CSC423/CSC623
Project: Design, development and implementation of a relational database
Part 3

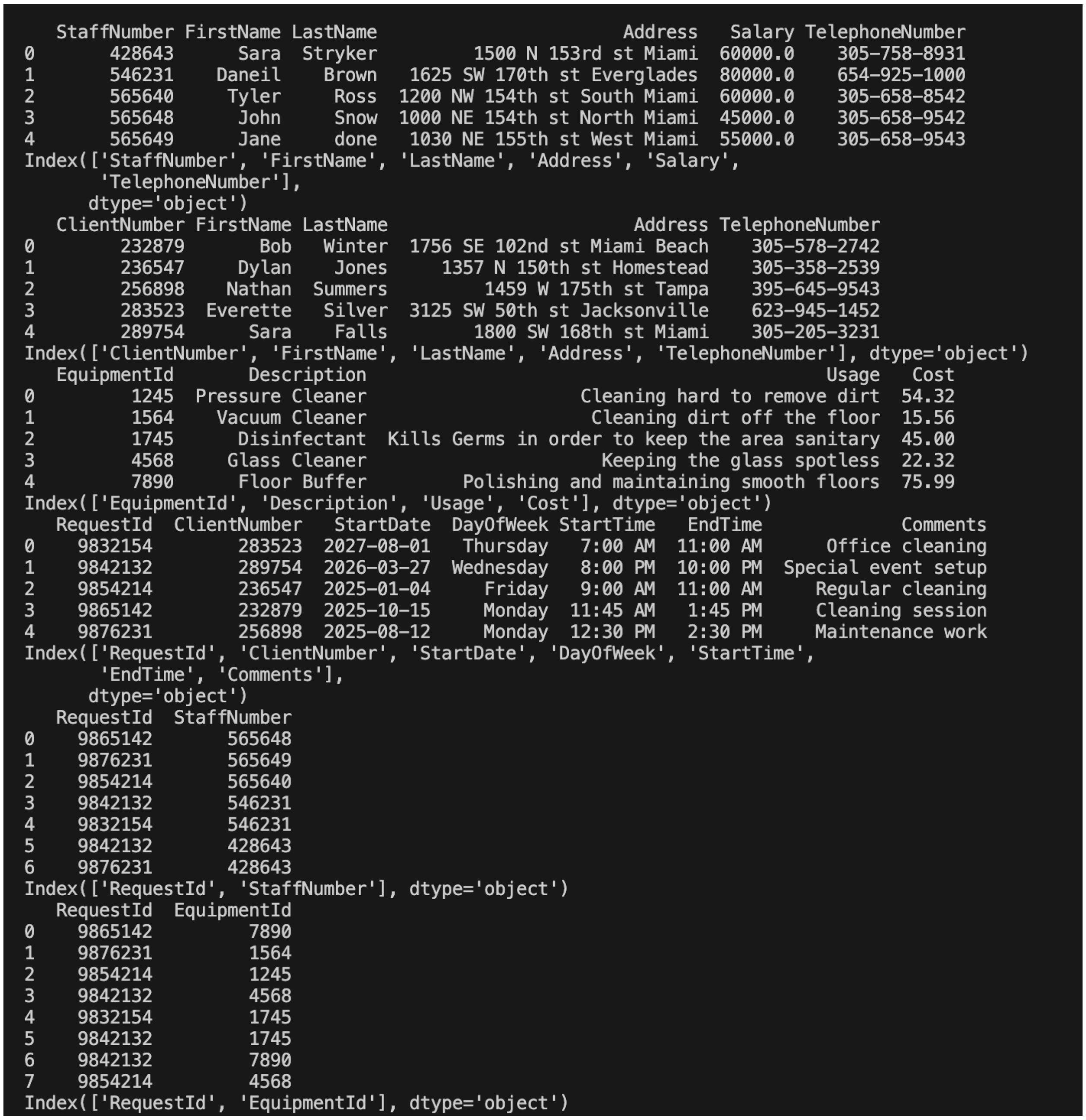
Pablo de la Fuente, Jesse Aronson, Tyler Tejera

Part 3: Translate the logical data model for the DBMS. (12/07/23)
a. Develop SQL code to create the entire database schema, reflecting the constraints identified in previous steps.
b. Create at least 5 tuples for each relation in your database.
c. Develop 5 SQL queries using embedded SQL (see embedded SQL tutorial).
d. Upload all the code and documentation to GitHub.

A. To develop the entire database schema, we created a total of 6 tables to reflect the requirements of the cleaning service company. Those include the 4 entities called Employee, Client, Equipment and request. Additionally, 2 more tables were added as intermediate tables to create the many-to-many relationships between request and equipment and between request and employee. The creation of the tables also include the constraints stated in part 2 of the project, such as not nullable variables, and that salary must be greater than 0. Finally, the creation of the tables also includes the addition of foreign keys to create the relationships between the tables. Those foreign keys are added between in the request table and in the intermediate tables. As explained in chapter 17: in many-to-many relationships types, we posted a copy of the primary key of the entities to act as foreign keys as well as to act in combination as primary keys of the intermediate tables.

B. For part B, we inserted at least 5 tuples for each table, with some additional tuples in the intermediate table to show that the relationships work effectively. Additionally, we tested the insertion of the tables using “SELECT * from tableName”.

The result was the following in the order of Employee, Client, Equipment, Request, RequestEmployee and RequestEquipment:



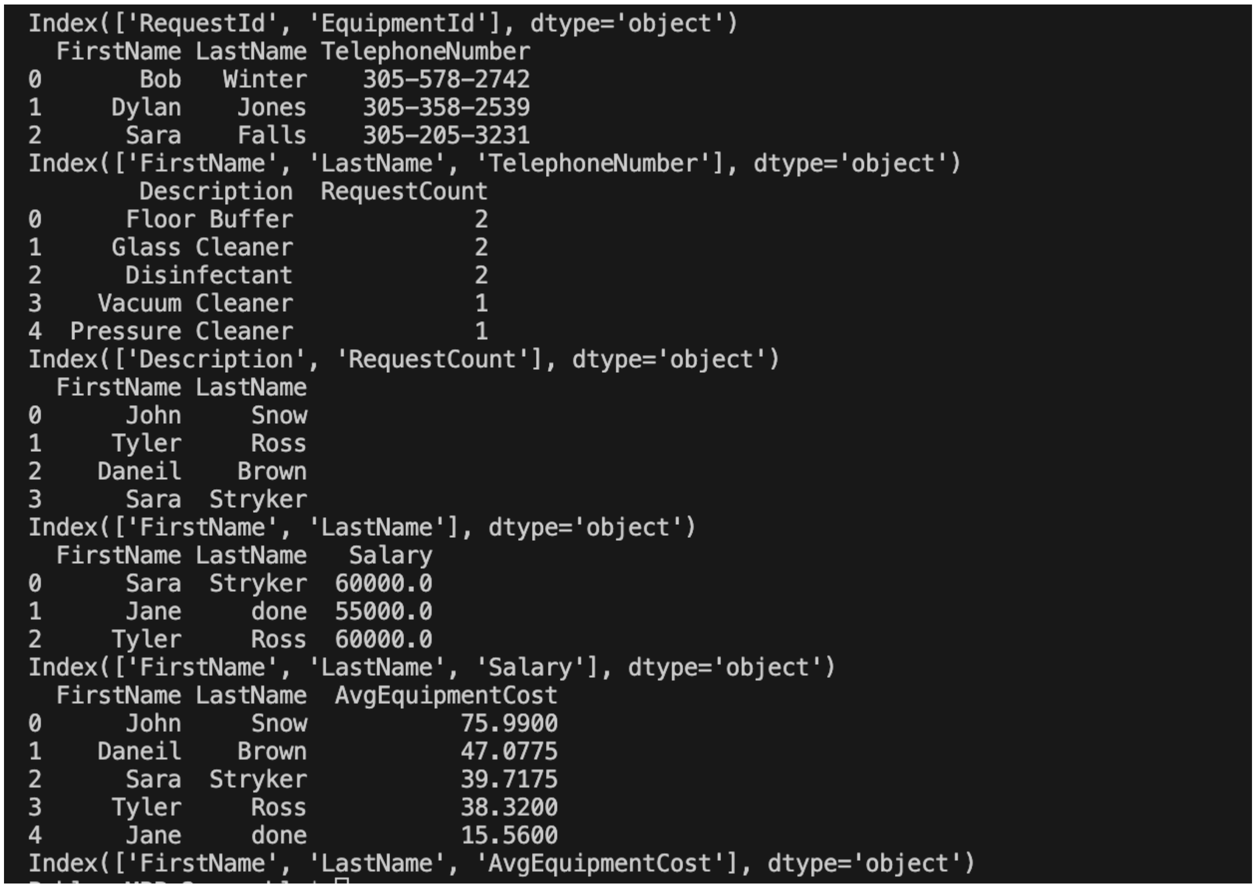
```
   StaffNumber FirstName LastName                        Address   Salary TelephoneNumber
0       428643      Sara  Stryker         1500 N 153rd st Miami  60000.0    305-758-8931
1       546231    Daneil    Brown   1625 SW 170th st Everglades  80000.0    654-925-1000
2       565640     Tyler     Ross  1200 NW 154th st South Miami  60000.0    305-658-8542
3       565648      John     Snow  1000 NE 154th st North Miami  45000.0    305-658-9542
4       565649      Jane     done   1030 NE 155th st West Miami  55000.0    305-658-9543
Index(['StaffNumber', 'FirstName', 'LastName', 'Address', 'Salary',
       'TelephoneNumber'],
      dtype='object')
   ClientNumber FirstName LastName                        Address TelephoneNumber
0        232879       Bob   Winter  1756 SE 102nd st Miami Beach    305-578-2742
1        236547     Dylan    Jones     1357 N 150th st Homestead    305-358-2539
2        256898    Nathan  Summers         1459 W 175th st Tampa    395-645-9543
3        283523  Everette   Silver  3125 SW 50th st Jacksonville    623-945-1452
4        289754      Sara    Falls        1800 SW 168th st Miami    305-205-3231
Index(['ClientNumber', 'FirstName', 'LastName', 'Address', 'TelephoneNumber'], dtype='object')
   EquipmentId       Description                                     Usage   Cost
0         1245  Pressure Cleaner                    Cleaning hard to remove dirt  54.32
1         1564    Vacuum Cleaner                    Cleaning dirt off the floor  15.56
2         1745      Disinfectant  Kills Germs in order to keep the area sanitary  45.00
3         4568     Glass Cleaner                      Keeping the glass spotless  22.32
4         7890      Floor Buffer         Polishing and maintaining smooth floors  75.99
Index(['EquipmentId', 'Description', 'Usage', 'Cost'], dtype='object')
   RequestId  ClientNumber   StartDate  DayOfWeek StartTime   EndTime            Comments
0    9832154        283523  2027-08-01   Thursday   7:00 AM  11:00 AM     Office cleaning
1    9842132        289754  2026-03-27  Wednesday   8:00 PM  10:00 PM  Special event setup
2    9854214        236547  2025-01-04     Friday   9:00 AM  11:00 AM     Regular cleaning
3    9865142        232879  2025-10-15     Monday  11:45 AM   1:45 PM     Cleaning session
4    9876231        256898  2025-08-12     Monday  12:30 PM   2:30 PM     Maintenance work
Index(['RequestId', 'ClientNumber', 'StartDate', 'DayOfWeek', 'StartTime',
       'EndTime', 'Comments'],
      dtype='object')
   RequestId  StaffNumber
0    9865142       565648
1    9876231       565649
2    9854214       565640
3    9842132       546231
4    9832154       546231
5    9842132       428643
6    9876231       428643
Index(['RequestId', 'StaffNumber'], dtype='object')
   RequestId  EquipmentId
0    9865142         7890
1    9876231         1564
2    9854214         1245
3    9842132         4568
4    9832154         1745
5    9842132         1745
6    9842132         7890
7    9854214         4568
Index(['RequestId', 'EquipmentId'], dtype='object')
```

C. For Part C we included 5 realistic and complicated example queries that can be used by the user. The 5 queries test all the

relationships and tables to make sure that everything is working correctly.

Here are the results of the 5 queries in order:

1. **Identify clients who have made requests for equipment with a cost greater than $50:**
2. **Find the equipment that has been requested the most and order the usage in descending order:**
3. **Retrieve the staff names of clients whose phone number starts with '305':**
4. **Find all staff who have been assigned to a request that involves cleaning equipment and include the salary:**
5. **Average Cost of Equipment Used by Each Employee:**



```
Index(['RequestId', 'EquipmentId'], dtype='object')
  FirstName LastName TelephoneNumber
0       Bob   Winter    305-578-2742
1     Dylan    Jones    305-358-2539
2      Sara    Falls    305-205-3231
Index(['FirstName', 'LastName', 'TelephoneNumber'], dtype='object')
        Description  RequestCount
0      Floor Buffer             2
1     Glass Cleaner             2
2      Disinfectant             2
3    Vacuum Cleaner             1
4  Pressure Cleaner             1
Index(['Description', 'RequestCount'], dtype='object')
  FirstName LastName
0      John     Snow
1     Tyler     Ross
2    Daneil    Brown
3      Sara  Stryker
Index(['FirstName', 'LastName'], dtype='object')
  FirstName LastName   Salary
0      Sara  Stryker  60000.0
1      Jane     done  55000.0
2     Tyler     Ross  60000.0
Index(['FirstName', 'LastName', 'Salary'], dtype='object')
  FirstName LastName  AvgEquipmentCost
0      John     Snow           75.9900
1    Daneil    Brown           47.0775
2      Sara  Stryker           39.7175
3     Tyler     Ross           38.3200
4      Jane     done           15.5600
Index(['FirstName', 'LastName', 'AvgEquipmentCost'], dtype='object')
```