```
# 1. Retrieve the names and email addresses of all students.
Execute
SELECT `Name`, email FROM student_table  35ms
```

tudent_table ×

Search results | Free | 1 | Cost: 44ms | 1 | Total 8

	Name varchar(225)	email varchar(225)
0	Filter	Filter
1	John Doe	john.doe@example.com
2	Jane Smith	jane.smith@example.com
3	Robert Johnson	robert.j@example.com
4	Emily White	emily.white@example.com
5	Michael Lee	michael.lee@example.com
6	Sarah Brown	sarah.brown@example.com
7	David Clark	david.clark@example.com
8	Melissa Turner	melissa.turner@example.co

```
5   #2.  Find the courses that have more than three credits.
    ▷ Execute
6   SELECT courseName FROM course_table WHERE credits>3  18ms
```

course_table

Search results | Free | 1 | Cost: 27ms | 1 | Total 4

	courseName varchar(225)
0	Filter
1	History
2	Chemistry
3	Physics
4	Biology

```
8   #3) List the exams scheduled after November 15, 2023.
    ▷ Execute
9   SELECT Exam_id,ExamDate,ExamTime FROM exam_table WHERE ExamDate>"2023-11-15"  15ms
```

exam_table

Search results | Free | 1 | Cost: 63ms | 1 | Total

	Exam_id int	ExamDate date	ExamTime time
0	Filter	Filter	Filter
1	204	2023-11-18	03:15:00
2	205	2023-11-20	01:00:00

```
11  #4) Get the faculty members who work in the "Mathematics" department.
    Execute
12  SELECT Faculty_id, `Name` FROM faculty_table WHERE `Department`="Mathematics"  6ms
```

faculty_table

Search results | Free | Cost: 22ms | 1 | Total 1

	Faculty_id int	Name varchar(25!
0	Filter	Filter
1	301	Dr. Smith

```
14  #5) Retrieve the courses that each student is enrolled in.
    Execute
15  SELECT student_Table.Name, course_Table.courseName
16  FROM student_table
17  JOIN enrollment_table ON student_Table.`Student_id` = enrollment_Table.`Student_id`
18  JOIN course_table ON enrollment_Table.course_id = course_Table.course_id;  18ms
```

Result

Search results | Free | Cost: 18ms | 1 | Total 8

	Name varchar	courseName varchar
0	Filter	Filter
1	John Doe	Mathematics
2	Jane Smith	History
3	Robert Johnson	Computer Science
4	Emily White	Literature
5	Michael Lee	Chemistry
6	Sarah Brown	Physics
7	David Clark	Economics
8	Melissa Turner	Biology

```
19
20  #6) Find the average score for each exam.
    ▷ Execute
21  SELECT Exam_id, AVG(Score) AS AverageScore
22  FROM examresults_table
23  GROUP BY `Exam_id`  35ms
```

examresults_table

Search results | Free | Cost: 50ms | 1 | Total 5

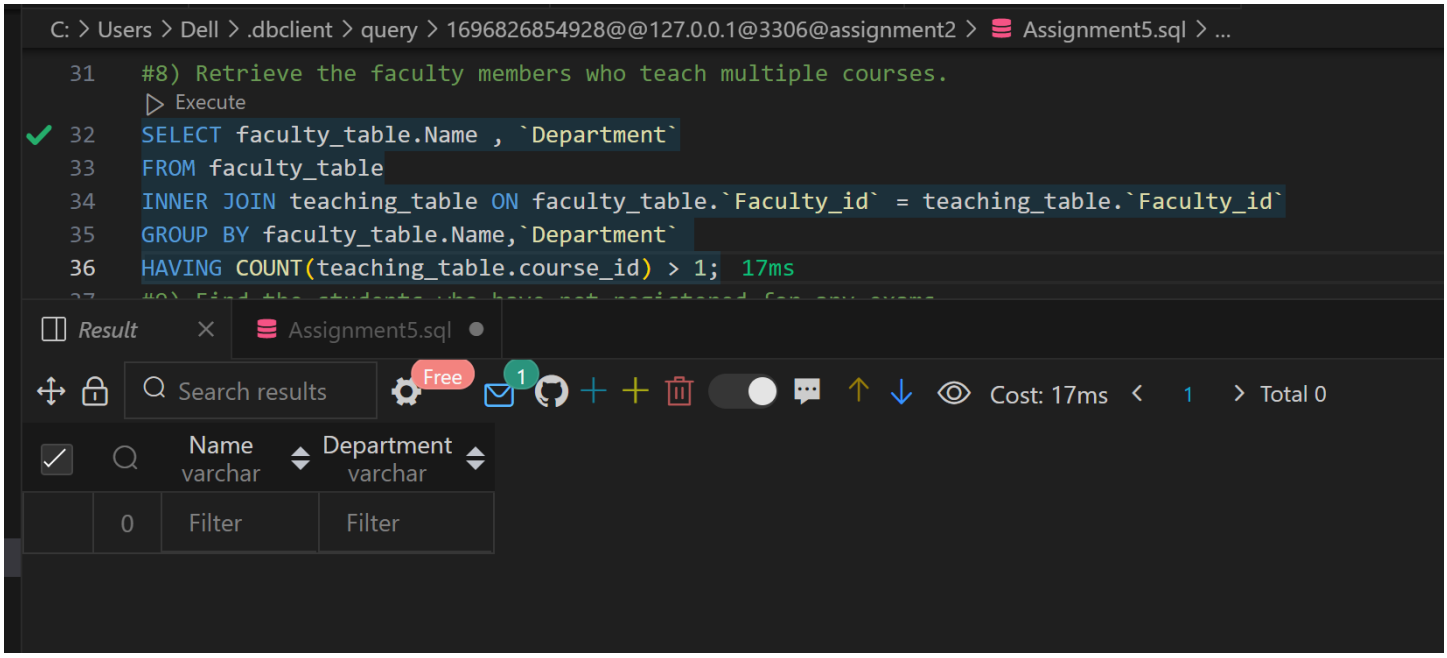
	Exam_id int	AverageScore newdecimal
0	Filter	Filter
1	201	89.333333
2	202	95.500000
3	203	89.000000
4	204	94.500000
5	205	91.000000

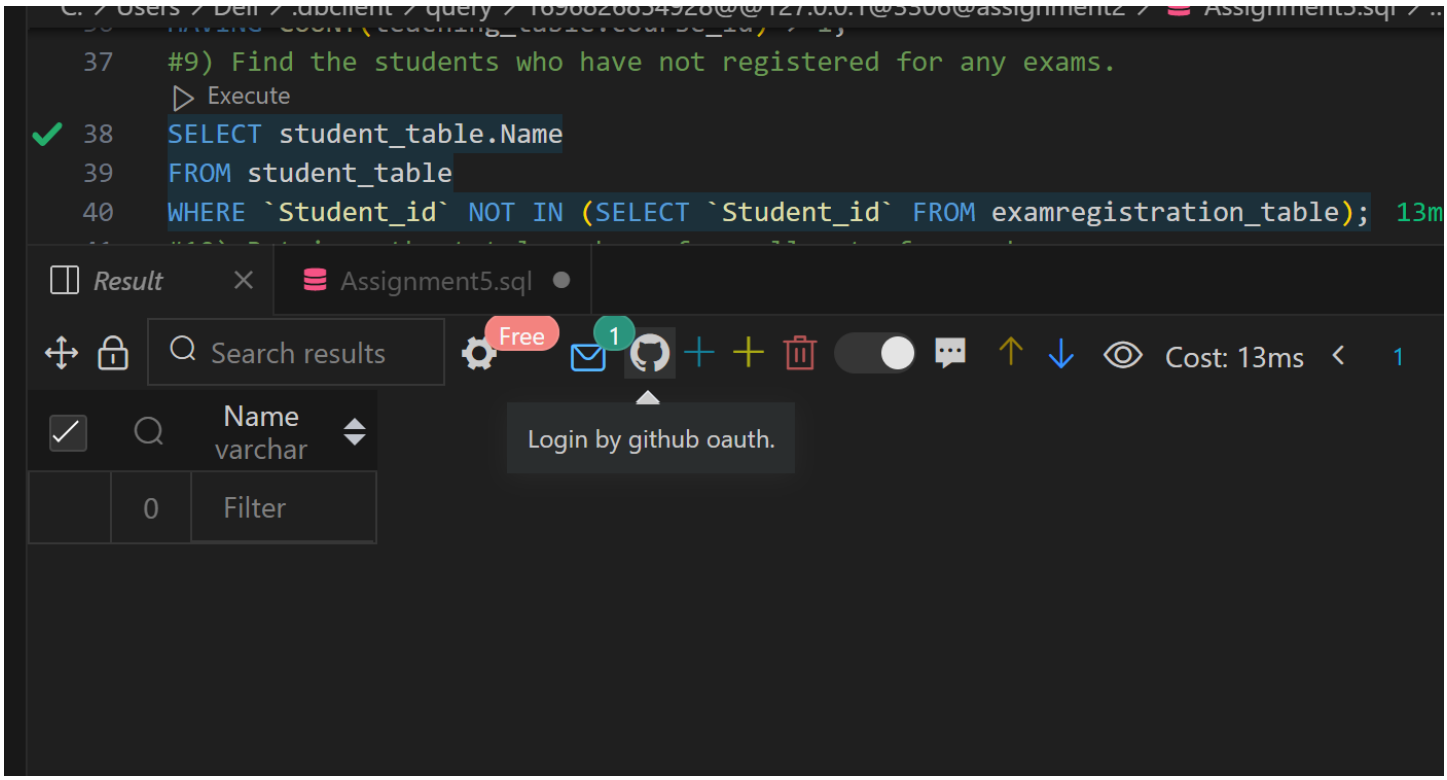
```
25  #7) List the students who scored above 90 on any exam.
    ▷ Execute
26  SELECT DISTINCT student_table.`Name`,examresults_table.Score
27  FROM student_table
28  JOIN examresults_table ON student_table.`Student_id`=examresults_table.`Student_id`
29  WHERE `Score`>90  32ms
30
```

Result

Search results | Free | Cost: 32ms | 1 | Total 4

	Name varchar	Score newdecima
0	Filter	Filter
1	John Doe	92.50
2	Robert Johnson	95.50
3	Michael Lee	94.50
4	Sarah Brown	91.00


C: > Users > Dell > .dbclient > query > 1696826854928@@127.0.0.1@3306@assignment2 > Assignment5.sql > ...
31 #8) Retrieve the faculty members who teach multiple courses.
Execute
32 SELECT faculty_table.Name , `Department`
33 FROM faculty_table
34 INNER JOIN teaching_table ON faculty_table.`Faculty_id` = teaching_table.`Faculty_id`
35 GROUP BY faculty_table.Name,`Department`
36 HAVING COUNT(teaching_table.course_id) > 1; 17ms
Result
Assignment5.sql
Search results
Free
1
Cost: 17ms
1
Total 0
Name
varchar
Department
varchar
0
Filter
Filter


37 #9) Find the students who have not registered for any exams.
Execute
38 SELECT student_table.Name
39 FROM student_table
40 WHERE `Student_id` NOT IN (SELECT `Student_id` FROM examregistration_table); 13m
Result
Assignment5.sql
Search results
Free
1
Cost: 13ms
1
Login by github oauth.
Name
varchar
0
Filter

C: > Users > Dell > .dbclient > query > 1696826854928@@127.0.0.1@3306@assignment2 > Assignment5.sql > ...

```
41  #10) Retrieve the total number of enrollments for each course.
    ▷ Execute
42  SELECT course_table.courseName, COUNT(enrollment_table.course_id) AS TotalEnrollments
43  FROM course_table
44  LEFT JOIN enrollment_table ON course_table.course_id = enrollment_table.course_id
45  GROUP By course_table.course_id;  9ms
```

Result | Assignment5.sql

Search results | Free | Cost: 9ms | 1 | Total 8

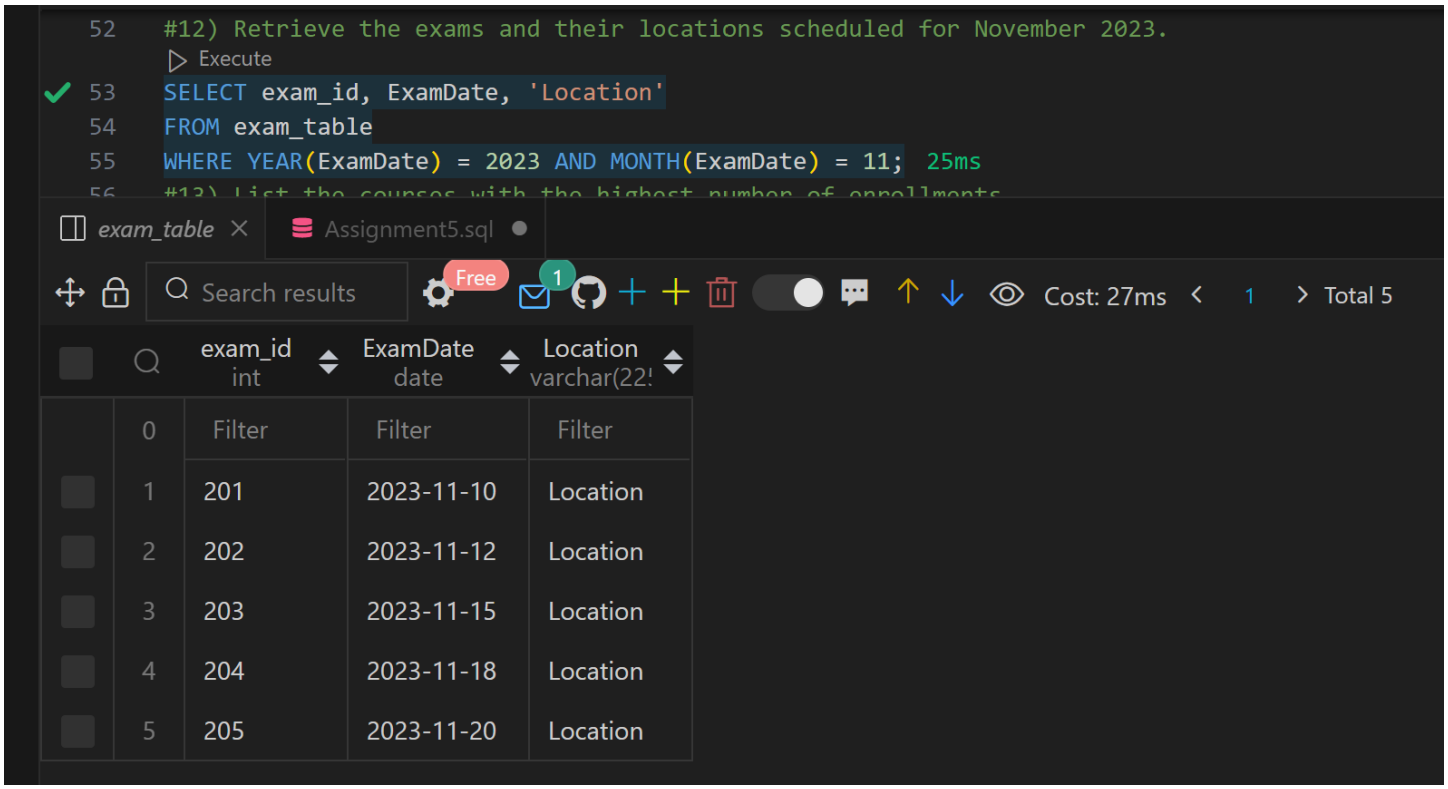
	courseName varchar	TotalEnrollments bigint
0	Filter	Filter
1	Mathematics	1
2	History	1
3	Computer Science	1
4	Literature	1
5	Chemistry	1
6	Physics	1
7	Economics	1
8	Biology	1

```
46  #11) Find the students who are enrolled in the "History" course.
    ▷ Execute
47  SELECT student_table.Name
48  FROM student_table
49  JOIN enrollment_table ON student_table.`Student_id` = enrollment_table.`Student_id`
50  JOIN course_table ON enrollment_table.course_id = course_table.course_id
51  WHERE course_table.`courseName` = "History";  18ms
```

Result | Assignment5.sql

Search results | Free | Cost: 18ms | 1 | Total 1

	Name varchar
0	Filter
1	Jane Smith


52 #12) Retrieve the exams and their locations scheduled for November 2023.
Execute
53 SELECT exam_id, ExamDate, 'Location'
54 FROM exam_table
55 WHERE YEAR(ExamDate) = 2023 AND MONTH(ExamDate) = 11; 25ms
exam_table
Assignment5.sql
Search results
Free
1
Cost: 27ms
1
Total 5
exam_id int
ExamDate date
Location varchar(22
0 Filter Filter Filter
1 201 2023-11-10 Location
2 202 2023-11-12 Location
3 203 2023-11-15 Location
4 204 2023-11-18 Location
5 205 2023-11-20 Location

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

C: > Users > Dell > .dbclient > query > 1696826854928@@127.0.0.1@3306@assignment2 > Assignment5.sql > ...

```
57  SELECT c.courseName, MAX(e.Enrollments) AS MaxEnrollments
58  FROM course_table c
59  LEFT JOIN (
60   SELECT course_id, COUNT(student_id) AS Enrollments
61   FROM enrollment_table e
62   GROUP BY course_id
63  ) e ON c.course_id= e.course_id
64  GROUP BY c.courseName;  3ms
```

Result

Search results | Free | 1 | Cost: 3ms | 1 | Total 8

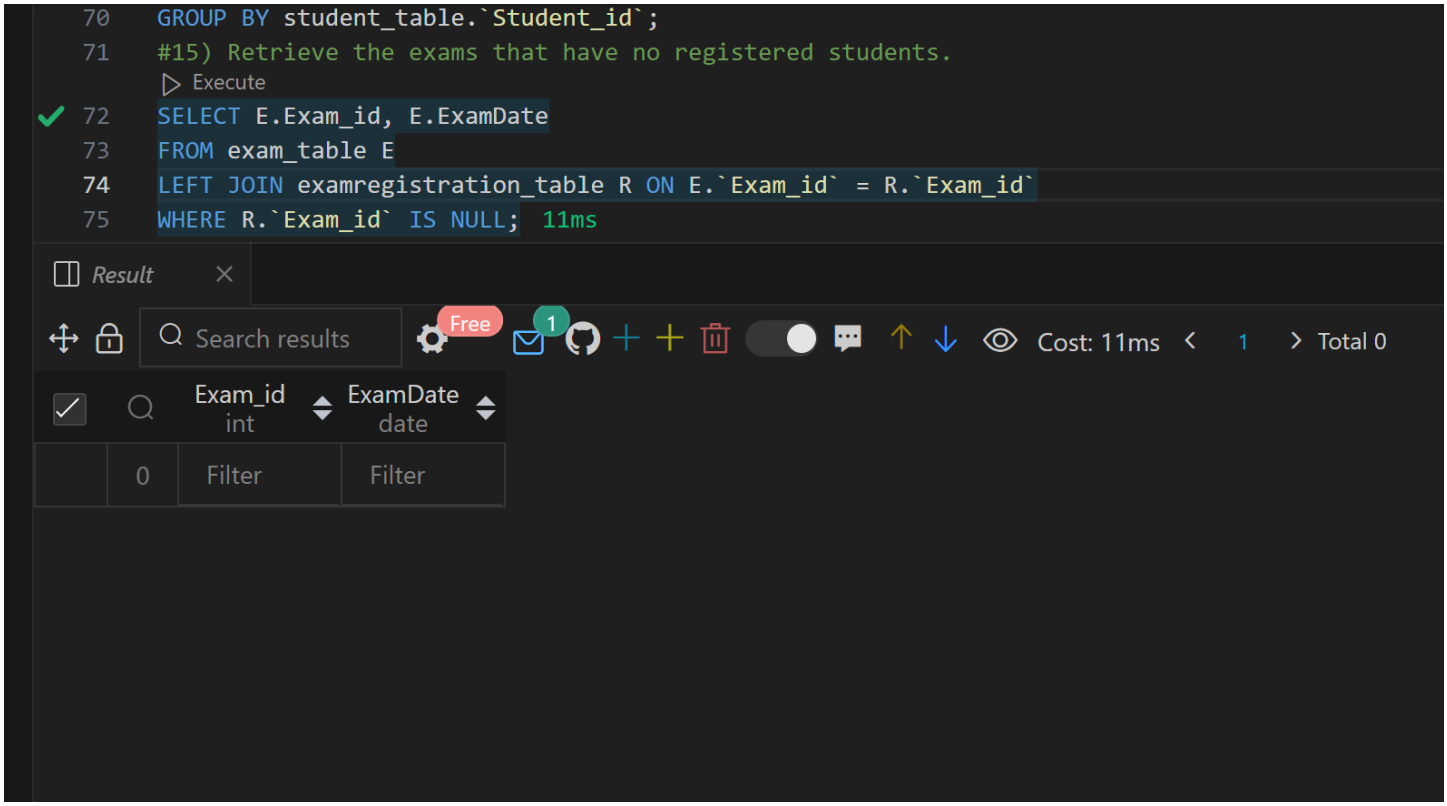
	courseName varchar	MaxEnrollments bigint
0	Filter	Filter
1	Mathematics	2
2	History	1
3	Computer Science	1
4	Literature	1
5	Chemistry	1
6	Physics	1
7	Economics	1
8	Biology	1

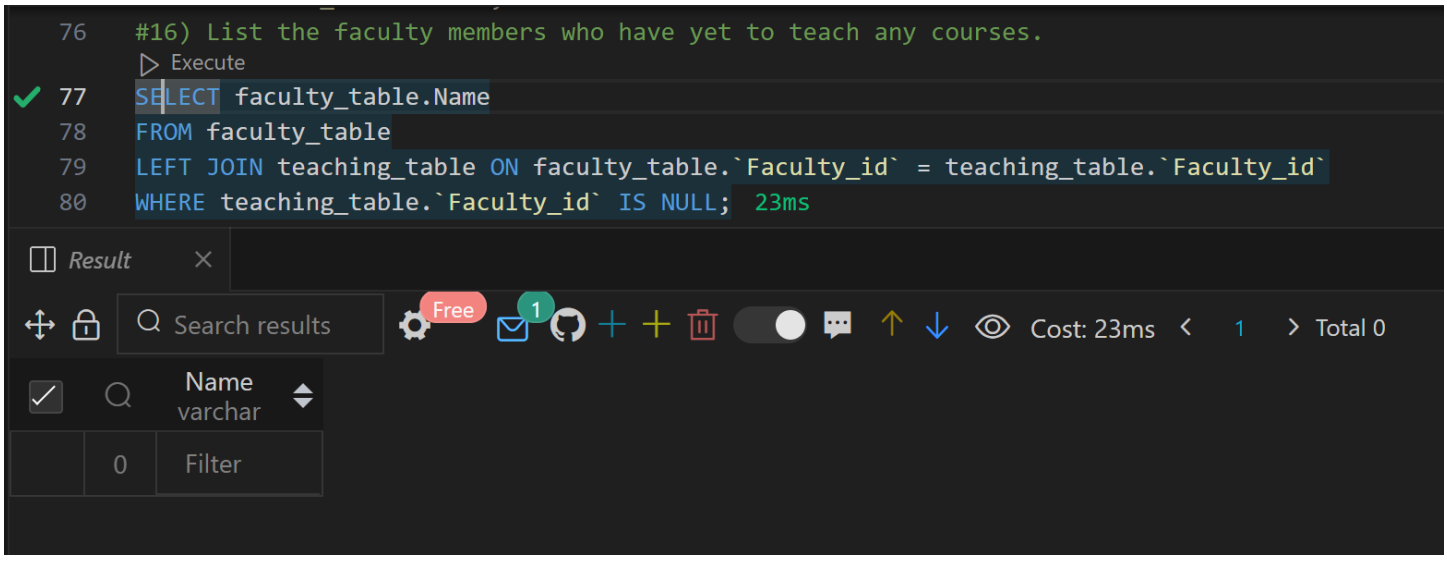
```sql
#14) Find the average score for each student.
Execute
SELECT student_table.Name, AVG(Score) AS AverageScore
FROM student_table
JOIN examresults_table ON student_table.`Student_id` = student_table.`Student_id`
GROUP BY student_table.`Student_id`;  20ms
```

Result

Search results | Free | 1 | Cost: 20ms | 1 | Total 8

	Name varchar	AverageScore newdecimal
0	Filter	Filter
1	John Doe	91.142857
2	Jane Smith	91.142857
3	Robert Johnson	91.142857
4	Emily White	91.142857
5	Michael Lee	91.142857
6	Sarah Brown	91.142857
7	David Clark	91.142857
8	Melissa Turner	91.142857


70 GROUP BY student_table.`Student_id`;
71 #15) Retrieve the exams that have no registered students.
Execute
72 SELECT E.Exam_id, E.ExamDate
73 FROM exam_table E
74 LEFT JOIN examregistration_table R ON E.`Exam_id` = R.`Exam_id`
75 WHERE R.`Exam_id` IS NULL; 11ms
Result
Search results
Free
Cost: 11ms
1
Total 0
Exam_id int
ExamDate date
0
Filter
Filter


76 #16) List the faculty members who have yet to teach any courses.
Execute
77 SELECT faculty_table.Name
78 FROM faculty_table
79 LEFT JOIN teaching_table ON faculty_table.`Faculty_id` = teaching_table.`Faculty_id`
80 WHERE teaching_table.`Faculty_id` IS NULL; 23ms
Result
Search results
Free
Cost: 23ms
1
Total 0
Name varchar
0
Filter

```sql
#17) Find the students who have registered for exams in both "Mathematics" and "Computer Science" departments.
SELECT student_table.Name
FROM student_table
JOIN enrollment_table ON student_table.`Student_id` =enrollment_table.`Student_id`
JOIN course_table ON enrollment_table.course_id=course_table.course_id
JOIN teaching_table ON course_table.course_id = teaching_table.course_id
JOIN faculty_table ON teaching_table.`Faculty_id` = faculty_table.`Faculty_id`
WHERE faculty_table.`Department` IN ('Mathematics', 'Computer Science')
GROUP BY student_table.`Student_id`
HAVING COUNT(DISTINCT faculty_table.`Department`) = 2;
```

Result

Search results | Free | Cost: 12ms | 1 | Total 0

Delete select rows

	Name varchar
0	Filter

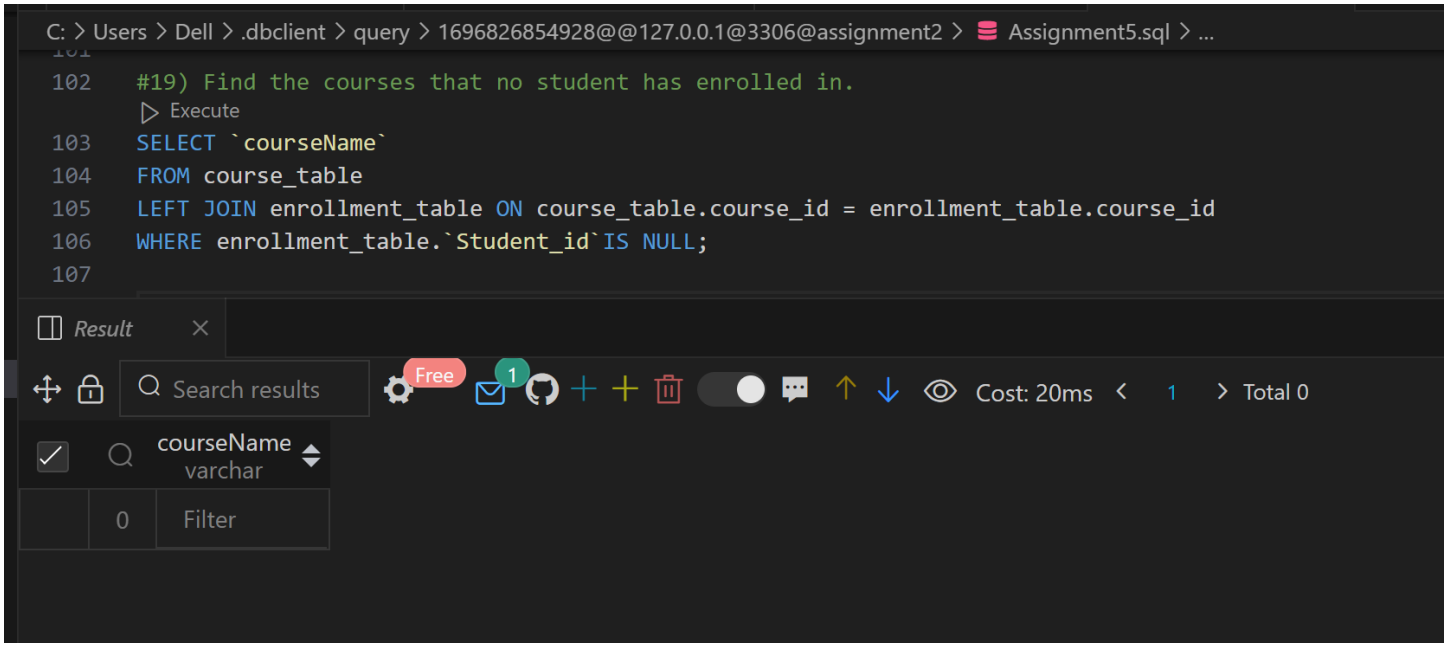
C: > Users > Dell > .dbclient > query > 1696826854928@@127.0.0.1@3306@assignment2 > Assignment5.sql > ...

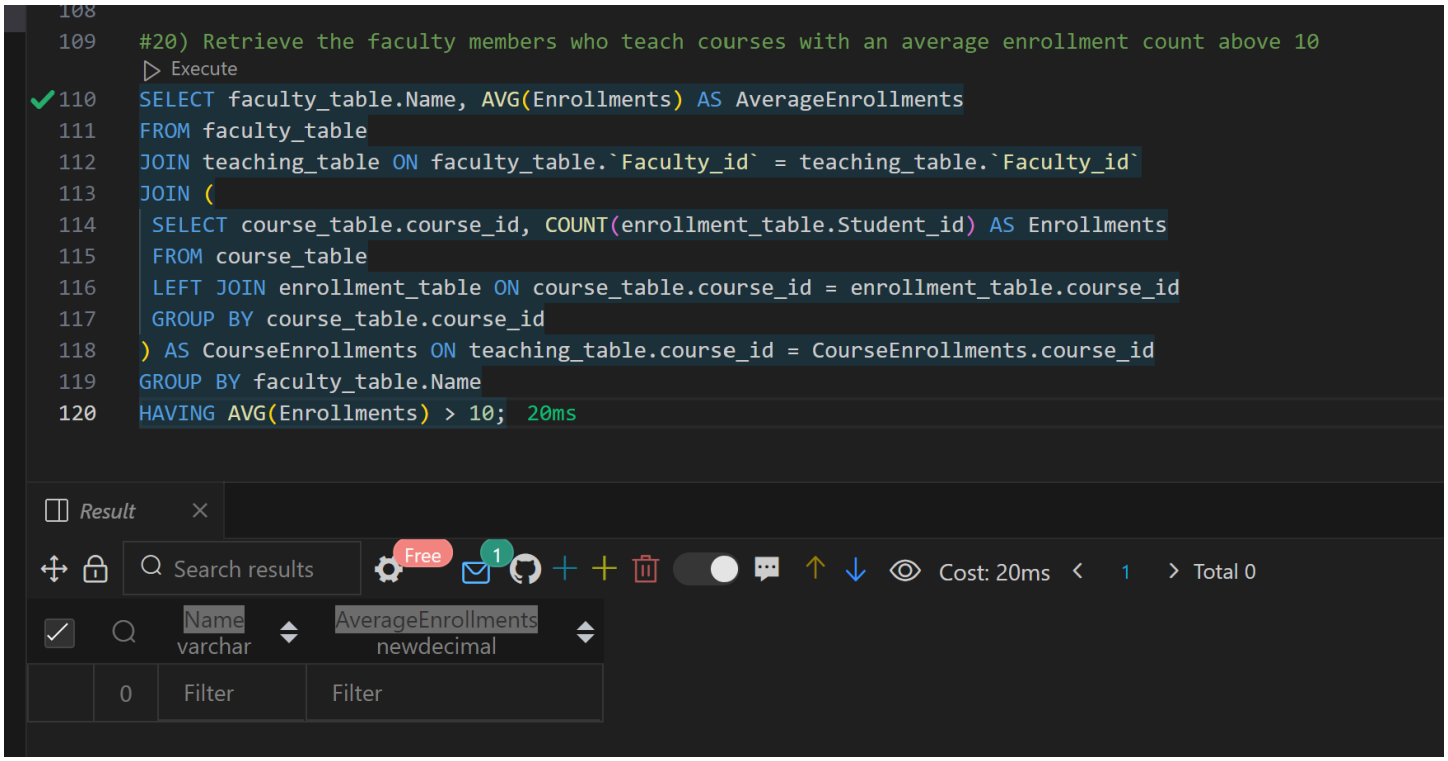
```sql
#18) Retrieve the students who scored the highest in each exam.
SELECT examresults_table.Exam_id, student_table.Name, examresults_table.Score
FROM examresults_table
JOIN student_table ON examresults_table.`Student_id` = student_table.`Student_id`
WHERE (examresults_table.`Exam_id`, examresults_table.Score) IN (
 SELECT `Exam_id`, MAX(Score)
 FROM examresults_table
 GROUP BY `Exam_id`);  33ms
```

Result

Search results | Free | Cost: 33ms | 1 | Total 5

	Exam_id int	Name varchar	Score newdecima
0	Filter	Filter	Filter
1	201	John Doe	92.50
2	202	Robert Johnson	95.50
3	203	Emily White	89.00
4	204	Michael Lee	94.50
5	205	Sarah Brown	91.00


C: > Users > Dell > .dbclient > query > 1696826854928@@127.0.0.1@3306@assignment2 > Assignment5.sql > ...
#19) Find the courses that no student has enrolled in.
Execute
SELECT `courseName`
FROM course_table
LEFT JOIN enrollment_table ON course_table.course_id = enrollment_table.course_id
WHERE enrollment_table.`Student_id`IS NULL;
Result
Search results
Free
Cost: 20ms
1
Total 0
courseName
varchar
0
Filter


#20) Retrieve the faculty members who teach courses with an average enrollment count above 10
Execute
SELECT faculty_table.Name, AVG(Enrollments) AS AverageEnrollments
FROM faculty_table
JOIN teaching_table ON faculty_table.`Faculty_id` = teaching_table.`Faculty_id`
JOIN (
SELECT course_table.course_id, COUNT(enrollment_table.Student_id) AS Enrollments
FROM course_table
LEFT JOIN enrollment_table ON course_table.course_id = enrollment_table.course_id
GROUP BY course_table.course_id
) AS CourseEnrollments ON teaching_table.course_id = CourseEnrollments.course_id
GROUP BY faculty_table.Name
HAVING AVG(Enrollments) > 10; 20ms
Result
Search results
Free
Cost: 20ms
1
Total 0
Name
varchar
AverageEnrollments
newdecimal
0
Filter
Filter