In [1]:

```
!pip install mysql-connector-python
```

```
Collecting mysql-connector-python
  Downloading mysql_connector_python-8.1.0-cp310-cp310-win_amd64.whl (10.9
MB)
     ---------------------------------------- 10.9/10.9 MB 2.3 MB/s eta 0:
00:00
Collecting protobuf<=4.21.12,>=4.21.1
  Downloading protobuf-4.21.12-cp310-abi3-win_amd64.whl (527 kB)
     -------------------------------------- 527.0/527.0 kB 1.8 MB/s eta 0:
00:00
Installing collected packages: protobuf, mysql-connector-python
Successfully installed mysql-connector-python-8.1.0 protobuf-4.21.12
```

In [18]:

```
import mysql.connector
```

In [19]:

```
conn=mysql.connector.connect(host="localhost",user="root",passwd="Dsk151312@")
print(conn)
```

```
<mysql.connector.connection_cext.CMySQLConnection object at 0x0000021ECE8B
AF20>
```

In [20]:

```
Mycurr=conn.cursor()
Mycurr.execute("show databases")
```

In [8]:

```
for x in Mycurr:
    print(x)
conn.close()
```

```
('information_schema',)
('mysql',)
('performance_schema',)
('shopdb',)
('student_management',)
('sys',)
```

In [35]:

```
import mysql.connector
conn=mysql.connector.connect(
host="localhost",
user="root",
password="Dsk151312@",
database="student_management")

cur=conn.cursor()

val=[(1,"Rohit","A Ch",20),(2,"Kumaresh","N M",20),(3,"Adhithyan","B",20),(4,"Dhrish","S
qry='insert into students values(%s,%s,%s,%s)'

cur.executemany(qry,val)
conn.commit()

cur.execute("Select * from students")

for x in cur:
    print(x)
conn.close()
```

```
(1, 'Rohit', 'A Ch', 20)
(2, 'Kumaresh', 'N M', 20)
(3, 'Adhithyan', 'B', 20)
(4, 'Dhrish', 'S Kumar', 20)
(5, 'Kirthana', 'Pati', 20)
```

In [38]:

```
import mysql.connector
conn=mysql.connector.connect(
host="localhost",
user="root",
password="Dsk151312@",
database="student_management")

cur=conn.cursor()

val=[(1,"Data science"),(2,"Computer Vision"),(3,"reinforcement Learning"),(4,"Tensorflo
qry='insert into courses values(%s,%s)'
cur.executemany(qry,val)
conn.commit()

cur.execute("Select * from courses")

for x in cur:
    print(x)
conn.close()
```

```
(1, 'Data science')
(2, 'Computer Vision')
(3, 'reinforcement Learning')
(4, 'Tensorflow')
```

In [ ]:

```
import mysql.connector
conn=mysql.connector.connect(
host="localhost",
user="root",
password="Dsk151312@",
database="student_management")

cur=conn.cursor()

val=[(1, 1, 1),
    (2, 2, 2),
    (3, 3, 1),
    (4, 4, 3),
    (5, 5, 4)]
qry='insert into enrollments values(%s,%s,%s)'

cur.executemany(qry,val)
conn.commit()


for x in cur:
    print(x)
conn.close()
```

In [45]:

```
import mysql.connector
conn=mysql.connector.connect(
host="localhost",
user="root",
password="Dsk151312@",
database="student_management")

cur=conn.cursor()

cur.execute("Select first_name,last_name,age from students")

for x in cur:
    print(x)
conn.close()
```

```
('Rohit', 'A Ch', 20)
('Kumaresh', 'N M', 20)
('Adhithyan', 'B', 20)
('Dhrish', 'S Kumar', 20)
('Kirthana', 'Pati', 20)
```

In [48]:

```
import mysql.connector
conn=mysql.connector.connect(
host="localhost",
user="root",
password="Dsk151312@",
database="student_management")

cur=conn.cursor()

cur.execute("Select first_name,last_name,age from students")

for x in cur:
    print(x)
conn.close()
```

```
('Rohit', 'A Ch', 20)
('Kumaresh', 'N M', 20)
('Adhithyan', 'B', 20)
('Dhrish', 'S Kumar', 20)
('Kirthana', 'Pati', 20)
```

In [51]:

```
# Find the names of students who are older than 25 years. ( here we will do age = 19)
import mysql.connector
conn=mysql.connector.connect(
host="localhost",
user="root",
password="Dsk151312@",
database="student_management")

cur=conn.cursor()

cur.execute("Select first_name,last_name from students where age>19")

for x in cur:
    print(x)
conn.close()
```

```
('Rohit', 'A Ch')
('Kumaresh', 'N M')
('Adhithyan', 'B')
('Dhrish', 'S Kumar')
('Kirthana', 'Pati')
```

In [54]:

```
# List the courses that have at least 5 enrollments. ( here i am changing it to <=5)
import mysql.connector
conn=mysql.connector.connect(
host="localhost",
user="root",
password="Dsk151312@",
database="student_management")

cur=conn.cursor()

cur.execute("SELECT course_name FROM courses WHERE course_id IN (SELECT course_id FROM e
for x in cur:
    print(x)
conn.close()
```

```
('Data science',)
('Computer Vision',)
('reinforcement Learning',)
('Tensorflow',)
```

In [63]:

```
# Retrieve the courses in which a specific student (by student ID) is enrolled.

import mysql.connector

conn = mysql.connector.connect(
    host="localhost",
    user="root",
    password="Dsk151312@",
    database="student_management"
)

cur = conn.cursor()
student_id = 1  # Replace with the desired student ID

query = """
    SELECT courses.course_name
    FROM courses
    INNER JOIN enrollments ON courses.course_id = enrollments.course_id
    WHERE enrollments.student_id = %s
"""

cur.execute(query, (student_id,))

for x in cur:
    print(x)

conn.close()
```

```
('Data science',)
```

In [67]:

```
# Find the average age of all students.
import mysql.connector

conn = mysql.connector.connect(
    host="localhost",
    user="root",
    password="Dsk151312@",
    database="student_management"
)

cur = conn.cursor()

cur.execute("Select avg(age) from students")

average_age = cur.fetchone()[0]

print("Average Age of Students:", average_age)
conn.close()
```

```
Average Age of Students: 20.0000
```

In [74]:

```
# List the students who are enrolled in a specific course (by course ID).

import mysql.connector

# Establish a connection to the MySQL database
conn = mysql.connector.connect(
    host="localhost",
    user="root",
    password="Dsk151312@",
    database="student_management"
)

# Create a cursor to interact with the database
cur = conn.cursor()

# Define the course ID you want to search for (replace 1 with your desired course ID)
course_id = 1

# Execute the SQL query to retrieve students enrolled in the specific course
query = f"SELECT students.first_name, students.last_name FROM students \
         INNER JOIN enrollments ON students.student_id = enrollments.student_id \
         WHERE enrollments.course_id = {course_id}"

cur.execute(query)

# Fetch all the rows of the result set
enrolled_students = cur.fetchall()

# Print the list of students enrolled in the specific course
for student in enrolled_students:
    print(f"{student[0]} {student[1]}")

# Close the cursor and connection
cur.close()
conn.close()
```

```
Rohit A Ch
Adhithyan B
```

In [76]:

```
# Retrieve the students who have the same first names (e.g., find all students with the 

import mysql.connector

# Establish a connection to the MySQL database
conn = mysql.connector.connect(
    host="localhost",
    user="root",
    password="Dsk151312@",
    database="student_management"
)

# Create a cursor to interact with the database
cur = conn.cursor()

# Define the first name you want to search for (replace 'John' with your desired first n
first_name_to_search = 'John'

# Execute the SQL query to retrieve students with the specified first name
query = f"SELECT first_name, last_name FROM students WHERE first_name = '{first_name_to_

cur.execute(query)

# Fetch all the rows of the result set
matching_students = cur.fetchall()

# Print the list of students with the specified first name
for student in matching_students:
    print(f"{student[0]} {student[1]}")

# Close the cursor and connection
cur.close()
conn.close()
```

In [77]:

```
# Count the number of courses each student is enrolled in and list the students along wi
import mysql.connector

# Establish a connection to the MySQL database
conn = mysql.connector.connect(
    host="localhost",
    user="root",
    password="Dsk151312@",
    database="student_management"
)

# Create a cursor to interact with the database
cur = conn.cursor()

# Execute the SQL query to count the number of courses for each student
query = """
    SELECT students.student_id, students.first_name, students.last_name, COUNT(enrollmen
    FROM students
    LEFT JOIN enrollments ON students.student_id = enrollments.student_id
    GROUP BY students.student_id, students.first_name, students.last_name
"""

cur.execute(query)

# Fetch all the rows of the result set
student_course_counts = cur.fetchall()

# Print the list of students along with their course counts
for student in student_course_counts:
    student_id, first_name, last_name, course_count = student
    print(f"Student ID: {student_id}, Name: {first_name} {last_name}, Course Count: {cou

# Close the cursor and connection
cur.close()
conn.close()
```

```
Student ID: 1, Name: Rohit A Ch, Course Count: 1
Student ID: 2, Name: Kumaresh N M, Course Count: 1
Student ID: 3, Name: Adhithyan B, Course Count: 1
Student ID: 4, Name: Dhrish S Kumar, Course Count: 1
Student ID: 5, Name: Kirthana Pati, Course Count: 1
```

In [79]:

```
# Find the courses with no enrollments.
import mysql.connector

# Establish a connection to the MySQL database
conn = mysql.connector.connect(
    host="localhost",
    user="root",
    password="Dsk151312@",
    database="student_management"
)

# Create a cursor to interact with the database
cur = conn.cursor()

# Execute the SQL query to find courses with no enrollments
query = """
    SELECT courses.course_id, courses.course_name
    FROM courses
    LEFT JOIN enrollments ON courses.course_id = enrollments.course_id
    WHERE enrollments.course_id IS NULL
"""

cur.execute(query)

# Fetch all the rows of the result set
courses_with_no_enrollments = cur.fetchall()

# Print the list of courses with no enrollments
for course in courses_with_no_enrollments:
    course_id, course_name = course
    print(f"Course ID: {course_id}, Course Name: {course_name} has no enrollments.")

# Close the cursor and connection
cur.close()
conn.close()
```

In [80]:

```
# Retrieve the courses with enrollments exceeding the average enrollment count.

import mysql.connector

# Establish a connection to the MySQL database
conn = mysql.connector.connect(
    host="localhost",
    user="root",
    password="Dsk151312@",
    database="student_management"
)

# Create a cursor to interact with the database
cur = conn.cursor()

# Execute a single SQL query to find courses with enrollments exceeding the average enro
query_courses_exceeding_avg = """
    SELECT courses.course_id, courses.course_name, COUNT(enrollments.enrollment_id) as e
    FROM courses
    INNER JOIN enrollments ON courses.course_id = enrollments.course_id
    GROUP BY courses.course_id, courses.course_name
    HAVING enrollment_count > (
        SELECT AVG(enrollment_id)
        FROM enrollments
    )
"""

cur.execute(query_courses_exceeding_avg)

# Fetch all the rows of the result set
courses_exceeding_avg = cur.fetchall()

# Print the list of courses with enrollments exceeding the average
for course in courses_exceeding_avg:
    course_id, course_name, enrollment_count = course
    print(f"Course ID: {course_id}, Course Name: {course_name}, Enrollment Count: {enrol

# Close the cursor and connection
cur.close()
conn.close()
```

In [ ]:

```

```