Christian Craig
DSC 450 - Hw3

DSC 450: Database Processing for Large-Scale Analytics

Assignment Module 3

Part 1

You were hired to do some data analysis for a local zoo. Below is the data table, including the necessary constraints and all the insert statements to populate the database.

```
-- Drop all the tables to clean up
DROP TABLE Animal;

-- ACategory: Animal category 'common', 'rare', 'exotic'.  May be NULL
-- TimeToFeed: Time it takes to feed the animal (hours)
CREATE TABLE Animal
(
  AID       NUMBER(3, 0),
  AName      VARCHAR2(30) NOT NULL,
  ACategory VARCHAR2(18),

  TimeToFeed NUMBER(4,2),

  CONSTRAINT Animal_PK
    PRIMARY KEY(AID)
);
INSERT INTO Animal VALUES(1, 'Galapagos Penguin', 'exotic', 0.5);
INSERT INTO Animal VALUES(2, 'Emperor Penguin', 'rare', 0.75);
INSERT INTO Animal VALUES(3, 'Sri Lankan sloth bear', 'exotic', 2.5);
INSERT INTO Animal VALUES(4, 'Grizzly bear', 'common', 3.0);
INSERT INTO Animal VALUES(5, 'Giant Panda bear', 'exotic', 1.5);
INSERT INTO Animal VALUES(6, 'Florida black bear', 'rare', 1.75);
INSERT INTO Animal VALUES(7, 'Siberian tiger', 'rare', 3.25);
INSERT INTO Animal VALUES(8, 'Bengal tiger', 'common', 2.75);
INSERT INTO Animal VALUES(9, 'South China tiger', 'exotic', 2.5);
INSERT INTO Animal VALUES(10, 'Alpaca', 'common', 0.25);
INSERT INTO Animal VALUES(11, 'Llama', NULL, 3.5);
```

Since none of the managers in the zoo know SQL, it is up to you to write the queries to answer the following list of questions.

1. Find all the animals (their names) that take less than 1.5 hours to feed
2. Find both the rare and exotic animals (in a single query)
3. Return the listings for all animals whose rarity is missing (NULL) in the database
4. Find the rarity rating of all animals that require between 1 and 2.5 hours to be fed
5. Find the minimum and maximum feeding time amongst all the animals in the zoo (in a single query)
6. Find the average feeding time for all of the rare animals

Christian Craig
DSC 450 - Hw3

7. Determine how many NULLs there are in the ACategory column using SQL

```
SELECT SUM(timetofeed)/COUNT(timetofeed) FROM Animal;
--7
SELECT COUNT(*) - COUNT(acategory)
FROM Animal;
--8
SELECT aname,acategory
FROM Animal
WHERE acategory != 'exotic' or acategory IS NULL;
```

ery Result

SQL | All Rows Fetched: 1 in 0.066 seconds

	COUNT(*)-COUNT(ACATEGORY)
1	1

8. Find all animals named 'Alpaca', 'Llama' or any other animals that are not listed as exotic

```
--8
SELECT aname,acategory
FROM Animal
WHERE acategory != 'exotic' or acategory IS NULL;
```

Query Result

SQL | All Rows Fetched: 7 in 0.036 seconds

	ANAME	ACATEGORY
1	Emperor Penguin	rare
2	Grizzly bear	common
3	Florida black bear	rare
4	Siberian tiger	rare
5	Bengal tiger	common
6	Alpaca	common
7	Llama	(null)

Christian Craig
DSC 450 - Hw3

```sql
-- Drop all the tables to clean up
DROP TABLE Animal;

-- ACategory: Animal category 'common', 'rare', 'exotic'.  May be NULL
-- TimeToFeed: Time it takes to feed the animal (hours)
CREATE TABLE Animal
(
  AID       NUMBER(3, 0),
  AName      VARCHAR2(30) NOT NULL,
  ACategory VARCHAR2(18),

  TimeToFeed NUMBER(4,2),

  CONSTRAINT Animal_PK
    PRIMARY KEY(AID)
);

INSERT INTO Animal VALUES(1, 'Galapagos Penguin', 'exotic', 0.5);
INSERT INTO Animal VALUES(2, 'Emperor Penguin', 'rare', 0.75);
INSERT INTO Animal VALUES(3, 'Sri Lankan sloth bear', 'exotic', 2.5);
INSERT INTO Animal VALUES(4, 'Grizzly bear', 'common', 3.0);
INSERT INTO Animal VALUES(5, 'Giant Panda bear', 'exotic', 1.5);
INSERT INTO Animal VALUES(6, 'Florida black bear', 'rare', 1.75);
INSERT INTO Animal VALUES(7, 'Siberian tiger', 'rare', 3.25);
INSERT INTO Animal VALUES(8, 'Bengal tiger', 'common', 2.75);
INSERT INTO Animal VALUES(9, 'South China tiger', 'exotic', 2.5);
INSERT INTO Animal VALUES(10, 'Alpaca', 'common', 0.25);
INSERT INTO Animal VALUES(11, 'Llama', NULL, 3.5);

SELECT * FROM Animal;
SELECT COUNT(*) FROM Animal;

--1
SELECT * FROM Animal WHERE timetofeed<1.5;
--2
SELECT * FROM Animal WHERE acategory = 'rare' OR acategory = 'exotic';
--3
 SELECT aname
 FROM Animal
 WHERE acategory IS NULL;
--4
 SELECT acategory, timetofeed
 FROM Animal
 WHERE timetofeed > 1 AND timetofeed < 2.5;
--5
 SELECT MIN(timetofeed),MAX(timetofeed)
 FROM Animal
--6
 SELECT SUM(timetofeed)/COUNT(timetofeed) FROM Animal;
 --7
 SELECT COUNT(*) - COUNT(acategory)
 FROM Animal;
 --8
 SELECT aname,acategory
 FROM Animal
 WHERE acategory != 'exotic' or acategory IS NULL;
```

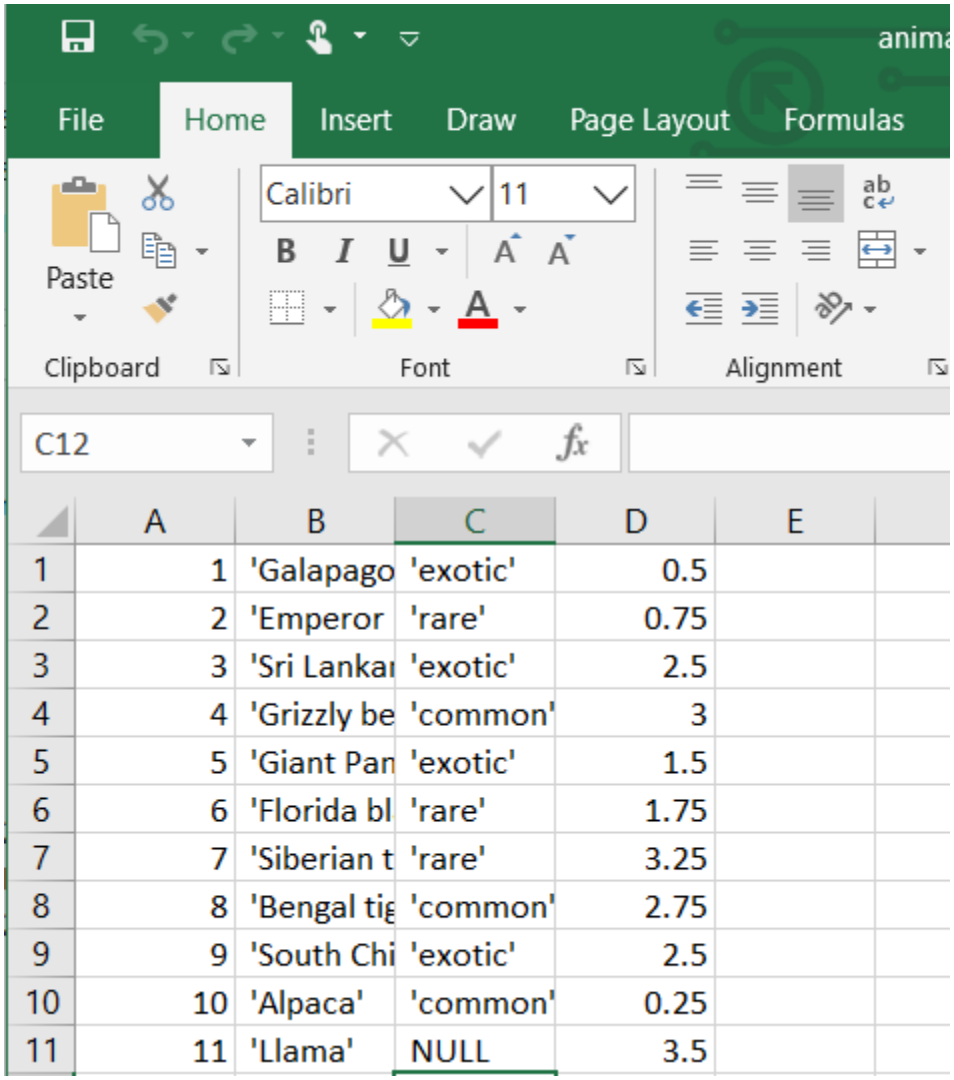
Part 2

a) Write python code that is going to export a table from SQLite database into a CSV file. You can use the attached SQLite_LoadAnimalTable.py to create and populate the table before you start. Once you have created the database using attached code, your python code solution should query the rows from the Animal table in SQLite database and write a new animal.txt file that is contains the comma-separated rows from the Animal table, e.g.,:
1, Galapagos Penguin, exotic, 0.5
2, Emperor Penguin, rare, 0.75
…
Not Entirety of code, just output of table in python and csv. Open py file to see all of code.

Christian Craig
DSC 450 - Hw3

```
>>> file = fd. read()
... allDataLine = file.split('\n')
... allDataLine = allDataLine[:-1]
...
>>> for line in allDataLine:
...     values = line.split(',')
...     print(values)
...
['1', " 'Galapagos Penguin'", " 'exotic'", '0.5']
['2', " 'Emperor Penguin'", " 'rare'", '0.75']
['3', " 'Sri Lankan sloth bear'", " 'exotic'", '2.5']
['4', " 'Grizzly bear'", " 'common'", '3']
['5', " 'Giant Panda bear'", " 'exotic'", '1.5']
['6', " 'Florida black bear'", " 'rare'", '1.75']
['7', " 'Siberian tiger'", " 'rare'", '3.25']
['8', " 'Bengal tiger'", " 'common'", '2.75']
['9', " 'South China tiger'", " 'exotic'", '2.5']
['10', " 'Alpaca'", " 'common'", '0.25']
['11', " 'Llama'", ' NULL', '3.5']
```

Christian Craig
DSC 450 - Hw3



	A	B	C	D	E
1	1	'Galapago	'exotic'	0.5	
2	2	'Emperor	'rare'	0.75	
3	3	'Sri Lankar	'exotic'	2.5	
4	4	'Grizzly be	'common'	3	
5	5	'Giant Pan	'exotic'	1.5	
6	6	'Florida bl	'rare'	1.75	
7	7	'Siberian t	'rare'	3.25	
8	8	'Bengal tig	'common'	2.75	
9	9	'South Chi	'exotic'	2.5	
10	10	'Alpaca'	'common'	0.25	
11	11	'Llama'	NULL	3.5	

Christian Craig
DSC 450 - Hw3

2a full code

```
#write lines in csv file
with fd as file:
    for ins in inserts:
        file.write(ins[ins.find("(") + 1:ins.find(")")])
        file.write('\n')


conn.commit()   # finalize inserted data
conn.close()    # close the connection


#to show values of txt / csv
'''
import os
os.getcwd() #'/Users/cjcbg/Desktop/DSC 450'
fd = open('animal.csv','r')

    #alternate read method

file = fd. read()
allDataLine = file.split('\n')
allDataLine = allDataLine[:-1]

    for line in allDataLine:
        values = line.split(',')
        print(values)
'''
```

Christian Craig
DSC 450 - Hw3

```
createtbl = """
CREATE TABLE Animal
(
  AID       NUMBER(3, 0),
  AName      VARCHAR2(30) NOT NULL,
  ACategory VARCHAR2(18),

  TimeToFeed NUMBER(4,2),

  CONSTRAINT Animal_PK
    PRIMARY KEY(AID)
);
"""

inserts = ["INSERT INTO Animal VALUES(1, 'Galapagos Penguin', 'exotic', 0.5);", "INSERT INTO Animal VALUES(2, 'Emperor Penguin', 'rare', 0.75);", "INSERT INTO

import sqlite3

conn = sqlite3.connect('dsc450.db') # open the connection
cursor = conn.cursor()


cursor.execute(createtbl)   # create the Animal table
#^don't think need the above code as animal table already created

import csv
import os

os.getcwd()
#os.chdir('/Users/cjcbg/Desktop/DSC 450')
fd = open('animal.csv', 'w')
animal_path = os.getcwd()+"\\animal_data.csv"
```

b) Write python code that is going to load the comma-separated animal.txt file you have created in part-a into the Animal table in SQLite database. Your code must read the animal.txt file and use executemany() to load the data in python (i.e., your solution cannot use the sample code from part 2-a to load the data). At the end of your code, you should verify how many rows were loaded by printing the output of
SELECT COUNT(*) FROM Animal;

```
<sqlite3.Cursor object at 0x043BEE20>
>>> count = cursor.execute("SELECT COUNT(*)FROM Animal;")
...
>>> countAnimal = count.fetchall()
...
>>> print(countAnimal)
...
[(11,)]
```

Christian Craig
DSC 450 - Hw3

2b full code

```
import sqlite3
conn = sqlite3.connect('dsc450.db') # open the connection
cursor = conn.cursor()

result = cursor.execute("select* from animal")

animalData=result.fetchall()



#load into database
import os
os.getcwd() #'/Users/cjcbg/Desktop/DSC 450'
fd = open('animal.csv','r')

    #alternate read method

file = fd. read()
allDataLine = file.split('\n')

#Send to a list
def tbl():
    lst = []
    for line in allDataLine:
        values = line.split(', ')
        lst.append(values)
    return lst

cursor.execute("DELETE FROM Animal") #Delete contents in table so no repeat ID's
tbl()[:-1] #remove blank space
cursor.executemany("Insert INTO Animal VALUES(?,?,?,?);",tbl()[:-1])
count = cursor.execute("SELECT COUNT(*)FROM Animal;")
countAnimal = count.fetchall()
```