CSC 623 Project Part 3
Binjie Shen, Yuming Wang, Max Ding

a)

Client table: all fields are required.
Tel should be a valid phone number formats.

```
def regex(e, item):
    return re.match(e, item) is not None

db_connect.create_function("REGEXP", 2, regex)
```

```
## create Client table
## tel format is U.S. phone number

query = """
    CREATE TABLE Client (
    clientNo INT PRIMARY KEY,
    fName VARCHAR(255) NOT NULL,
    lName VARCHAR(255) NOT NULL,
    address VARCHAR(255) NOT NULL,
    tel VARCHAR(20) NOT NULL CHECK (tel REGEXP '^[0-9]{10}$')
    );

"""

cursor.execute(query)

```

Employee table: all fields are required.
Tel should be a valid phone number formats.
Salary might have a minimum value (set 1200)

```
## create Employee table
## tel format is U.S. phone number
## minimum salary per month is 1200

query = """
    CREATE TABLE Employee (
    staffNo INT PRIMARY KEY,
    fName VARCHAR(255) NOT NULL,
    lName VARCHAR(255) NOT NULL,
    address VARCHAR(255) NOT NULL,
    salary DECIMAL(10,2) NOT NULL CHECK (salary > 1200),
    tel VARCHAR(20) NOT NULL CHECK (tel REGEXP '^[0-9]{10}$')
    );

"""

cursor.execute(query)

```

```
<sqlite3.Cursor at 0x1eba03fe340>
```

Requirement table: reqID, startDate, startTime, duration, clientNo should be required
startDate and startTime should be valid datas and times.
Duration should have a range (set 2 to 8)

```
## create Requirement table
## startDate should greater than current date
## duration is between 2 to 8 hours

query = """
    CREATE TABLE Requirement (
    reqID INT PRIMARY KEY,
    startDate DATE NOT NULL CHECK (startDate > CURRENT_DATE),
    startTime TIME NOT NULL,
    duration INT NOT NULL CHECK (duration BETWEEN 2 AND 8),
    comment TEXT,
    clientNo INT NOT NULL,
    FOREIGN KEY (clientNo) REFERENCES Client(clientNo)
    );

"""
cursor.execute(query)
```

```
<sqlite3.Cursor at 0x1eba03fe340>
```

Equipment table: equipID, description, and cost should be required.
Cost should be a positive number (set cost > 0)

```
## create Equipment table
## The cost should be a positive number
query = """
    CREATE TABLE Equipment (
    equipID INT PRIMARY KEY,
    description TEXT NOT NULL,
    usage TEXT,
    cost DECIMAL(10,2) NOT NULL CHECK (cost > 0)
    );

"""
cursor.execute(query)
```

```
<sqlite3.Cursor at 0x1eba03fe340>
```

Assignment Table:

```
## create Assignment table
query = """
    CREATE TABLE Assignment (
    staffNo INT,
    reqID INT,
    PRIMARY KEY (staffNo, reqID),
    FOREIGN KEY (staffNo) REFERENCES Employee(staffNo),
    FOREIGN KEY (reqID) REFERENCES Requirement(reqID)
    );

"""
cursor.execute(query)
```

```
<sqlite3.Cursor at 0x1eba03fe340>
```

Usage table:

```
## create Usage table
query = """
    CREATE TABLE Usage (
    equipID INT,
    reqID INT,
    PRIMARY KEY (equipID, reqID),
    FOREIGN KEY (equipID) REFERENCES Equipment(equipID),
    FOREIGN KEY (reqID) REFERENCES Requirement(reqID)
    );

"""
cursor.execute(query)
```

```
<sqlite3.Cursor at 0x1eba03fe340>
```

Create a trigger to ensure employee is not assigned to a past requirement:

```
## create a trigger to ensure employee is not assigned to a past requirement
query = """
    CREATE TRIGGER check_assignment
    BEFORE INSERT ON Assignment
    FOR EACH ROW
    BEGIN
        SELECT
        CASE
            WHEN (SELECT startDate FROM Requirement WHERE reqID = NEW.reqID) < date('now') OR
                 ((SELECT startDate FROM Requirement WHERE reqID = NEW.reqID) = date('now') AND
                  (SELECT startTime FROM Requirement WHERE reqID = NEW.reqID) < time('now')) THEN
                RAISE(ABORT, 'Cannot assign employee to a past requirement')
        END;
    END;

"""
cursor.execute(query)
```

```
<sqlite3.Cursor at 0x1eba03fe340>
```

b)
Client:

```
query = """
    INSERT INTO Client (clientNo, fName, lName, address, tel) VALUES
    (1, 'Binjie', 'Shen', '123 7th Ave, Miami, FL 33132', '3051234567'),
    (2, 'John', 'Wick', '456 8th Ave , Miami, FL 33176', '3052345678'),
    (3, 'Bill', 'Gates', '789 10th Ave, Orlando, FL 31236', '6811234567'),
    (4, 'Eva', 'Green', '101 6th Ave, Huntington, WV 12345', '9631234567'),
    (5, 'Gal', 'Gadot', '202 9th Ave, New York, NY 23456', '5671234567');
"""
cursor.execute(query)
```

```
<sqlite3.Cursor at 0x1eba03fe340>
```

```
query = """
    SELECT *
    FROM Client
    """
cursor.execute(query)

column_names = [row[0] for row in cursor.description]

table_data = cursor.fetchall()
df = pd.DataFrame(table_data, columns=column_names)

print(df)
print(df.columns)
```

```
   clientNo   fName  lName                           address         tel
0         1  Binjie   Shen       123 7th Ave, Miami, FL 33132  3051234567
1         2    John   Wick      456 8th Ave , Miami, FL 33176  3052345678
2         3    Bill  Gates    789 10th Ave, Orlando, FL 31236  6811234567
3         4     Eva  Green  101 6th Ave, Huntington, WV 12345  9631234567
4         5     Gal  Gadot    202 9th Ave, New York, NY 23456  5671234567
Index(['clientNo', 'fName', 'lName', 'address', 'tel'], dtype='object')
```

Employee:

```
query = """
    INSERT INTO Employee (staffNo, fName, lName, address, salary, tel) VALUES
    (1, 'Mark', 'Taylor', '111 SW 9th Ct, Miami, FL 33125', 3000, '5556789012'),
    (2, 'Linda', 'Wilson', '222 8th Ct, Miami, FL 12345', 2000, '5557890123'),
    (3, 'Steve', 'Martin', '333 SW St, Doral, FL 33178 ', 2500, '5568901234'),
    (4, 'Angela', 'White', '444 Cherry St, Tamarac, FL 15619', 1900, '5559212345'),
    (5, 'Tom', 'Jones', '555 Aspen St, Delray Beach, FL 33999', 2600, '1550123456'),
    (6, 'Mary', 'Garcia', '666 5th Ave, Delray Beach, FL 33678', 2700, '3550123456'),
    (7, 'John', 'Jackson', '777 Cortez Ct, Delray Beach, FL 33444', 2750, '9550123456'),
    (8, 'William', 'Thomas', '888 Via Leonardo, Lake Worth, FL 33467', 3120, '5150123456');

"""
cursor.execute(query)
```

```
<sqlite3.Cursor at 0x1eba03fe340>
```

```
query = """
    SELECT *
    FROM Employee
    """
cursor.execute(query)

column_names = [row[0] for row in cursor.description]

table_data = cursor.fetchall()
df = pd.DataFrame(table_data, columns=column_names)

print(df)
print(df.columns)
```

```
   staffNo    fName    lName                                address  salary  \
0        1     Mark   Taylor          111 SW 9th Ct, Miami, FL 33125    3000
1        2    Linda   Wilson             222 8th Ct, Miami, FL 12345    2000
2        3    Steve   Martin             333 SW St, Doral, FL 33178     2500
3        4   Angela    White        444 Cherry St, Tamarac, FL 15619    1900
4        5      Tom    Jones    555 Aspen St, Delray Beach, FL 33999    2600
5        6     Mary   Garcia     666 5th Ave, Delray Beach, FL 33678    2700
6        7     John  Jackson   777 Cortez Ct, Delray Beach, FL 33444    2750
7        8  William   Thomas  888 Via Leonardo, Lake Worth, FL 33467    3120

          tel
0  5556789012
1  5557890123
2  5568901234
3  5559212345
4  1550123456
5  3550123456
6  9550123456
7  5150123456
Index(['staffNo', 'fName', 'lName', 'address', 'salary', 'tel'], dtype='object')
```

Requirement:

```
query = """
    INSERT INTO Requirement (reqID, startDate, startTime, duration, comment, clientNo) VALUES
    (1, '2023-12-10', '10:00', 2, 'Corridor', 1),
    (2, '2023-12-11', '14:00', 5, 'Hall', 2),
    (3, '2023-12-12', '09:00', 3, 'Meeting room', 3),
    (4, '2023-12-13', '13:00', 4, 'Bathroom', 4),
    (5, '2023-12-14', '15:00', 3, 'Pantry', 5);

"""
cursor.execute(query)
```

```
<sqlite3.Cursor at 0x1eba03fe340>
```

```
query = """
    SELECT *
    FROM Requirement
    """
cursor.execute(query)

column_names = [row[0] for row in cursor.description]

table_data = cursor.fetchall()
df = pd.DataFrame(table_data, columns=column_names)

print(df)
print(df.columns)
```

```
   reqID   startDate startTime  duration       comment  clientNo
0      1  2023-12-10     10:00         2      Corridor         1
1      2  2023-12-11     14:00         5          Hall         2
2      3  2023-12-12     09:00         3  Meeting room         3
3      4  2023-12-13     13:00         4      Bathroom         4
4      5  2023-12-14     15:00         3        Pantry         5
Index(['reqID', 'startDate', 'startTime', 'duration', 'comment', 'clientNo'], dtype='object')
```

Equipment:

```
query = """
    INSERT INTO Equipment (equipID, description, usage, cost) VALUES
    (1, 'Vacuum', 'Clean dust', 1000),
    (2, 'Sponge mop', 'Clean floor', 30),
    (3, 'Squeegee', 'Clean glass', 20),
    (4, 'Warning sign', 'Warn people that the floor is wet', 10),
    (5, 'Mop', '', 15),
    (6, 'Toilet brush', 'Clean toilet', 10),
    (7, 'Plunger', 'Clean toilet', 5);
"""
cursor.execute(query)
```

```
<sqlite3.Cursor at 0x1eba03fe340>
```

```
query = """
    SELECT *
    FROM Equipment
    """
cursor.execute(query)

column_names = [row[0] for row in cursor.description]

table_data = cursor.fetchall()
df = pd.DataFrame(table_data, columns=column_names)

print(df)
print(df.columns)
```

```
   equipID   description                          usage  cost
0        1       Vacuum                    Clean dust  1000
1        2    Sponge mop                   Clean floor    30
2        3      Squeegee                   Clean glass    20
3        4  Warning sign  Warn people that the floor is wet    10
4        5           Mop                                15
5        6  Toilet brush                  Clean toilet    10
6        7       Plunger                  Clean toilet     5
Index(['equipID', 'description', 'usage', 'cost'], dtype='object')
```

Assignment:

```
query = """
    INSERT INTO Assignment (staffNo, reqID) VALUES
    (1, 1),
    (2, 2),
    (3, 3),
    (4, 4),
    (5, 5),
    (6, 1),
    (7, 2),
    (8, 2);
"""
cursor.execute(query)
```

```
<sqlite3.Cursor at 0x1eba03fe340>
```

```
query = """
    SELECT *
    FROM Assignment
    """
cursor.execute(query)

column_names = [row[0] for row in cursor.description]

table_data = cursor.fetchall()
df = pd.DataFrame(table_data, columns=column_names)

print(df)
print(df.columns)
```

```
   staffNo  reqID
0        1      1
1        2      2
2        3      3
3        4      4
4        5      5
5        6      1
6        7      2
7        8      2
Index(['staffNo', 'reqID'], dtype='object')
```

Usage:

```
query = """
    INSERT INTO Usage (equipID, reqID) VALUES
    (1, 1),
    (2, 2),
    (3, 3),
    (4, 2),
    (5, 5),
    (6, 4),
    (7, 4);
"""
cursor.execute(query)
```

```
<sqlite3.Cursor at 0x1eba03fe340>
```

```
query = """
    SELECT *
    FROM Usage
    """
cursor.execute(query)

column_names = [row[0] for row in cursor.description]

table_data = cursor.fetchall()
df = pd.DataFrame(table_data, columns=column_names)

print(df)
print(df.columns)
```

```
   equipID  reqID
0        1      1
1        2      2
2        3      3
3        4      2
4        5      5
5        6      4
6        7      4
Index(['equipID', 'reqID'], dtype='object')
```

c)
Display all the elements within the requirements of the task that requires more than one piece of equipment.

```
## Display all the elements within the requirements of the task that requires more than one piece of equipment.
query = """
   SELECT R.*
    FROM Requirement R
    JOIN (
        SELECT reqID
        FROM Usage
        GROUP BY reqID
        HAVING COUNT(equipID) > 1
    ) AS U ON R.reqID = U.reqID
"""
cursor.execute(query)
```

```
<sqlite3.Cursor at 0x1eba03fe340>
```

```
column_names = [row[0] for row in cursor.description]

table_data = cursor.fetchall()
df = pd.DataFrame(table_data, columns=column_names)

print(df)
print(df.columns)
```

```
   reqID   startDate startTime  duration   comment  clientNo
0      2  2023-12-11     14:00         5      Hall         2
1      4  2023-12-13     13:00         4  Bathroom         4
Index(['reqID', 'startDate', 'startTime', 'duration', 'comment', 'clientNo'], dtype='object')
```

Display all the elements within the requirements of the task that requires more than one employee.

```
## Display all the elements within the requirements of the task that requires more than one employee.
query = """
   SELECT R.*
    FROM Requirement R
    JOIN (
        SELECT reqID
        FROM Assignment
        GROUP BY reqID
        HAVING COUNT(staffNo) > 1
    ) AS A ON R.reqID = A.reqID
"""
cursor.execute(query)
```

```
<sqlite3.Cursor at 0x1eba03fe340>
```

```
column_names = [row[0] for row in cursor.description]

table_data = cursor.fetchall()
df = pd.DataFrame(table_data, columns=column_names)

print(df)
print(df.columns)
```

```
   reqID   startDate startTime  duration   comment  clientNo
0      1  2023-12-10     10:00         2  Corridor         1
1      2  2023-12-11     14:00         5      Hall         2
Index(['reqID', 'startDate', 'startTime', 'duration', 'comment', 'clientNo'], dtype='object')
```

Display all the elements of the requirement whose start time is in the afternoon.

```
## Display all the elements of the requirement whose start time is in the afternoon.
query = """
   SELECT * FROM Requirement
    WHERE substr(startTime, 1, 2) >= '12'
"""
cursor.execute(query)
```

```
<sqlite3.Cursor at 0x1eba03fe340>
```

```
column_names = [row[0] for row in cursor.description]

table_data = cursor.fetchall()
df = pd.DataFrame(table_data, columns=column_names)

print(df)
print(df.columns)
```

```
   reqID   startDate startTime  duration   comment  clientNo
0      2  2023-12-11     14:00         5      Hall         2
1      4  2023-12-13     13:00         4  Bathroom         4
2      5  2023-12-14     15:00         3    Pantry         5
Index(['reqID', 'startDate', 'startTime', 'duration', 'comment', 'clientNo'], dtype='object')
```

Display all the elements of the employee whose job is in the morning.

```
## Display all the elements of the employee whose job is in the morning.
query = """
    SELECT E.*
     FROM Employee E
     JOIN Assignment A ON E.staffNo = A.staffNo
     JOIN Requirement R ON A.reqID = R.reqID
     WHERE substr(R.startTime, 1, 2) < '12'
"""
cursor.execute(query)
```

```
<sqlite3.Cursor at 0x1eba03fe340>
```

```
column_names = [row[0] for row in cursor.description]

table_data = cursor.fetchall()
df = pd.DataFrame(table_data, columns=column_names)

print(df)
print(df.columns)
```

```
   staffNo  fName   lName                            address  salary  \
0        1   Mark  Taylor       111 SW 9th Ct, Miami, FL 33125    3000
1        3  Steve  Martin          333 SW St, Doral, FL 33178     2500
2        6   Mary  Garcia  666 5th Ave, Delray Beach, FL 33678    2700

          tel
0  5556789012
1  5568901234
2  3550123456
Index(['staffNo', 'fName', 'lName', 'address', 'salary', 'tel'], dtype='object')
```

Display the description of the equipment used in the requirement labeled with the comment 'Bathroom'.

```
## Display the description of the equipment used in the requirement labeled with the comment 'Bathroom'.
query = """
    SELECT E.description
    FROM Equipment E
    JOIN Usage U ON E.equipID = U.equipID
    JOIN Requirement R ON U.reqID = R.reqID
    WHERE R.comment = 'Bathroom'
"""
cursor.execute(query)
```

```
<sqlite3.Cursor at 0x1eba03fe340>
```

```
column_names = [row[0] for row in cursor.description]

table_data = cursor.fetchall()
df = pd.DataFrame(table_data, columns=column_names)

print(df)
print(df.columns)
```

```
    description
0  Toilet brush
1       Plunger
Index(['description'], dtype='object')
```

d)
GitHub link: https://github.com/Binjs313/CSC623_finalProject.git