CSC 455 HW#3
Due Date:04-30-2017 Email: kying@mail.depaul.edu

Part 1: Write the SQL queries from ZooDatabased.sql to answer questions:

```
/*Part1-1*/
select aname, zookeepid
from handles, animal
where aid=ANIMALID;

/*Part1-2*/
select aname, zookeepid
from animal left outer join handles
ON aid=ANIMALID;

/*Part1-3*/
select zname, SUM(timetofeed)
from zookeeper, handles, animal
where aid=animalid and zid=zookeepid
group by zname;

/*Part1-4*/
select Assigned, zname, aname
from zookeeper, handles, animal
where zid=zookeepid and aid=animalid
order by assigned ASC;

/*Part1-5*/
select aname
from animal,  handles
where aid=animalid
group by aname
HAVING count(animalid) >=1;

/*Part1-6*/
select aname
from animal outer left join  handles
ON aid=animalid
group by aname
```

CSC 455 HW#3
Due Date:04-30-2017 Email: kying@mail.depaul.edu
Part 2-A:
Write a python script that is going to read the queries that you have created in Part-1 from a SQL file, execute each SQL query against SQLite database and print the output of that query.

```
#Provided
import sqlite3
from sqlite3 import OperationalError

conn = sqlite3.connect('csc455_HW3.db')
c = conn.cursor()

# Open and read the file as a single buffer
fd = open('ZooDatabase.sql', 'r')
# Read as a single document (not individual lines)
sqlFile = fd.read()
fd.close()

# all SQL commands (split on ';' which separates them)
sqlCommands = sqlFile.split(';')

# Execute every command from the input file (separated by ";")
for command in sqlCommands:
    # This will skip and report errors
    # For example, if the tables do not yet exist, this will skip over
    # the DROP TABLE commands
    try:
        c.execute(command)
    except OperationalError as msg:
        print ("Command skipped: ", msg)

c.close()
conn.commit()
conn.close()
```

In []:

```
#Part2-A: Python script to read queries from Part-1 from SQL file,
#Execute each SQL query against SQLite database and print the output of the query
import sqlite3
from sqlite3 import OperationalError

fd = open ('csc455_wk4_assign_part2a.sql')
content=fd.read()
fd.close()
content=content.strip().split(';')

#Part2-A: The for lopp to read query against the SQL db to output the result
for i in range(len(content)-1):
    data=c.execute(content[i]).fetchall()
    print('Part2-',i+1)
    for u in range(len(data)):
        print(data[u])
    print('\t')
```

The following is the result queries against the db based on the provided condition

Out []:

```
Part2- 1
('Galapagos Penguin', 1)
('Emperor Penguin', 1)
('Alpaca', 1)
('Sri Lankan sloth bear', 2)
('Grizzly bear', 2)
('Giant Panda bear', 2)
('Siberian tiger', 3)
('Bengal tiger', 3)
('South China tiger', 3)
('Alpaca', 3)

Part2- 2
('Galapagos Penguin', 1, 0.5)
('Emperor Penguin', 1, 0.75)
('Sri Lankan sloth bear', 2, 2.5)
('Grizzly bear', 2, 3)
('Giant Panda bear', 2, 1.5)
('Florida black bear', None, 1.75)
('Siberian tiger', 3, 3.5)
('Bengal tiger', 3, 2.75)
('South China tiger', 3, 2.25)
```

```
('Alpaca', 1, 0.25)
('Alpaca', 3, 0.25)
('Llama', None, 3.5)

Part2- 3
('Jim Carrey', 1.5)
('Rob Schneider', 8.75)
('Tina Fey', 7.0)

Part2- 4
('01-Jan-2000', 'Jim Carrey', 'Galapagos Penguin')
('01-Jan-2000', 'Jim Carrey', 'Alpaca')
('01-Jan-2000', 'Rob Schneider', 'Siberian tiger')
('02-Jan-2000', 'Jim Carrey', 'Emperor Penguin')
('02-Jan-2000', 'Tina Fey', 'Sri Lankan sloth bear')
('03-Jan-2000', 'Tina Fey', 'Giant Panda bear')
('03-Jan-2000', 'Rob Schneider', 'Bengal tiger')
('04-Jan-2000', 'Tina Fey', 'Grizzly bear')
('04-Jan-2000', 'Rob Schneider', 'Alpaca')
('05-Jan-2000', 'Rob Schneider', 'South China tiger')

Part2- 5
('Alpaca',)
('Bengal tiger',)
('Emperor Penguin',)
('Galapagos Penguin',)
('Giant Panda bear',)
('Grizzly bear',)
('Siberian tiger',)
('South China tiger',)
('Sri Lankan sloth bear',)

Part2- 6
('Bengal tiger',)
('Emperor Penguin',)
('Florida black bear',)
('Galapagos Penguin',)
('Giant Panda bear',)
('Grizzly bear',)
('Llama',)
('Siberian tiger',)
('South China tiger',)
('Sri Lankan sloth bear',)
```

```
#Part 2-B: Create the table and use python to automate loading of the following
file into SQLite:
#Tools: ipython Notebook
#Method: Use pandas library to read and load the table from the URL
#Assumption:
#Replace the nan by None which is nonetype
#Primary Key: License Number (No Schema is formatted since no Functional Depend
ency provided)
#INTEGER Type: License_Number
#DATE Type: Status_Date, Original Issue_Date
#String Type: Renewed, Status,Driver_Type, License_Type, Name, Sex, Chauffeur_C
ity, Chauffeur_State, Record_Number

# Set up the Sqlite3
import sqlite3
from sqlite3 import OperationalError
#Create hw3_2b DataBase
conn = sqlite3.connect('hw3_2b.db')
c = conn.cursor()

#Import pandas and numpy library
import pandas as pd
import numpy as np

#Use pd.read_csv to read the URL and save to content list
content=pd.read_csv("http://rasinsrv07.cstcis.cti.depaul.edu/CSC455/Public_Chau
ffeurs_Short_hw3.csv"
                    ,index_col=False, header=0)

#Replace all the pd nan by None value in order to insert to sQL
content = content.where((pd.notnull(content)), None)
```

*** The followings are screenshots checking the table that we read***

```
In [6]: #Check out first 5 rows of Table
content[0:5]
```

Out[6]:

	License Number	Renewed	Status	Status Date	Driver Type	License Type	Original Issue Date	Name	Sex	Chauffeur City	Chauffeur State	Record Number
0	103344	None	DENIED	03/21/2014	Livery Only	None	None	ABDALLAH ATIEH	MALE	TINLEY PARK	IL	14-00367283
1	101721	None	INACTIVE	10/02/2014	Taxi	None	None	ABDIRISAK I ABDI	MALE	CHICAGO	IL	14-01682108
2	101711	None	INACTIVE	10/02/2014	Taxi	None	None	ZHENGQIANGU FU	MALE	CHICAGO	IL	14-01682846
3	101710	None	INACTIVE	10/02/2014	Taxi	None	None	ROBERT LEE ROSS	MALE	CHICAGO	IL	14-01682727
4	101709	None	INACTIVE	10/02/2014	Taxi	None	None	ABDIRAHIM ABDULLAHI HAJI	MALE	CHICAGO	IL	14-01682591

```
In [7]: #Check out last 5 rows of Table
content[-5:]
```

Out[7]:

	License Number	Renewed	Status	Status Date	Driver Type	License Type	Original Issue Date	Name	Sex	Chauffeur City	Chauffeur State	Record Number
995	100711	None	ACTIVE	05/09/2014	Taxi	PERMANENT	05/09/2014	ABBES MOSTEFAOUI	MALE	CHICAGO	IL	14-00641227
996	100710	None	ACTIVE	05/19/2014	Taxi	PERMANENT	05/19/2014	DOVRANGELDI DANATAROV	MALE	CHICAGO	IL	14-00641198
997	100709	09/2014	ACTIVE	05/09/2014	Taxi	PERMANENT	05/09/2014	CLAUDIA B RODRIGUEZ-SILVA	FEMALE	ORLAND PARK	IL	14-00641172
998	100708	None	ACTIVE	05/09/2014	Livery Only	PERMANENT	05/09/2014	JIAHAO ZHANG	MALE	CHICAGO	IL	14-00641142
999	100707	None	ACTIVE	05/12/2014	Taxi	PERMANENT	05/12/2014	BRIAN B SIGLER	MALE	CHICAGO	IL	14-00641116

```
In [8]: # Check row[0] value
content.ix[0]
```

```
Out[8]: License Number                  103344
        Renewed                           None
        Status                          DENIED
        Status Date                 03/21/2014
        Driver Type                Livery Only
        License Type                      None
        Original Issue Date               None
        Name                   ABDALLAH ATIEH
        Sex                               MALE
        Chauffeur City             TINLEY PARK
        Chauffeur State                     IL
        Record Number             14-00367283
        Name: 0, dtype: object
```

```
In [9]: #Check None type to verify
type(content.ix[0][1])
```

```
Out[9]: NoneType
```

```
#Create Table 'License' Script
Li = '''CREATE TABLE License
(
    License_Number INTEGER NOT NULL,
    Renewed varchar(10),
    Status varchar(8),
    Status_Date DATE,
    Driver_Type varchar(15),
    License_Type varchar(10),
    Original_Issue_Date DATE,
    Name varchar(25),
    Sex varchar(50),
    Chauffeur_City varchar(50),
    Chauffeur_State varchar(50),
    Record_Number varchar(50)

);'''

#Drop the Table 'License'
c.execute("DROP TABLE License")
#Populate the Table 'License'
c.execute(Li)
```

```
#INSERT statement from the Table which is called 'content', Total 12 attributes
for i in range(len(content)):
    c.execute("INSERT INTO License VALUES (?,?,?,?,?,?,?,?,?,?,?,?)",
              (int(content.ix[i][0]),
               content.ix[i][1],
               content.ix[i][2],
               content.ix[i][3],
               content.ix[i][4],
               content.ix[i][5],
               content.ix[i][6],
               content.ix[i][7],
               content.ix[i][8],
               content.ix[i][9],
               content.ix[i][10],
               content.ix[i][11]))
```

The following screenshot is used for checking None (Nonetype) value from the table column 2 (Attribute= Renewed), the screenshot is not showing all the rows because the record is too long

In [13]:
```
#Double check the records where the column "Renewed" is NULL
data=c.execute("select * from License where Renewed is NULL").fetchall()
for u in range(len(data)):
    print(data[u])
```

```
(103344, None, 'DENIED', '03/21/2014', 'Livery Only', None, None, 'ABDALLAH ATIEH', 'MALE', 'TINLEY PARK', 'IL'
, '14-00367283')
(101721, None, 'INACTIVE', '10/02/2014', 'Taxi', None, None, 'ABDIRISAK I ABDI', 'MALE', 'CHICAGO', 'IL', '14-0
1682108')
(101711, None, 'INACTIVE', '10/02/2014', 'Taxi', None, None, 'ZHENGQIANGU FU', 'MALE', 'CHICAGO', 'IL', '14-016
82846')
(101710, None, 'INACTIVE', '10/02/2014', 'Taxi', None, None, 'ROBERT LEE ROSS', 'MALE', 'CHICAGO', 'IL', '14-01
682727')
(101709, None, 'INACTIVE', '10/02/2014', 'Taxi', None, None, 'ABDIRAHIM ABDULLAHI HAJI', 'MALE', 'CHICAGO', 'IL
', '14-01682591')
(101708, None, 'INACTIVE', '10/02/2014', 'Livery Only', None, None, 'YAHYA H ELKHATIB', 'MALE', 'OAK LAWN', 'IL
', '14-01678733')
(101707, None, 'INACTIVE', '10/02/2014', 'Taxi', None, None, 'SAUGAT CHAPAIN', 'MALE', 'CHICAGO', 'IL', '14-016
82383')
(101706, None, 'INACTIVE', '10/02/2014', 'Taxi', None, None, 'SYED MUBASHIR ALI', 'MALE', 'CHICAGO', 'IL', '14-
01682274')
(101705, None, 'INACTIVE', '10/02/2014', 'Taxi', None, None, 'HASSAN M DARIYAI', 'MALE', 'CHICAGO', 'IL', '14-0
1681608')
(101704, None, 'INACTIVE', '10/02/2014', 'Taxi', None, None, 'ASFAW ADMASSU BAYOU', 'MALE', 'CHICAGO', 'IL', '1
4-01677906')
(101703, None, 'INACTIVE', '10/02/2014', 'Taxi', None, None, 'WAYNE SHINAGAWA', 'MALE', 'CHICAGO', 'IL', '14-01
677462')
(101702, None, 'INACTIVE', '10/02/2014', 'Taxi', None, None, 'KHAMIT SEITOV', 'MALE', 'CHICAGO', 'IL', '14-0168
1399')
(101701, None, 'INACTIVE', '10/02/2014', 'Livery Only', None, None, 'CLYDE A CHRISTIAN JR', 'MALE', 'CHICAGO',
'IL', '14-01680087')
(101700, None, 'INACTIVE', '10/02/2014', 'Livery Only', None, None, 'MURAD TAIYM', 'MALE', 'TINLEY PARK', 'IL',
'14-01679989')
(101699, None, 'INACTIVE', '10/02/2014', 'Taxi', None, None, 'TEMIRLAN KOCHKOROV', 'MALE', 'CHICAGO', 'IL', '14
-01678635')
```

*** The following screenshot is showing the Table with query against the sql where *Chauffeur_City=='OAK LAWN' (Column 10)*

```
In [14]: #Double check and output the table agein to query the Chauffeur_City=='OAK LAWN'
         data=c.execute("select * from License where Chauffeur_City=='OAK LAWN';").fetchall()
         for l in data:
             print(l)

(101708, None, 'INACTIVE', '10/02/2014', 'Livery Only', None, None, 'YAHYA H ELKHATIB', 'MALE', 'OAK LAWN', 'IL
', '14-01678733')
(101438, None, 'ACTIVE', '08/18/2014', 'Livery Only', 'PERMANENT', '08/20/2014', 'ADNAN MUSTAFA', 'MALE', 'OAK
LAWN', 'IL', '14-01361216')
(101300, None, 'ACTIVE', '07/28/2014', 'Livery Only', 'PERMANENT', '07/28/2014', 'MOHMOUD NAJI', 'MALE', 'OAK L
AWN', 'IL', '14-01174866')
(101290, '08/2014', 'ACTIVE', '08/18/2014', 'Taxi', 'PERMANENT', '07/25/2014', 'SALEH A. MUTHANNA', 'MALE', 'OA
K LAWN', 'IL', '14-01148236')
(101254, None, 'ACTIVE', '07/22/2014', 'Taxi', 'PERMANENT', '07/22/2014', 'RIYAD Y. SAID', 'MALE', 'OAK LAWN',
'IL', '14-01132513')
(101208, None, 'ACTIVE', '07/11/2014', 'Livery Only', 'PERMANENT', '07/11/2014', 'ALLEN SHELBY', 'MALE', 'OAK L
AWN', 'IL', '14-01073052')
(101127, None, 'ACTIVE', '07/02/2014', 'Livery Only', 'PERMANENT', '07/02/2014', 'MORAD MOHD ALSARAS', 'MALE',
'OAK LAWN', 'IL', '14-00997175')
```

CSC 455 HW#3
Due Date:04-30-2017 Email: kying@mail.depaul.edu
Part 3: Using the company.sql database, write the SQL queries.

```
/*Part3-1*/
select e2.Fname,e2.MINIT, e2.lname, ' superised by ', e1.Fname,e1.MINIT,e1.lname
from employee e2, employee e1
where e2.super_ssn = e1.ssn and e1.fname='Franklin' and e1.minit='T' and e1.lname='Wong';


/*Part3-2*/
select pname, pnumber, sum(hours)
from Project, Works_on
where pno=pnumber
group by pname,pnumber;

/*Part3-3*/
select dname, AVG(salary)
from department, employee
where DNO = Dnumber
group by dnumber,dname
order by dnumber;

/*Part3-4*/
select AVG(Salary)
from employee
where SEX='F';

/*Part3-5*/
select dname, avg(salary),count(ssn) AS number_of_employees
from department, employee
where dnumber=dno
group by dname
having avg(salary)>42000;

/*Part3-6*/
select fname, minit,lname,(select max(salary) from employee)-salary AS diff
from employee
where (select max(salary) from employee)-salary<25000;
```