Weeks 9&10 Exercises

Author: Jordan,Andrew

Date: 08/12/2022

In [30]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

%pwd
```

Out[30]:

```
'C:\\Users\\Andrew\\Documents\\Grad School\\DSC 540 - Data Preparation\\Assignments'
```

Activity 11, page 320

In [31]:

```
#Connect to the supplied petsDB database
import sqlite3
conn=sqlite3.connect("data/petsdb")
```

In [32]:

```
#write a function to check whether the connection has been successful
def did_it_connect(conn):
    try:
        print('Yep')
    except Exception as ex:
        print('Nope')
```

In [33]:

```
did_it_connect(conn)
```

```
Yep
```

In [34]:

```
#Find the different age groups in the persons database

#Set c
c = conn.cursor()
```

In [37]:

```
#Loop through table getting count of people by age
for people, age in c.execute("SELECT count(*), age FROM persons GROUP BY age"):
    print('Age:', age, 'count:', people)
```

```
Age: 5 count: 2
Age: 6 count: 1
Age: 7 count: 1
Age: 8 count: 3
Age: 9 count: 1
Age: 11 count: 2
Age: 12 count: 3
Age: 13 count: 1
Age: 14 count: 4
Age: 16 count: 2
Age: 17 count: 2
Age: 18 count: 3
Age: 19 count: 1
Age: 22 count: 3
Age: 23 count: 2
Age: 24 count: 3
Age: 25 count: 2
Age: 27 count: 1
Age: 30 count: 1
Age: 31 count: 3
Age: 32 count: 1
Age: 33 count: 1
Age: 34 count: 2
Age: 35 count: 3
Age: 36 count: 3
Age: 37 count: 1
Age: 39 count: 2
Age: 40 count: 1
Age: 42 count: 1
Age: 44 count: 2
Age: 48 count: 2
Age: 49 count: 1
Age: 50 count: 1
Age: 51 count: 2
Age: 52 count: 2
Age: 53 count: 2
Age: 54 count: 2
Age: 58 count: 1
Age: 59 count: 1
Age: 60 count: 1
Age: 61 count: 1
Age: 62 count: 2
Age: 63 count: 1
Age: 65 count: 2
```

```
Age: 66 count: 2
Age: 67 count: 1
Age: 68 count: 3
Age: 69 count: 1
Age: 70 count: 1
Age: 71 count: 4
Age: 72 count: 1
Age: 73 count: 5
Age: 74 count: 3
```

In [38]:

```
#Find the age group that has the maximum number of people

#Loop through table getting count by age and sorting descending
for people, age in c.execute("SELECT count(*), age FROM persons GROUP BY age ORDER BY count(*) DESC"):
    print('Age', age, 'count:', people)
    break #Break to only print top entry
```

```
Age 73 count: 5
```

In [40]:

```
#Find the people who do not have a last name

#Loop through table getting first names with null last name
for name in c.execute("SELECT first_name FROM persons WHERE last_name IS null"):
    print('First name:', name[0])
```

```
First name: Erica
First name: Jordi
First name: Chasity
First name: Gregg
First name: Cary
First name: Francisca
First name: Raleigh
First name: Maria
First name: Mariane
First name: Mona
First name: Kayla
First name: Karlie
First name: Morris
First name: Sandy
First name: Hector
First name: Hiram
First name: Tressa
First name: Berry
First name: Pearline
First name: Maynard
First name: Dorian
First name: Mylene
First name: Lafayette
First name: Tara
First name: Destiny
First name: Lesly
First name: Perry
First name: Maritza
First name: Grant
First name: Laury
First name: Name
First name: Estefania
First name: Destiney
First name: Jaquelin
First name: Alfonzo
First name: Lisandro
First name: Priscilla
First name: Elenora
First name: Rudolph
First name: Ona
First name: Rebeca
First name: Sigurd
First name: Alice
First name: Dane
First name: Judge
First name: Allene
First name: Jalen
First name: Myron
First name: Travon
First name: Shayna
First name: Myah
First name: Letha
First name: Felton
First name: London
First name: Koby
First name: Golden
First name: Anissa
First name: Sid
First name: Ernesto
First name: Josianne
```

In [41]:

```
#Find out how many people have more than one pet


#Find duplicate owner ids
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(row[0], 'people have more than one pet')
```

```
43 people have more than one pet
```

In [43]:

```
#Find out how many pets have received treatment


#Find all treatment_done with 1 value
res = c.execute("SELECT count(*) FROM pets WHERE treatment_done=1")

for row in res:
    print(row[0], 'pets have received treatment')
```

```
36 pets have received treatment
```

In [44]:

```
#Find out how many pets have received treatment and the type of pet is known


#Find all treatment_done with 1 value and pet_type not null
res = c.execute("SELECT count(*) FROM pets WHERE treatment_done=1 AND pet_type IS NOT null")

for row in res:
    print(row[0], 'pets have receieved treatment with a known type of pet')
```

```
16 pets have receieved treatment with a known type of pet
```

In [47]:

```
#Find out how many pets are from the city called east port

#Join tables and find results where city=eastport
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[0], 'pets are from East Port')
```

```
49 pets are from East Port
```

In [48]:

```
#Find out how many pets are fro mthe city called east port and who received a treatment


#Join tables and find results where city=eastport and treatment_done=1
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[0], 'treated pets are from East Port')
```

```
11 treated pets are from East Port
```