DSC 450: Database Processing for Large-Scale Analytics Assignment Module 5

Part 1

Using the company.sql database (posted in with this assignment), write the following SQL queries.

1. Find the names of all employees who are directly supervised by 'Franklin T Wong' (you cannot use Franklin's SSN in the query).

```
SELECT FNAME, MINIT, LNAME
FROM Employee
WHERE DNO IN (SELECT DNUMBER
FROM Department
WHERE DNUMBER = 5 AND LNAME!='Wong');

FNAME MINIT LNAME

1 John B Smith
2 Ramesh K Narayan
3 Joyce A English
4 Melissa M Jones
```

2. For each project, list the project name, project number, and the total hours per week (by all employees) spent on that project.

```
--2
SELECT PNAME, PNUMBER, (SELECT SUM(HOURS)
FROM Works_On
WHERE PNUMBER = PNO) AS TotalHoursWK
FROM Project
ORDER BY TotalHoursWK desc;
```

	PNAME		
1	ProductX	1	62
2	Newbenefits	30	60
3	Computerization	10	55
4	ProductY	2	38
5	Reorganization	20	35
6	ProductZ	3	5

3. For each department, retrieve the department name and the average salary of all employees working in that department. Order the output by department number in ascending order.

```
--3
SELECT DNUMBER, DNAME, (SELECT AVG(SALARY)
FROM Employee
WHERE DNUMBER = DNO) AS AVG_SAL
FROM Department
ORDER BY DNUMBER asc;

DNUMBER DNAME
AVG_SAL

1 1 Headquarters 55000
2 4 Administration 28000
3 5 Research 32100
```

4. Retrieve the average salary of all female employees.

```
SELECT AVG(SALARY)
FROM Employee
WHERE Sex = 'F';

AVG(SALARY)
1 28625
```

5. For each department whose average salary is greater than \$42,000, retrieve the department name and the number of employees in that department.

```
--5

SELECT DNAME, (SELECT COUNT(FNAME)

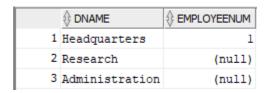
FROM Employee

WHERE DNUMBER = DNO

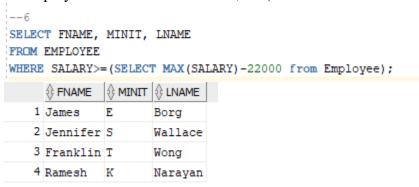
HAVING AVG(SALARY)>42000) AS EmployeeNum

FROM Department

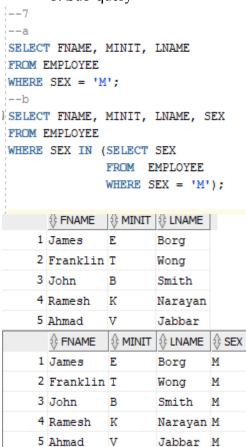
ORDER BY EmployeeNum;
```



6. Retrieve the names of employees whose salary is within \$22,000 of the salary of the employee who is paid the most in the company (e.g., if the highest salary in the company is \$85,000, retrieve the names of all employees that make at least \$63,000.).

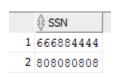


- 7. Find all male employees using:
 - a. Plain SELECT query
 - b. Sub-query



8. Find all employees who are not assigned to any project using SET operation in SQL

```
--8
SELECT SSN FROM EMPLOYEE
MINUS
SELECT ESSN FROM WORKS_ON
```



Part 2

Create the table and use python to automate loading of the following file into SQLite: http://rasinsrv07.cstcis.cti.depaul.edu/CSC455/Public_Chauffeurs_Short_hw3.csv
It contains comma-separated data, with two changes: NULL may now be represented by NULL string or an empty string (e.g., either ,NULL, or ,,) and some of the names have the following form "Last, First" instead of "First Last", which is problematic because when you split the string on a comma, you end up with too many values to insert.

```
You can use csvreader to automatically solve the problem for you: import csv fd = open('Public_Chauffeurs_Short_hw3.csv', 'r') reader = csv.reader(fd) for row in reader: print(row) fd.close()
```

```
|Chauffeurs =
CREATE TABLE Chauffeurs (
    LicenseNumber NUMBER,
    RENEWED
             DATE,
    STATUS
              VARCHAR2(15),
    StatusDate DATE,
   DriverType VARCHAR2(30),
    LicenseType VARCHAR2(30),
    OGIssueDate VARCHAR2 DATE,
    NAME VARCHAR2(75),
    SEX VARCHAR2(15),
    ChauffeurCity VARCHAR2(50),
    ChauffeurState VARCHAR2(2),
    RecordNumber VARCHAR2(2) NOT NULL UNIQUE,
    Constraint Chauffeur_PK
        PRIMARY KEY(RecordNumber)
```

```
import csv
fd = open('Public_Chauffeurs_Short_hw3.csv', 'r')
reader = csv.reader(fd)
lst = []
for row in reader:
    lst.append(row)
    print(lst)

import sqlite3
conn = sqlite3.connect('dsc450.db')
cursor = conn.cursor()

cursor.execute(Chauffeurs) #chauffeurs tbl

C = cursor.executemany("INSERT OR IGNORE INTO Chauffeurs VALUES (?,?,?,?,?,?,?,?,?,?,?)",lst)
```

```
C = cursor.executemany("INSERT OR IGNORE INTO Chauffeurs VALUES (?,?,?,?,?,?,?,?,?,?)",lst)

query1 = '''SELECT * FROM Chauffeurs'''

for lines in cursor.execute(query1).fetchall():
    print(lines)

fd.close()
close = conn.close()
```

Print goes on longer

Part 3

We are going to work with a small extract of tweets (about 200 of them), available here: http://rasinsrv07.cstcis.cti.depaul.edu/CSC455/Assignment4.txt

NOTE 1: I do not recommend trying to copy-paste this text, because there is absolutely no knowing what might come out from paste on your system. You should be able to use "Save as..." function in your browser.

NOTE 2: The input data is separated by a string "EndOfTweet" which serves as a delimiter. The text itself consists of a single line, so using readline() or even readlines() will still only give you one row which needs to be split by the custom delimiter (i.e. .split('EndOfTweet')).

- a. Create a SQL table to contain the following attributes of a tweet:

 "created_at", "id_str", "text", "source", "in_reply_to_user_id", "in_reply_to_screen_name",

 "in_reply_to_status_id", "retweet_count", "contributors". Please assign reasonable data types to
 each attribute and use SQLite for this assignment.
- b. Write python code to read through the Assignment4.txt file and populate your table from part a. Make sure your python code reads through the file and loads the data properly (including NULLs).

```
CREATE TABLE Tweets (
    created_at VARCHAR2(25),
    id_str VARCHAR2(25),
    text VARCHAR2(280),
    source VARCHAR2(100),
    in_reply_to_user_id VARCHAR2(30),
    in_reply_to_screen_name VARCHAR2(30),
    in_reply_to_status_id VARCHAR2 (30),
    retweet_count NUMBER,
    contributors VARCHAR2(15),
    Constraint Tweets_PK
        PRIMARY KEY(id_str)

);
```

```
close = conn.close()
tweetData.close()
```

```
>>> query1 = '''SELECT * FROM Tweets'''
...
>>> for lines in cursor.execute(query1).fetchall():
... print(lines)
...
('Tue Nov 05 00:00:04 +0000 2013', '39751360971874048', 'JUST GOT HOME AND SAW WE BROKE THE VEVO RECORD! IF YOU HELPED I LOVE YOU', '<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>', None, None, None, None, None, None, None, None)
('Tue Nov 05 00:00:04 +0000 2013', '397513609732845568', 'We are here to play tough cricket: Richardson <a href="http://t.co/aNFIdXmo8h"/">http://t.co/aNFIdXmo8h</a>', '<a href="http://scoop.so" rel="nofollow">SCOOP Hot News India</a>', None, None, None, None, None, None, None)
('Tue Nov 05 00:00:04 +0000 2013', '397513609732816896', 'I really love kids, always have always will.', '<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>', None, None,
```

Keeps going