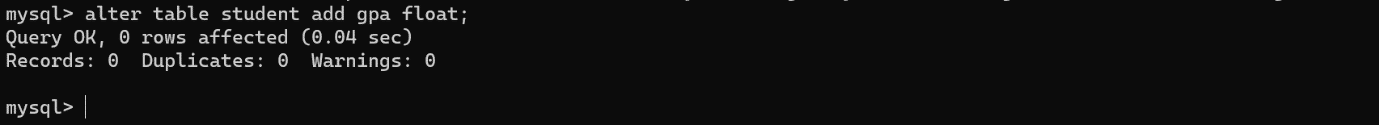
mysql> create table professor(professorID int(5) primary key, professorName varchar(50));



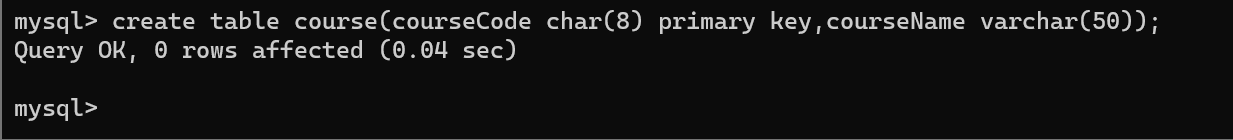
mysql> create table student(studentID char(9) primary key,studentName varchar(50),phoneNumber int(10), address varchar(200));

mysql> alter table student add gpa float;

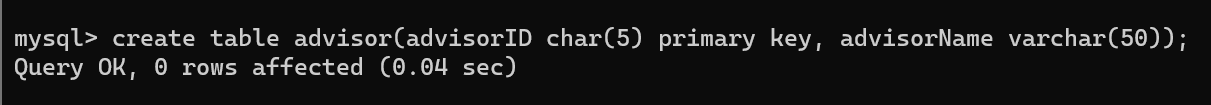




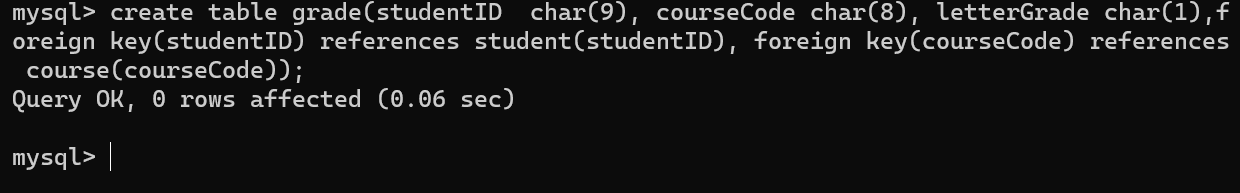
mysql> create table course(courseCode char(8) primary key,courseName varchar(50));



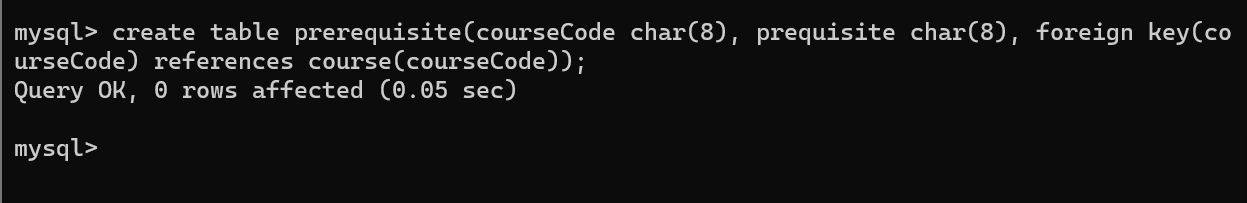
mysql> create table advisor(advisorID char(5) primary key, advisorName varchar(50));



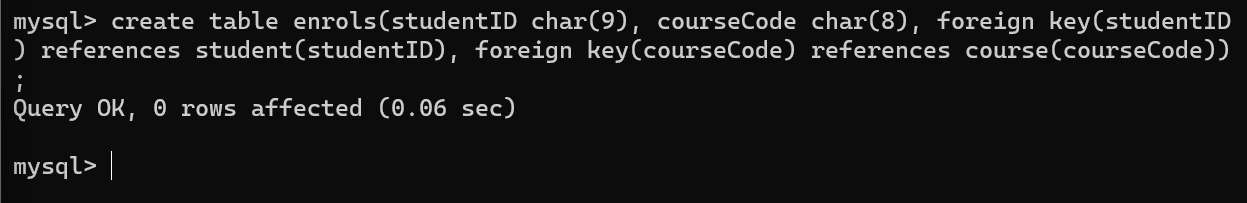
mysql> create table grade(studentID char(9), courseCode char(8), letterGrade char(1),foreign key(studentID) references student(studentID), foreign key(courseCode) references course(courseCode));



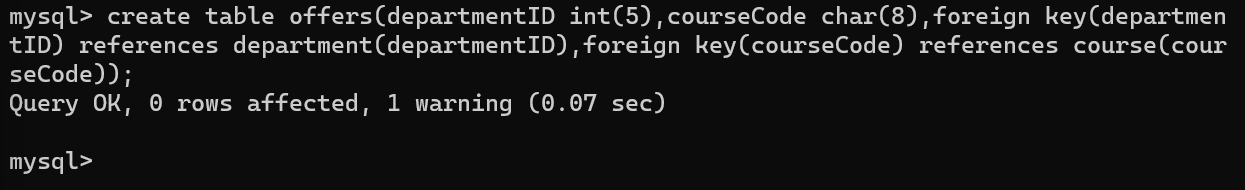
mysql> create table prerequisite(courseCode char(8), prequisite char(8), foreign key(courseCode) references course(courseCode));



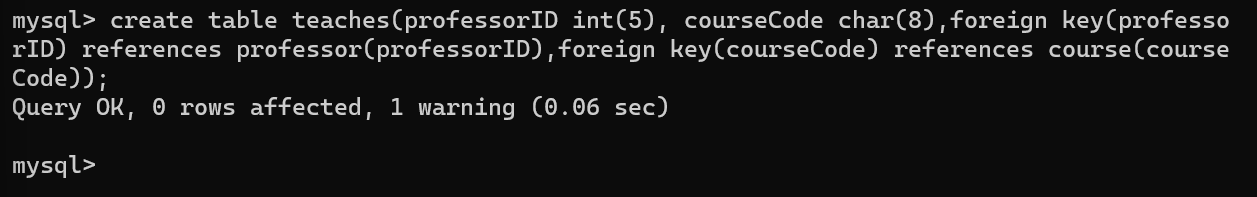
mysql> create table enrols(studentID char(9), courseCode char(8), foreign key(studentID) references student(studentID), foreign key(courseCode) references course(courseCode));



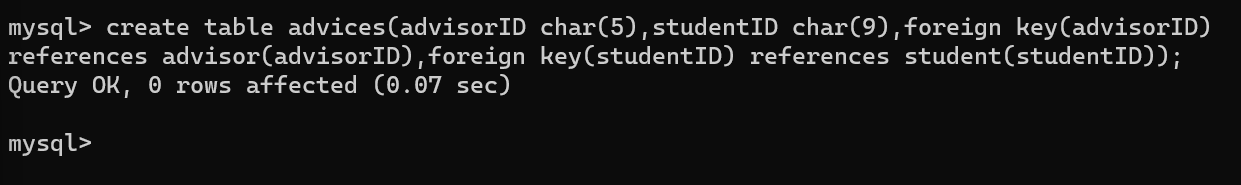
mysql> create table offers(departmentID int(5),courseCode char(8),foreign key(departmentID) references department(departmentID),foreign key(courseCode) references course(courseCode));



mysql> create table teaches(professorID int(5), courseCode char(8),foreign key(professorID) references professor(professorID),foreign key(courseCode) references course(courseCode));



mysql> create table advices(advisorID char(5),studentID char(9),foreign key(advisorID) references advisor(advisorID),foreign key(studentID) references student(studentID));



1. **DML**: Insert records into these tables.

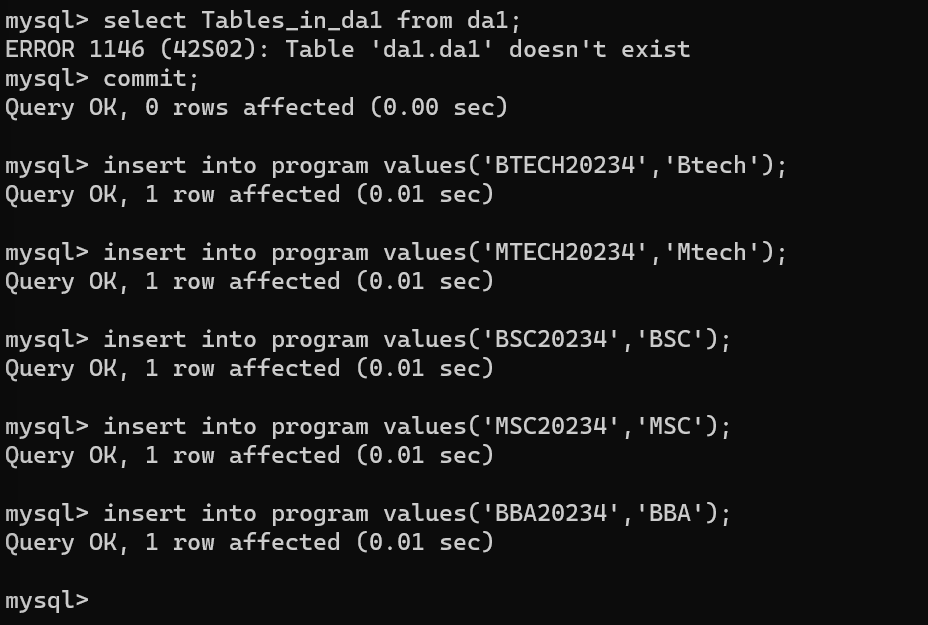
mysql> insert into program values('BTECH20234','Btech');

mysql> insert into program values('MTECH20234','Mtech');

mysql> insert into program values('BSC20234','BSC');

mysql> insert into program values('MSC20234','MSC');

mysql> insert into program values('BBA20234','BBA');



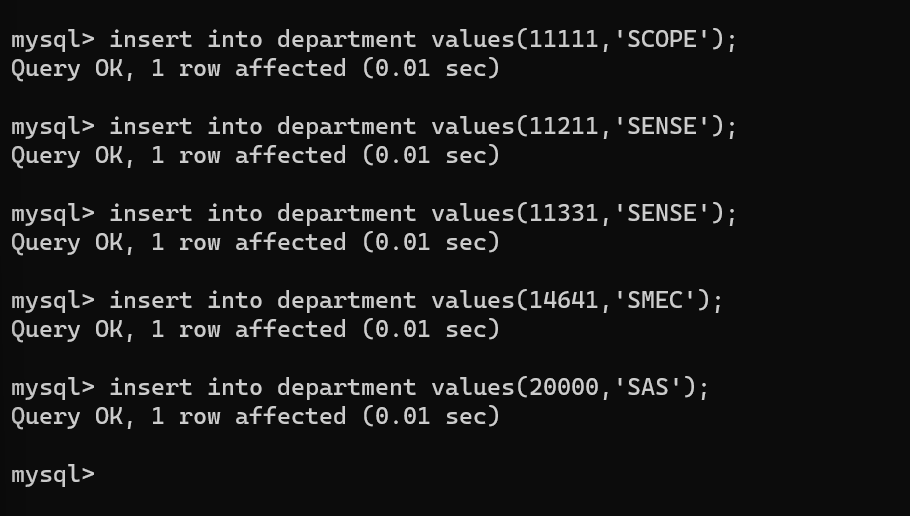
mysql> insert into department values(11111,'SCOPE');

mysql> insert into department values(11211,'SENSE');

mysql> insert into department values(11331,'SENSE');

mysql> insert into department values(14641,'SMEC');

mysql> insert into department values(20000,'SAS');



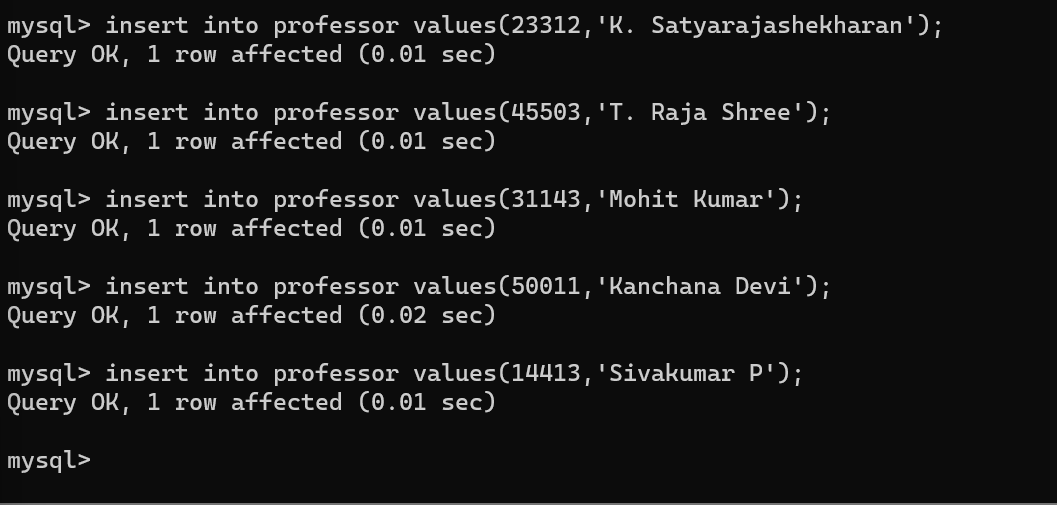
mysql> insert into professor values(23312,'K. Satyarajashekharan');

mysql> insert into professor values(45503,'T. Raja Shree');

mysql> insert into professor values(31143,'Mohit Kumar');

mysql> insert into professor values(50011,'Kanchana Devi');

mysql> insert into professor values(14413,'Sivakumar P');



mysql> insert into student values('22BCE5007','Rohit Sinha',9353958508,'VIT Chennai',8.99);

mysql> insert into student values('22BCE1959','Dhairya Gupta',9942853377,'VIT Chennai',9.42);

mysql> insert into student values('22BEC1822','Rakesh Arora',8722388999,'Bangalore',7.33);

mysql> insert into student values('22BEC1311','Meena prasad',9886484269,'Bangalore',9.00);

mysql> insert into student values('22BEE1541','Rahul Sudev',9449016169,'Kerala',9.80);

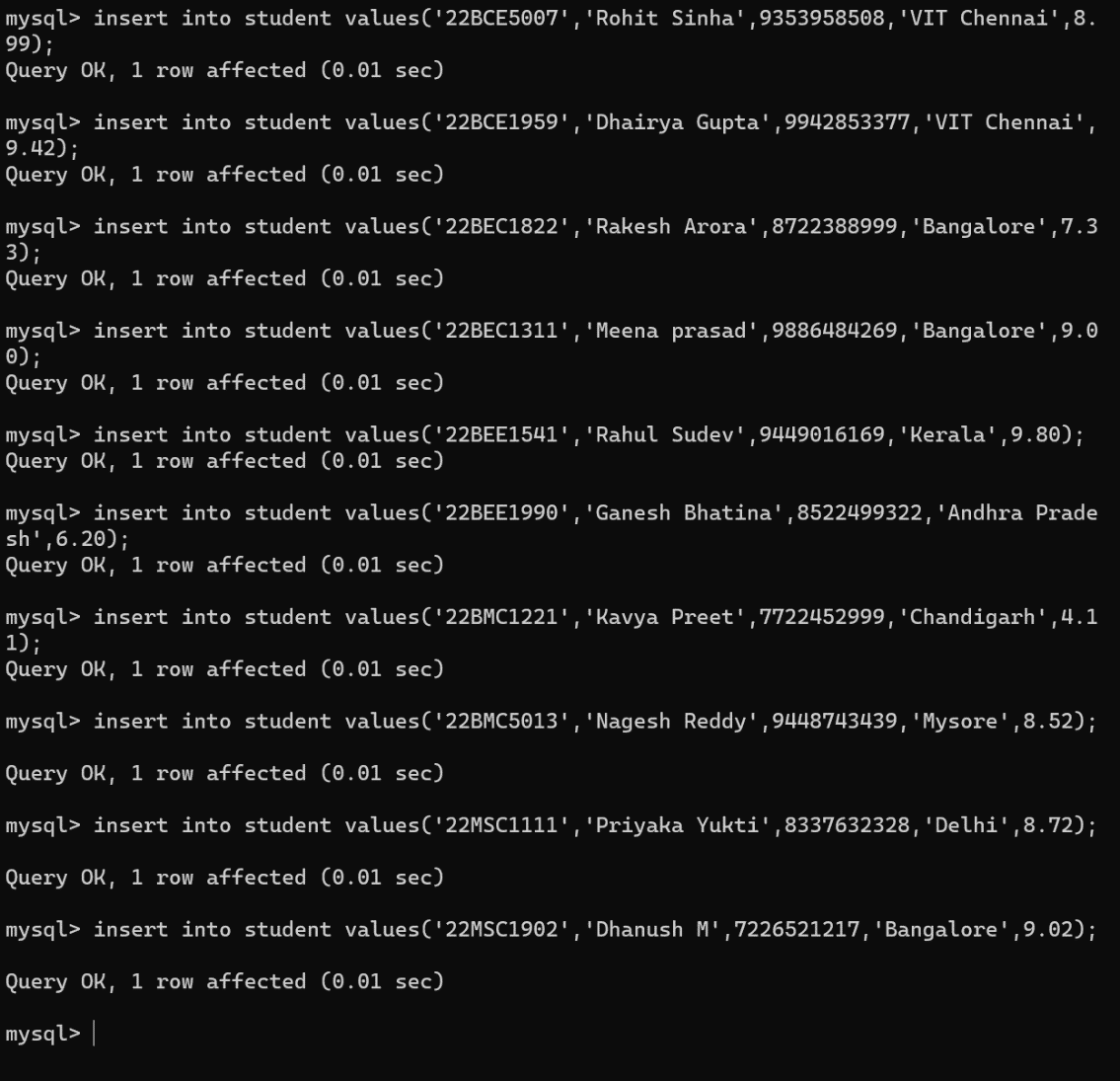
mysql> insert into student values('22BEE1990','Ganesh Bhatina',8522499322,'Andhra Pradesh',6.20);

mysql> insert into student values('22BMC1221','Kavya Preet',7722452999,'Chandigarh',4.11);

mysql> insert into student values('22BMC5013','Nagesh Reddy',9448743439,'Mysore',8.52);

mysql> insert into student values('22MSC1111','Priyaka Yukti',8337632328,'Delhi',8.72);

mysql> insert into student values('22MSC1902','Dhanush M',7226521217,'Bangalore',9.02);



mysql> insert into course values('BCSE102L','Structured And Object Oriented Programming');

mysql> insert into course values('BCSE202L','Data Structures And Algorithms');

mysql> insert into course values('BECE102L','Digital System Design');

mysql> insert into course values('BECE201L','Microprocessors And Microcontrollers');

mysql> insert into course values('BEEE101L','Basic Electrical Engineering');

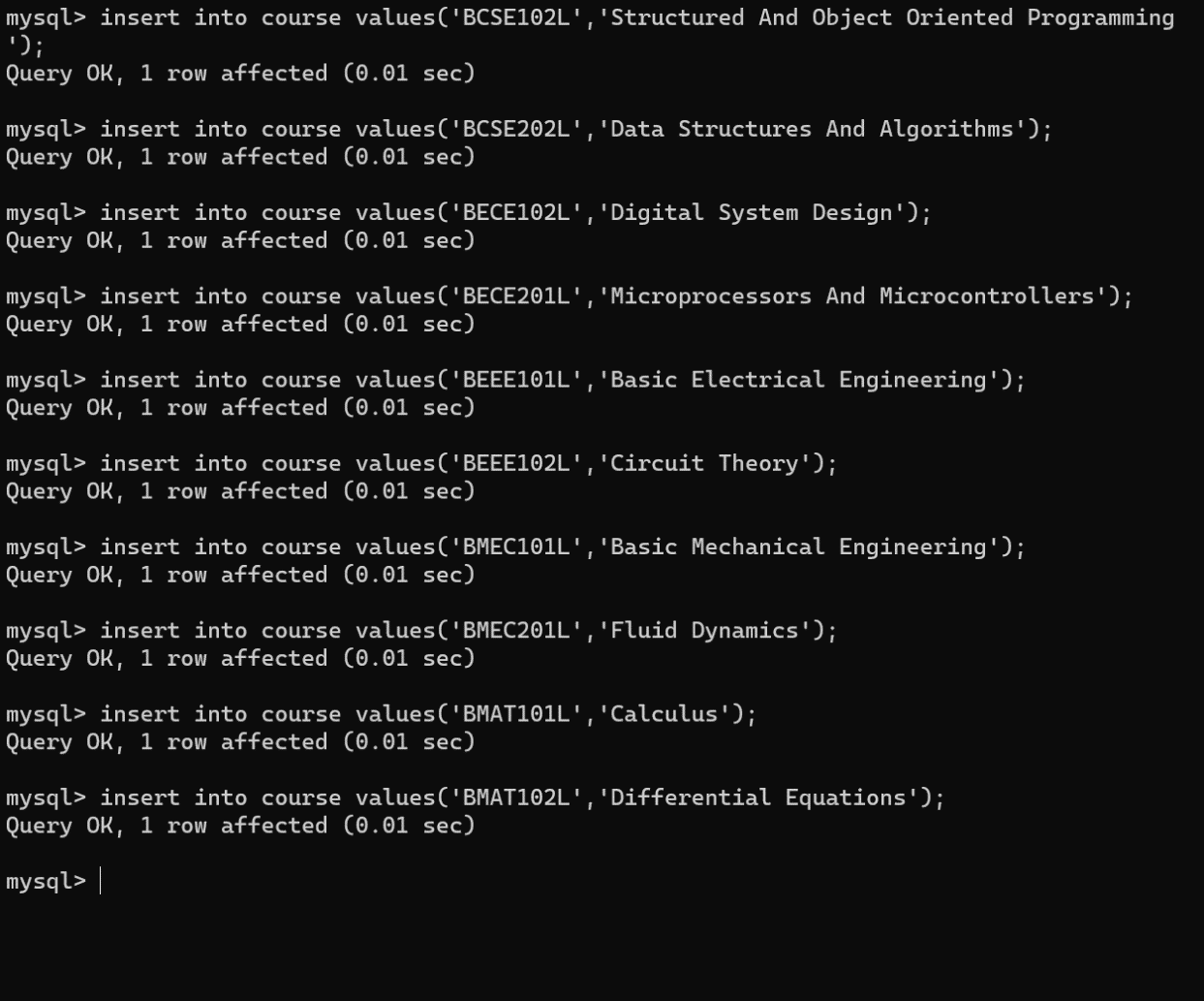
mysql> insert into course values('BEEE102L','Circuit Theory');

mysql> insert into course values('BMEC101L','Basic Mechanical Engineering');

mysql> insert into course values('BMEC201L','Fluid Dynamics');

mysql> insert into course values('BMAT101L','Calculus');

mysql> insert into course values('BMAT102L','Differential Equations');



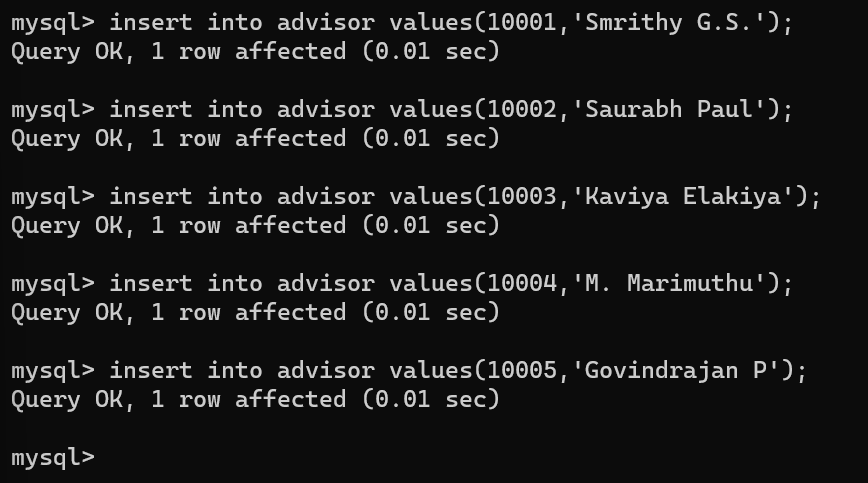
mysql> insert into advisor values(10001,'Smrithy G.S.');

mysql> insert into advisor values(10002,'Saurabh Paul');

mysql> insert into advisor values(10003,'Kaviya Elakiya');

mysql> insert into advisor values(10004,'M. Marimuthu');

mysql> insert into advisor values(10005,'Govindrajan P');



mysql> insert into grade values('22BCE5007','BCSE102L','S');

mysql> insert into grade values('22BCE5007','BCSE202L','S');

mysql> insert into grade values('22BCE1959','BCSE102L','A');

mysql> insert into grade values('22BCE1959','BCSE202L','A');

mysql> insert into grade values('22BE^C59','BCSE202L','A');

mysql> insert into grade values('22BEC1311','BECE102L','S');

mysql> insert into grade values('22BEC1311','BECE201L','B');

mysql> insert into grade values('22BEC1822','BECE102L','B');

mysql> insert into grade values('22BEC1822','BECE201L','C');

mysql> insert into grade values('22BEE1541','BEEE101L','S');

mysql> insert into grade values('22BEE1541','BEEE102L','S');

mysql> insert into grade values('22BEE1990','BEEE101L','C');

mysql> insert into grade values('22BEE1990','BEEE102L','D');

mysql> insert into grade values('22BMC1221','BMEC101L','E');

mysql> insert into grade values('22BMC1221','BMEC201L','E');

mysql> insert into grade values('22BMC5013','BMEC101L','A');

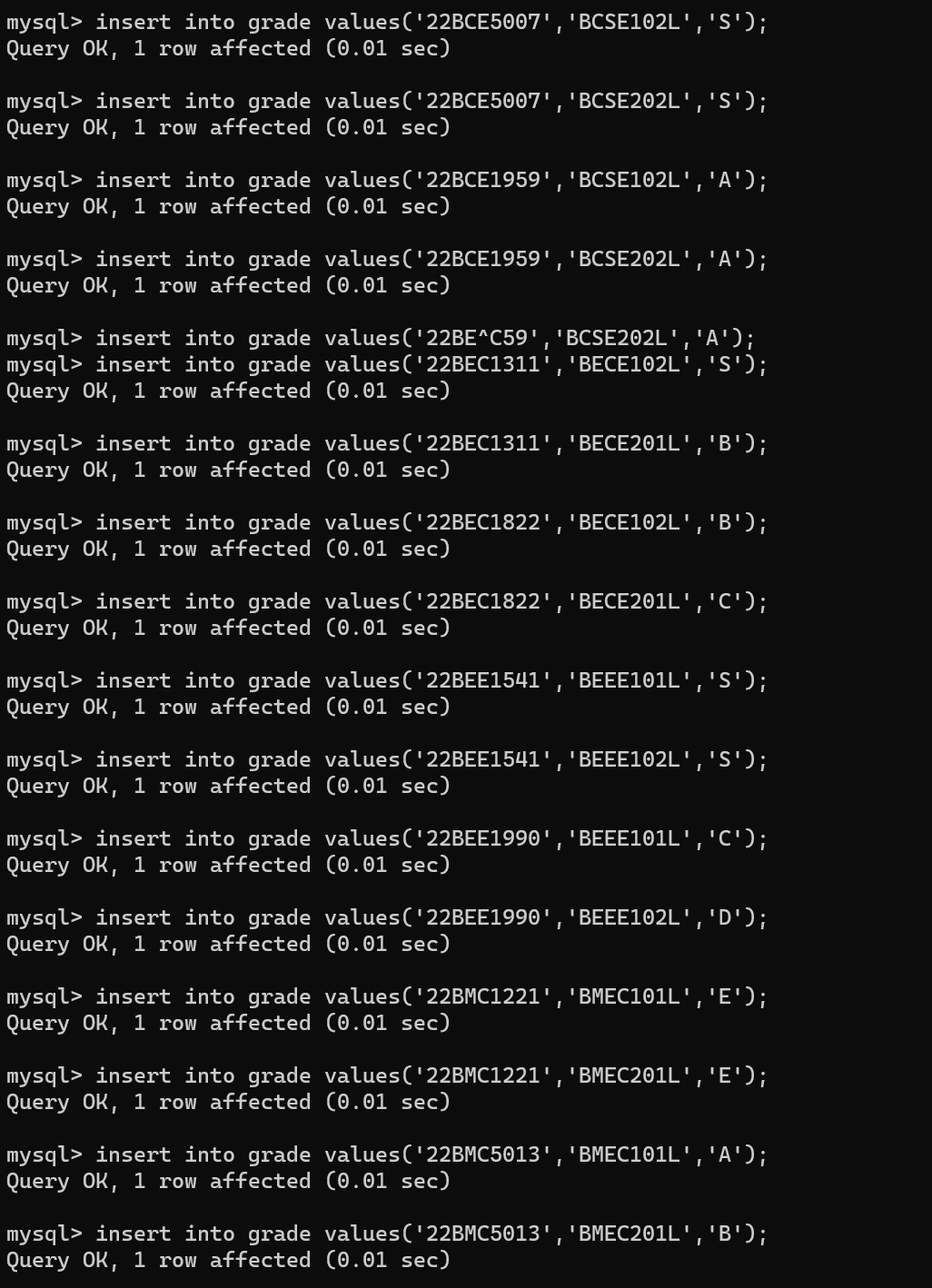
mysql> insert into grade values('22BMC5013','BMEC201L','B');

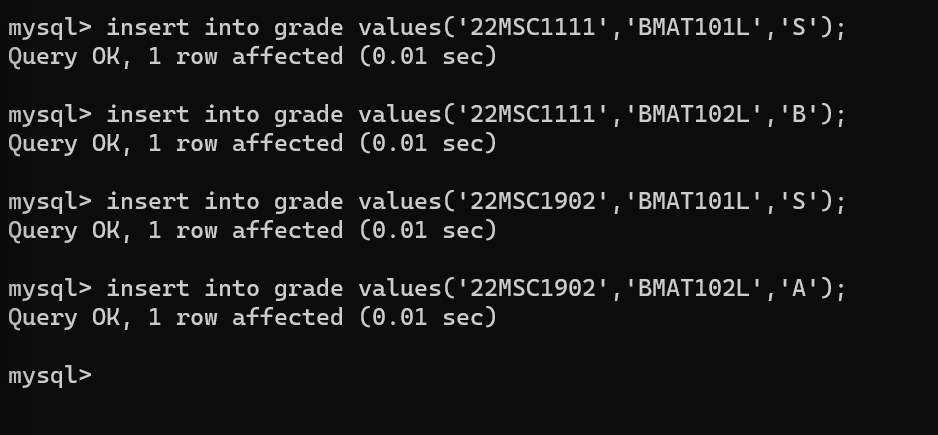
mysql> insert into grade values('22MSC1111','BMAT101L','S');

mysql> insert into grade values('22MSC1111','BMAT102L','B');

mysql> insert into grade values('22MSC1902','BMAT101L','S');

mysql> insert into grade values('22MSC1902','BMAT102L','A');





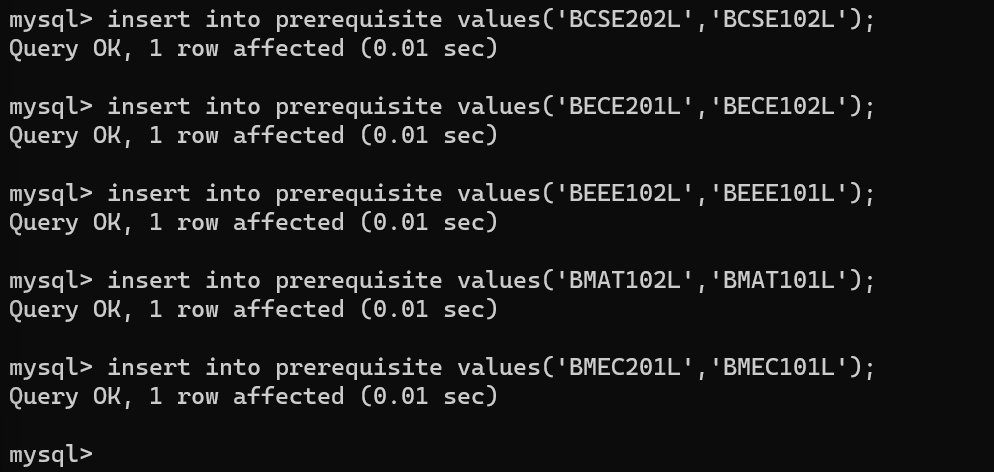
mysql> insert into prerequisite values('BCSE202L','BCSE102L');

mysql> insert into prerequisite values('BECE201L','BECE102L');

mysql> insert into prerequisite values('BEEE102L','BEEE101L');

mysql> insert into prerequisite values('BMAT102L','BMAT101L');

mysql> insert into prerequisite values('BMEC201L','BMEC101L');



mysql> insert into enrols values('22BCE1959','BCSE102L');

mysql> insert into enrols values('22BCE1959','BCSE202L');

mysql> insert into enrols values('22BCE5007','BCSE102L');

mysql> insert into enrols values('22BCE5007','BCSE202L');

mysql> insert into enrols values('22BEC1311','BECE102L');

mysql> insert into enrols values('22BEC1311','BECE201L');

mysql> insert into enrols values('22BEC1822','BECE102L');

mysql> insert into enrols values('22BEC1822','BECE201L');

mysql> insert into enrols values('22BEE1541','BEEE101L');

mysql> insert into enrols values('22BEE1541','BEEE102L');

mysql> insert into enrols values('22BEE1990','BEEE101L');

mysql> insert into enrols values('22BEE1990','BEEE102L');

mysql> insert into enrols values('22BMC1221','BMAT101L');

mysql> insert into enrols values('22BMC1221','BMAT102L');

mysql> insert into enrols values('22BMC5013','BMAT101L')

mysql> insert into enrols values('22BMC5013','BMAT102L');

mysql> update enrols set courseCode = 'BMEC101L' where courseCode = 'BMAT101L';

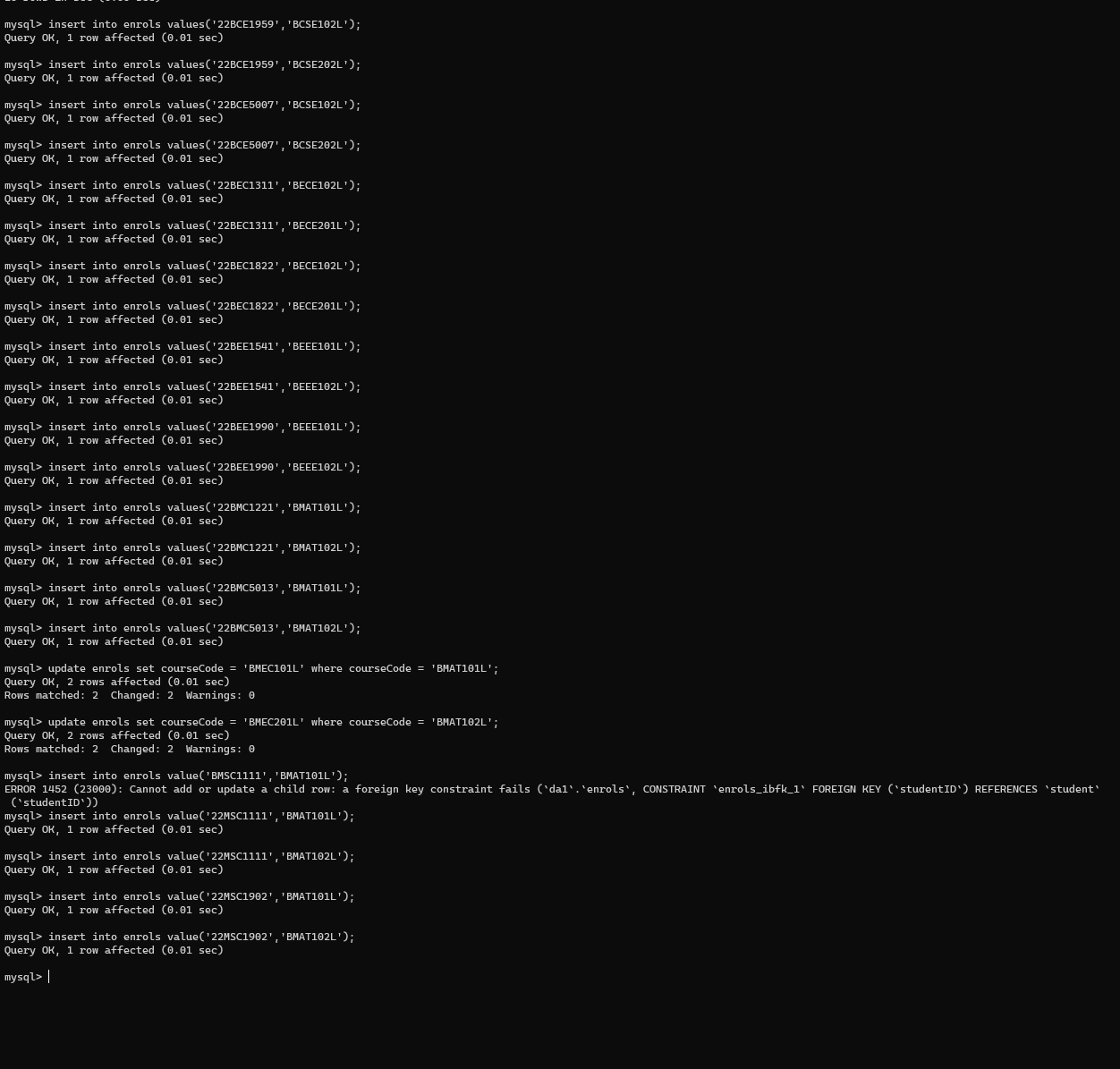
mysql> update enrols set courseCode = 'BMEC201L' where courseCode= 'BMAT102L';

mysql> insert into enrols value('22MSC1111','BMAT101L');

mysql> insert into enrols value('22MSC1111','BMAT102L');

mysql> insert into enrols value('22MSC1902','BMAT101L');

mysql> insert into enrols value('22MSC1902','BMAT102L');



mysql> insert into offers values(11111,'BCSE102L');

mysql> insert into offers values(11111,'BCSE202L');

mysql> insert into offers values(11211,'BECE102L');

mysql> insert into offers values(11211,'BECE201L');

mysql> update department set departmentName = 'SELECT' where departmentID = 11331;

mysql> insert into offers values(11331,'BEEE101L');

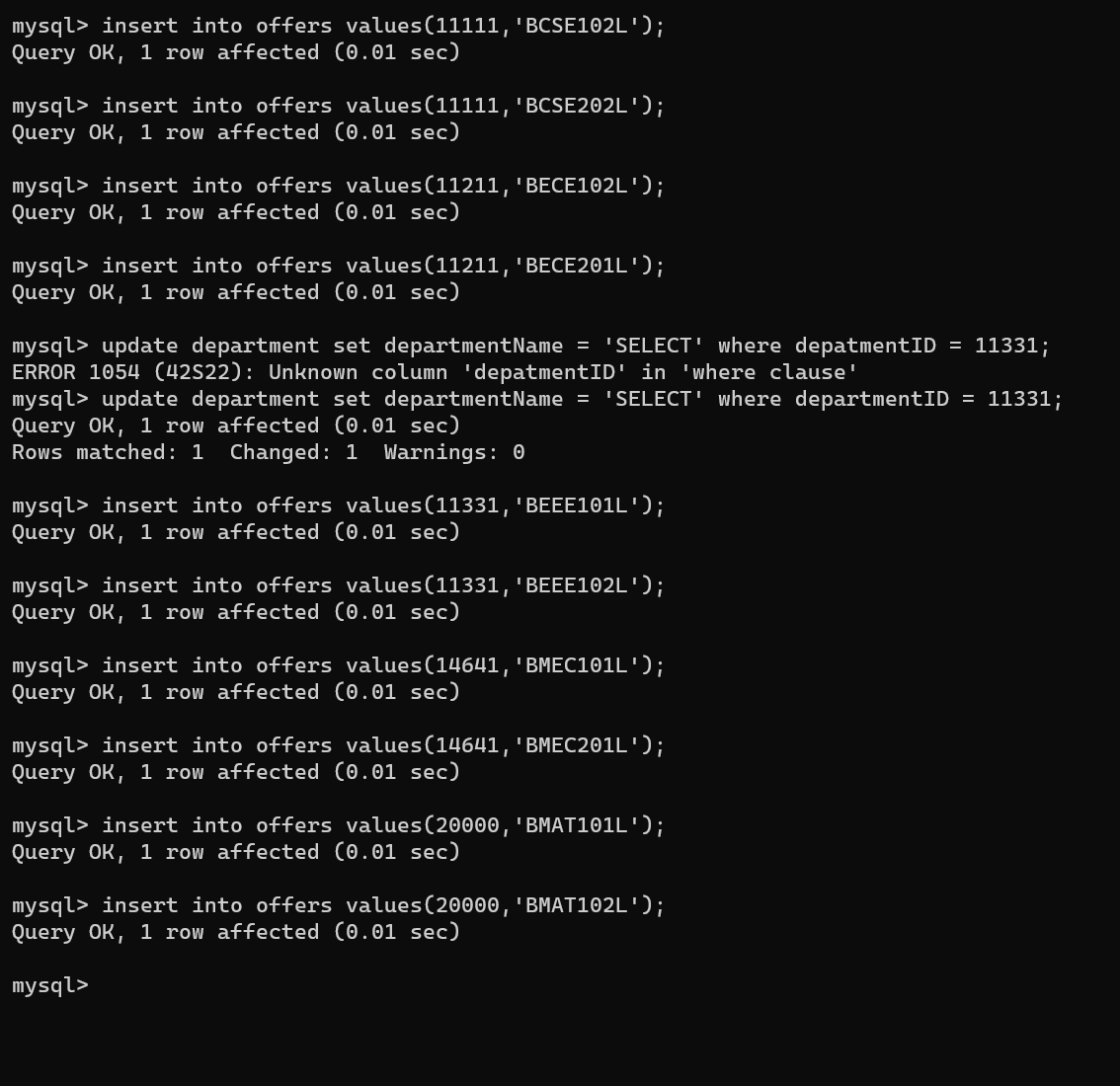
mysql> insert into offers values(11331,'BEEE102L');

mysql> insert into offers values(14641,'BMEC101L');

mysql> insert into offers values(14641,'BMEC201L');

mysql> insert into offers values(20000,'BMAT101L');

mysql> insert into offers values(20000,'BMAT102L');



mysql> insert into teaches values(14413,'BCSE102L');

mysql> insert into teaches values(14413,'BCSE202L');

mysql> insert into teaches values(23312,'BECE102L');

mysql> insert into teaches values(23312,'BECE201L');

mysql> insert into teaches values(45503,'BEEE101L');

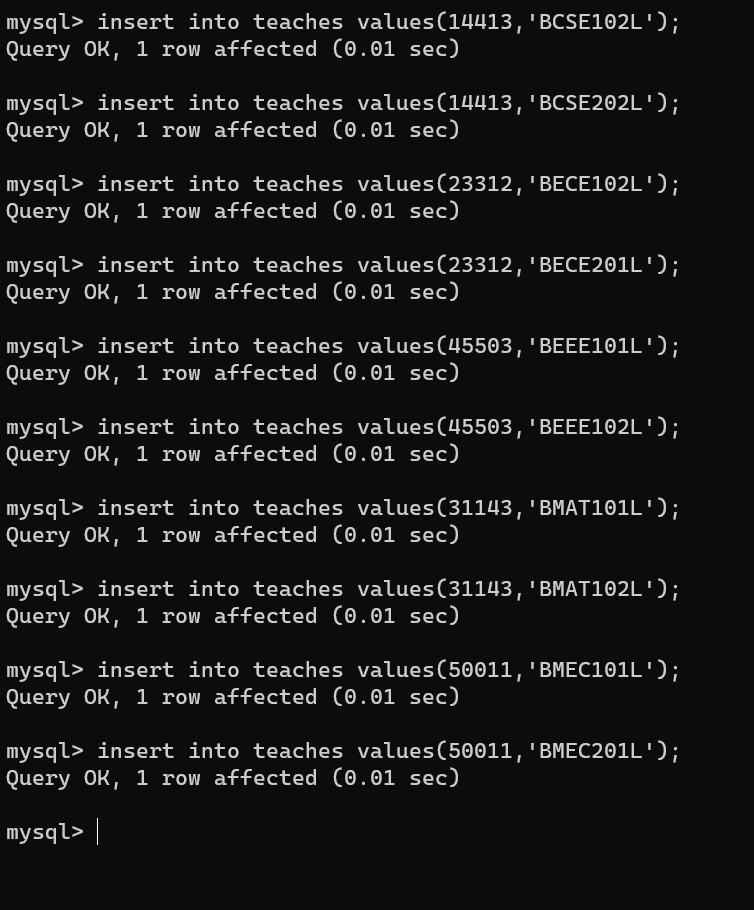
mysql> insert into teaches values(45503,'BEEE102L');

mysql> insert into teaches values(31143,'BMAT101L');

mysql> insert into teaches values(31143,'BMAT102L');

mysql> insert into teaches values(50011,'BMEC101L');

mysql> insert into teaches values(50011,'BMEC201L');



mysql> insert into advices values(10001,'22BCE5007');

mysql> insert into advices values(10001,'22BCE1959');

mysql> insert into advices values(10002,'22BEC1311');

mysql> insert into advices values(10002,'22BEC1822');

mysql> insert into advices values(10003,'22BEE1541');

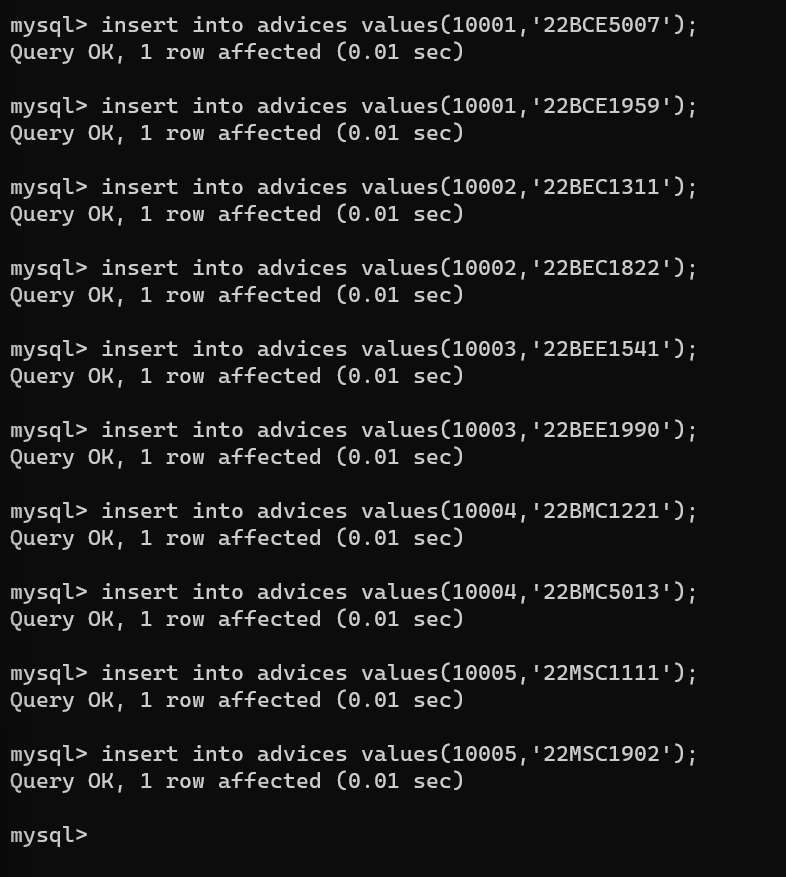
mysql> insert into advices values(10003,'22BEE1990');

mysql> insert into advices values(10004,'22BMC1221');

mysql> insert into advices values(10004,'22BMC5013');

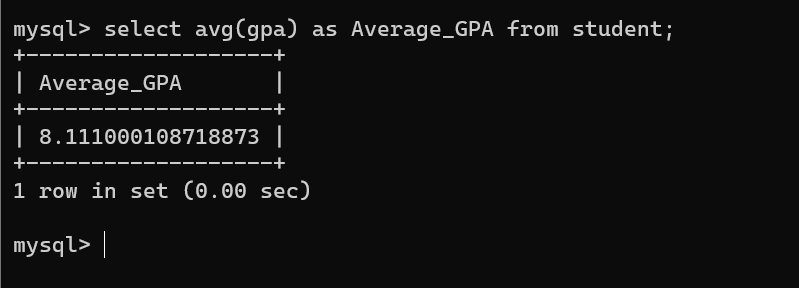
mysql> insert into advices values(10005,'22MSC1111');

mysql> insert into advices values(10005,'22MSC1902');



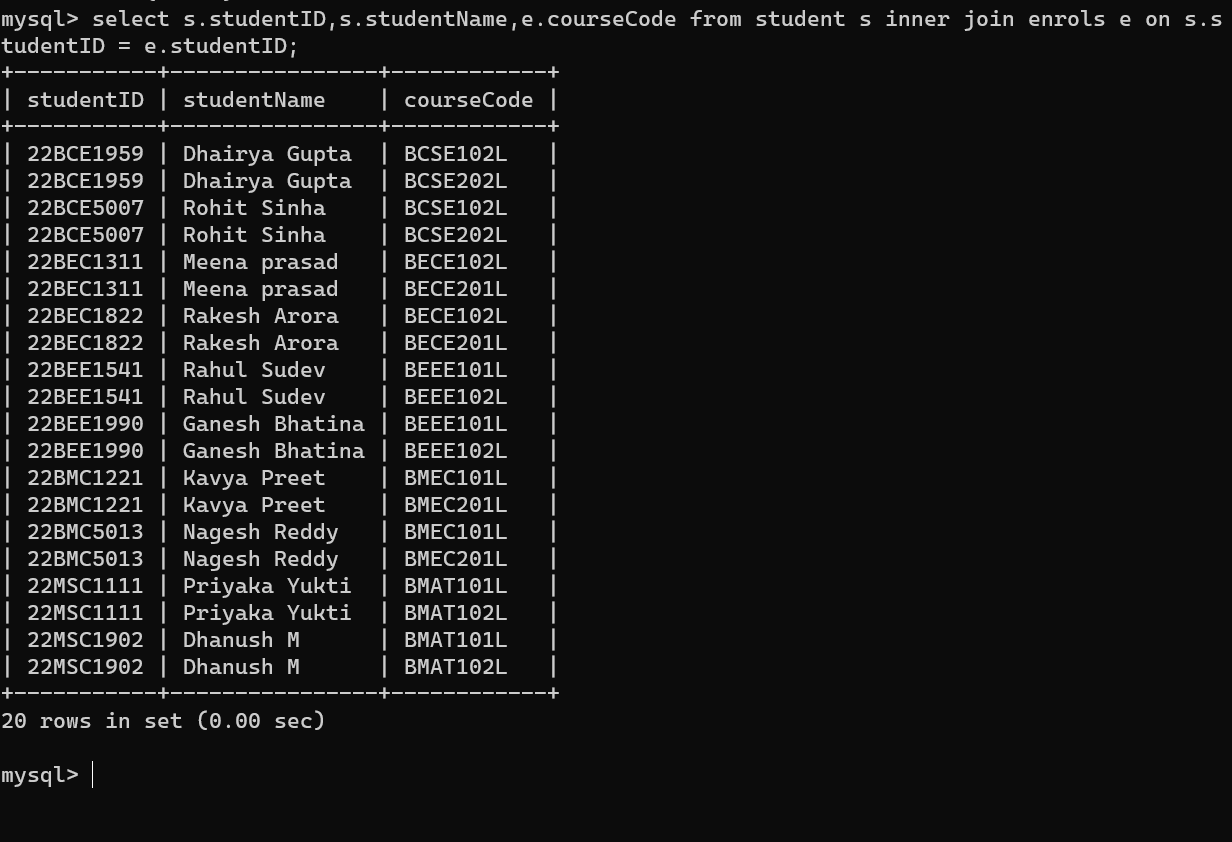
1. **Aggregation function**: Find the average GPA of students.

mysql> select avg(gpa) as Average\_GPA from student;



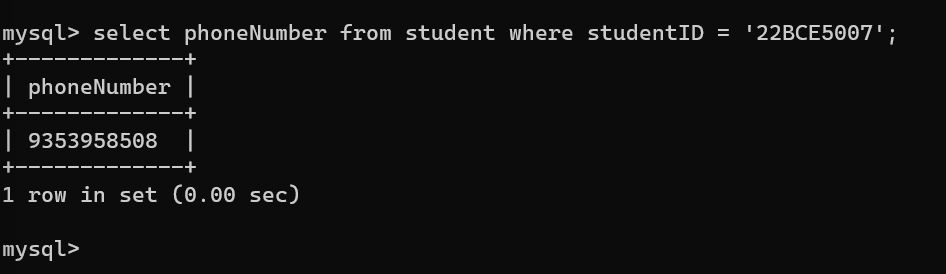
1. **Joins**: Retrieve a list of students along with their courses.

mysql> select s.studentID,s.studentName,e.courseCode from student s inner join enrols e on s.studentID = e.studentID;



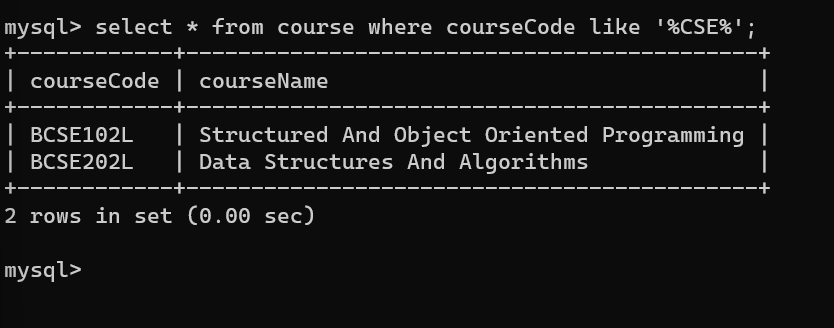
1. **Filter**: Find the phone number of a specific student.

mysql> select phoneNumber from student where studentID = '22BCE5007';



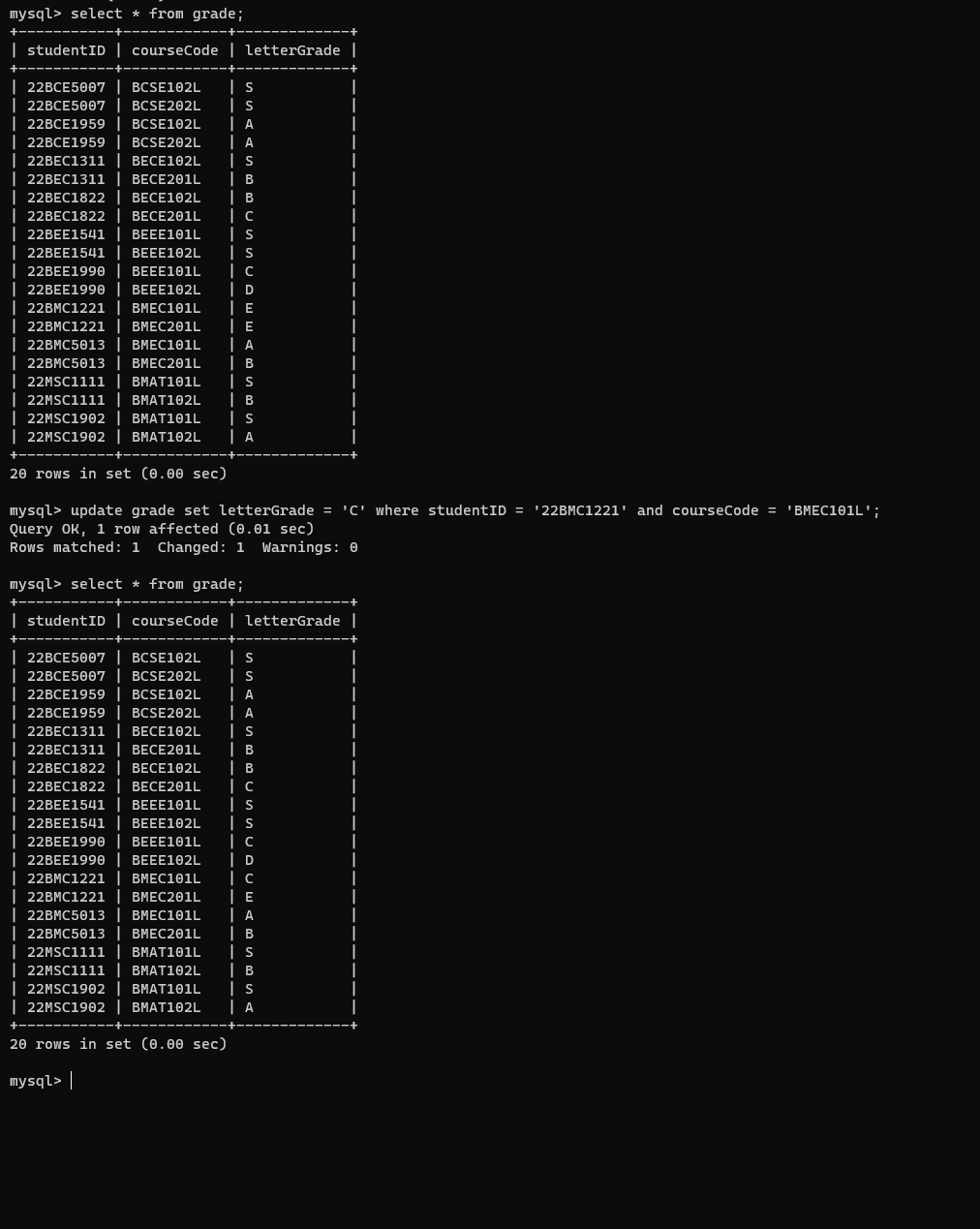
1. **Filter**: List of CSE courses offered.

mysql> select \* from course where courseCode like '%CSE%';



1. **Update**: Change the grade of a student in a specific course.

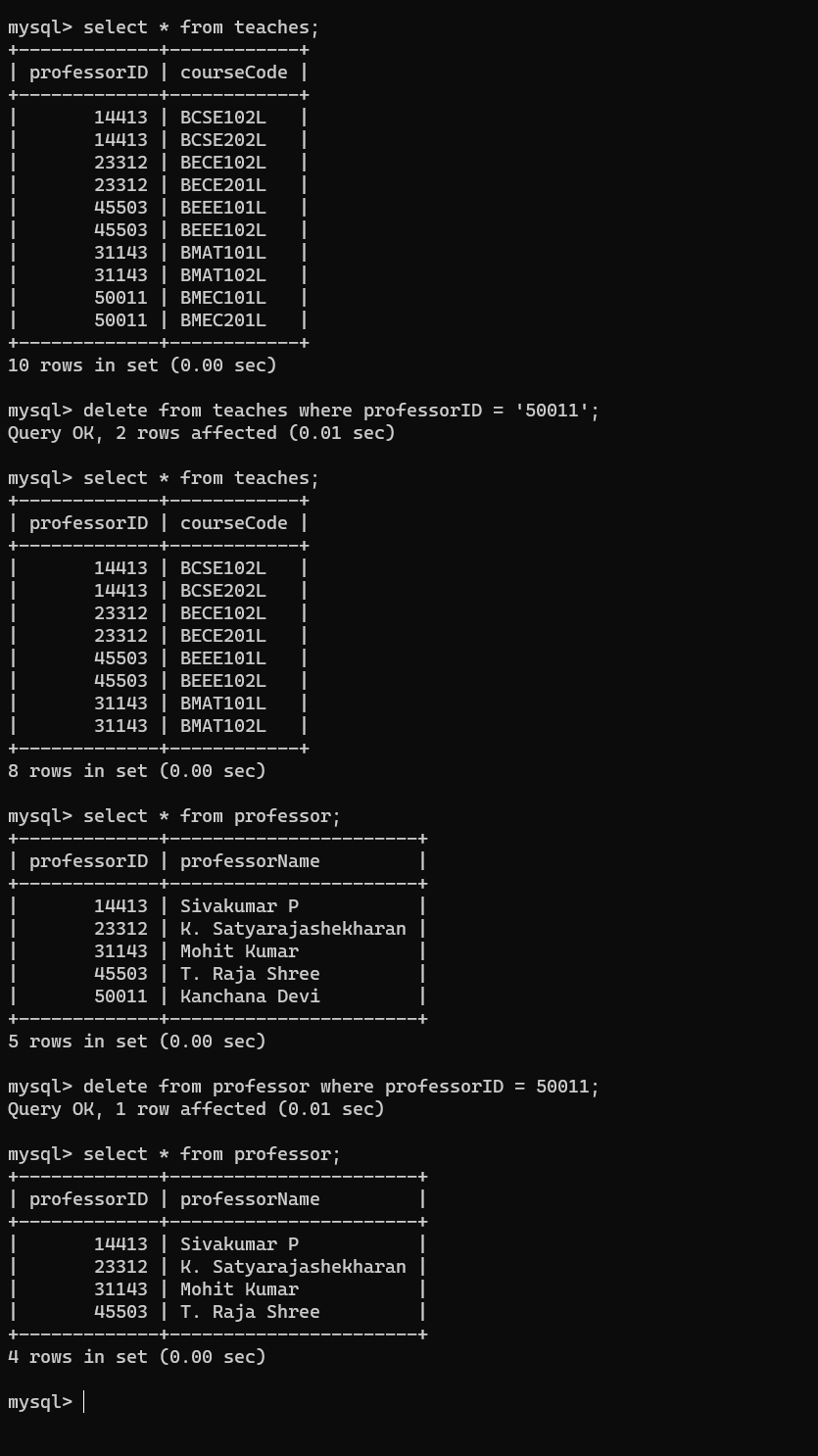
mysql> update grade set letterGrade = 'C' where studentID = '22BMC1221' and courseCode = 'BMEC101L';



1. **Delete**: Remove a professor's record along with their associated courses.

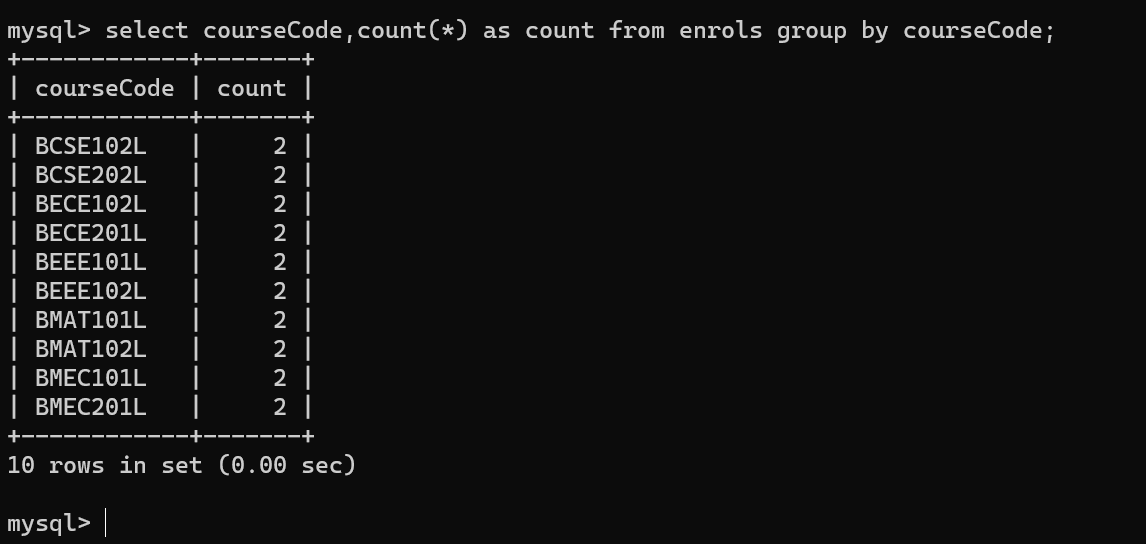
mysql> delete from teaches where professorID = '50011';

mysql> delete from professor where professorID = 50011;



1. **Group by**: Count the number of students enrolled in each course.

mysql> select courseCode,count(\*) as count from enrols group by courseCode;



1. **Subquery**: List students who haven't completed a prerequisite course.

mysql> select studentID,count(\*) as Number\_Of\_Courses from enrols group by studentID having count(\*) = (select count(\*) from enrols group by studentID having count(\*) < 2);

