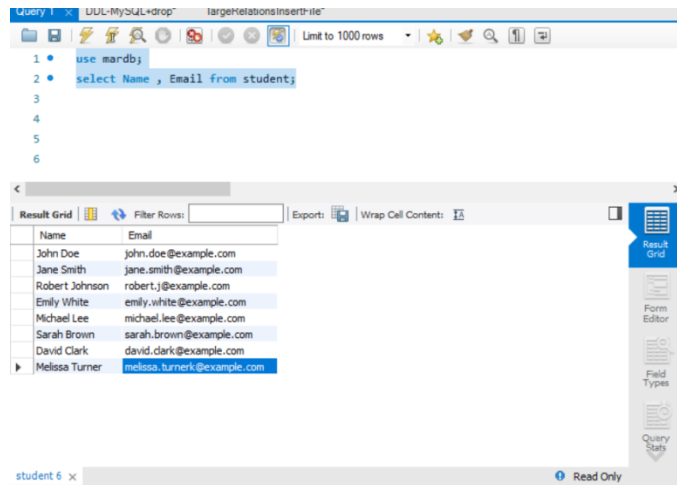


Assignment-5

1) Retrieve the name and email address of all students.

Ans) use mardb;

select Name , Email from student;



The screenshot shows a MySQL query editor window with the following SQL code:

```
1 use mardb;
2 select Name , Email from student;
3
4
5
6
```

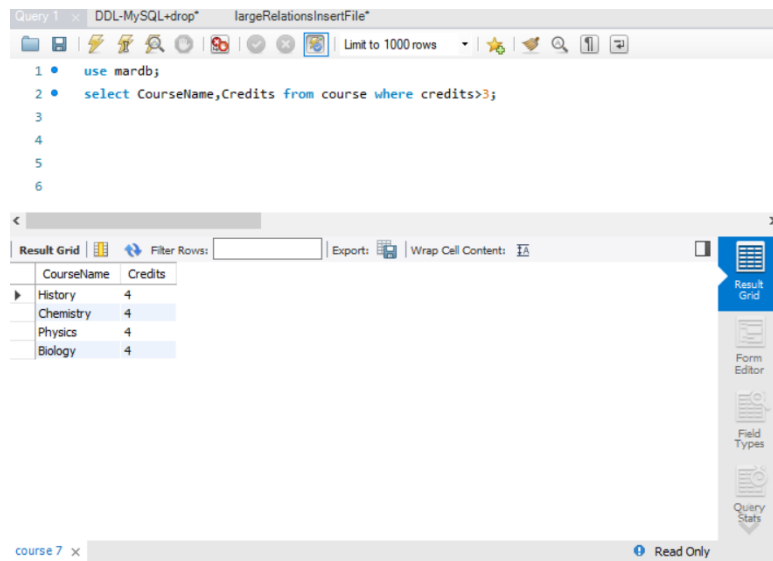
The results are displayed in a table with the following data:

Name	Email
John Doe	john.doe@example.com
Jane Smith	jane.smith@example.com
Robert Johnson	robert.j@example.com
Emily White	emily.white@example.com
Michael Lee	michael.lee@example.com
Sarah Brown	sarah.brown@example.com
David Clark	david.clark@example.com
Melissa Turner	melissa.turner@example.com

2) Find the courses that have more than three credits.

Ans) use mardb;

select CourseName,Credits from course where credits>3;



The screenshot shows a MySQL query editor window with the following SQL code:

```
1 use mardb;
2 select CourseName,Credits from course where credits>3;
3
4
5
6
```

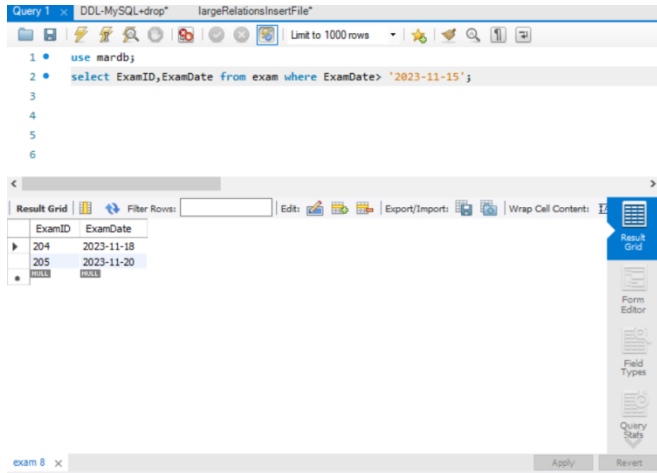
The results are displayed in a table with the following data:

CourseName	Credits
History	4
Chemistry	4
Physics	4
Biology	4

3) List the exams scheduled after November 15,2023

Ans) use mardb;

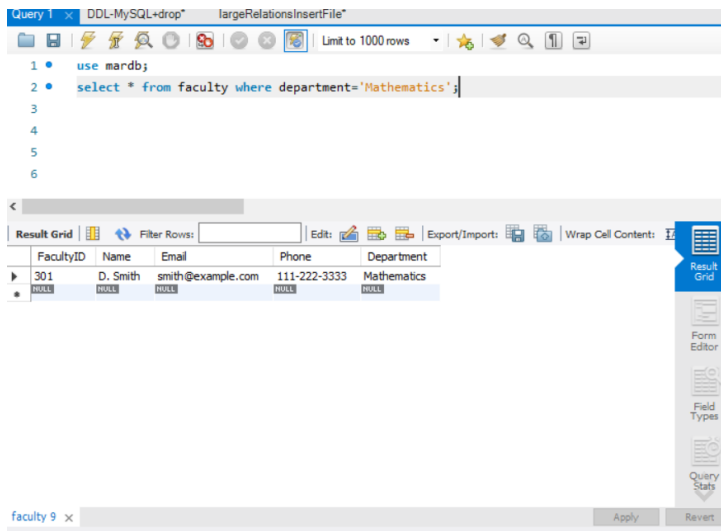
```
select ExamID,ExamDate from exam where ExamDate> '2023-11-15';
```



4) Get the faculty members who work in the “Mathematics” department.

Ans) use mardb;

```
select * from faculty where department='Mathematics';
```



5) Retrieve the courses that each student is enrolled in.

Ans) use mardb;

```
SELECT Student.Name, Course.CourseName
```

```
FROM Student
```

```
JOIN Enrollment ON Student.StudentID = Enrollment.StudentID
```

```
JOIN Course ON Enrollment.CourseID = Course.CourseID;
```

Query 1 x DDL-MySQL-drop* largeRelationsInsertFile*

Limit to 1000 rows

```

1 • use mardb;
2 • SELECT Student.Name, Course.CourseName
3 FROM Student
4 JOIN Enrollment ON Student.StudentID = Enrollment.StudentID
5 JOIN Course ON Enrollment.CourseID = Course.CourseID;
6
7

```

Result Grid

Name	CourseName
John Doe	Mathematics
John Doe	History
Jane Smith	Mathematics
Robert Johnson	Computer Science
Emily White	Literature
Michael Lee	Chemistry
Sarah Brown	Physics
David Clark	Economics
Melissa Turner	Biology

Result 10 x Read Only

6) Find the average score for each exam.

Ans) use mardb;

```

SELECT ExamID, AVG(Score) AS AverageScore
FROM ExamResults
GROUP BY ExamID;

```

Query 1 x DDL-MySQL-drop* largeRelationsInsertFile*

Limit to 1000 rows

```

1 • use mardb;
2 • SELECT ExamID, AVG(Score) AS AverageScore
3 FROM ExamResults
4 GROUP BY ExamID;
5
6
7

```

Result Grid

ExamID	AverageScore
201	89.333333
202	95.500000
203	89.000000
204	94.500000
205	91.000000

Result 11 x Read Only

7) List the students who scored above 90 on any exam.

Ans) use mardb;

```

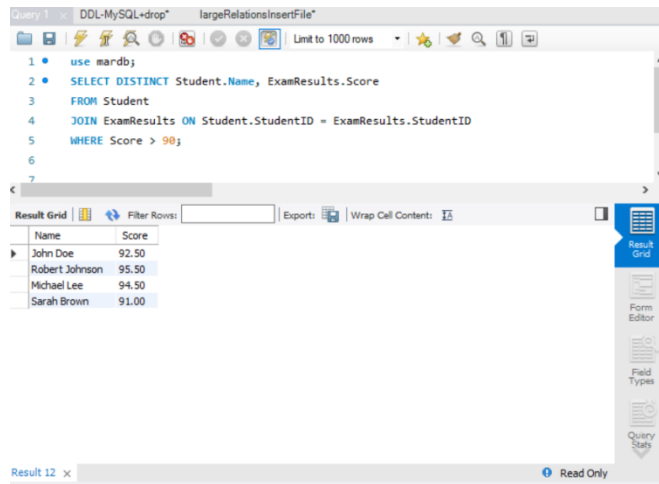
SELECT DISTINCT Student.Name, ExamResults.Score

FROM Student

JOIN ExamResults ON Student.StudentID = ExamResults.StudentID

WHERE Score > 90;

```



8) Retrieve the faculty members who teach multiple courses.

Ans) use mardb;

```

SELECT Faculty.Name, COUNT(*) AS CourseCount

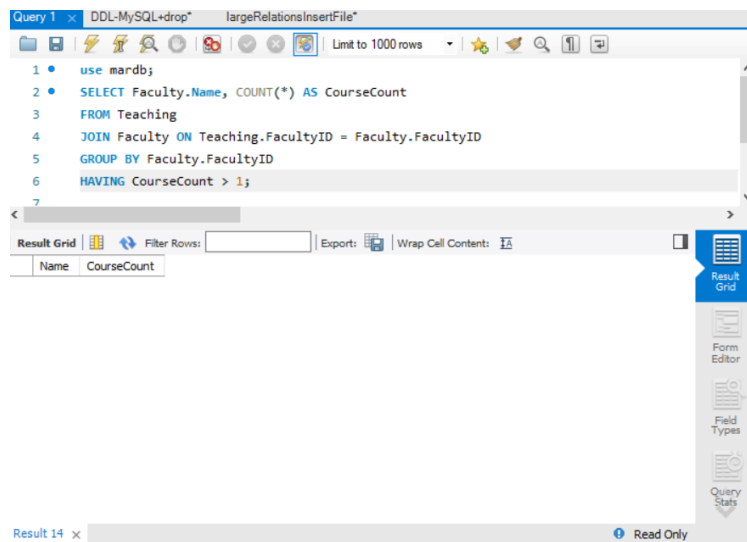
FROM Teaching

JOIN Faculty ON Teaching.FacultyID = Faculty.FacultyID

GROUP BY Faculty.FacultyID

HAVING CourseCount > 1;

```



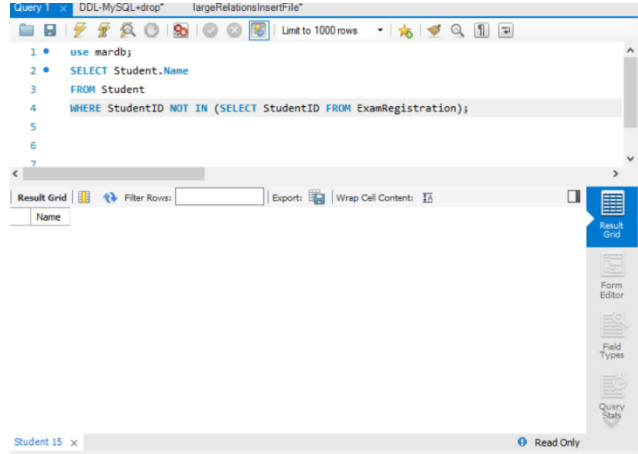
9) Find the students who have not registered in any exam.

Ans) use mardb;

```
SELECT Student.Name
```

```
FROM Student
```

```
WHERE StudentID NOT IN (SELECT StudentID FROM ExamRegistration);
```



10) Retrieve the number of enrollments for each course.

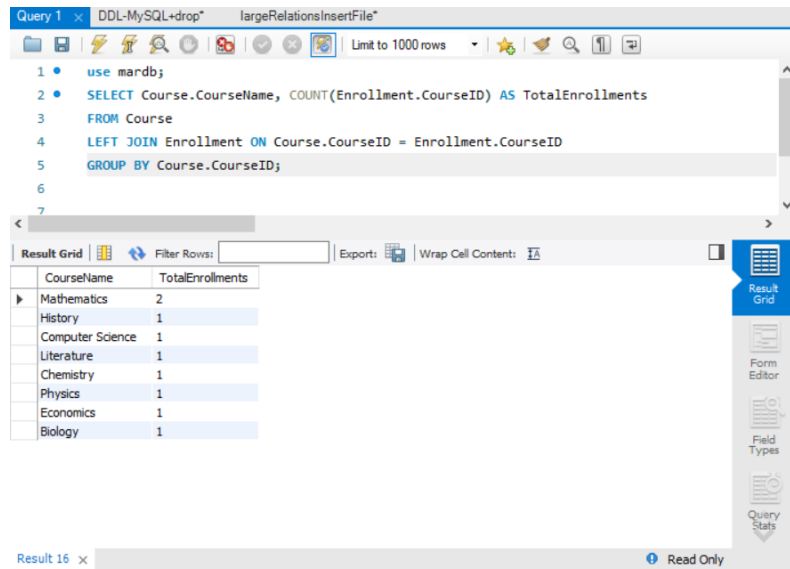
Ans) use mardb;

```
SELECT Course.CourseName, COUNT(Enrollment.CourseID) AS TotalEnrollments
```

```
FROM Course
```

```
LEFT JOIN Enrollment ON Course.CourseID = Enrollment.CourseID
```

```
GROUP BY Course.CourseID;
```



11) Find the students who are enrolled in the "History" course.

Ans) use mardb;

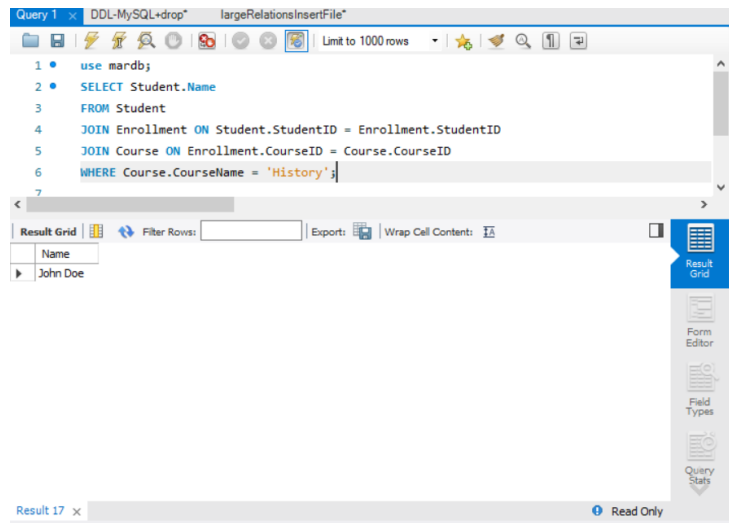
```
SELECT Student.Name
```

```
FROM Student
```

```
JOIN Enrollment ON Student.StudentID = Enrollment.StudentID
```

```
JOIN Course ON Enrollment.CourseID = Course.CourseID
```

```
WHERE Course.CourseName = 'History';
```



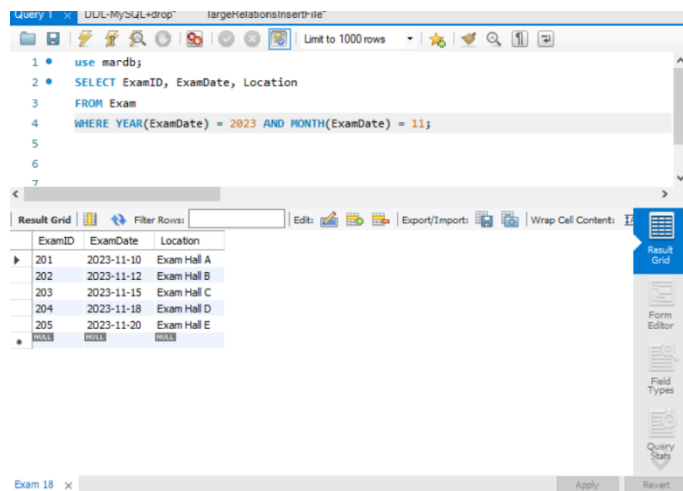
12) Retrieve the exams and their locations scheduled for November 2023.

Ans) use mardb;

```
SELECT ExamID, ExamDate, Location
```

```
FROM Exam
```

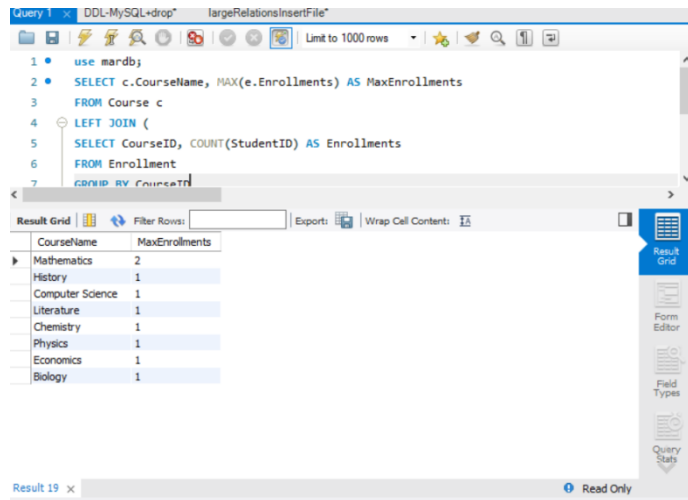
```
WHERE YEAR(ExamDate) = 2023 AND MONTH(ExamDate) = 11;
```



13) List the courses with the highest number of enrollments.

Ans) use mardb;

```
SELECT c.CourseName, MAX(e.Enrollments) AS MaxEnrollments
FROM Course c
LEFT JOIN (
SELECT CourseID, COUNT(StudentID) AS Enrollments
FROM Enrollment
GROUP BY CourseID
) e ON c.CourseID = e.CourseID
GROUP BY c.CourseName;
```



The screenshot shows a MySQL query editor window with the following SQL query:

```
1 use mardb;
2 SELECT c.CourseName, MAX(e.Enrollments) AS MaxEnrollments
3 FROM Course c
4 LEFT JOIN (
5 SELECT CourseID, COUNT(StudentID) AS Enrollments
6 FROM Enrollment
7 GROUP BY CourseID
8 ) e ON c.CourseID = e.CourseID
9 GROUP BY c.CourseName;
```

The result grid displays the following data:

CourseName	MaxEnrollments
Mathematics	2
History	1
Computer Science	1
Literature	1
Chemistry	1
Physics	1
Economics	1
Biology	1

14) Find the average score for each student.

Ans) use mardb;

```
SELECT Student.Name, AVG(Score) AS AverageScore
FROM Student
JOIN ExamResults ON Student.StudentID = ExamResults.StudentID
GROUP BY Student.StudentID;
```

Query 1: DDL-MySQL-drop* largeRelationsInsertFile*

```

1 use mardb;
2 SELECT Student.Name, AVG(Score) AS AverageScore
3 FROM Student
4 JOIN ExamResults ON Student.StudentID = ExamResults.StudentID
5 GROUP BY Student.StudentID;

```

Result Grid

Name	AverageScore
John Doe	92.500000
Jane Smith	88.000000
Robert Johnson	95.500000
Emily White	89.000000
Michael Lee	94.500000
Sarah Brown	91.000000
David Clark	87.500000

Result 20 x Read Only

15) Retrieve the exams that have no registered students.

Ans) use mardb;

```

SELECT E.ExamID, E.ExamDate
FROM Exam E
LEFT JOIN ExamRegistration ER ON E.ExamID = ER.ExamID
WHERE ER.ExamID IS NULL;

```

Query 1: DDL-MySQL-drop* largeRelationsInsertFile*

```

1 use mardb;
2 SELECT E.ExamID, E.ExamDate
3 FROM Exam E
4 LEFT JOIN ExamRegistration ER ON E.ExamID = ER.ExamID
5 WHERE ER.ExamID IS NULL;

```

Result Grid

ExamID	ExamDate
--------	----------

Result 21 x Read Only

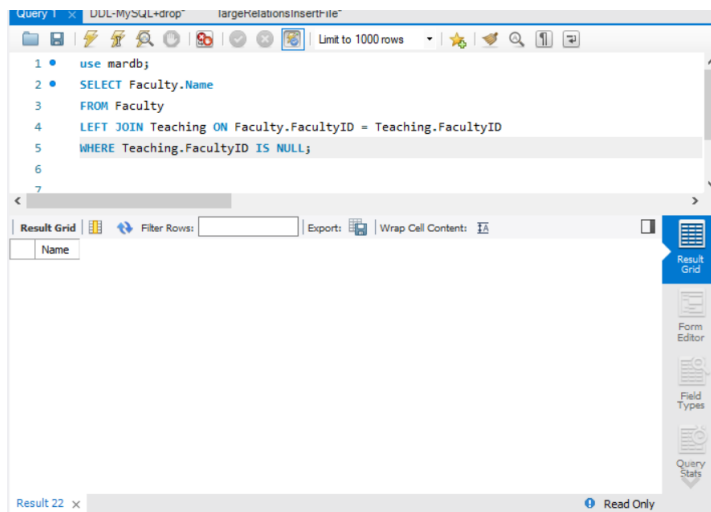
16) List the faculty members who have yet to teach any courses.

Ans) use mardb;

```

SELECT Faculty.Name
FROM Faculty
LEFT JOIN Teaching ON Faculty.FacultyID = Teaching.FacultyID
WHERE Teaching.FacultyID IS NULL;

```

17) Find the students who have registered for exams in both “Mathematics” and “Computer Science” departments.

Ans) use mardb;

SELECT Student.Name

FROM Student

JOIN Enrollment ON Student.StudentID = Enrollment.StudentID

JOIN Course ON Enrollment.CourseID = Course.CourseID

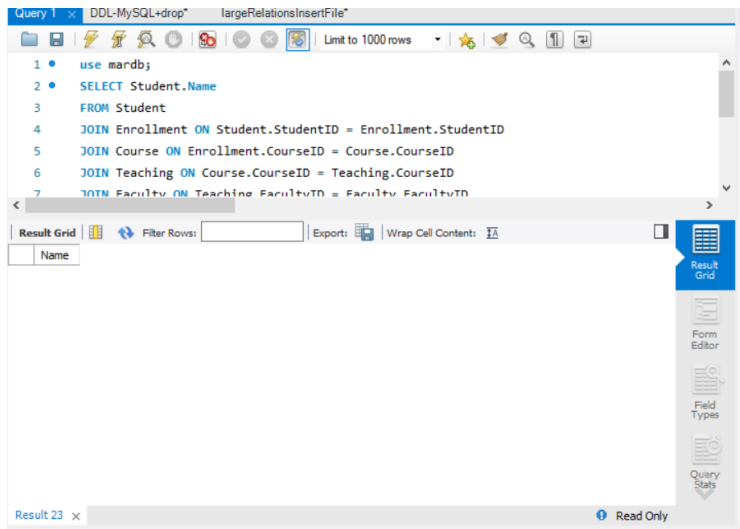
JOIN Teaching ON Course.CourseID = Teaching.CourseID

JOIN Faculty ON Teaching.FacultyID = Faculty.FacultyID

WHERE Faculty.Department IN ('Mathematics', 'Computer Science')

GROUP BY Student.StudentID

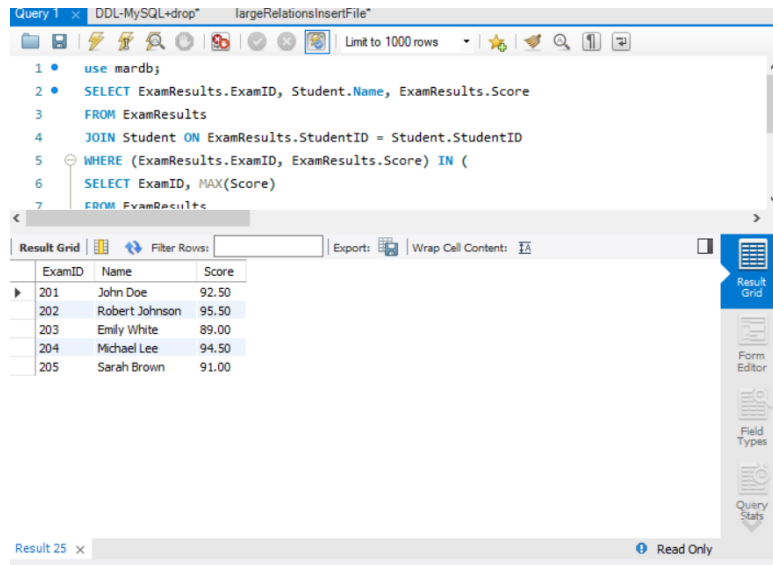
HAVING COUNT(DISTINCT Faculty.Department) = 2;



18) Retrieve the students who scored the highest in each exam.

Ans) use mardb;

```
SELECT ExamResults.ExamID, Student.Name, ExamResults.Score
FROM ExamResults
JOIN Student ON ExamResults.StudentID = Student.StudentID
WHERE (ExamResults.ExamID, ExamResults.Score) IN (
SELECT ExamID, MAX(Score)
FROM ExamResults
GROUP BY ExamID);
```



The screenshot shows a MySQL query editor window with a query and its results. The query is as follows:

```
1 use mardb;
2 SELECT ExamResults.ExamID, Student.Name, ExamResults.Score
3 FROM ExamResults
4 JOIN Student ON ExamResults.StudentID = Student.StudentID
5 WHERE (ExamResults.ExamID, ExamResults.Score) IN (
6 SELECT ExamID, MAX(Score)
7 FROM ExamResults
8 GROUP BY ExamID);
```

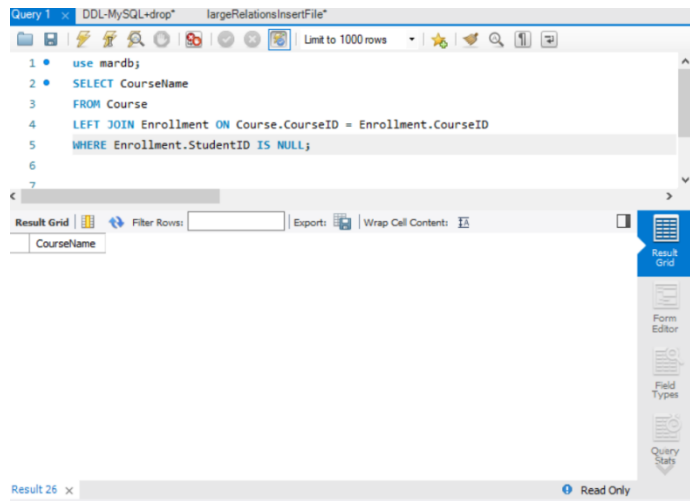
The results are displayed in a table with the following data:

ExamID	Name	Score
201	John Doe	92.50
202	Robert Johnson	95.50
203	Emily White	89.00
204	Michael Lee	94.50
205	Sarah Brown	91.00

19) Find the courses that no student has enrolled in.

Ans) use mardb;

```
SELECT CourseName
FROM Course
LEFT JOIN Enrollment ON Course.CourseID = Enrollment.CourseID
WHERE Enrollment.StudentID IS NULL;
```



20) Retrieve the faculty members who teachers courses with an average enrollment count above 10.

Ans) use mardb;

```

SELECT Faculty.Name, AVG(Enrollments) AS AverageEnrollments
FROM Faculty
JOIN Teaching ON Faculty.FacultyID = Teaching.FacultyID
JOIN (
SELECT Course.CourseID, COUNT(Enrollment.StudentID) AS Enrollments
FROM Course
LEFT JOIN Enrollment ON Course.CourseID = Enrollment.CourseID
GROUP BY Course.CourseID
) AS CourseEnrollments ON Teaching.CourseID = CourseEnrollments.CourseID
GROUP BY Faculty.Name
HAVING AVG(Enrollments) > 10;

```

