SAN JOSE STATE UNIVERSITY - CMPE 180B - Database Systems

Phuong Duy Lam, Nguyen

SJSU ID: 018229432

HOMEWORK 1

Due date: 03/07/2025 11:59PM

Problem 4

In [5]:
```
#Install pysqlite3 for python and import pandas to use later
#pip install pysqlite3
from sqlite3 import dbapi2 as sqlite3
print(sqlite3.sqlite_version)
import pandas as pd
from IPython.display import display, HTML
```

```
3.45.3
```

In [6]:
```
dbname = "homework1-4.db"

def printSqlResults(cursor, tblName):
  try:
    df = pd.DataFrame(cursor.fetchall(), columns=[i[0] for i in cursor.description])
    display(HTML("<b><font color=Green> " + tblName + "</font></b>" + df.to_html(index=False)))
  except:
    pass

def runSql(caption, query):
  conn = sqlite3.connect(dbname) # Connect to the database
  cursor = conn.cursor() # Create a cursor (think: it's like a "pointer")
  cursor.execute(query) # Execute the query
  printSqlResults(cursor, caption) # Print the results
  conn.close()

def runSql_withCommit(caption, query):
  conn = sqlite3.connect(dbname) # Connect to the database
  cursor = conn.cursor() # Create a cursor (think: it's like a "pointer")
  cursor.execute(query) # Execute the query
  printSqlResults(cursor, caption) # Print the results
  conn.commit()
  conn.close()

def runStepByStepSql(query, fromline):
  lines = query.strip().split('\n')
  for lineidx in range(fromline, len(lines)):
    partial_query = '\n'.join(lines[:lineidx])
    caption = 'Query till line:' + partial_query
    runSql(caption, partial_query + ';')
```

In [7]:
```
conn = sqlite3.connect(dbname)
cursor = conn.cursor()

# Create the STUDENT table
cursor.execute("""
CREATE TABLE IF NOT EXISTS STUDENT (
    Name VARCHAR(100) NOT NULL,
    Student_number INTEGER PRIMARY KEY,
    Class INTEGER,
    Major VARCHAR(10)
);
""")

# Create the COURSE table
cursor.execute("""
CREATE TABLE IF NOT EXISTS COURSE (
    Course_number VARCHAR(10) PRIMARY KEY,
    Course_name VARCHAR(100) NOT NULL,
    Credit_hours INTEGER,
    Department VARCHAR(10)
);
""")

# Create the SECTION table
cursor.execute("""
CREATE TABLE IF NOT EXISTS SECTION (
    Section_identifier INTEGER PRIMARY KEY,
    Course_number VARCHAR(10) NOT NULL,
    Semester VARCHAR(10),
    Year INTEGER,
    Instructor VARCHAR(100),
    FOREIGN KEY (Course_number) REFERENCES COURSE(Course_number)
);
""")

# Create the GRADE_REPORT table
cursor.execute("""
CREATE TABLE IF NOT EXISTS GRADE_REPORT (
    Student_number INTEGER NOT NULL,
    Section_identifier INTEGER NOT NULL,
    Grade VARCHAR(2),
    PRIMARY KEY (Student_number, Section_identifier),
    FOREIGN KEY (Student_number) REFERENCES STUDENT(Student_number),
    FOREIGN KEY (Section_identifier) REFERENCES SECTION(Section_identifier)
);
""")

# Create the PREREQUISITE table
cursor.execute("""
CREATE TABLE IF NOT EXISTS PREREQUISITE (
    Course_number VARCHAR(10) NOT NULL,
    Prerequisite_number VARCHAR(10) NOT NULL,
    PRIMARY KEY (Course_number, Prerequisite_number),
    FOREIGN KEY (Course_number) REFERENCES COURSE(Course_number),
    FOREIGN KEY (Prerequisite_number) REFERENCES COURSE(Course_number)
);
""")

conn.commit()  # Save the changes
conn.close()


# --- DATA INSERTION ---
conn = sqlite3.connect(dbname)
cursor = conn.cursor()

# Clear existing data (for repeatable testing)
cursor.execute("DELETE FROM STUDENT;")
cursor.execute("DELETE FROM COURSE;")
cursor.execute("DELETE FROM SECTION;")
cursor.execute("DELETE FROM GRADE_REPORT;")
cursor.execute("DELETE FROM PREREQUISITE;")

# Insert data into STUDENT
cursor.execute("""
INSERT INTO STUDENT (Name, Student_number, Class, Major) VALUES
('Smith', 17, 1, 'CS'),
('Brown', 8, 2, 'CS');
""")

# Insert data into COURSE
cursor.execute("""
INSERT INTO COURSE (Course_name, Course_number, Credit_hours, Department) VALUES
('Intro to Computer Science', 'CS1310', 4, 'CS'),
('Data Structures', 'CS3320', 4, 'CS'),
('Discrete Mathematics', 'MATH2410', 3, 'MATH'),
('Database', 'CS3380', 3, 'CS');
""")

# Insert data into SECTION
cursor.execute("""
INSERT INTO SECTION (Section_identifier, Course_number, Semester, Year, Instructor) VALUES
(85, 'MATH2410', 'Fall', 07, 'King'),
(92, 'CS1310', 'Fall', 07, 'Anderson'),
(102, 'CS3320', 'Spring', 08, 'Knuth'),
(112, 'MATH2410', 'Fall', 08, 'Chang'),
(119, 'CS1310', 'Fall', 08, 'Anderson'),
(135, 'CS3380', 'Fall', 08, 'Stone');
""")

# Insert data into GRADE_REPORT
cursor.execute("""
INSERT INTO GRADE_REPORT (Student_number, Section_identifier, Grade) VALUES
(17, 112, 'B'),
(17, 119, 'C'),
(8, 85, 'A'),
(8, 92, 'A'),
(8, 102, 'B'),
(8, 135, 'A');
""")

# Insert data into PREREQUISITE
cursor.execute("""
INSERT INTO PREREQUISITE (Course_number, Prerequisite_number) VALUES
('CS3380', 'CS3320'),
('CS3380', 'MATH2410'),
('CS3320', 'CS1310');
""")


conn.commit()
conn.close()


# --- VERIFY (Optional) ---
conn = sqlite3.connect(dbname)
cursor = conn.cursor()
runSql("Students", "SELECT * FROM STUDENT;")
runSql("Courses", "SELECT * FROM COURSE;")
runSql("Sections", "SELECT * FROM SECTION;")
runSql("Grade Reports", "SELECT * FROM GRADE_REPORT;")
runSql("Prerequisites", "SELECT * FROM PREREQUISITE;")
conn.close()
```

Students

Student_number	Name	Class	Major
8	Brown	2	CS
17	Smith	1	CS

Courses

Course_number	Course_name	Credit_hours	Department
CS1310	Intro to Computer Science	4	CS
CS3320	Data Structures	4	CS
MATH2410	Discrete Mathematics	3	MATH
CS3380	Database	3	CS

Sections

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	7	King
92	CS1310	Fall	7	Anderson
102	CS3320	Spring	8	Knuth
112	MATH2410	Fall	8	Chang
119	CS1310	Fall	8	Anderson
135	CS3380	Fall	8	Stone

Grade Reports

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

Prerequisites

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Specify the following queries in SQL on the database schema in Figure 1.2.

a. Retrieve the number of all straight-A students (students who have a grade of A in all their courses). b. Retrieve the names and major departments of all students who do not have a grade of A in any of their courses.

In [48]:
```
qry_problem4a = """
SELECT COUNT(DISTINCT S.Student_number) AS Straight_A_Students
FROM STUDENT S
WHERE NOT EXISTS (SELECT *
    FROM GRADE_REPORT G
    WHERE G.Grade <> 'A')
;
"""
runSql('Problem 4a', qry_problem4a)
```

Problem 4a

Straight_A_Students
0

In [50]:
```
qry_problem4b = """
SELECT S.Name, S.Major
FROM STUDENT S
WHERE S.Student_number NOT IN (SELECT G.Student_number
    FROM GRADE_REPORT G
    WHERE G.Grade = 'A')
;
"""
runSql('Problem 4b', qry_problem4b)
```

Problem 4b

Name	Major
Smith	CS

In [ ]:
```

```