learn-co-curriculum / dsc-filtering-and-ordering-lab Public

View license

0 stars 202 forks

Star Watch

Code Issues Pull requests Actions Projects Security Insights

solution

This branch is 7 commits ahead, 8 commits behind master. Contribute

hoffm386 Merge pull request #1 from hoffm386/curriculum ... on Jan 21 10

View code

README.md

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`.

```
# Relevant imports
import pandas as pd
import sqlite3

# Create a connection
conn = sqlite3.connect('dogs.db')

# Select all
pd.read_sql("SELECT * FROM dogs;", conn)
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	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

Queries

Display the outputs for each of the following query descriptions.

Select the name and breed for all female dogs

▶ **Click for hint:**

```
pd.read_sql("""
SELECT name, breed
  FROM dogs
```

```
 WHERE gender = 'F';
""", conn)
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	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

▶ **Click for hint:**

```
pd.read_sql("""
SELECT COUNT(*) AS num_dogs
  FROM dogs
 WHERE name IS NULL;
""", conn)
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	num_dogs
0	1

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

▶ **Click for hint:**

```
pd.read_sql("""
SELECT name
  FROM dogs
 WHERE name LIKE '%ff%'
    OR name LIKE '%oo%';
""", conn)
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	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.

▶ **Click for hint:**

```
pd.read_sql("""
SELECT name
  FROM dogs
 ORDER BY name;
""", conn)
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	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

```
pd.read_sql("""
SELECT name, breed
```

```
  FROM dogs
 WHERE hungry = 1
 ORDER BY age;
""", conn)
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	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

▶ **Click for hint:**

```
pd.read_sql("""
SELECT name, age, temperament
  FROM dogs
 ORDER BY age DESC
 LIMIT 1;
""", conn)
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	name	age	temperament
0	Pickles	13	mischievous

Select the name and age of the three youngest dogs

```
pd.read_sql("""
SELECT name, age
  FROM dogs
 ORDER BY age
 LIMIT 3;
""", conn)
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	name	age
0	Snoopy	3
1	Clifford	4
2	None	4

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

▶ **Click for hint:**

```
pd.read_sql("""
SELECT name, breed
  FROM dogs
 WHERE age BETWEEN 5 AND 10
 ORDER BY age DESC;
""", conn)
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	name	breed
0	McGruff	bloodhound
1	Snowy	fox terrier
2	Lassie	collie
3	Scooby	great dane
4	Little Ann	coonhound

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.

```
pd.read_sql("""
SELECT name, age, hungry
  FROM dogs
 WHERE hungry = 1
   AND age BETWEEN 2 AND 7
 ORDER BY name;
""", conn)
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	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

```
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.

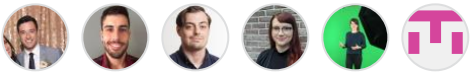
Releases

No releases published

Packages

No packages published

Contributors 6



Languages

- **Jupyter Notebook** 64.8%
- **Python** 35.2%