

ASSIGNMENT-4

1) Retrieve the names and email addresses of all students.

The screenshot shows the VS Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** Assignment2
- Explorer:** Shows files like connect.py, course_Table.py, create_db.py, database.py, enrollment_Table.py, exam_Table.py, examregistration_Table..., examresult_Table.py, Faculty_Table.py, insert_Table1.py, insert_Table2.py, insert_Table3.py, insert_Table4.py, insert_Table5.py, insert_Table6.py, insert_Table7.py, insert_Table8.py, query.py, query.sql, student_Table.py, Teaching_Table.py.
- Editor:** query.sql (Active Connection)

```
1  #1) Retrieve the names and email addresses of all students.  
2  | Execute  
3  use database;  
4  | Execute  
5  select Name , Email from student_table;
```
- Terminal:** student_table
- Output:** Shows the results of the query:

	Name	Email
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

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

The screenshot shows the VS Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** Assignment2
- Explorer:** Shows files like connect.py, course_Table.py, create_db.py, database.py, enrollment_Table.py, exam_Table.py, examregistration_Table..., examresult_Table.py, Faculty_Table.py, insert_Table1.py, insert_Table2.py, insert_Table3.py, insert_Table4.py, insert_Table5.py, insert_Table6.py, insert_Table7.py, insert_Table8.py, query.py, query.sql, student_Table.py, Teaching_Table.py.
- Editor:** query.sql (Active Connection)

```
6  #2)Find the courses that have more than three credits.  
7  | Execute  
8  select courseName,credits from course_table where credits>3; 79ms
```
- Terminal:** course_table
- Output:** Shows the results of the query:

	courseName	credits
1	History	4
2	Chemistry	4
3	Physics	4
4	Biology	4

3) List the exams scheduled after November 15, 2023.

The screenshot shows a code editor interface with several files listed in the Explorer pane. In the center, there is a code editor window with the following content:

```
File Edit Selection View Go Run ... ⏪ ⏪ Assignment2
EXPLORER ... query.py query.sql exam_Table.py
ASSIGNMENT2 ...
connect.py course_Table.py
course_Table.py create_db.py
create_db.py database.py
database.py enrollment_Table.py
enrollment_Table.py exam_Table.py
exam_Table.py examregistration_Table...
examregistration_Table... examresult_Table.py
examresult_Table.py Faculty_Table.py
Faculty_Table.py insert_Table1.py
insert_Table1.py insert_Table2.py
insert_Table2.py insert_Table3.py
insert_Table3.py insert_Table4.py
insert_Table4.py insert_Table5.py
insert_Table5.py insert_Table6.py
insert_Table6.py insert_Table7.py
insert_Table7.py insert_Table8.py
insert_Table8.py query.py
query.sql student_Table.py
student_Table.py Teaching_Table.py
Teaching_Table.py
```

```
query.sql > ...
6 #2)Find the courses that have more than three credits.
7 #select courseName,credits from course_table where credits>3;
8 #3) List the exams scheduled after November 15, 2023.
9 #select Exam_id,ExamDate from exam_Table where ExamDate> '2023-11-15';
10
11
12
```

Below the code editor is a results pane titled "exam_Table" showing the following data:

Exam_id	ExamDate
1	2023-11-18
2	2023-11-20

Cost: 117ms Total 2

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

The screenshot shows a code editor interface with several files listed in the Explorer pane. In the center, there is a code editor window with the following content:

```
File Edit Selection View Go Run ... ⏪ ⏪ Assignment2
EXPLORER ... query.py query.sql Faculty_Table.py exam_Table.py
ASSIGNMENT2 ...
connect.py course_Table.py
course_Table.py create_db.py
create_db.py database.py
database.py enrollment_Table.py
enrollment_Table.py exam_Table.py
exam_Table.py examregistration_Table...
examregistration_Table... examresult_Table.py
examresult_Table.py Faculty_Table.py
Faculty_Table.py insert_Table1.py
insert_Table1.py insert_Table2.py
insert_Table2.py insert_Table3.py
insert_Table3.py insert_Table4.py
insert_Table4.py insert_Table5.py
insert_Table5.py insert_Table6.py
insert_Table6.py insert_Table7.py
insert_Table7.py insert_Table8.py
insert_Table8.py query.py
query.sql student_Table.py
student_Table.py Teaching_Table.py
Teaching_Table.py
```

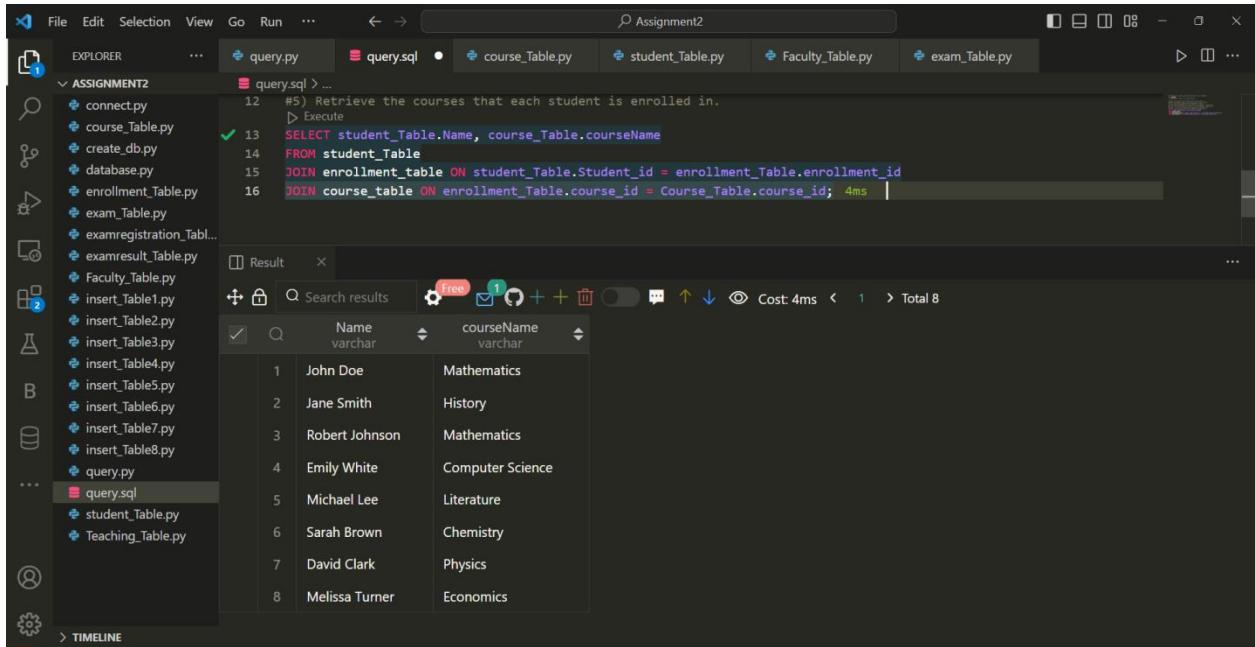
```
query.sql > ...
6 #2)Find the courses that have more than three credits.
7 #select courseName,credits from course_table where credits>3;
8 #3) List the exams scheduled after November 15, 2023.
9 #select Exam_id,ExamDate from exam_Table where ExamDate> '2023-11-15';
10 #4) Get the faculty members who work in the "Mathematics" department.
11 #select * from faculty_table where Department='Mathematics';
12
```

Below the code editor is a results pane titled "faculty_table" showing the following data:

Faculty_id	Name	email	Phone	Department
301	Dr. Smith	smith@example.com	111-222-3333	Mathematics

Cost: 88ms Total 1

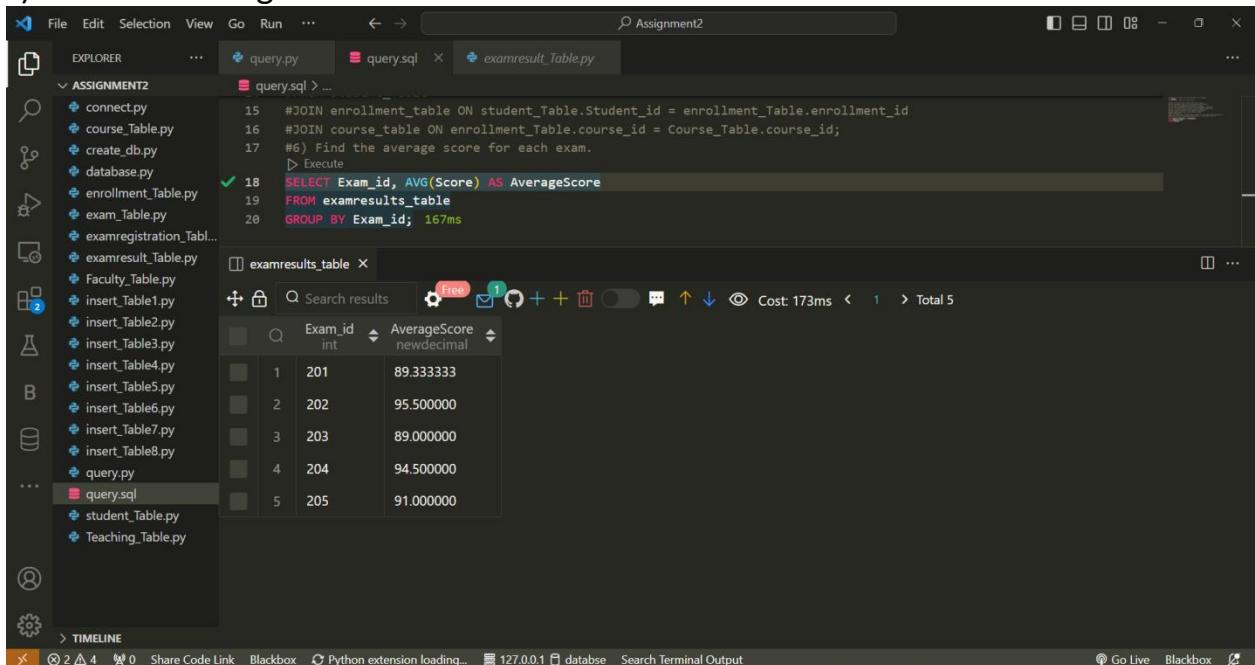
5) Retrieve the courses that each student is enrolled in.



```
SELECT student_Table.Name, course_Table.courseName
FROM student_Table
JOIN enrollment_table ON student_Table.Student_id = enrollment_Table.enrollment_id
JOIN course_table ON enrollment_Table.course_id = Course_Table.course_id;
```

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

6) Find the average score for each exam.



```
SELECT Exam_id, AVG(Score) AS AverageScore
FROM examsresults_table
GROUP BY Exam_id;
```

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

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

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows files like database.py, enrollment_Table.py, exam_Table.py, examregistration_Table..., examresult_Table.py, Faculty_Table.py, insert_Table1.py, insert_Table2.py, insert_Table3.py, insert_Table4.py, insert_Table5.py, insert_Table6.py, insert_Table7.py, insert_Table8.py, query.py, and query.sql.
- Code Editor:** Displays a SQL script named query.sql with the following code:

```

21  #6) Find the average score for each exam.
22  #SELECT Exam_id, AVG(Score) AS AverageScore
23  #FROM examresults_table
24  #GROUP BY Exam_id;
25
26  #7) List the students who scored above 90 on any exam.
27  SELECT DISTINCT student_Table.Name, examresults_Table.Score
28  FROM student_Table
29  JOIN examresults_Table ON student_Table.Student_id = examresults_Table.Student_id
30  WHERE Score > 90; 107ms
31

```
- Results Panel:** Shows the output of the query with the following data:

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

8) Retrieve the faculty members who teach multiple courses.

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows files like database.py, enrollment_Table.py, exam_Table.py, examregistration_Table..., examresult_Table.py, Faculty_Table.py, insert_Table1.py, insert_Table2.py, insert_Table3.py, insert_Table4.py, insert_Table5.py, insert_Table6.py, insert_Table7.py, insert_Table8.py, query.py, and query.sql.
- Code Editor:** Displays a SQL script named query.sql with the following code:

```

27  #SELECT DISTINCT student_Table.Name, examresults_Table.Score
28  #FROM student_Table
29  #JOIN examresults_Table ON student_Table.Student_id = examresults_Table.Student_id
30  #WHERE Score > 90;
31  #8) Retrieve the faculty members who teach multiple courses.
32  SELECT Faculty_table.Name, COUNT(*) AS CourseCount
33  FROM teaching_table
34  JOIN faculty_table ON teaching_table.Faculty_id = faculty_table.Faculty_id
35  GROUP BY Faculty_table.Faculty_id
36  HAVING CourseCount > 1; 56ms
37  #9) Find the students who have not registered for any exams.
38  #SELECT student_Table.Name
39

```
- Results Panel:** Shows the output of the query with the following data:

	Name	CourseCount
1	Prof. Adams	2

9) Find the students who have not registered for any exams.

The screenshot shows the Visual Studio Code interface with the title bar "Assignment2". The Explorer sidebar on the left lists files under "ASSIGNMENT2" such as database.py, enrollment_Table.py, exam_Table.py, examregistration_Table..., examresult_Table.py, Faculty_Table.py, insert_Table1.py, insert_Table2.py, insert_Table3.py, insert_Table4.py, insert_Table5.py, insert_Table6.py, insert_Table7.py, insert_Table8.py, query.py, query.sql, student_Table.py, and Teaching_Table.py. The "TIMELINE" section indicates that the active editor cannot provide timeline information. The main editor area contains a SQL script with the following code:

```
26 #7) List the students who scored above 90 on any exam.
27 #SELECT DISTINCT student_Table.Name, examresults_Table.Score
28 #FROM student_Table
29 #JOIN examresults_Table ON student_Table.Student_id = examresults_Table.Student_id
30 #WHERE Score > 90;
31
32 #9) Find the students who have not registered for any exams.
33 ▷ Execute
34 SELECT student_Table.Name
35 FROM student_table
36 WHERE Student_id NOT IN (SELECT Student_id FROM examregistration_table); 138ms
37 ##FROM course_Table
```

The "Result" tab shows the output of the last executed query:

Name	varchar

The status bar at the bottom shows "12°C Cloudy" and the date "18-11-2023".

10) Retrieve the total number of enrollments for each course.

The screenshot shows the Visual Studio Code interface with the title bar "Assignment2". The Explorer sidebar on the left lists files under "ASSIGNMENT2" such as connect.py, course_Table.py, create_db.py, database.py, enrollment_Table.py, exam_Table.py, examregistration_Table..., examresult_Table.py, Faculty_Table.py, insert_Table1.py, insert_Table2.py, insert_Table3.py, insert_Table4.py, insert_Table5.py, insert_Table6.py, insert_Table7.py, insert_Table8.py, query.py, and query.sql. The "TIMELINE" section shows file save history. The main editor area contains a SQL script with the following code:

```
40 #WHERE Student_id NOT IN (SELECT Student_id FROM examregistration_table);
41 #10) Retrieve the total number of enrollments for each course.
42 ▷ Execute
43 SELECT Course_table.courseName, COUNT(enrollment_table.course_id) AS TotalEnrollments
44 FROM course_Table
45 LEFT JOIN enrollment_Table ON course_Table.course_id = enrollment_Table.course_id
46 GROUP BY course_Table.course_id; 40ms
47 #11) Find the students who are enrolled in the "History" course.
```

The "Result" tab shows the output of the last executed query:

courseName	TotalEnrollments
Mathematics	2
History	1
Computer Science	1
Literature	1
Chemistry	1
Physics	1
Economics	1
Biology	1

The status bar at the bottom shows "12°C Cloudy" and the date "18-11-2023".

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

```
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
WHERE Course_table.courseName = 'History';
```

Name
John Doe

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

```
SELECT Exam_id, ExamDate, Location
FROM exam_table
WHERE YEAR(ExamDate) = 2023 AND MONTH(ExamDate) = 11;
```

Exam_id	ExamDate	Location
1	2023-11-10	Exam Hall A
2	2023-11-12	Exam Hall B
3	2023-11-15	Exam Hall C
4	2023-11-18	Exam Hall D
5	2023-11-20	Exam Hall E

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

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows files in the `ASSIGNMENT2` folder, including `create_db.py`, `database.py`, `enrollment_Table.py`, `exam_Table.py`, `examregistration_Table.py`, `examresult_Table.py`, `Faculty_Table.py`, `insert_Table1.py`, `insert_Table2.py`, `insert_Table3.py`, `insert_Table4.py`, `insert_Table5.py`, `insert_Table6.py`, `insert_Table7.py`, `insert_Table8.py`, `query.py`, `query.sql` (selected), `student_Table.py`, and `Teaching_Table.py`. A timeline for `query.sql` is also visible.
- Code Editor:** Displays the following SQL query:


```

56    #15) List the courses with the highest number of enrollments.
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
62        GROUP BY course_id
63    ) e ON c.course_id = e.course_id
64    GROUP BY c.courseName; 96ms
      
```
- SQL Explorer:** Shows the results of the query in a table:

	courseName	MaxEnrollments
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
- Bottom Bar:** Includes icons for Share Code Link, Generate Commit Message, Blackbox, Python extension loading..., 127.0.0.1 database, Search Terminal Output, CRLF, SQL, Go Live, Blackbox, Prettier, and system status like battery level and date/time.

14) Find the average score for each student.

The screenshot shows the VS Code interface with the following details:

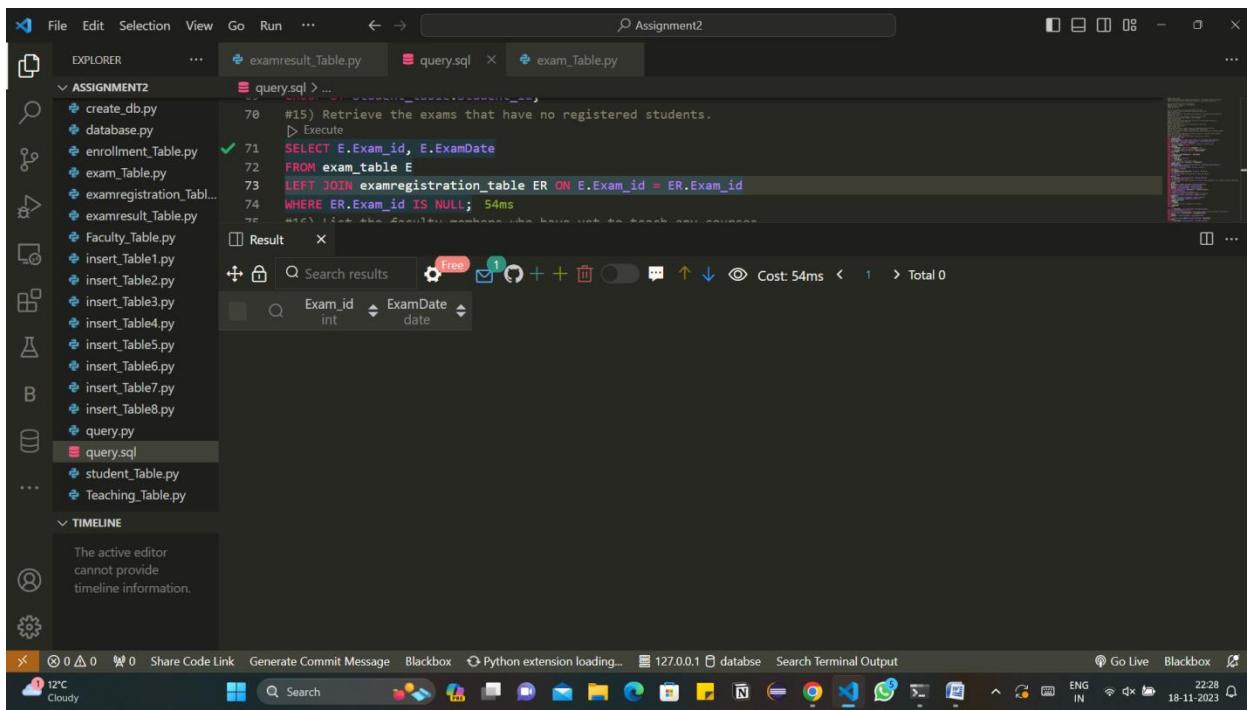
- File Explorer:** Shows files in the `ASSIGNMENT2` folder, including `create_db.py`, `database.py`, `enrollment_Table.py`, `exam_Table.py`, `examregistration_Table.py`, `examresult_Table.py`, `Faculty_Table.py`, `insert_Table1.py`, `insert_Table2.py`, `insert_Table3.py`, `insert_Table4.py`, `insert_Table5.py`, `insert_Table6.py`, `insert_Table7.py`, `insert_Table8.py`, `query.py`, `query.sql` (selected), `student_Table.py`, and `Teaching_Table.py`. A timeline for `query.sql` is also visible.
- Code Editor:** Displays the following SQL query:


```

65    #14) Find the average score for each student.
66    SELECT Student_table.Name, AVG(Score) AS AverageScore
67    FROM student_table
68    JOIN examsresults_table ON Student_table.Student_id = ExamResults_table.Student_id
69    GROUP BY Student_table.Student_id; 47ms
      
```
- SQL Explorer:** Shows the results of the query in a table:

	Name	AverageScore
1	John Doe	92.500000
2	Jane Smith	88.000000
3	Robert Johnson	95.500000
4	Emily White	89.000000
5	Michael Lee	94.500000
6	Sarah Brown	91.000000
7	David Clark	87.500000
- Bottom Bar:** Includes icons for Share Code Link, Generate Commit Message, Blackbox, Python extension loading..., 127.0.0.1 database, Search Terminal Output, CRLF, SQL, Go Live, Blackbox, Prettier, and system status like battery level and date/time.

15) Retrieve the exams that have no registered students.



File Edit Selection View Go Run ...

Assignment2

EXPLORER

ASSIGNMENT2

- create_db.py
- database.py
- enrollment_Table.py
- exam_Table.py
- examregistration_Table...
- examresult_Table.py
- Faculty_Table.py
- insert_Table1.py
- insert_Table2.py
- insert_Table3.py
- insert_Table4.py
- insert_Table5.py
- insert_Table6.py
- insert_Table7.py
- insert_Table8.py
- query.py
- query.sql
- student_Table.py
- Teaching_Table.py

TIMELINE

The active editor cannot provide timeline information.

query.sql > ...

```
70 #15) Retrieve the exams that have no registered students.
71
72 FROM exam_table E
73 LEFT JOIN examregistration_table ER ON E.Exam_id = ER.Exam_id
74 WHERE ER.Exam_id IS NULL; 54ms
```

Result

Exam_id	ExamDate

Search results

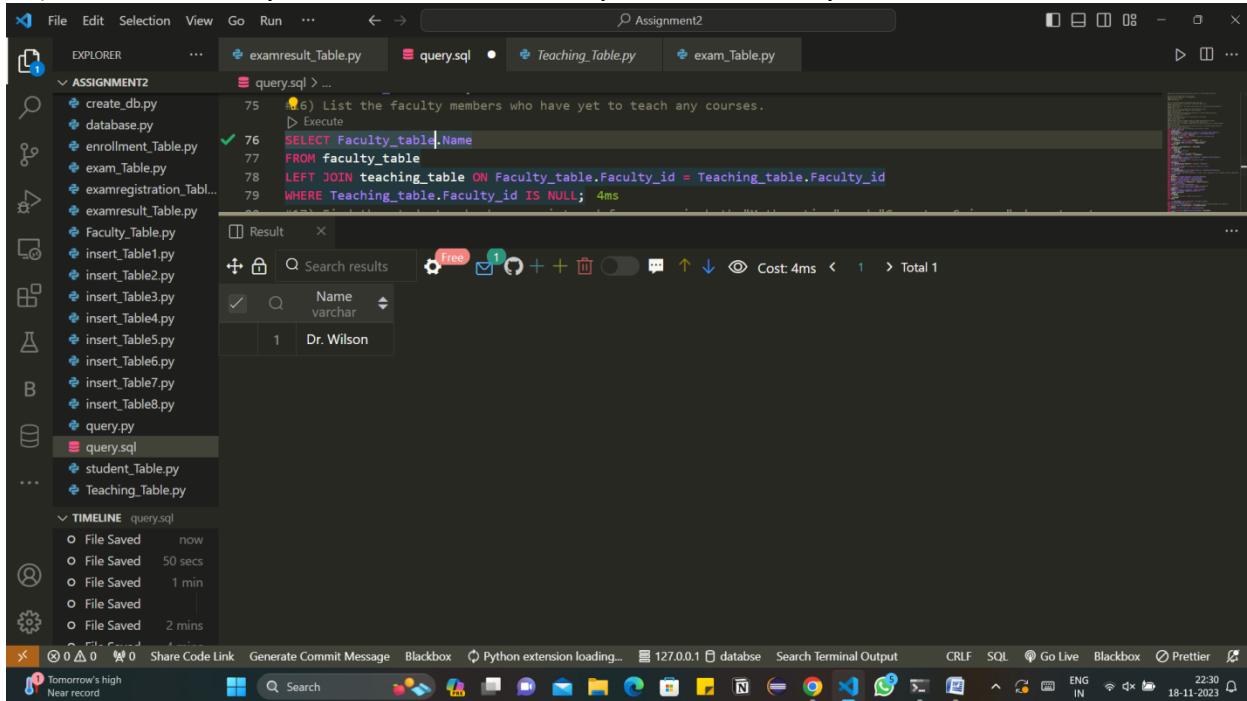
Cost: 54ms < 1 > Total 0

127.0.0.1 database Search Terminal Output

Share Code Link Generate Commit Message Blackbox Python extension loading... 127.0.0.1 database Search Terminal Output Go Live Blackbox

12°C Cloudy 18-11-2023

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



File Edit Selection View Go Run ...

Assignment2

EXPLORER

ASSIGNMENT2

- create_db.py
- database.py
- enrollment_Table.py
- exam_Table.py
- examregistration_Table...
- examresult_Table.py
- Faculty_Table.py
- insert_Table1.py
- insert_Table2.py
- insert_Table3.py
- insert_Table4.py
- insert_Table5.py
- insert_Table6.py
- insert_Table7.py
- insert_Table8.py
- query.py
- query.sql
- student_Table.py
- Teaching_Table.py

TIMELINE query.sql

- File Saved now
- File Saved 50 secs
- File Saved 1 min
- File Saved
- File Saved 2 mins

query.sql > ...

```
75 #16) List the faculty members who have yet to teach any courses.
76
77 FROM faculty_table
78 LEFT JOIN teaching_table ON Faculty_table.Faculty_id = Teaching_table.Faculty_id
79 WHERE Teaching_table.Faculty_id IS NULL; 4ms
```

Result

Name
Dr. Wilson

Search results

Cost: 4ms < 1 > Total 1

127.0.0.1 database Search Terminal Output CRLF SQL Go Live Blackbox Prettier

Tomorrow's high Near record 18-11-2023

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

```

    80 #17) Find the students who have registered for exams in both "Mathematics" and "Computer Science" departments.
    81 SELECT Student_table.Name
    82 FROM student_table
    83 JOIN enrollment_table ON Student_table.Student_id = Enrollment_table.Student_id
    84 JOIN course_table ON Enrollment_table.course_id = course_table.course_id
    85 JOIN teaching_table ON course_table.course_id = Teaching_table.course_id
    86 JOIN faculty_table ON Teaching_table.Faculty_id = Faculty_table.Faculty_id
    87 WHERE Faculty_table.Department IN ('Mathematics', 'Computer Science')
    88 GROUP BY Student_table.Student_id
    89 HAVING COUNT(DISTINCT Faculty_table.Department) = 2; 107ms
  
```

Result

Name	varchar

Search results

File Saved now

File Saved 1 min

File Saved 2 mins

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

```

    90 #18) Retrieve the students who scored the highest in each exam.
    91 SELECT ExamResults_table.Exam_id, Student_table.Name, ExamResults_table.Score
    92 FROM ExamResults_table
    93 JOIN student_table ON ExamResults_table.Student_id = Student_table.Student_id
    94 WHERE (ExamResults_table.Exam_id, ExamResults_table.Score) IN (
    95     SELECT Exam_id, MAX(Score)
    96     FROM ExamResults_table
    97     GROUP BY Exam_id); 72ms
  
```

Result

Exam_id	Name	Score
1	John Doe	92.50
2	Robert Johnson	95.50
3	Emily White	89.00
4	Michael Lee	94.50
5	Sarah Brown	91.00

File Saved now

File Saved 41 secs

File Saved 1 min

File Saved 2 mins

File Saved 3 mins

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

```
#19) Find the courses that no student has enrolled in.  
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; 33ms
```

The screenshot shows the Visual Studio Code interface with the 'query.sql' file open in the editor. The code is a SQL query to find courses that have no students enrolled. The result pane shows an empty table with columns 'CourseName' and 'varchar'. The timeline pane shows the history of saves for the file.

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

```
#20) Retrieve the faculty members who teach courses with an average enrollment count above 10.  
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; 36ms
```

The screenshot shows the Visual Studio Code interface with the 'query.sql' file open in the editor. The code is a complex SQL query to find faculty members whose average course enrollment is above 10. The result pane shows an empty table with columns 'Name' and 'AverageEnrollments'. The timeline pane shows the history of saves for the file.