

DSC 540 Brandon Mather Activity 11:

```
In [2]: import sqlite3
```

```
In [3]: #Connect to petsDB and check if the connection has been successful  
conn = sqlite3.connect("petsdb")
```

```
In [4]: 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
```

```
In [5]: print(is_opened(conn))
```

True

```
In [6]: conn.close()
```

```
In [7]: print(is_opened(conn))
```

Connection closed Cannot operate on a closed database.
False

```
In [8]: #Find the different age groups in the persons database  
conn = sqlite3.connect("petsdb")
```

```
In [9]: c = conn.cursor()
```

```
In [10]: 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

```
In [11]: #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(*)"):
    print("Highest number of people {} came from {} age group".format(ppl, age))
    break
```

Highest number of people 5 came from 73 age group

```
In [12]: #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,)
```

```
In [17]: #Find out how many people have more then one pet
res = c.execute("SELECT count(*) FROM (SELECT count(owner_id) FROM pets GROUP BY owner_id) WHERE count > 1")
for row in res:
    print("{} People has more than one pets".format(row[0]))

43 People has more than one pets
```

```
In [18]: #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,)
```

```
In [19]: #Find out how many pets have recieved 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,)
```

```
In [20]: #Find out how many pets are from the called east port
res = c.execute("SELECT count(*) FROM pets JOIN persons ON pets.owner_id = persons.id")
for row in res:
    print(row)

(49,)
```

```
In [21]: #Find out how many pets are from the city called east port and who recieved a treatment
res = c.execute("SELECT count(*) FROM pets JOIN persons ON pets.owner_id = persons.id WHERE pets.city = 'East Port' AND pets.treatment_done = 1")
for row in res:
    print(row)

(11,)
```