Filtering and Ordering - Lab Assignment

Introduction

In this lab, you will write more `SELECT` statements to solidify your ability to query a SQL database. You will also write more specific queries using the tools you learned in the previous lesson.

Objectives

You will be able to:

- Write SQL queries to filter and order results
- Order the results of your queries by using `ORDER BY` (`ASC` & `DESC`)
- Limit the number of records returned by a query using `LIMIT`
- Filter results using `BETWEEN` and `IS NULL`

Famous Dogs

Here's a database full of famous dogs! The `dogs` table is populated with the following data:

name	age	gender	breed	temperament	hungry
Snoopy	3	M	beagle	friendly	1
McGruff	10	M	bloodhound	aware	0
Scooby	6	M	great dane	hungry	1
Little Ann	5	F	coonhound	loyal	0
Pickles	13	F	black lab	mischievous	1
Clifford	4	M	big red	smiley	1
Lassie	7	F	collie	loving	1
Snowy	8	F	fox terrier	adventurous	0
NULL	4	M	golden retriever	playful	1

Connecting to the Database

First, import sqlite3 and establish a connection to the database **dogs.db**. Then, create a cursor object so that you can pass SQL queries to the database.

In [1]:
```
import sqlite3
!DIR
con = sqlite3.connect('data/dogs.db')
cursor = con.cursor()
```

```
 Volume in drive C is Windows
 Volume Serial Number is AE06-1B20

 Directory of C:\Users\nzcoo\OneDrive\Desktop\DS\DS311-Technologies-in-Data-Ana
lytic\Week_4_SQL_Queries\Lab_Assignment

02/27/2022  01:07 AM    <DIR>          .
02/27/2022  01:09 AM    <DIR>          ..
02/23/2022  05:36 AM    <DIR>          .ipynb_checkpoints
02/23/2022  05:36 AM    <DIR>          data
02/25/2022  04:53 AM                 0 data.sqlite
02/23/2022  05:36 AM    <DIR>          img
02/26/2022  12:21 AM            24,170 SQL_Lab_Assignment_1.ipynb
02/27/2022  01:07 AM             7,042 SQL_Lab_Assignment_2.ipynb
02/23/2022  05:36 AM             6,633 SQL_Lab_Assignment_3_Optional.ipynb
02/23/2022  05:36 AM             6,175 SQL_Lab_Exercise.ipynb
               5 File(s)         44,020 bytes
               5 Dir(s)  84,589,862,912 bytes free
```

Queries

Display the outputs for each of the following query descriptions.

Select the name and breed for all female dogs

In [23]:
```
dogs_query = \
"""
SELECT*
FROM dogs
"""
cursor.execute(dogs_query)
cursor.fetchall()
```

Out[23]:
```
[(1, 'Snoopy', 3, 'M', 'beagle', 'friendly', 1),
 (2, 'McGruff', 10, 'M', 'bloodhound', 'aware', 0),
 (3, 'Scooby', 6, 'M', 'great dane', 'hungry', 1),
 (4, 'Little Ann', 5, 'F', 'coonhound', 'loyal', 0),
 (5, 'Pickles', 13, 'F', 'black lab', 'mischievous', 1),
 (6, 'Clifford', 4, 'M', 'big red', 'smiley', 1),
 (7, 'Lassie', 7, 'F', 'collie', 'loving', 1),
 (8, 'Snowy', 8, 'F', 'fox terrier', 'adventurous', 0),
 (9, None, 4, 'M', 'golden retriever', 'playful', 1)]
```

In [11]:
```
cursor.description
```

Out[11]:
```
(('id', None, None, None, None, None, None),
 ('name', None, None, None, None, None, None),
 ('age', None, None, None, None, None, None),
 ('gender', None, None, None, None, None, None),
 ('breed', None, None, None, None, None, None),
 ('temperament', None, None, None, None, None, None),
 ('hungry', None, None, None, None, None, None))
```

In [72]:
```
dogs_query = \
"""
SELECT name, breed
FROM dogs
WHERE gender = 'F'

"""
cursor.execute(dogs_query)
cursor.fetchall()
```

Out[72]:
```
[('Little Ann', 'coonhound'),
 ('Pickles', 'black lab'),
 ('Lassie', 'collie'),
 ('Snowy', 'fox terrier')]
```

Select the names of all dogs listed in alphabetical order. Notice that SQL lists the nameless dog first.

In [53]:
```
dogs_query = \
"""
SELECT name
FROM dogs
ORDER BY name ASC

"""
cursor.execute(dogs_query)
cursor.fetchall()
```

Out[53]:
```
[(None,),
 ('Clifford',),
 ('Lassie',),
 ('Little Ann',),
 ('McGruff',),
 ('Pickles',),
 ('Scooby',),
 ('Snoopy',),
 ('Snowy',)]
```

Select any dog that doesn't have a name

In [63]:
```
dogs_query = \
"""
SELECT name, age, gender, breed, temperament
FROM dogs
WHERE name IS NULL

"""
cursor.execute(dogs_query)
cursor.fetchall()
```

Out[63]:
```
[(None, 4, 'M', 'golden retriever', 'playful')]
```

Select the name and breed of only the hungry dogs and list them from youngest to oldest

In [71]:
```
dogs_query = \
"""
SELECT name, breed
FROM dogs
WHERE hungry is 1
ORDER BY age ASC

"""
cursor.execute(dogs_query)
cursor.fetchall()
```

Out[71]:
```
[('Snoopy', 'beagle'),
 ('Clifford', 'big red'),
 (None, 'golden retriever'),
 ('Scooby', 'great dane'),
 ('Lassie', 'collie'),
 ('Pickles', 'black lab')]
```

Select the oldest dog's name, age, and temperament

In [67]:
```
dogs_query = \
"""
SELECT name, age, temperament
FROM dogs
WHERE age = (SELECT MAX(age) FROM dogs)

"""
cursor.execute(dogs_query)
cursor.fetchall()
```

Out[67]:
```
[('Pickles', 13, 'mischievous')]
```

Select the three youngest dogs

In [68]:
```
dogs_query = \
"""
SELECT age, name
FROM dogs
ORDER BY age ASC
LIMIT 3

"""
cursor.execute(dogs_query)
cursor.fetchall()
```

Out[68]:
```
[(3, 'Snoopy'), (4, 'Clifford'), (4, None)]
```

Select the name and breed of the dogs who are between five and ten years old, ordered from oldest to youngest

In [70]:
```
dogs_query = \
"""
SELECT name, breed
FROM dogs
WHERE age BETWEEN 5 AND 10

"""
cursor.execute(dogs_query)
cursor.fetchall()
```

Out[70]:
```
[('McGruff', 'bloodhound'),
 ('Scooby', 'great dane'),
 ('Little Ann', 'coonhound'),
 ('Lassie', 'collie'),
 ('Snowy', 'fox terrier')]
```

Select the name, age, and hungry columns for hungry dogs between the ages of two and seven. This query should also list these dogs in alphabetical order.

In [ ]:
```

```

In [73]:
```
dogs_query = \
"""
SELECT name, age, hungry
FROM dogs
WHERE age BETWEEN 2 AND 7
ORDER BY name ASC
"""
cursor.execute(dogs_query)
cursor.fetchall()
```

Out[73]:
```
[(None, 4, 1),
 ('Clifford', 4, 1),
 ('Lassie', 7, 1),
 ('Little Ann', 5, 0),
 ('Scooby', 6, 1),
 ('Snoopy', 3, 1)]
```

Summary

Great work! In this lab you practiced writing more complex SQL statements to not only query specific information but also define the quantity and order of your results.