National Institute of Technology Meghalaya

Assignment No: 05

Student Name: Subhasish Dutta

Roll Number: T23CS001

Programme: Master of Technology

Department: Computer Science & Engineering

Semester: 1

Course Name: ADVANCED DBMS LAB

Course Code: CS553

Assignment 05

Question No : 01

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

MySQL Code

```
import mysql.connector

# Connect to the MySQL server
conn = mysql.connector.connect(host='localhost', user='root',
password='shuvo634', database='shuvodb')
mycursor = conn.cursor()

# Define the SQL query to retrieve names and email addresses of all students
select_query = "SELECT Name, Email FROM Student"

# Execute the SQL query
mycursor.execute(select_query)

# Fetch all the results
results = mycursor.fetchall()

# Print the names and email addresses of all students
for row in results:
    name, email = row
    print("Name: {}, Email: {}".format(name, email))

# Close the connection
conn.close()
```

Output

```
PS C:\Users\Digital Outlet\Desktop\python_conncet_to_mysql> & "C:/Users/Digital Outlet/Ap
ital Outlet/Desktop/python_conncet_to_mysql/Assignment 4.py"
PS C:\Users\Digital Outlet\Desktop\python_conncet_to_mysql> & "C:/Users/Digital Outlet/Ap
ital Outlet/Desktop/python_conncet_to_mysql/Assignment 5.py"
Name: John Doe, Email: john.doe@example.com
Name: Jane Smith, Email: jane.smith@example.com
Name: Robert Johnson, Email: robert.j@example.com
Name: Emily White, Email: emily.white@example.com
Name: Michael Lee, Email: michael.lee@example.com
Name: Sarah Brown, Email: sarah.brown@example.com
Name: David Clark, Email: david clark@example.com
Name: Melissa Turner, Email: melissa.turner@example.com
PS C:\Users\Digital Outlet\Desktop\python_conncet_to_mysql>
```

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

MySQL Code

```
import mysql.connector

# Connect to the MySQL server
conn = mysql.connector.connect(host='localhost', user='root',
password='shuvo634', database='shuvodb')
mycursor = conn.cursor()

# Define the SQL query to find courses with more than three credits
select_query = "SELECT CourseName FROM Course WHERE Credits > 3"

# Execute the SQL query
mycursor.execute(select_query)

# Fetch all the results
results = mycursor.fetchall()

# Print the courses with more than three credits
for row in results:
    course_name = row[0]
    print("Course with more than three credits: {}".format(course_name))
```

```
# Close the connection
conn.close()
```

Output

```
PS C:\Users\Digital Outlet\Desktop\python_conncet_to_mysql> & "C:/Users/Digital Outlet/AppData/Local/Programs/Python/Python311/python.e
xe" "c:/Users/Digital Outlet/Desktop/python_conncet_to_mysql/Assignment 5-2.py"
Course with more than three credits: History
Course with more than three credits: Chemistry
Course with more than three credits: Physics
Course with more than three credits: Biology
```

3) List the exams scheduled after November 15, 2023

MySQL Code

```
import mysql.connector

# Connect to the MySQL server
conn = mysql.connector.connect(host='localhost', user='root',
password='shuvo634', database='shuvodb')
mycursor = conn.cursor()

# Define the SQL query to list exams scheduled after November 15, 2023
select_query = "SELECT * FROM Exam WHERE ExamDate > '2023-11-15'"

# Execute the SQL query
mycursor.execute(select_query)

# Fetch all the results
results = mycursor.fetchall()

# Print the exams scheduled after November 15, 2023
for row in results:
    exam_id, exam_date, exam_time, location = row
    print("Exam ID: {}, Date: {}, Time: {}, Location: {}".format(exam_id,
exam_date, exam_time, location))

# Close the connection
conn.close()
```

Output

```
PS C:\Users\Digital Outlet\Desktop\python_conncet_to_mysql> & "C:/Users/Digital Outlet/AppData/Local/Programs/Python/Python311/python.exe" "c:/Users/Digital Outlet/Desktop/python_conncet_to_mysql/Assignment 5-3.py"
Exam ID: 204, Date: 2023-11-18, Time: 15:15:00, Location: Exam Hall D
Exam ID: 205, Date: 2023-11-20, Time: 13:00:00, Location: Exam Hall E
PS C:\Users\Digital Outlet\Desktop\python_conncet_to_mysql>
```

---

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

MySQL Code

```python
import mysql.connector

# Connect to the MySQL server
conn = mysql.connector.connect(host='localhost', user='root',
password='shuvo634', database='shuvodb')
mycursor = conn.cursor()

# Define the SQL query to get faculty members in the "Mathematics" department
select_query = "SELECT Name, Email FROM Faculty WHERE Department = 'Mathematics'"

# Execute the SQL query
mycursor.execute(select_query)

# Fetch all the results
results = mycursor.fetchall()

# Print the faculty members in the "Mathematics" department
for row in results:
    name, email = row
    print("Name: {}, Email: {}".format(name, email))

# Close the connection
conn.close()
```

Output:-

```
PS C:\Users\Digital Outlet\Desktop\python_conncet_to_mysql> & "C:/Users/Digital Outlet/AppData/Local/Programs/Python/Python311/python.exe" "c:/Users/Digital Outlet/Desktop/python_conncet_to_mysql/Assignment 5-4.py"
Name: Dr. Smith, Email: smith@example.com
PS C:\Users\Digital Outlet\Desktop\python_conncet_to_mysql>
```

5) Retrieve the courses that each student is enrolled in

MySQL Code

```
import mysql.connector

# Connect to the MySQL server
conn = mysql.connector.connect(host='localhost', user='root',
password='shuvo634', database='shuvodb')
mycursor = conn.cursor()

# Define the SQL query to retrieve the courses that each student is enrolled in
select_query = """
SELECT Student.Name, Course.CourseName
FROM Student
JOIN Enrollment ON Student.StudentID = Enrollment.StudentID
JOIN Course ON Enrollment.CourseID = Course.CourseID
"""

# Execute the SQL query
mycursor.execute(select_query)

# Fetch all the results
results = mycursor.fetchall()

# Create a dictionary to store courses for each student
student_courses = {}

# Populate the dictionary with student-course information
for row in results:
    student_name, course_name = row
    if student_name not in student_courses:
        student_courses[student_name] = []
    student_courses[student_name].append(course_name)
```

```python
# Print the courses for each student
for student, courses in student_courses.items():
    print("Student: {}".format(student))
    for course in courses:
        print(" - Course: {}".format(course))

# Close the connection
conn.close()
```

Output

```
Student: Jane Smith
 - Course: Mathematics
Student: Robert Johnson
 - Course: Computer science
Student: Emily White
 - Course: Literature
Student: Michael Lee
 - Course: Chemistry
Student: Sarah Brown
 - Course: Physics
Student: David Clark
 - Course: Economics
Student: Melissa Turner
 - Course: Biology
PS C:\Users\Digital Outlet\Desktop\python_conncet_to_mysql>
```

6) **Find the average score for each exam**

MySQL Code

```python
import mysql.connector

# Connect to the MySQL server
conn = mysql.connector.connect(host='localhost', user='root',
password='shuvo634', database='shuvodb')
mycursor = conn.cursor()
```

```python
# Define the SQL query to find the average score for each exam
select_query = """
SELECT ExamID, AVG(score) as AverageScore
FROM ExamResults
GROUP BY ExamID
"""

# Execute the SQL query
mycursor.execute(select_query)

# Fetch all the results
results = mycursor.fetchall()

# Print the average score for each exam
for row in results:
    exam_id, average_score = row
    print("Exam ID: {}, Average Score: {}".format(exam_id, average_score))

# Close the connection
conn.close()
```

Output

```
PS C:\Users\Digital Outlet\Desktop\python_conncet_to_mysql> & "C:/Users/Digital Outlet/AppData/Local/Programs/Python/Python311/python.exe" "c:/Users/Digital Outlet/Desktop/python_conncet_to_mysql/Assignment 5-6.py"
Exam ID: 201, Average Score: 89.333333
Exam ID: 202, Average Score: 95.500000
Exam ID: 203, Average Score: 89.000000
Exam ID: 204, Average Score: 94.500000
Exam ID: 205, Average Score: 91.000000
PS C:\Users\Digital Outlet\Desktop\python_conncet_to_mysql>
```

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

MySQL Code

```
import mysql.connector

# Connect to the MySQL server
conn = mysql.connector.connect(host='localhost', user='root',
password='shuvo634', database='shuvodb')
mycursor = conn.cursor()

# Define the SQL query to list students who scored above 90 on any exam
select_query = """
SELECT DISTINCT Student.Name, ExamResults.score
FROM Student
JOIN ExamResults ON Student.StudentID = ExamResults.StudentID
WHERE ExamResults.score > 90
"""

# Execute the SQL query
mycursor.execute(select_query)

# Fetch all the results
results = mycursor.fetchall()

# Print the students who scored above 90 on any exam
for row in results:
    student_name, score = row
    print("Student: {}, Score: {}".format(student_name, score))

# Close the connection
conn.close()
```

Output

```
Exam ID: 205, Average Score: 91.000000
PS C:\Users\Digital Outlet\Desktop\python_conncet_to_mysql> & "C:/Users/Digital Outlet/AppData/Local/Programs/Python/Python311/python.exe" "c:/Users/Digital Outlet/Desktop/python_conncet_to_mysql/Assignment 5-7.py"
Student: John Doe, Score: 92.50
Student: Robert Johnson, Score: 95.50
Student: Michael Lee, Score: 94.50
Student: Sarah Brown, Score: 91.00
PS C:\Users\Digital Outlet\Desktop\python_conncet_to_mysql>
```

8) **Retrieve the faculty members who teach multiple courses**

MySQL Code

```
import mysql.connector

# Connect to the MySQL server
conn = mysql.connector.connect(host='localhost', user='root',
password='shuvo634', database='shuvodb')
mycursor = conn.cursor()

# Define the SQL query to retrieve faculty members who teach multiple courses
select_query = """
SELECT Faculty.Name, COUNT(Teaching.CourseID) as CourseCount
FROM Faculty
JOIN Teaching ON Faculty.FacultyID = Teaching.FacultyID
GROUP BY Faculty.Name
HAVING CourseCount > 1
"""

# Execute the SQL query
mycursor.execute(select_query)

# Fetch all the results
results = mycursor.fetchall()

# Print the faculty members who teach multiple courses
for row in results:
    faculty_name, course_count = row
    print("Faculty: {}, Number of Courses Taught: {}".format(faculty_name,
course_count))

# Close the connection
conn.close()
```

Output

```
PS C:\Users\Digital Outlet\Desktop\python_conncet_to_mysql> & "C:/Users/Digital Outlet/AppData/Local/Programs/Python/Python311/python.exe" "c:/Users/Dig
ital Outlet/Desktop/python_conncet_to_mysql/Assignment 5-8.py"
PS C:\Users\Digital Outlet\Desktop\python_conncet_to_mysql>
```

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

MySQL Code

```
import mysql.connector

# Connect to the MySQL server
conn = mysql.connector.connect(host='localhost', user='root',
password='shuvo634', database='shuvodb')
mycursor = conn.cursor()

# Define the SQL query to find students who have not registered for any exams
select_query = """
SELECT Student.Name
FROM Student
WHERE Student.StudentID NOT IN (SELECT DISTINCT StudentID FROM ExamRegistration)
"""

# Execute the SQL query
mycursor.execute(select_query)

# Fetch all the results
results = mycursor.fetchall()

# Print the students who have not registered for any exams
if results:
    for row in results:
        student_name = row[0]
        print("Student: {}".format(student_name))
else:
    print("No students found who have not registered for any exams.")

# Close the connection
conn.close()
```

Output

```
PS C:\Users\Digital Outlet\Desktop\python_conncet_to_mysql> & "C:/Users/Digital Outlet/AppData/Local/Programs/Python/Python311/python.exe" "c:/Users/Digital Outlet/Desktop/python_conncet_to_mysql/Assignment 5-9.py"
No students found who have not registered for any exams.
PS C:\Users\Digital Outlet\Desktop\python_conncet_to_mysql>
```

10) Retrieve the total number of enrollments for each course

MySQL Code

```
import mysql.connector

# Connect to the MySQL server
conn = mysql.connector.connect(host='localhost', user='root',
password='shuvo634', database='shuvodb')
mycursor = conn.cursor()

# Define the SQL query to retrieve the total number of enrollments for each
course
select_query = """
SELECT Course.CourseName, COUNT(Enrollment.CourseID) AS TotalEnrollments
FROM Course
LEFT JOIN Enrollment ON Course.CourseID = Enrollment.CourseID
GROUP BY Course.CourseName
"""

# Execute the SQL query
mycursor.execute(select_query)

# Fetch all the results
results = mycursor.fetchall()

# Print the total number of enrollments for each course
for row in results:
    course_name, total_enrollments = row
    print("Course: {}, Total Enrollments: {}".format(course_name,
total_enrollments))

# Close the connection
conn.close()
```

Output

```
PS C:\Users\Digital Outlet\Desktop\python_conncet_to_mysql> & "C:/Users/Digital Outlet/AppData/Local/Programs/Python/Python311/python.exe" "c:/Users/Dig
ital Outlet/Desktop/python_conncet_to_mysql/Assignment 5-10.py"
Course: Mathematics, Total Enrollments: 2
Course: History, Total Enrollments: 1
Course: Computer science, Total Enrollments: 1
Course: Literature, Total Enrollments: 1
Course: Chemistry, Total Enrollments: 1
Course: Physics, Total Enrollments: 1
Course: Economics, Total Enrollments: 1
Course: Biology, Total Enrollments: 1
PS C:\Users\Digital Outlet\Desktop\python_conncet_to_mysql>
```

---

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

MySQL Code

```python
import mysql.connector

# Connect to the MySQL server
conn = mysql.connector.connect(host='localhost', user='root',
password='shuvo634', database='shuvodb')
mycursor = conn.cursor()

# Define the SQL query to find the students enrolled in the "History" course
select_query = """
SELECT Student.Name
FROM Student
JOIN Enrollment ON Student.StudentID = Enrollment.StudentID
JOIN Course ON Enrollment.CourseID = Course.CourseID
WHERE Course.CourseName = 'History'
"""

# Execute the SQL query
mycursor.execute(select_query)

# Fetch all the results
results = mycursor.fetchall()

# Print the students enrolled in the "History" course
if results:
    for row in results:
        student_name = row[0]
        print("Student enrolled in History: {}".format(student_name))
else:
    print("No students found who are enrolled in History.")
```

```
# Close the connection
conn.close()
```

Ouput

```
PS C:\Users\Digital Outlet\Desktop\python_conncet_to_mysql> & "C:/Users/Digital Outlet/AppData/Local/Programs/Python/Python311/python.exe" "c:/Users/Dig
ital Outlet/Desktop/python_conncet_to_mysql/Assignment 5-11.py"
Student enrolled in History: John Doe
PS C:\Users\Digital Outlet\Desktop\python_conncet_to_mysql>
```

---

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

MySQL Code :

```
import mysql.connector

# Connect to the MySQL server
conn = mysql.connector.connect(host='localhost', user='root',
password='shuvo634', database='shuvodb')
mycursor = conn.cursor()

# Define the SQL query to retrieve exams and their locations scheduled for
November 2023
select_query = """
SELECT ExamDate, ExamTime, Location
FROM Exam
WHERE ExamDate >= '2023-11-01' AND ExamDate <= '2023-11-30'
"""

# Execute the SQL query
mycursor.execute(select_query)

# Fetch all the results
results = mycursor.fetchall()

# Print the exams and their locations for November 2023
for row in results:
    exam_date, exam_time, location = row
```

```
    print("Exam Date: {}, Exam Time: {}, Location: {}".format(exam_date,
exam_time, location))

# Close the connection
conn.close()
```

Output

```
PS C:\Users\Digital Outlet\Desktop\python_conncet_to_mysql> & "C:/Users/Digital Outlet/AppData/Local/Programs/Python/Python311/python.exe" "c:/Users/Dig
ital Outlet/Desktop/python_conncet_to_mysql/Assignment 5-12.py"
Exam Date: 2023-11-10, Exam Time: 9:00:00, Location: Exam Hall A
Exam Date: 2023-11-12, Exam Time: 14:00:00, Location: Exam Hall B
Exam Date: 2023-11-15, Exam Time: 10:30:00, Location: Exam Hall C
Exam Date: 2023-11-18, Exam Time: 15:15:00, Location: Exam Hall D
Exam Date: 2023-11-20, Exam Time: 13:00:00, Location: Exam Hall E
PS C:\Users\Digital Outlet\Desktop\python_conncet_to_mysql>
```

13. List the courses with the highest number of enrollment

MySQL Code

```
import mysql.connector

# Connect to the MySQL server
conn = mysql.connector.connect(host='localhost', user='root',
password='shuvo634', database='shuvodb')
mycursor = conn.cursor()

# Define the SQL query to list the courses with the highest number of enrollments
select_query = """
SELECT Course.CourseName, COUNT(Enrollment.CourseID) AS TotalEnrollments
FROM Course
LEFT JOIN Enrollment ON Course.CourseID = Enrollment.CourseID
GROUP BY Course.CourseName
ORDER BY TotalEnrollments DESC
LIMIT 1
"""

# Execute the SQL query
mycursor.execute(select_query)

# Fetch the result
result = mycursor.fetchone()
```

```
# Print the course with the highest number of enrollments
if result:
    course_name, total_enrollments = result
    print("Course with the highest enrollments: {}, Total Enrollments:
{}".format(course_name, total_enrollments))
else:
    print("No courses found with enrollments.")

# Close the connection
conn.close()
```

Output

```
PS C:\Users\Digital Outlet\Desktop\python_conncet_to_mysql> & "C:/Users/Digital Outlet/AppData/Local/Programs/Python/Python311/python.exe" "c:/Users/Dig
ital Outlet/Desktop/python_conncet_to_mysql/Assignment 5-13.py"
Course with the highest enrollments: Mathematics, Total Enrollments: 2
PS C:\Users\Digital Outlet\Desktop\python_conncet_to_mysql>
```

14) Find the average score for each student

MySQL Code

```
import mysql.connector

# Connect to the MySQL server
conn = mysql.connector.connect(host='localhost', user='root',
password='shuvo634', database='shuvodb')
mycursor = conn.cursor()

# Define the SQL query to find the average score for each student
select_query = """
SELECT Student.Name, AVG(ExamResults.score) AS AverageScore
FROM Student
LEFT JOIN ExamResults ON Student.StudentID = ExamResults.StudentID
GROUP BY Student.Name
"""

# Execute the SQL query
mycursor.execute(select_query)
```

```python
# Fetch all the results
results = mycursor.fetchall()

# Print the average score for each student
for row in results:
    student_name, average_score = row
    print("Student: {}, Average Score: {}".format(student_name, average_score))

# Close the connection
conn.close()
```

Output

```
PS C:\Users\Digital Outlet\Desktop\python_conncet_to_mysql> & "C:/Users/Digital Outlet/AppData/Local/Programs/Python/Python311/python.exe" "c:/Users/Dig
ital Outlet/Desktop/python_conncet_to_mysql/Assignment 5-14.py"
Student: John Doe, Average Score: 92.500000
Student: Jane Smith, Average Score: 88.000000
Student: Robert Johnson, Average Score: 95.500000
Student: Emily White, Average Score: 89.000000
Student: Michael Lee, Average Score: 94.500000
Student: Sarah Brown, Average Score: 91.000000
Student: David Clark, Average Score: 87.500000
Student: Melissa Turner, Average Score: None
PS C:\Users\Digital Outlet\Desktop\python_conncet_to_mysql>
```

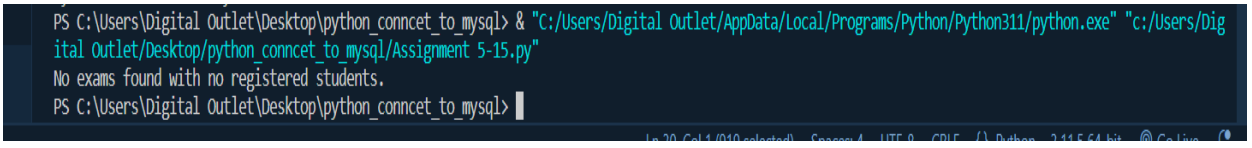
15) Retrieve the exams that have no registered students

MySQL Code

```python
import mysql.connector

# Connect to the MySQL server
conn = mysql.connector.connect(host='localhost', user='root',
password='shuvo634', database='shuvodb')
mycursor = conn.cursor()

# Define the SQL query to retrieve exams with no registered students
select_query = """
SELECT Exam.ExamID, Exam.ExamDate, Exam.ExamTime, Exam.Location
FROM Exam
WHERE Exam.ExamID NOT IN (SELECT DISTINCT ExamID FROM ExamRegistration)
```

```
"""

# Execute the SQL query
mycursor.execute(select_query)

# Fetch all the results
results = mycursor.fetchall()

# Print the exams with no registered students
if results:
    for row in results:
        exam_id, exam_date, exam_time, location = row
        print("Exam ID: {}, Exam Date: {}, Exam Time: {}, Location:
{}".format(exam_id, exam_date, exam_time, location))
else:
    print("No exams found with no registered students.")

# Close the connection
conn.close()
```

Output



---

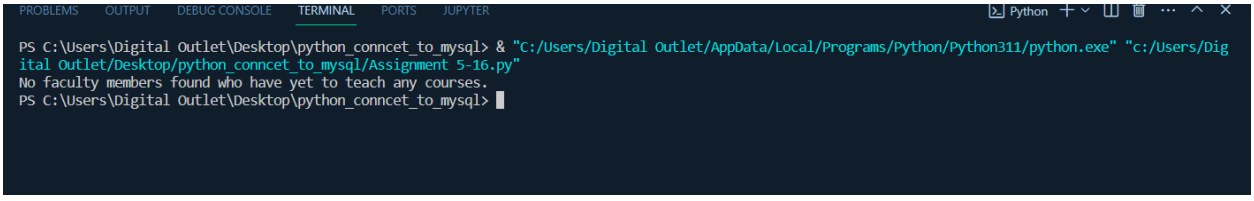
16) List the faculty members who have yet to teach any courses

MySQL Code

```
import mysql.connector

# Connect to the MySQL server
conn = mysql.connector.connect(host='localhost', user='root',
password='shuvo634', database='shuvodb')
mycursor = conn.cursor()

# Define the SQL query to list faculty members who have yet to teach any courses
```

```
select_query = """
SELECT Faculty.Name
FROM Faculty
WHERE Faculty.FacultyID NOT IN (SELECT DISTINCT FacultyID FROM Teaching)
"""

# Execute the SQL query
mycursor.execute(select_query)

# Fetch all the results
results = mycursor.fetchall()

# Print the faculty members who have yet to teach any courses
if results:
    for row in results:
        faculty_name = row[0]
        print("Faculty member who has yet to teach any courses:
{}".format(faculty_name))
else:
    print("No faculty members found who have yet to teach any courses.")

# Close the connection
conn.close()
```

Output



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

MySQL Code

```
import mysql.connector

# Connect to the MySQL server
conn = mysql.connector.connect(host='localhost', user='root',
password='shuvo634', database='shuvodb')
mycursor = conn.cursor()

# Define the SQL query to find students who have registered for exams in both
"Mathematics" and "Computer Science" departments
select_query = """
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;
"""

# Execute the SQL query
mycursor.execute(select_query)

# Fetch all the results
results = mycursor.fetchall()

# Print the students who have registered for exams in both "Mathematics" and
"Computer Science" departments
if results:
    for row in results:
        student_name = row[0]
        print("Student registered for exams in both Mathematics and Computer
Science: {}".format(student_name))
else:
    print("No students found who have registered for exams in both departments.")

# Close the connection
conn.close()
```

Output

```
PS C:\Users\Digital Outlet\Desktop\python_conncet_to_mysql> & "C:/Users/Digital Outlet/AppData/Local/Programs/Python/Python311/python.e
xe" "c:/Users/Digital Outlet/Desktop/python_conncet_to_mysql/Assignment 5-17.py"
No students found who have registered for exams in both departments.
PS C:\Users\Digital Outlet\Desktop\python_conncet_to_mysql>
```

---

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

MySQL Code

```
import mysql.connector

# Connect to the MySQL server
conn = mysql.connector.connect(host='localhost', user='root',
password='shuvo634', database='shuvodb')
mycursor = conn.cursor()

# Define the SQL query to retrieve students who scored the highest in each exam
select_query = """
SELECT E.ExamID, E.ExamDate, S.Name, ER.StudentID, MAX(EResults.score) AS
Max_Score
FROM Exam E
INNER JOIN ExamRegistration ER ON E.ExamID = ER.ExamID
INNER JOIN Student S ON ER.StudentID = S.StudentID
INNER JOIN ExamResults EResults ON ER.ExamID = EResults.ExamID AND ER.StudentID =
EResults.StudentID
GROUP BY E.ExamID, E.ExamDate, S.Name, ER.StudentID
"""

# Execute the SQL query
mycursor.execute(select_query)

# Fetch all the results
results = mycursor.fetchall()

# Print the students who scored the highest in each exam
for row in results:
```

```
    exam_id, exam_date, student_name, student_id, max_score = row
    print("Exam ID: {}, Exam Date: {}, Student Name: {}, Student ID: {}, Max
Score: {}".format(exam_id, exam_date, student_name, student_id, max_score))

# Close the connection
conn.close()
```

Output

```
PS C:\Users\Digital Outlet\Desktop\python_conncet_to_mysql> & "C:/Users/Digital Outlet/AppData/Local/Programs/Python/Python311/python.exe" "c:/Users/Dig
ital Outlet/Desktop/python_conncet_to_mysql/Assignment 5-18.py"
Exam ID: 201, Exam Date: 2023-11-10, Student Name: John Doe, Student ID: 1, Max Score: 92.50
Exam ID: 201, Exam Date: 2023-11-10, Student Name: Jane Smith, Student ID: 2, Max Score: 88.00
Exam ID: 201, Exam Date: 2023-11-10, Student Name: David Clark, Student ID: 7, Max Score: 87.50
Exam ID: 202, Exam Date: 2023-11-12, Student Name: Robert Johnson, Student ID: 3, Max Score: 95.50
Exam ID: 203, Exam Date: 2023-11-15, Student Name: Emily White, Student ID: 4, Max Score: 89.00
Exam ID: 204, Exam Date: 2023-11-18, Student Name: Michael Lee, Student ID: 5, Max Score: 94.50
Exam ID: 205, Exam Date: 2023-11-20, Student Name: Sarah Brown, Student ID: 6, Max Score: 91.00
PS C:\Users\Digital Outlet\Desktop\python_conncet_to_mysql>
```

---

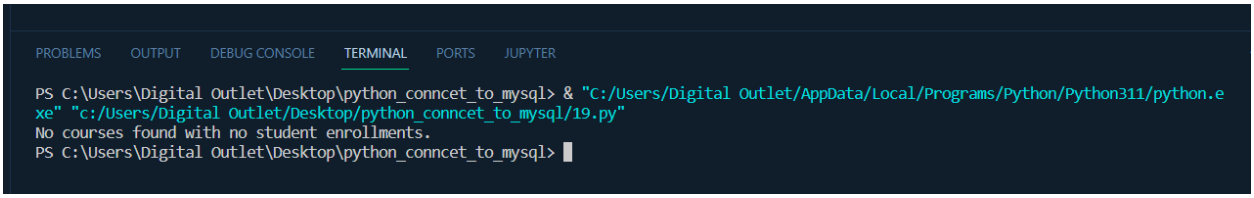
19) Find the courses that no student has enrolled in

MySQL Code

```
import mysql.connector

# Connect to the MySQL server
conn = mysql.connector.connect(host='localhost', user='root',
password='shuvo634', database='shuvodb')
mycursor = conn.cursor()

# Define the SQL query to find courses with no student enrollments
select_query = '''
SELECT CourseName
FROM Course
LEFT JOIN Enrollment ON Course.CourseID = Enrollment.CourseID
WHERE Enrollment.StudentID IS NULL;
'''

# Execute the SQL query
mycursor.execute(select_query)

# Fetch and print the results
results = mycursor.fetchall()
```

```
if len(results) == 0:
    print("No courses found with no student enrollments.")
else:
    print("Courses with no student enrollments:")
    for result in results:
        print(result[0])

# Close the connection
conn.close()
```

Output



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

MySQL Code

```
import mysql.connector

# Connect to the MySQL server
conn = mysql.connector.connect(host='localhost', user='root',
password='shuvo634', database='shuvodb')
mycursor = conn.cursor()

# Define the SQL query to retrieve faculty members with an average enrollment
count above 10
select_query = '''
SELECT F.Name AS FacultyName, F.Department, AVG(EnrollmentCount) AS
AverageEnrollment
FROM Faculty F
```

```
JOIN Teaching T ON F.FacultyID = T.FacultyID
JOIN Course C ON T.CourseID = C.CourseID
LEFT JOIN (
    SELECT CourseID, COUNT(StudentID) AS EnrollmentCount
    FROM Enrollment
    GROUP BY CourseID
) AS E ON C.CourseID = E.CourseID
GROUP BY F.FacultyID
HAVING AVG(EnrollmentCount) > 10;
'''

# Execute the SQL query
mycursor.execute(select_query)

# Fetch and print the results
results = mycursor.fetchall()
for result in results:
    faculty_name, department, average_enrollment = result
    print(f"Faculty Name: {faculty_name}, Department: {department}, Average
Enrollment: {average_enrollment:.2f}")

# Close the connection
conn.close()
```

Output

```
mysql.connector.errors.DatabaseError: 1111 (HY000): Invalid use of group function
PS C:\Users\Digital Outlet\Desktop\python_conncet_to_mysql> & "C:/Users/Digital Outlet/AppData/Local/Programs/Python/Python311/python.e
xe" "c:/Users/Digital Outlet/Desktop/python_conncet_to_mysql/Assignment 5-20.py"
PS C:\Users\Digital Outlet\Desktop\python_conncet_to_mysql>
```