AbhijitMandal_DSC540_Week11-12Ex

May 30, 2021

0.0.1 DSC 540 Week 11-12

Abhijit Mandal

0.0.2 Activity 11: Retrieving Data Correctly From Databases

- Connect to petsDB and check whether the connection has been successful.
- Find the different age groups in the persons database.
- Find the age group that has the maximum number of people.
- Find the people who do not have a last name.
- Find out how many people have more than one pet.
- Find out how many pets have received treatment.
- Find out how many pets have received treatment and the type of pet is known.
- Find out how many pets are from the city called east port.
- Find out how many pets are from the city called east port and who received a treatment.

0.0.3 Load the necessary libraries.

[7]:
```python
import sqlite3
```

[8]:
```python
#Connecting to Pets Db
conn = sqlite3.connect("petsdb")
```

[9]:
```python
#  function to make sure the connection is successful
def is_opened(conn):
    try:
        conn.execute("SELECT * FROM persons LIMIT 1")
        return True
    except sqlite3.ProgrammingError as e:
        print("Connection closed {}".format(e))
        return False
```

[10]:
```python
print(is_opened(conn))
```

```
True
```

[11]:
```python
conn.close()
```

[12]:
```python
print(is_opened(conn))
```

```
Connection closed Cannot operate on a closed database.
False
```

[13]:

```
#Find the different age groups in the persons database.
conn = sqlite3.connect("petsdb")
c = conn.cursor()
for ppl, age in c.execute("SELECT count(*), age FROM persons GROUP BY age"):
    print("We have {} people aged {}".format(ppl, age))
```

```
We have 2 people aged 5
We have 1 people aged 6
We have 1 people aged 7
We have 3 people aged 8
We have 1 people aged 9
We have 2 people aged 11
We have 3 people aged 12
We have 1 people aged 13
We have 4 people aged 14
We have 2 people aged 16
We have 2 people aged 17
We have 3 people aged 18
We have 1 people aged 19
We have 3 people aged 22
We have 2 people aged 23
We have 3 people aged 24
We have 2 people aged 25
We have 1 people aged 27
We have 1 people aged 30
We have 3 people aged 31
We have 1 people aged 32
We have 1 people aged 33
We have 2 people aged 34
We have 3 people aged 35
We have 3 people aged 36
We have 1 people aged 37
We have 2 people aged 39
We have 1 people aged 40
We have 1 people aged 42
We have 2 people aged 44
We have 2 people aged 48
We have 1 people aged 49
We have 1 people aged 50
We have 2 people aged 51
We have 2 people aged 52
We have 2 people aged 53
We have 2 people aged 54
We have 1 people aged 58
We have 1 people aged 59
We have 1 people aged 60
We have 1 people aged 61
We have 2 people aged 62
We have 1 people aged 63
We have 2 people aged 65
We have 2 people aged 66
We have 1 people aged 67
We have 3 people aged 68
We have 1 people aged 69
We have 1 people aged 70
We have 4 people aged 71
We have 1 people aged 72
We have 5 people aged 73
We have 3 people aged 74
```

[14]:

```python
# Find the age group that has the maximum number of people.
for ppl, age in c.execute("SELECT count(*), age FROM persons GROUP BY age ORDER BY count(*) DESC"):
    print("Highest number of people {} came from {} age group".format(ppl, age))
    break
```

```
Highest number of people 5 came from 73 age group
```

[15]:

```python
#Find the people who do not have a last name.
res = c.execute("SELECT count(*) FROM persons WHERE last_name IS null")
for row in res:
    print(row)
```

```
(60,)
```

[16]:

```python
# Find out how many people have more than one pet.
res = c.execute("SELECT count(*) FROM (SELECT count(owner_id) FROM pets GROUP BY owner_id HAVING count(owner_id) >1)")
for row in res:
    print("{} People has more than one pets".format(row[0]))
```

```
43 People has more than one pets
```

[17]:

```python
#Find out how many pets have received treatment.
res = c.execute("SELECT count(*) FROM pets WHERE treatment_done=1")
for row in res:
    print(row)
```

```
(36,)
```

[18]:

```python
# Find out how many pets have received treatment and the type of pet is known.
res = c.execute("SELECT count(*) FROM pets WHERE treatment_done=1 AND pet_type IS NOT null")
```

```
for row in res:
    print(row)
```

```
(16,)
```

[19]:

```
# Find out how many pets are from the city called east port.
res = c.execute("SELECT count(*) FROM pets JOIN persons ON pets.owner_id =␣
 ↪persons.id WHERE persons.city='east port'")
for row in res:
    print(row)
```

```
(49,)
```

[20]:

```
# Find out how many pets are from the city called east port and who received a␣
 ↪treatment.
res = c.execute("SELECT count(*) FROM pets JOIN persons ON pets.owner_id =␣
 ↪persons.id WHERE persons.city='east port' AND pets.treatment_done=1")
for row in res:
    print(row)
```

```
(11,)
```