Filtering, Ordering, and Limiting Data with SQL - Lab

Introduction

In this lab, you will practice writing SQL `SELECT` queries that limit results based on conditions, using `WHERE`, `ORDER BY`, and `LIMIT`.

Objectives

You will practice the following:

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

The Data

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

In the cell below, import `pandas` and `sqlite3`. Then establish a connection to the database `dogs.db`.

Look at all of the data in the table by selecting all columns from the `dogs` table with `pd.read_sql`.

```
# Your code here; imports, create a connection, select all
# first, import pandas
import pandas as pd
# import sqlite3
import sqlite3
# creating a connection
conn = sqlite3.connect('dogs.db')
# print feedback
print('Successfully imported the `dogs` database!')

Successfully imported the `dogs` database!
```

Queries

Display the outputs for each of the following query descriptions.

Select the `name` and `breed` for all `female` dogs

```
# brian-answer # general table
pd.read_sql('''
    SELECT name, breed FROM dogs
    WHERE gender = 'F';
''', conn)

         name        breed
0  Little Ann    coonhound
1     Pickles    black lab
2      Lassie       collie
3       Snowy  fox terrier
```

Select the `number of dogs` that do `not` have a name

```
# brian-answer
pd.read_sql('''
    SELECT COUNT(*) AS number_of_dogs_without_name FROM dogs
    WHERE name IS NULL;
''', conn)

   number_of_dogs_without_name
0                            1
```

Select the names of all dogs that contain the double letters `ff` or `oo`

```
# brian-answer
pd.read_sql('''
    SELECT name FROM dogs
    WHERE name LIKE "%FF%" OR name LIKE '%oo%';
''', conn)
```

```
       name
0    Snoopy
1   McGruff
2    Scooby
3  Clifford
```

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

```
# brian-added
pd.read_sql('''
    SELECT name FROM dogs
    ORDER BY name ASC;
''', conn)
```

```
         name
0        None
1    Clifford
2      Lassie
3  Little Ann
4     McGruff
5     Pickles
6      Scooby
7      Snoopy
8       Snowy
```

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

```
# brian-added
pd.read_sql('''
    SELECT name, breed FROM dogs
    WHERE hungry IS 1
    ORDER BY age ASC;
''', conn)
```

```
       name             breed
0    Snoopy            beagle
1  Clifford           big red
2      None  golden retriever
3    Scooby        great dane
4    Lassie            collie
5   Pickles         black lab
```

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

```
pd.read_sql('''
    SELECT * FROM dogs
''', conn)

   id        name  age gender             breed  temperament  hungry
0   1      Snoopy    3      M            beagle     friendly       1
1   2     McGruff   10      M        bloodhound        aware       0
2   3      Scooby    6      M        great dane       hungry       1
3   4  Little Ann    5      F         coonhound        loyal       0
4   5     Pickles   13      F         black lab  mischievous       1
5   6    Clifford    4      M           big red       smiley       1
6   7      Lassie    7      F            collie       loving       1
7   8       Snowy    8      F       fox terrier  adventurous       0
8   9        None    4      M  golden retriever      playful       1

# brian-answer
pd.read_sql('''
    SELECT name, age, temperament FROM dogs
    ORDER BY age DESC
    LIMIT 1;
''', conn)

      name  age  temperament
0  Pickles   13  mischievous
```

Select the `name` and `age` of the `three youngest dogs`

```
# brian-answer
pd.read_sql('''
    SELECT name, age FROM dogs
    ORDER BY age DESC
    LIMIT 3;
''', conn)

      name  age
0  Pickles   13
1  McGruff   10
2    Snowy    8
```

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

```
# brian-answer
pd.read_sql('''
    SELECT name, breed, age FROM dogs
    WHERE age BETWEEN 5 AND 10
    ORDER BY age DESC;
''', conn)
```

```
          name        breed  age
0      McGruff   bloodhound   10
1        Snowy  fox terrier    8
2       Lassie       collie    7
3       Scooby   great dane    6
4   Little Ann     coonhound    5
```

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`.

```
# Your code here
pd.read_sql('''
    SELECT name, age, hungry FROM dogs
    WHERE hungry IS 1
    AND
    age BETWEEN 2 AND 7
    ORDER BY name ASC;
''', conn)
```

```
       name  age  hungry
0      None    4       1
1  Clifford    4       1
2    Lassie    7       1
3    Scooby    6       1
4    Snoopy    3       1
```

Close the Database Connection

```
# closing the connection
conn.close()
```

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.