CSC423 Project Case Study: SuperMaids Cleaning Company

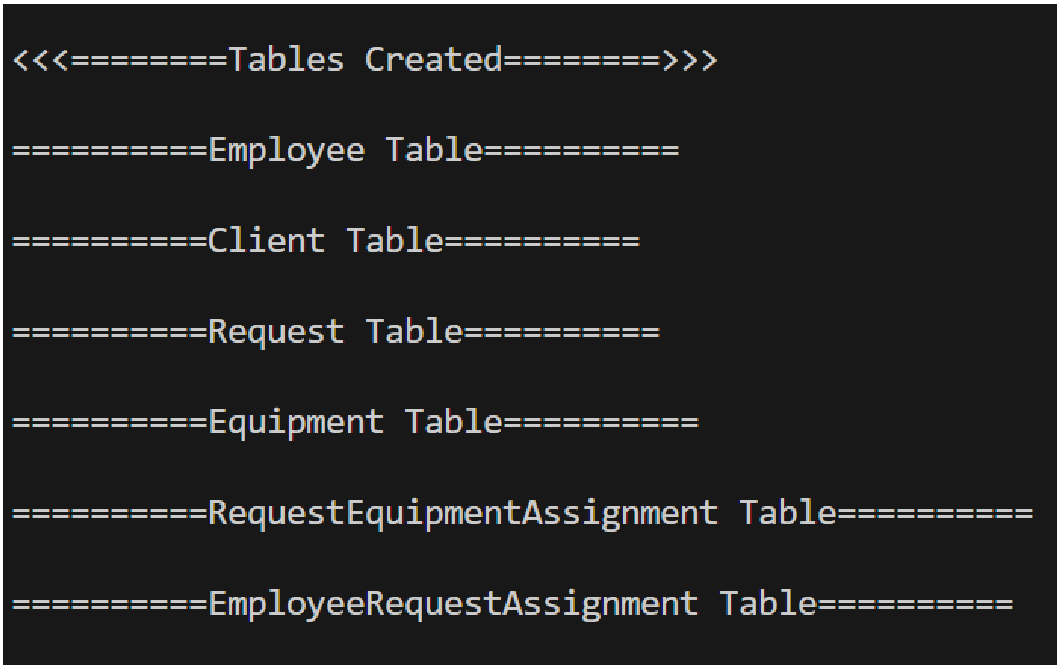
Team 9:
Zubaer Chowdhury
Oscar Arana
Justin Prince

Part 3:

GitHub: The code generated during step 3 (SQL statements + program) must be uploaded to a GitHub repository. The link to the repository must be provided in the last report. The GitHub repository must also include all the documentation (i.e., reports) generated in the three steps.

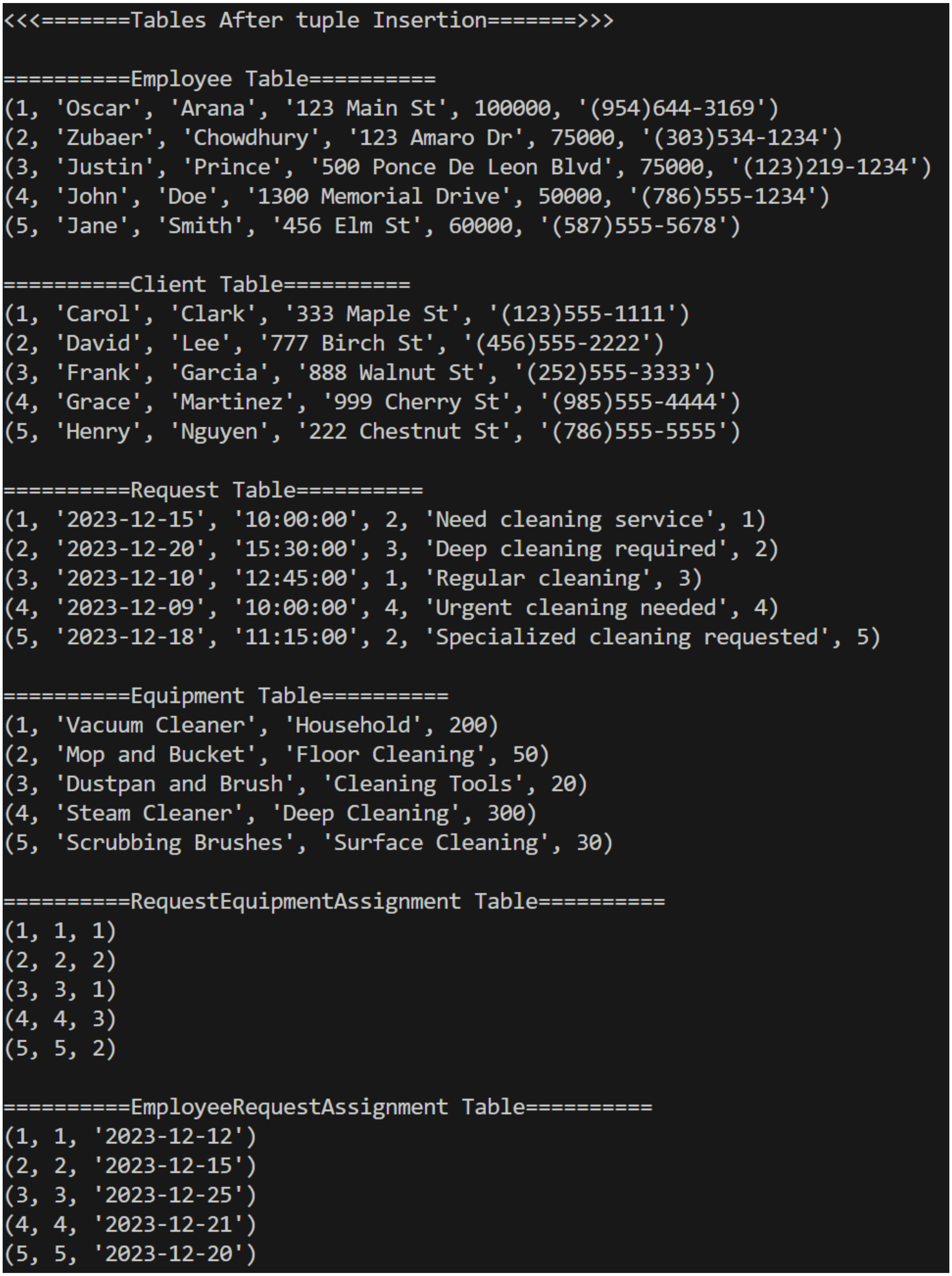
Translate the logical data model for the DBMS.
a. Develop SQL code to create the entire database schema, reflecting the constraints identified in previous steps.

(SQL code of database given in `.py` file of Github repo)
Ans a: `cleaning_service.db` view:



```
<<<========Tables Created========>>>

==========Employee Table==========

==========Client Table==========

==========Request Table==========

==========Equipment Table==========

==========RequestEquipmentAssignment Table==========

==========EmployeeRequestAssignment Table==========
```

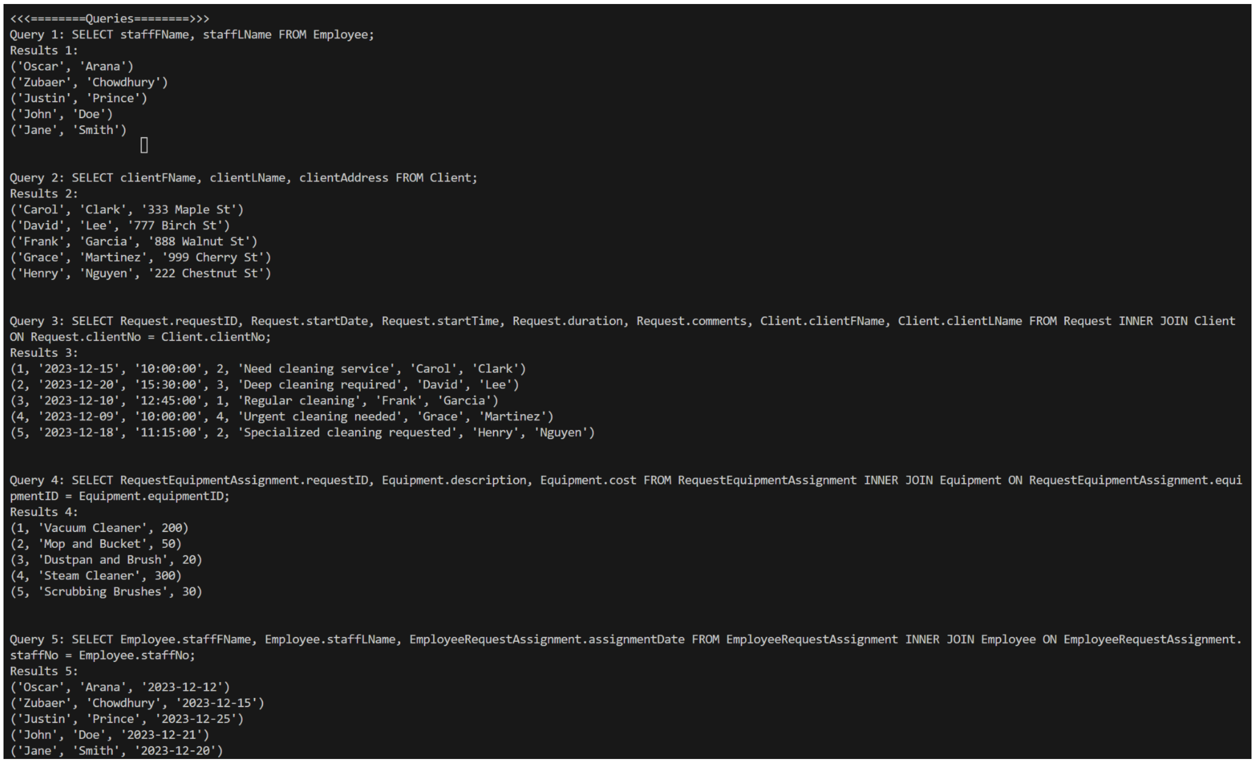
b. Create at least 5 tuples for each relation in your database.
Ans b: `cleaning_service.db` view:



```
<<<=======Tables After tuple Insertion=======>>>

==========Employee Table==========
(1, 'Oscar', 'Arana', '123 Main St', 100000, '(954)644-3169')
(2, 'Zubaer', 'Chowdhury', '123 Amaro Dr', 75000, '(303)534-1234')
(3, 'Justin', 'Prince', '500 Ponce De Leon Blvd', 75000, '(123)219-1234')
(4, 'John', 'Doe', '1300 Memorial Drive', 50000, '(786)555-1234')
(5, 'Jane', 'Smith', '456 Elm St', 60000, '(587)555-5678')

==========Client Table==========
(1, 'Carol', 'Clark', '333 Maple St', '(123)555-1111')
(2, 'David', 'Lee', '777 Birch St', '(456)555-2222')
(3, 'Frank', 'Garcia', '888 Walnut St', '(252)555-3333')
(4, 'Grace', 'Martinez', '999 Cherry St', '(985)555-4444')
(5, 'Henry', 'Nguyen', '222 Chestnut St', '(786)555-5555')

==========Request Table==========
(1, '2023-12-15', '10:00:00', 2, 'Need cleaning service', 1)
(2, '2023-12-20', '15:30:00', 3, 'Deep cleaning required', 2)
(3, '2023-12-10', '12:45:00', 1, 'Regular cleaning', 3)
(4, '2023-12-09', '10:00:00', 4, 'Urgent cleaning needed', 4)
(5, '2023-12-18', '11:15:00', 2, 'Specialized cleaning requested', 5)

==========Equipment Table==========
(1, 'Vacuum Cleaner', 'Household', 200)
(2, 'Mop and Bucket', 'Floor Cleaning', 50)
(3, 'Dustpan and Brush', 'Cleaning Tools', 20)
(4, 'Steam Cleaner', 'Deep Cleaning', 300)
(5, 'Scrubbing Brushes', 'Surface Cleaning', 30)

==========RequestEquipmentAssignment Table==========
(1, 1, 1)
(2, 2, 2)
(3, 3, 1)
(4, 4, 3)
(5, 5, 2)

==========EmployeeRequestAssignment Table==========
(1, 1, '2023-12-12')
(2, 2, '2023-12-15')
(3, 3, '2023-12-25')
(4, 4, '2023-12-21')
(5, 5, '2023-12-20')
```

c. Develop 5 SQL queries using embedded SQL (see embedded SQL tutorial).

Ans c: `cleaning_service.db` view:



```
<<<========Queries========>>>
Query 1: SELECT staffFName, staffLName FROM Employee;
Results 1:
('Oscar', 'Arana')
('Zubaer', 'Chowdhury')
('Justin', 'Prince')
('John', 'Doe')
('Jane', 'Smith')


Query 2: SELECT clientFName, clientLName, clientAddress FROM Client;
Results 2:
('Carol', 'Clark', '333 Maple St')
('David', 'Lee', '777 Birch St')
('Frank', 'Garcia', '888 Walnut St')
('Grace', 'Martinez', '999 Cherry St')
('Henry', 'Nguyen', '222 Chestnut St')


Query 3: SELECT Request.requestID, Request.startDate, Request.startTime, Request.duration, Request.comments, Client.clientFName, Client.clientLName FROM Request INNER JOIN Client
ON Request.clientNo = Client.clientNo;
Results 3:
(1, '2023-12-15', '10:00:00', 2, 'Need cleaning service', 'Carol', 'Clark')
(2, '2023-12-20', '15:30:00', 3, 'Deep cleaning required', 'David', 'Lee')
(3, '2023-12-10', '12:45:00', 1, 'Regular cleaning', 'Frank', 'Garcia')
(4, '2023-12-09', '10:00:00', 4, 'Urgent cleaning needed', 'Grace', 'Martinez')
(5, '2023-12-18', '11:15:00', 2, 'Specialized cleaning requested', 'Henry', 'Nguyen')


Query 4: SELECT RequestEquipmentAssignment.requestID, Equipment.description, Equipment.cost FROM RequestEquipmentAssignment INNER JOIN Equipment ON RequestEquipmentAssignment.equi
pmentID = Equipment.equipmentID;
Results 4:
(1, 'Vacuum Cleaner', 200)
(2, 'Mop and Bucket', 50)
(3, 'Dustpan and Brush', 20)
(4, 'Steam Cleaner', 300)
(5, 'Scrubbing Brushes', 30)


Query 5: SELECT Employee.staffFName, Employee.staffLName, EmployeeRequestAssignment.assignmentDate FROM EmployeeRequestAssignment INNER JOIN Employee ON EmployeeRequestAssignment.
staffNo = Employee.staffNo;
Results 5:
('Oscar', 'Arana', '2023-12-12')
('Zubaer', 'Chowdhury', '2023-12-15')
('Justin', 'Prince', '2023-12-25')
('John', 'Doe', '2023-12-21')
('Jane', 'Smith', '2023-12-20')
```

d. Upload all the code and documentation to GitHub.

Ans d: Pdf reports of all three parts of the project, the SQL `.py` code along with a `README.md`, which gives an overview of the entire project, have been uploaded to the Github repo: https://github.com/osarana/Supermaids