In [1]:

```python
import mysql.connector

mydb = mysql.connector.connect(
  host="localhost",
  user="root",
  password="Yankejing020711"   #REPLACE THIS WITH THE PASSWORD YOU SET
)

print(mydb)

if mydb.is_connected():
    print("CONNECTION SUCCESSFUL")
```

```
<mysql.connector.connection_cext.CMySQLConnection object at 0x1061fa4e0>
CONNECTION SUCCESSFUL
```

In [2]:

```python
mydb = mysql.connector.connect(
  host="localhost",
  user="root",
  password="Yankejing020711", #REPLACE THIS WITH YOUR PASSWORD
  database = "data1050f24" #connecting to testDatabase
)
```

1. Find the id, title and section_id of all the courses being taught in Fall 2023.

In [3]:

```python
mycursor = mydb.cursor()

mycursor.execute(
    """
    SELECT S.course_id, C.title, S.sec_id
    FROM course AS C
    JOIN section AS S
    ON C.course_id = S.course_id
    WHERE S.semester = 'Fall'
    AND S.year = 2023;
    """
)
```

In [4]:

```python
# fetch all the rows from the result of the query
result = mycursor.fetchall()
# print
for row in result:
    print(row)
```

```
('Data2050', 'Practicum', '1')
('CSCI1270', 'Database Management Systems', '1')
('CSCI1270', 'Database Management Systems', '2')
('Data1030', 'Hands-on data science', '1')
('Psyc2134', 'Intro to Cognitive Science', '1')
('Data1050', 'Data Engineering', '1')
```

2. Find the id, title, section_id, and instructor name of all courses being taught in Fall 2023

In [5]:

```
query="DROP VIEW IF EXISTS instructorname;"
mycursor=mydb.cursor()
mycursor.execute(query)
```

In [6]:

```
mycursor = mydb.cursor()

mycursor.execute(
    """
    CREATE VIEW instructorname AS
    SELECT I.name, T.course_id, T.sec_id, T.semester, T.year
    FROM teaches AS T
    JOIN instructor AS I
    ON T.ID = I.ID;
    """
)
```

In [7]:

```
query="DROP VIEW IF EXISTS coursetitle;"
mycursor=mydb.cursor()
mycursor.execute(query)
```

In [8]:

```
mycursor = mydb.cursor()

mycursor.execute(
    """
    CREATE VIEW coursetitle AS
    SELECT S.course_id, S.sec_id, S.semester, S.year, C.title
    FROM section AS S
    JOIN course AS C
    ON S.course_id = C.course_id;
    """
)
```

In [9]:

```
mycursor = mydb.cursor()

mycursor.execute(
    """
    SELECT C.course_id, C.title, C.sec_id, I.name
    FROM coursetitle AS C
    JOIN instructorname AS I
    ON C.course_id = I.course_id
    WHERE C.semester = 'Fall'
    AND C.year = 2023;
    """
)
```

In [10]:

```
# fetch all the rows from the result of the query
result = mycursor.fetchall()
# print
```

```
for row in result:
    print(row)
```

```
('Data2050', 'Practicum', '1', 'Joe Smith')
('Data2050', 'Practicum', '1', 'Carmen Herman')
('Data2050', 'Practicum', '1', 'Joe Smith')
('CSCI1270', 'Database Management Systems', '1', 'Ying Zheng')
('CSCI1270', 'Database Management Systems', '1', 'Calista Holistic')
('CSCI1270', 'Database Management Systems', '2', 'Ying Zheng')
('CSCI1270', 'Database Management Systems', '2', 'Calista Holistic')
('Data1030', 'Hands-on data science', '1', 'Carmen Herman')
('Psyc2134', 'Intro to Cognitive Science', '1', 'Taylor Slow')
('Psyc2134', 'Intro to Cognitive Science', '1', 'Rashid Irani')
('Data1050', 'Data Engineering', '1', 'Carmen Herman')
('Data1050', 'Data Engineering', '1', 'Joe Smith')
```

3. Write a SQL instruction to change (i.e., update) the time_slot_id of all courses which are 6 to 2. You may not delete these and insert new ones; use the update command.

In [11]:

```
# Clear any pending results
mycursor.fetchall()

# Perform the update
mycursor.execute(
    """
    UPDATE time_slot
    SET time_slot_id = 2
    WHERE time_slot_id = 6;
    """
)

# Commit the transaction
mydb.commit()
```

In [12]:

```
query = "SELECT * from time_slot"
mycursor.execute(query)
for x in mycursor:
    print(x)
```

```
(1, 'Monday', datetime.timedelta(seconds=32400), datetime.timedelta(seconds=
39600))
(2, 'Monday', datetime.timedelta(seconds=39600), datetime.timedelta(seconds=
3600))
(2, 'Tuesday', datetime.timedelta(seconds=39600), datetime.timedelta(seconds
=3600))
(3, 'Monday', datetime.timedelta(seconds=3600), datetime.timedelta(seconds=1
0800))
(4, 'Monday', datetime.timedelta(seconds=10800), datetime.timedelta(seconds=
18000))
(5, 'Tuesday', datetime.timedelta(seconds=32400), datetime.timedelta(seconds
=39600))
(7, 'Tuesday', datetime.timedelta(seconds=3600), datetime.timedelta(seconds=
10800))
(8, 'Wednesday', datetime.timedelta(seconds=32400), datetime.timedelta(secon
ds=39600))
(9, 'Wednesday', datetime.timedelta(seconds=39600), datetime.timedelta(secon
ds=3600))
(10, 'Wednesday', datetime.timedelta(seconds=3600), datetime.timedelta(secon
ds=10800))
(11, 'Wednesday', datetime.timedelta(seconds=10800), datetime.timedelta(seco
nds=18000))
(12, 'Thursday', datetime.timedelta(seconds=32400), datetime.timedelta(secon
ds=39600))
(13, 'Thursday', datetime.timedelta(seconds=39600), datetime.timedelta(secon
ds=3600))
(14, 'Thursday', datetime.timedelta(seconds=3600), datetime.timedelta(second
s=10800))
(15, 'Friday', datetime.timedelta(seconds=32400), datetime.timedelta(seconds
=39600))
(16, 'Friday', datetime.timedelta(seconds=39600), datetime.timedelta(seconds
=3600))
(17, 'Friday', datetime.timedelta(seconds=3600), datetime.timedelta(seconds=
10800))
(18, 'Friday', datetime.timedelta(seconds=10800), datetime.timedelta(seconds
=18000))
```

4. Find the course id, title, section and start time of all courses taught in Fall 2022

In [13]:

```
query="DROP VIEW IF EXISTS starttime;"
mycursor=mydb.cursor()
mycursor.execute(query)
```

In [14]:

```
mycursor = mydb.cursor()

mycursor.execute(
        """
        CREATE VIEW starttime AS
        SELECT S.course_id, T.start_time
        FROM section AS S
        JOIN time_slot AS T
        ON S.time_slot_id = T.time_slot_id
        """
)
```

In [15]:

```
mycursor = mydb.cursor()

mycursor.execute(
        """
        SELECT T.course_id, C.title, C.sec_id, T.start_time
        FROM starttime AS T
        JOIN coursetitle AS C
        ON T.course_id = C.course_id
        WHERE (C.semester = 'Fall') AND (C.year = 2022);
        """
)
```

In [16]:

```
# fetch all the rows from the result of the query
result = mycursor.fetchall()
# print
for row in result:
    print(row)
```

```
('Data1030', 'Hands-on data science', '1', datetime.timedelta(seconds=3960
0))
('Data1030', 'Hands-on data science', '1', datetime.timedelta(seconds=3960
0))
('Data1030', 'Hands-on data science', '1', datetime.timedelta(seconds=3960
0))
('Data1030', 'Hands-on data science', '1', datetime.timedelta(seconds=3960
0))
('Data1050', 'Data Engineering', '1', datetime.timedelta(seconds=3600))
('Data1050', 'Data Engineering', '1', datetime.timedelta(seconds=32400))
```

5. Write a SQL instruction so that all students with ID 1238 and 1333 who have a null grade in any course, have their grade changed to 'A'.

In [21]:

```
mycursor = mydb.cursor()
query = "TRUNCATE TABLE takes;"
mycursor.execute(query)
```

In [23]:

```
import pandas as pd
import numpy as np
data = pd.read_csv("~/Desktop/DATA1050/data/takes.csv")
data = data.replace({np.nan: None})

for i,row in data.iterrows():
            sql = "INSERT INTO data1050F24.takes VALUES (%s,%s,%s,%s,%s,%s)"
            mycursor.execute(sql, tuple(row))
            #print("Record inserted")
            # the connection is not autocommitted by default, so we
            # must commit to save our changes
            mydb.commit()
```

In [24]:

```
mycursor = mydb.cursor()

mycursor.execute(
    """
    CREATE TEMPORARY TABLE originalcredits AS
    SELECT ID, course_id, sec_id, semester, year, grade
    FROM takes
    """
)
```

In [25]:

```
mycursor = mydb.cursor()

query = """
SELECT * FROM originalcredits;
"""
mycursor.execute(query)

for x in mycursor:
    print(x)
```

```
('1122', 'CSCI1270', '2', 'Fall', 2023, None)
('1122', 'Data1050', '1', 'Fall', 2023, None)
('1238', 'CSCI1270', '1', 'Fall', 2023, None)
('1238', 'Data1030', '1', 'Fall', 2023, None)
('1238', 'Data1050', '1', 'Fall', 2022, 'A')
('1333', 'APMA1650', '1', 'Spring', 2023, 'B')
('1333', 'CSCI1270', '1', 'Fall', 2023, None)
('1333', 'Data1030', '1', 'Fall', 2022, 'A')
('1333', 'Data1050', '1', 'Fall', 2022, 'A')
('1333', 'Data2050', '1', 'Fall', 2023, None)
('1845', 'CSCI1270', '2', 'Fall', 2023, None)
('1845', 'Data1050', '1', 'Fall', 2023, None)
('1888', 'CSCI1270', '1', 'Fall', 2023, None)
('1888', 'Data1050', '1', 'Fall', 2023, None)
('1934', 'Data1050', '1', 'Fall', 2023, None)
('1970', 'Data1050', '1', 'Fall', 2023, None)
('1999', 'CSCI1270', '1', 'Fall', 2023, None)
('1999', 'Data1050', '1', 'Fall', 2023, None)
('2142', 'CSCI1270', '1', 'Fall', 2023, None)
('2142', 'Data1050', '1', 'Fall', 2023, None)
('2143', 'Data1050', '1', 'Fall', 2023, None)
('2946', 'Data1050', '1', 'Fall', 2023, None)
('3303', 'Data1050', '1', 'Fall', 2023, None)
```

In [26]:

```
mycursor = mydb.cursor()

mycursor.execute(
    """
    UPDATE takes
    SET grade = 'A'
    WHERE grade IS NULL AND ID IN (1238, 1333);
    """
)
```

In [27]:

```
mycursor = mydb.cursor()

query = """
SELECT * FROM takes;
"""
mycursor.execute(query)
```

```
for x in mycursor:
    print(x)
```

```
('1122', 'CSCI1270', '2', 'Fall', 2023, None)
('1122', 'Data1050', '1', 'Fall', 2023, None)
('1238', 'CSCI1270', '1', 'Fall', 2023, 'A')
('1238', 'Data1030', '1', 'Fall', 2023, 'A')
('1238', 'Data1050', '1', 'Fall', 2022, 'A')
('1333', 'APMA1650', '1', 'Spring', 2023, 'B')
('1333', 'CSCI1270', '1', 'Fall', 2023, 'A')
('1333', 'Data1030', '1', 'Fall', 2022, 'A')
('1333', 'Data1050', '1', 'Fall', 2022, 'A')
('1333', 'Data2050', '1', 'Fall', 2023, 'A')
('1845', 'CSCI1270', '2', 'Fall', 2023, None)
('1845', 'Data1050', '1', 'Fall', 2023, None)
('1888', 'CSCI1270', '1', 'Fall', 2023, None)
('1888', 'Data1050', '1', 'Fall', 2023, None)
('1934', 'Data1050', '1', 'Fall', 2023, None)
('1970', 'Data1050', '1', 'Fall', 2023, None)
('1999', 'CSCI1270', '1', 'Fall', 2023, None)
('1999', 'Data1050', '1', 'Fall', 2023, None)
('2142', 'CSCI1270', '1', 'Fall', 2023, None)
('2142', 'Data1050', '1', 'Fall', 2023, None)
('2143', 'Data1050', '1', 'Fall', 2023, None)
('2946', 'Data1050', '1', 'Fall', 2023, None)
('3303', 'Data1050', '1', 'Fall', 2023, None)
```

6. After executing this instruction write a query to make appropriate changes to the tot_credit of those students (that is, increment it by the credits assigned to those courses where the grade was changed.)

In [28]:
```
query="DROP TABLE IF EXISTS updatecourse;"
mycursor=mydb.cursor()
mycursor.execute(query)
```

In [29]:
```
mycursor = mydb.cursor()

mycursor.execute(
    """
    CREATE TEMPORARY TABLE updatecourse AS
    SELECT T.course_id, T.ID
    FROM takes AS T
    JOIN originalcredits AS O
    ON (T.ID, T.course_id, T.sec_id, T.semester, T.year) = (O.ID, O.course_i
    WHERE (T.grade <> O.grade OR (T.grade IS NULL AND O.grade IS NOT NULL) C
    """
)
```

In [30]:
```
mycursor = mydb.cursor()

query = """
SELECT * FROM updatecourse;
"""
```

```
mycursor.execute(query)

for x in mycursor:
    print(x)
```

```
('CSCI1270', '1238')
('Data1030', '1238')
('CSCI1270', '1333')
('Data2050', '1333')
```

In [31]:

```
mycursor = mydb.cursor()

mycursor.execute(
    """
    UPDATE student
    SET tot_cred = tot_cred +
    (SELECT SUM(C.credits)
    FROM course AS C
    JOIN updatecourse AS U
    ON C.course_id = U.course_id
    WHERE U.ID = '1238')
    WHERE ID = '1238';
    """
)
```

In [32]:

```
mycursor = mydb.cursor()

mycursor.execute(
    """
    UPDATE student
    SET tot_cred = tot_cred +
    (SELECT SUM(C.credits)
    FROM course AS C
    JOIN updatecourse AS U
    ON C.course_id = U.course_id
    WHERE U.ID = '1333')
    WHERE ID = '1333';
    """
)
```

In [33]:

```
mycursor = mydb.cursor()

query = """
SELECT ID, name, dept_name, tot_cred
FROM student
WHERE ID IN ('1238', '1333');
"""
mycursor.execute(query)

for x in mycursor:
    print(x)
```

```
('1238', 'John Smith', 'DSI', Decimal('6.00'))
('1333', 'Mukesh Modi', 'DSI', Decimal('9.00'))
```

7. After executing the instructions for 4 and 5, write a query to return the ID and names of those students who have the most credits.

In [34]:

```
mycursor = mydb.cursor()

mycursor.execute(
    """
    SELECT name, ID
    FROM student
    WHERE tot_cred = (SELECT MAX(tot_cred) FROM student)
    """
)

for x in mycursor:
    print(x)
```

```
('Ivan Petrov', '1221')
('Mukesh Modi', '1333')
('Jane Doe', '1645')
('Steve Brown', '2946')
```

mycursor = mydb.cursor() query = "TRUNCATE TABLE student;" mycursor.execute(query)

import pandas as pd data = pd.read_csv("~/Desktop/DATA1050/data/student.csv")

for i,row in data.iterrows(): sql = "INSERT INTO data1050F24.student VALUES (%s,%s,%s,%s)" mycursor.execute(sql, tuple(row)) #print("Record inserted") # the connection is not autocommitted by default, so we # must commit to save our changes mydb.commit()

mycursor = mydb.cursor()

query = """DROP VIEW IF EXISTS coursecredits; DROP VIEW IF EXISTS coursetitle; DROP VIEW IF EXISTS data1050list; DROP VIEW IF EXISTS instructorname; DROP VIEW IF EXISTS originaltakes; DROP VIEW IF EXISTS starttime;""" mycursor.execute(query)

In [ ]: