

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 2.5 hours to feed

Name: Syed Noor Razi Ali

Student Id: 2070326

```
select * from ANIMAL where TimeToFeed < 2.5;
```

Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0.023 seconds

AID	ANAME	ACATEGORY	TIMETOFEED
1	1 Galapagos Penguin	exotic	0.5
2	2 Emperor Penguin	rare	0.75
3	5 Giant Panda bear	exotic	1.5
4	6 Florida black bear	rare	1.75
5	10 Alpaca	common	0.25

2. Find both the rare and exotic animals (in a single query, not two different queries)

```
select * from ANIMAL where ACategory = 'rare' OR ACategory = 'exotic';
```

```
select * from ANIMAL where ACategory IS NULL;
```

```
select * from ANIMAL where ACategory = 'rare' and TimeToFeed > 1.5 and TimeToFeed < 2.5 ;
```

```
select min(TimeToFeed), max(TimeToFeed) from ANIMAL;
```

```
select avg(TimeToFeed), ACategory from ANIMAL where ACategory = 'exotic' group by ACategory;
```

```
select * from ANIMAL where AName = 'Alpaca' OR AName = 'Llama' OR NOT ACategory = 'exotic';
```

Script Output x Query Result x

SQL | All Rows Fetched: 7 in 0.065 seconds

AID	ANAME	ACATEGORY	TIMETOFEED
1	1 Galapagos Penguin	exotic	0.5
2	2 Emperor Penguin	rare	0.75
3	3 Sri Lankan sloth bear	exotic	2.5
4	5 Giant Panda bear	exotic	1.5
5	6 Florida black bear	rare	1.75
6	7 Siberian tiger	rare	3.25
7	9 South China tiger	exotic	2.5

3. Return the listings for all animals whose rarity is missing (NULL) in the database

Name: Syed Noor Razi Ali

Student Id: 2070326

```
select * from ANIMAL where ACategory IS NULL;

select * from ANIMAL where ACategory = 'rare' and TimeToFeed > 1.5 and TimeToFeed < 2.5 ;

select min(TimeToFeed), max(TimeToFeed) from ANIMAL;

select avg(TimeToFeed), ACategory from ANIMAL where ACategory = 'exotic' group by ACategory;
```

Script Output x Query Result x

SQL | All Rows Fetched: 1 in 0.069 seconds

AID	ANAME	ACATEGORY	TIMETOFEED
1	11 Llama	(null)	3.5

4. Find the rarity rating of all animals that require between 1.5 and 2.5 hours to be fed

```
select * from ANIMAL where ACategory = 'rare' and TimeToFeed > 1.5 and TimeToFeed < 2.5 ;
```

Script Output x Query Result x

SQL | All Rows Fetched: 1 in 0.021 seconds

AID	ANAME	ACATEGORY	TIMETOFEED
1	6 Florida black bear	rare	1.75

5. Find the minimum and maximum feeding time amongst all the animals in the zoo (in a single SQL query, not two different queries)

Name: Syed Noor Razi Ali

Student Id: 2070326

```
select min(TimeToFeed), max(TimeToFeed) from ANIMAL;
```

Script Output x		Query Result x	
SQL All Rows Fetched: 1 in 0.024 seconds			
	MIN(TIMETOFEED)	MAX(TIMETOFEED)	
1	0.25	3.5	

6. Find the average feeding time for all of the exotic animals

```
select avg(TimeToFeed), ACategory from ANIMAL where ACategory = 'exotic' group by ACategory;
```

Script Output x		Query Result x	
SQL All Rows Fetched: 1 in 0.021 seconds			
	AVG(TIMETOFEED)	ACATEGORY	
1	1.75	exotic	

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

Name: Syed Noor Razi Ali
Student Id: 2070326

<pre>select count(*) from ANIMAL where ACategory IS NULL ;</pre>	
Script Output x Query Result x	
SQL All Rows Fetched: 1 in 0.057 seconds	
COUNT(*)	
1	1

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

```

select min(TimeToFeed), max(TimeToFeed) from ANIMAL;

select avg(TimeToFeed), ACategory from ANIMAL where ACategory = 'exotic' group by ACategory;

select * from ANIMAL where AName = 'Alpaca' OR AName = 'Llama' OR NOT ACategory = 'exotic';

```

Script Output x

Query Result x

AID	ANAME	ACATEGORY	TIMETOFEED
1	2 Emperor Penguin	rare	0.75
2	4 Grizzly bear	common	3
3	6 Florida black bear	rare	1.75
4	7 Siberian tiger	rare	3.25
5	8 Bengal tiger	common	2.75
6	10 Alpaca	common	0.25
7	11 Llama	(null)	3.5

Part 2

- a) Write python code that is going to export a table from a 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 the data into a new animal.txt file that contains the comma-separated rows from the Animal table, e.g.,:

Name: Syed Noor Razi Ali

Student Id: 2070326

1, Galapagos Penguin, exotic, 0.5

2, Emperor Penguin, rare, 0.75

```
In [1]: import sqlite3
```

```
In [40]: conn = sqlite3.connect('dsc450.db') # open the connection
         cursor = conn.cursor()
```

```
In [41]: conn.execute('DROP TABLE ANIMAL;')
```

```
Out[41]: <sqlite3.Cursor at 0x21888be1b90>
```

```
In [42]: createtable = """
         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)
         );
         """
```

```
In [43]: inserts = ["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);"]
```

```
In [44]: cursor.execute(createtable)
```

```
Out[44]: <sqlite3.Cursor at 0x21888be1650>
```

```
In [45]: for ins in inserts: #insert the rows
         cursor.execute(ins)
```

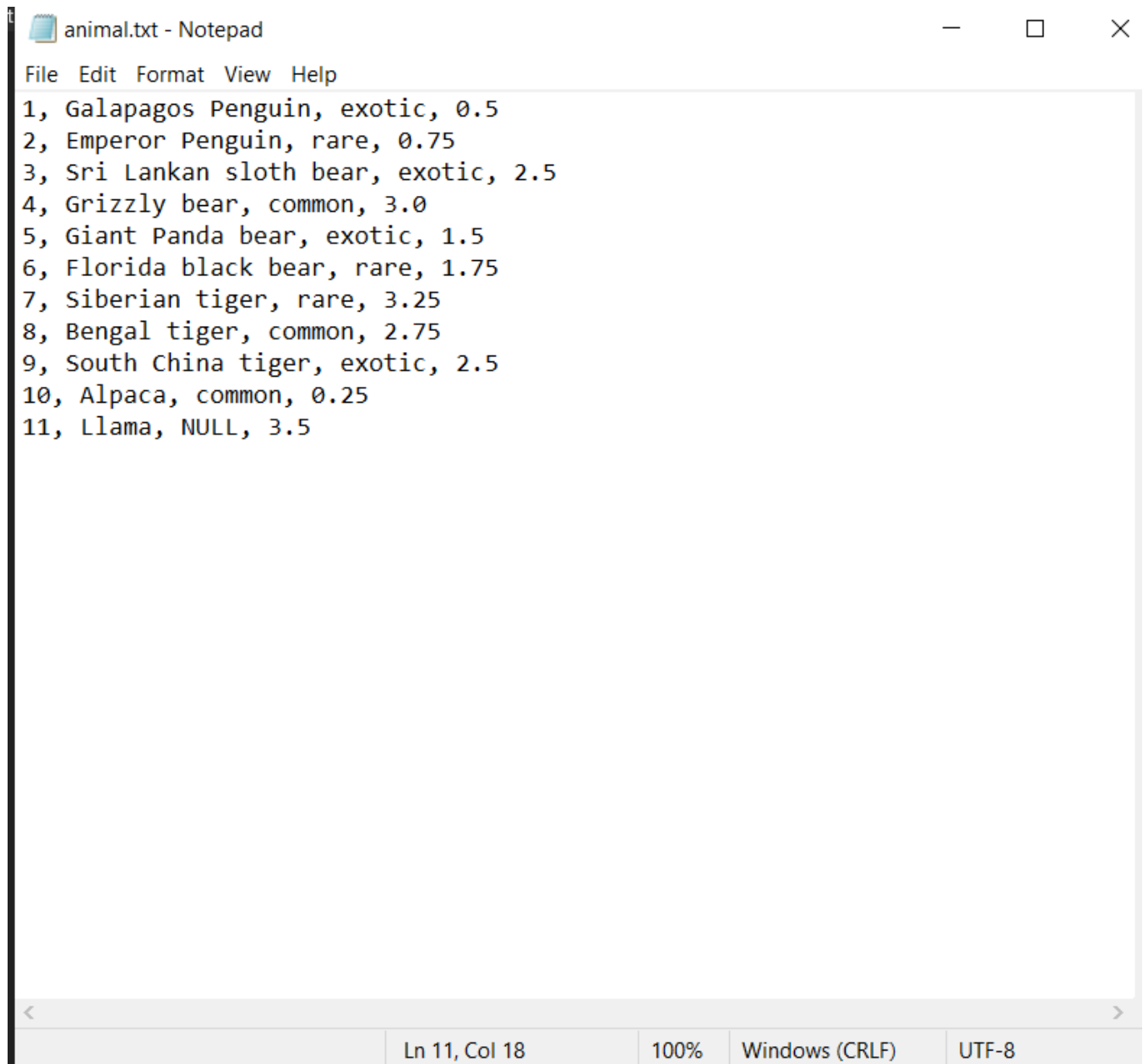
```
In [46]: conn.commit() # finalize inserted data
         conn.close() # close the connection
```

```
In [49]: def queryToTxt(query):
         """Creates txt file with results (in tuple form) of SQL query"""
         conn = sqlite3.connect('dsc450.db')
         cursor = conn.cursor()
         result = cursor.execute(query)
         resultTxt = str(result.fetchall())
         conn.commit()
         conn.close()
         resultTxt = resultTxt.replace(' ', ('', ''))
         # print(resultTxt)
         resultVals = resultTxt.split(',')
         f = open('C:/Users/razia/OneDrive/Desktop/animal.txt', 'w')
         for i in resultVals:
             if i[0] == '[':
                 i = i[1:]
                 f.write(i + '\n')
             elif i[-1] == ']':
                 i = i[:-1]
                 f.write(i + '\n')
             else:
                 f.write(i + '\n')
         f.close()
         return
```

...

Name: Syed Noor Razi Ali

Student Id: 2070326



```
File Edit Format View Help
1, Galapagos Penguin, exotic, 0.5
2, Emperor Penguin, rare, 0.75
3, Sri Lankan sloth bear, exotic, 2.5
4, Grizzly bear, common, 3.0
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
```

Ln 11, Col 18 100% Windows (CRLF) UTF-8

- 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 has to be different from the sample code from part 2-a to load the data). You can either drop the table from 2-a or create a different database for this part of the assignment.
- At the end of your code, you should verify how many rows were loaded by printing the output of `SELECT COUNT(*) FROM Animal;`

Name: Syed Noor Razi Ali

Student Id: 2070326

In [73]: `import sqlite3`

In [74]: `def ex():
 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)
);
 """`

In [75]: `conn = sqlite3.connect('dsc450.db') # open the connection
cursor = conn.cursor()`

In [76]: `conn.execute('DROP TABLE ANIMAL;')`

Out[76]: `<sqlite3.Cursor at 0x2188ae2e30>`

In [77]: `cursor.execute(createtbl)`

Out[77]: `<sqlite3.Cursor at 0x2188b4bc00>`

In [80]: `insert = "INSERT INTO Animal VALUES ('%d', '%s', '%s', '%d');"
with open(r'C:\Users\razia\OneDrive\Desktop\animal.txt', 'r') as inFile:
 allData = inFile.readlines()
 for row in allData:
 value = row.strip().split(',')
 cursor.execute(insert % (int(value[0]), value[1], value[2], float(value[3])))`

In [81]: `res = cursor.execute('SELECT COUNT(*) FROM Animal')`

In [82]: `inFile.close()
conn.commit()`

In [83]: `ans = res.fetchone()[0]`

In [86]: `print('There are total %s rows in table Animal.' %(ans))`

There are total 11 rows in table Animal.

In [85]: `conn.close()`

In [72]: `ex()`