

James Frierson, Cecile Darwiche, Riyan Ibadah
CSC 423 Database Systems
Group Project Part 3

a. Develop SQL code to create the entire database schema, reflecting the constraints identified in previous steps

Client Table: We started by creating a Client table to hold client information. It has fields for client number, first and last name, address, and a contact number. The client number is the primary key, ensuring each client has a unique identifier.

Equipment Table: Next, we created an Equipment table to keep track of equipment owned by the business. This includes an equipment ID, description, usage, and cost. We imposed constraints on cost and usage to prevent them from having negative values.

Employee Table: The Employee table records details about the staff, including an employee ID, name, address, salary, and phone number. A salary constraint ensures that no negative values are entered.

Requirement Table: The Table Requirement keeps track of requirements related to client requests. This includes a requirement ID, start date and time, duration, a client number linking back to the Client table, and comments. A foreign key constraint is used to link clientNum to the Client table, which helps keep consistency across the database.

```

query = """
CREATE TABLE Client(
  ClientNum INT NOT NULL,
  fName VARCHAR(50),
  lName VARCHAR(50),
  address VARCHAR(100),
  number VARCHAR(10),
  PRIMARY KEY(ClientNum)
);
"""

#ON DELETE CASCADE will delete all of the children of the main tables
# Execute query, the result is stored in cursor
cursor.execute(query)

query = """
CREATE TABLE Equipment(
  eqID INT,
  description VARCHAR(500),
  usage INT NOT NULL,
  cost FLOAT,
  CONSTRAINT AllowedCost CHECK (cost >= 0)
  CONSTRAINT TimesUsed CHECK (usage >= 0)
  PRIMARY KEY (eqID)
);
"""

cursor.execute(query)

#CONSTRAINT FOR PHONE NUMBER BEING 10 DIGITS?
query = """
CREATE TABLE Employee(
  staffNum INT,
  fName VARCHAR(50),
  lName VARCHAR(50),
  address VARCHAR(100),
  salary FLOAT CHECK(salary > 0),
  number INT,
  PRIMARY KEY(staffNum)
);
"""

cursor.execute(query)

## DOES REQUIREMENT HAVE CLIENTNUM FOREIGN KEY HERE LIKE ER DIAGRAM??
query = """
CREATE TABLE Requirement(
  reqID INT,
  startD DATE,
  startT TIME,
  duration TIME,
  clientNum INT,
  comments VARCHAR(500),
  PRIMARY KEY (reqID)
  FOREIGN KEY (clientNum) REFERENCES Client(clientNum)
);
"""

cursor.execute(query)

```

B. Create at least 5 tuples for each relation in your database

Employee Data Insertion: We inserted records into the Employee table. Each tuple contains a staff number, a first name, a last name, an address, a salary, and a phone number. These records provide a basic dataset of employees for the business.

Equipment Data Insertion: For the Equipment table, we added entries representing different pieces of equipment. Each tuple includes an equipment ID, a description, the number of times the equipment has been used (usage), and its cost.

Client Data Insertion: In the Client table, we inserted client records. Each record consists of a client number, first name, last name, address, and a contact number, creating a dataset of clients associated with the business.

Requirement Data Insertion: We inserted data into the Requirement table. Each record includes a requirement ID, a start date, a start time, a duration, a client number (linking to a record in the Client table), and comments related to the requirement.

```
query = """
INSERT INTO Employee
VALUES
    (1, 'Alice', 'Johnson', '789 Oak St', 30000, 3456789012),
    (2, 'Bob', 'Williams', '101 Pine St', 32000, 4567890123),
    (3, 'John', 'Ronaldo', '134 Po St', 35000, 4566960123),
    (4, 'Lionel', 'Messi', '563 Po St', 33000, 6566960123),
    (5, 'Lebron', 'James', '431 Spruce St', 31000, 4566966423);
"""

cursor.execute(query)

query= """
INSERT INTO Equipment
VALUES
    (1, 'Vacuum Cleaner', 20, 150.00),
    (2, 'Mop', 9, 20.00),
    (3, 'Broom', 26, 25.00),
    (4, 'Duster', 11, 50.00),
    (5, 'Plunger', 32, 20.00);
"""

cursor.execute(query)

query = """
INSERT INTO Client
VALUES
    (1, 'John', 'Doe', '123 Main St', 1234567890),
    (2, 'Jane', 'Smith', '456 Elm St', 2345678901),
    (3, 'Anne', 'Wolf', '573 Palm Dr', 3057836721),
    (4, 'Maria', 'Wolf', '7025 89th St', 3055312756),
    (5, 'Carly', 'Hess', '562 45th Pl', 7864539877);
"""

cursor.execute(query)

query = """
INSERT INTO Requirement
VALUES
    (1, '2024-01-03', '10:30:00', 'Clean bathroom', 4 , 5),
    (2, '2023-12-02', '09:00:00', 'Wash windows', 2, 4),
    (3, '2023-10-02', '07:00:00', 'Wash floor', 2 , 3),
    (4, '2022-10-22', '05:00:00', 'Clean Kitchen', 1 , 2),
    (5, '2023-05-16', '11:15:00', 'Clean floor tile', 3 , 1);
"""

cursor.execute(query)
```

```

ClientNum  fName  lName  address  number
0         1   John   Doe    123 Main St  1234567890
1         2    Jane   Smith  456 Elm St  2345678901
2         3    Anne   Wolf   573 Palm Dr  3057836721
3         4   Maria   Wolf   7025 89th St  3055312756
4         5    Carly  Hess   562 45th Pl  7864539877
Index(['ClientNum', 'fName', 'lName', 'address', 'number'], dtype='object')

staffNum  fName  lName  address  salary  number
0         1  Alice  Johnson  789 Oak St  30000.0  3456789012
1         2    Bob  Williams  101 Pine St  32000.0  4567890123
2         3   John  Ronaldo  134 Po St  35000.0  4566960123
3         4  Lionel  Messi   563 Po St  33000.0  6566960123
4         5  LeBron  James  431 Spruce St  31000.0  4566966423
Index(['staffNum', 'fName', 'lName', 'address', 'salary', 'number'], dtype='object')

reqID     startD  startT  duration  clientNum  comments
0         1  2024-01-03  10:30:00  Clean bathroom  4         5
1         2  2023-12-02  09:00:00  Wash windows    2         4
2         3  2023-10-02  07:00:00  Wash floor      2         3
3         4  2022-10-22  05:00:00  Clean Kitchen   1         2
4         5  2023-05-16  11:15:00  Clean floor tile  3         1
Index(['reqID', 'startD', 'startT', 'duration', 'clientNum', 'comments'], dtype='object')

eqID     description  usage  cost
0         1 Vacuum Cleaner  20  150.0
1         2           Mop    9   20.0
2         3          Broom   26   25.0
3         4          Duster  11   50.0
4         5         Plunger  32   20.0
Index(['eqID', 'description', 'usage', 'cost'], dtype='object')

```

C. Develop 5 SQL queries using embedded SQL

In part c of the SQL code, we developed five SQL queries that serve different purposes within the database:

Retrieve Client Names with Their Cleaning Start Date and Time: The first query gets the first and last names of clients along with the start date and time of their cleaning requirements. This is achieved through a JOIN operation between the Client and Requirement tables on the clientNum field.

Calculate Total Salary Expense for All Employees: The second query calculates the sum of all salaries for employees. This is a simple aggregation query that uses the SUM function on the salary column of the Employee table.

Retrieve All Requirements Along with Client Information: The third query is designed to fetch all details from the Requirement table and the associated client's first and last name from the Client table. This again involves a JOIN operation between the Requirement and Client tables.

Find Clients with No Cleaning Requirements: The fourth query identifies clients who do not have any cleaning requirements. This is done using a LEFT JOIN between the Client and Requirement tables and checking for NULL in the Requirement table's reqID field, which would indicate no linked requirements.

Number of Services Scheduled in 2024: The final query aims to count all services scheduled for the year 2024 by searching for a pattern in the startD field of the Requirement table.

```

#Retrieve Client Name with Their Cleaning Start Date and Time
query = """
    SELECT Client.fName, Client.lName, Requirement.StartD, Requirement.startT
    FROM Client
    JOIN Requirement ON Client.clientNum = Requirement.clientNum;
    """
cursor.execute(query)
print_table(cursor)

#calculate total salary expense for all employees
query = """
SELECT SUM(E.salary) AS TotalSalaryExpense
FROM Employee E;
    """
cursor.execute(query)
print_table(cursor)

#retrieve all requirements along with client information
query = """
    SELECT
        R.reqID,
        R.startD,
        R.startT,
        R.duration,
        R.comments,
        C.fName AS ClientFirstName,
        C.lName AS ClientLastName
    FROM Requirement R
    JOIN Client C ON R.ClientNum = C.ClientNum;
    """
cursor.execute(query)
print_table(cursor)

#Find clients with no cleaning requirements
query = """
SELECT
    C.ClientNum,
    C.fName,
    C.lName
FROM Client C
LEFT JOIN Requirement R ON C.ClientNum = R.ClientNum
WHERE R.reqID IS NULL;
    """
cursor.execute(query)
print_table(cursor)

#Number of services scheduled in 2024
query = """
SELECT *
    FROM Requirement r
    WHERE startD LIKE '%2024%';
    """
cursor.execute(query)
print_table(cursor)

```

```

    fName lName      startD      startT
0 Maria   Wolf  2024-01-03  10:30:00
1 Jane   Smith  2023-12-02  09:00:00
2 Jane   Smith  2023-10-02  07:00:00
3 John    Doe   2022-10-22  05:00:00
4 Anne   Wolf  2023-05-16  11:15:00
Index(['fName', 'lName', 'startD', 'startT'], dtype='object')

    TotalSalaryExpense
0                161000.0
Index(['TotalSalaryExpense'], dtype='object')

    reqID      startD      startT      duration  comments  ClientFirstName  ClientLastName
0      1  2024-01-03  10:30:00  Clean bathroom      5           Maria           Wolf
1      2  2023-12-02  09:00:00  Wash windows      4           Jane           Smith
2      3  2023-10-02  07:00:00  Wash floor        3           Jane           Smith
3      4  2022-10-22  05:00:00  Clean Kitchen      2           John            Doe
4      5  2023-05-16  11:15:00  Clean floor tile    1           Anne           Wolf
Index(['reqID', 'startD', 'startT', 'duration', 'comments', 'ClientFirstName',
      'ClientLastName'],
      dtype='object')

    ClientNum  fName lName
0           5  Carly  Hess
Index(['ClientNum', 'fName', 'lName'], dtype='object')

    reqID      startD      startT      duration  clientNum  comments
0      1  2024-01-03  10:30:00  Clean bathroom      4           5
Index(['reqID', 'startD', 'startT', 'duration', 'clientNum', 'comments'], dtype='object')

```

Upload all the code and documentation to GitHub

Link: <https://github.com/RiyanIbad/Database-final>