Activity 11: Retrieving Data Correctly From Databases

1. Connect to petsDB and check whether the connection has been successful.

In [1]:
```
import sqlite3
```

In [2]:
```
con = sqlite3.connect("petsdb")
```

In [3]:
```
# Function to ensure a successful connection

def is_opened(con):
    try:
        con.execute("SELECT * FROM persons LIMIT 1")
        return True
    except sqlite3.ProgrammingError as err:
        print('Connection closed : ', err)
        return False

print(is_opened(con))
```

```
True
```

In [4]:
```
con.close()
```

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

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

2. Find the different age groups in the persons database.

In [6]:
```
with sqlite3.connect('petsdb') as con: #Connect to the petsdb
    cur = con.cursor()
    rows= cur.execute("PRAGMA table_info([persons])") #Get table info
    for row in rows:
        print(row)
```

```
(0, 'id', 'INTEGER', 0, None, 0)
(1, 'first_name', 'TEXT', 0, None, 0)
(2, 'last_name', 'TEXT', 0, None, 0)
(3, 'age', 'INTEGER', 0, None, 0)
(4, 'city', 'TEXT', 0, None, 0)
(5, 'zip_code', 'INTEGER', 0, None, 0)
```

In [7]:
```
with sqlite3.connect('petsdb') as con:
    cur = con.cursor()
    rows= cur.execute("SELECT age, count(1) FROM persons GROUP BY age")
    print('AGE', '   COUNT', '\n---    ', '-----')
    for row in rows:
        print(row[0], '\t', row[1])
```

```
AGE     COUNT
---     -----
5         2
6         1
```

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

3. Find the age group that has the maximum number of people.

In [8]:
```
for age, people in cur.execute("SELECT age, count(1) FROM persons GROUP BY age ORDER BY
    print("We have {} people aged {}".format(people, age))
```

```
We have 5 people aged 73
```

4. Find the people who do not have a last name.

In [9]:
```
rows= cur.execute("select * from persons where last_name is null")
print(len(cur.fetchall()))
```

```
60
```

5. Find out how many people have more than one pet.

In [10]:
```
rows= cur.execute("select count(1) from (select * from persons where last_name is null)"
#print(len(cur.fetchall()))
for count in rows:
    print(count[0])
```

```
60
```

6. Find out how many pets have received treatment.

In [11]:
```
#Understand the table fields
rows= cur.execute("PRAGMA table_info([pets])")
for row in rows:
    print(row)
```

```
(0, 'owner_id', 'INTEGER', 0, None, 0)
(1, 'pet_name', 'TEXT', 0, None, 0)
(2, 'pet_type', 'REAL', 0, None, 0)
(3, 'treatment_done', 'INTEGER', 0, None, 0)
```

In [12]:
```
#Check the different values for treatment_done
rows= cur.execute("select treatment_done, count(1) from pets group by treatment_done")
for treatment, count in rows:
    print(treatment , ' ' , count)
```

```
0   114
1   36
```

In [13]:
```
#Get the number of pets that received treatment
rows= cur.execute("select count(1) from pets where treatment_done = 1")
for count in rows:
    print("Number of pets that received treatment = {}".format(count[0]))
```

```
Number of pets that received treatment = 36
```

7. Find out how many pets have received treatment and the type of pet is known.

In [14]:
```
#Check the different values for pet_type
rows= cur.execute("select pet_type, count(1) from pets group by pet_type")
for pet_type, count in rows:
    print(pet_type , ' ' , count)
```

```
None   82
1.0   68
```

In [15]:
```
#Get the number of pets that received treatment whose type is known
rows= cur.execute("select count(1) from pets where treatment_done = 1 and pet_type = 1")
for count in rows:
    print("There are {} pets which received treatment and the type of pet is known".form
```

```
There are 16 pets which received treatment and the type of pet is known
```

8. Find out how many pets are from the city called east port.

In [16]:
```
rows= cur.execute("select count(a.owner_id) from pets as a inner join persons as b on a.
for count in rows:
    print("There are {} pets from the city east port".format(count[0]))
```

```
There are 49 pets from the city east port
```

9. Find out how many pets are from the city called east port and who received a treatment.

In [17]:
```
rows= cur.execute("select count(a.owner_id) from pets as a inner join persons as b on a.
for count in rows:
    print("There are {} pets from the city east port that received treatment".format(cou
```

```
There are 11 pets from the city east port that received treatment
```