Sarah Cummings
Assignment 3 CSC455

Part 1 -SQL queries for the following:

1. List the animals (animal names) and the ID of the zoo keeper assigned to them.

```
SELECT Animal.AName, Handles.ZooKeepID
FROM Animal, Handles
WHERE Animal.AID = Handles.ANIMALID;
```

2. Now repeat the previous query and make sure that the animals without a handler also appear in the answer.

```
SELECT Animal.AName, Handles.ZooKeepID
FROM Animal FULL OUTER JOIN Handles
ON Animal.AID = Handles.ANIMALID;
```

3. Report, for every zoo keeper name, the total number of hours they spend feeding all animals in their care.

```
SELECT ZooKeeper.ZName, SUM(Animal.TimeToFeed)
 FROM ZooKeeper, Animal, Handles
WHERE ZooKeeper.ZID = Handles.ZooKeepID AND
Animal.AID=Handles.AnimalID
```

4. Report every handling assignment (as a list of assignment date, zoo keeper name and animal name). Sort the result of the query by the assignment date in an ascending order.

```
SELECT Handles.ASSIGNED, ZooKeeper.ZName, Animal.AName
FROM Handles, ZooKeeper, Animal
WHERE ZooKeeper.ZID=Handles.ZooKeepID AND
Animal.AID= Handles.AnimalID
ORDER BY Handles.ASSIGNED ASC;
```

5. Find the names of animals that have at least 1 zoo keeper assigned to them.

```
SELECT Animal.ANAME
FROM Animal, Handles
WHERE Animal.AID = Handles.AnimalID
AND Handles.ZooKeepID IS NOT NULL;
```

6. Find the names of animals that have 0 or 1 (i.e., less than 2) zoo keepers assigned to them.

```
SELECT Animal.AName,
FROM Animal FULL OUTER JOIN Handles
ON Animal.AID = Handles.ANIMALID
WHERE Handles.ZooKeepID IS NULL
```

Part 2- Run part 1 queries from a SQL file and print the results in Python

```
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('SarahC_Assignment3.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).fetchall()
    except OperationalError:
        print ("Command skipped: "+ command)

print("Answer to Part 1 Number 1: ")
print( c.execute(sqlCommands[30]).fetchall())

#I can't get #2 to work because FULL OUTER JOINS aren't supported in sqllite
#I can't get #3 to work in general (take off points for it in part 1, not thi part!)


print("Answer to Part 1 Number 4: ")
print(c.execute(sqlCommands[33]).fetchall())

print("Answer to Part 1 Number 5: ")
print(c.execute(sqlCommands[34]).fetchall())

#I get how to go about this assignment but my queries are wrong in part 1 :(
c.close()
conn.commit()
conn.close()
```

Part 3 – SQL Queries using Company.sql for the following:

1. Find the names of all employees who are directly supervised by 'Franklin T Wong'.

```
SELECT Employee.Fname, Employee.Lname
FROM Employee
WHERE Employee.Dno = 5 AND Employee.Lname IS NOT ‘Wong’
```

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

```
SELECT Project.Pname, SUM(Works_On.Hours)
FROM Project, Works_On
WHERE Project.Pnumber=Works_On.Pnumber
GROUP BY Project.Pname
```

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.

```
SELECT Department.Dname, AVG(Employee.Salary)
FROM Department, Employee
WHERE Employee.Dno=Department.Dnumber
ORDER BY Department.Dnumber ASC
```

4. Retrieve the average salary of all female employees.

```
SELECT AVG(Employee.Salary)
FROM Employee
Where Employee.Sex = ‘F’
```

5. For each department whose average salary is greater than $43,000, retrieve the department name and the number of employees in that department.

```
SELECT Dname, count(*)
FROM, Department, Employee
WHERE Avg(Salarary) >=43000 IN

(SELECT Dname, Avg(Salary)
FROM Employee, Department
Where Employee.Dno= Department.Dnumber
GROUP BY Dname)
```

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 $82,000, retrieve the names of all employees that make at least $60,000.).

```
SELECT Employee.Fname, Employee.Minit, Employee.Lname
FROM Employee
WHERE Employee.Salary >= MAX(Salary) - 22000
```

kj kj