Michael Perrine

Week 11-12 Assignment

DSC 540

Professor Williams

Activity 8.01

In [5]:

```
# This code imports the necessary libraries
import sqlite3
```

In [ ]:

```
# This code connects to the pets database
pets = sqlite3.connect('petsdb')
```

In [ ]:

```
# This code creates a function to open the database
# and includes a try block
def is_opened(pets):
    try:
        pets.execute('SELECT * FROM persons LIMIT 1')
        return True
    except sqlite3.ProgrammingError as e:
        print('Connection ended {}'.format(e))
        return False
    print(is_opened(pets))
```

In [ ]:

```
# This code closes the pets database
pets.close()
```

In [ ]:

```
# This code confirms the database is closed
print(is_opened(pets))
```

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

In [ ]:

```
# This code connects to the pets database and creates
# the cursor object to call data
pets= sqlite3.connect('petsdb')
p = pets.cursor()
```

1. What is the count of people belonging to different age groups in the persons table?

In [ ]:

```
# This code selects people and ages and groups them by age
for ppl, age in p.execute('SELECT count(*), age FROM persons GROUP BY age'):
    print('There are {} people aged {}'.format(ppl, age))
```

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

2. Which age group has the maximum number of people?

In [18]:

```
# This code orders people from highest to lowest
for ppl, age in p.execute('SELECT count(*), age FROM persons GROUP BY age ORDER BY
    print('The highest number of people is {} and came from the {} age group'.forma
    break
```

```
The highest number of people is 5 and came from the 73 age group
```

3. How many people do not have a last name?

In [ ]:

```
# This code counts the number of entries with no last name
l_name = p.execute('SELECT count(*) FROM persons WHERE last_name IS null')
for row in l_name:
    print(row)
```

```
(60,)
```

4. How many people have more than one pet?

In [20]:

```
# This code counts entries with multiple pets
p_count = p.execute('SELECT count(*) FROM (SELECT count(owner_id) FROM pets GROUP B
for row in p_count:
    print('{} people with multiple pets'.format(row[0]))
```

```
43 people with multiple pets
```

5. How many pets have received treatment?

In [22]:

```
# This code filters pets that received treatment
p_treat = p.execute('SELECT count(*) FROM pets WHERE treatment_done = 1')
for row in p_treat:
    print(row)
```

```
(36,)
```

6. How many pets have received treatment, and the type of pet is known?

In [24]:

```
# This code counts the number of known pets that
# received treatment
p_known = p.execute('SELECT count(*) FROM pets WHERE treatment_done =1 AND pet_type
for row in p_known:
    print(row)
```

```
(16,)
```

7. How many pets are from the city called East Port?

In [25]:

```
# This code filters and counts pets from East Port
p_city = p.execute("SELECT count(*) FROM pets JOIN persons ON pets.owner_id = perso
for row in p_city:
    print(row)
```

```
(49,)
```

8. How many pets are from the city called East Port, and who received treatment?

```
In [26]: # This code filters pets from East Port and
         # received treatment
         p_treat_city = p.execute('SELECT count(*) FROM pets JOIN persons ON pets.owner_id =
         for row in p_treat_city:
             print(row)
```

```
(11,)
```