DSC540-T302 Data Preparation

Weeks 11 & 12: Storing Data and Final Project

Saravanan Janarthanan

Assignment

Activity 8.01: Retrieving Data Accurately from Databases

The goal of this activity is to fetch data and retrieve information from two tables, persons and pets, which are a part of the petsdb database.

Perform queries to answer the questions

- 1. What is the count of people belonging to different age groups in the persons table?
- 1. Which age group has the maximum number of people?
- 1. How many people do not have a last name?
- 1. How many people have more than one pet?
- 1. How many pets have received treatment?
- 1. How many pets have received treatment, and the type of pet is known?
- 1. How many pets are from the city called east port?
- 1. How many pets are from the city called east port, and who received treatment?

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

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

To answer the same get the count of each age from persons table and use GROUP BY age and print the same

In [2]:
```
# Obtain the connection to petsdb.  Petsdb is already loaded with table data.  using t
# open the a connection to it.
conn = sqlite3.connect("petsdb")

# create a cursor from the connection to obatin resultset
cursr = conn.cursor()
```

In [3]:
```
# construct the query to fetch groupe age count
query1 = 'SELECT count(*), age from persons GROUP BY age'

# Execute the query and print the age and count values
for count_ppl, age in cursr.execute(query1):

    print("At age ", age, " ther are ", count_ppl, " persons")
```

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

2 Which age group has the maximum number of people?

To find the max value among the age count, query the count value for each age by grouping it and the sort by count value in descending to have the high count value in the top

Print the first row only to answer the question

In [4]:
```
# construct the query to fetch groupe age count and then sort the order in descending
# to have max values in top to low values in the bottom
query2 = "SELECT age, count(*) as ppl_count FROM persons GROUP BY age  ORDER BY ppl_cc

# Execute the query and print the age and count with max values
for age, count_ppl in cursr.execute(query2):
    print("Age ", age, " has ", count_ppl, " persons which is max among the age group"
    break
```

```
Age  73  has  5  persons which is max among the age group
```

3 How many people do not have a last name?

Execute a query to find count of persons from persons table that has the column last_name column filled with null value which implies that those records have no last_name value.

In [5]:
```
# construct the query to fetch count of persons whose last name value has nulls values
query3 = "SELECT count(*) FROM persons WHERE last_name is null"

# Execute the query and print the count of persons with no last name
rows = cursr.execute(query3)

# Iterate the cursor to print rows from query result
for row in rows:
    print(row[0] , " people do no have a last name")
```

```
60  people do no have a last name
```

4 How many people have more than one pet?

From the pets table find the count of owner ids or owners those have more than one pet. This query provides a list of records grouped by ownerid with petcount. USe this result and query over to aggregate the count numbers to obtain owner count.

In [6]:
```
# construct the sub query to aggrgate the count list from inner query
query4 = "SELECT count(*) FROM (SELECT count(owner_id) as ownr_count FROM pets GROUP B

# Execute the query and print the aggregated owner count
rows = cursr.execute(query4)

for row in rows:
    print(row[0] , " people have more than one pet")
```

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

5 How many pets have received treatment?

From pets table find the count the records that has treatment_done column value with 1

In [7]:
```
# construct the query to fetch count of pets that has treatment_done value with 1
query5 = "SELECT count(*) FROM pets WHERE treatment_done = 1"

# Execute the query and print count of pets from query5
rows = cursr.execute(query5)
```

```
for row in rows:
    print(row[0] , " pets has received treatment")
```

```
36  pets has received treatment
```

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

Use an AND operator to combine both conditions , treatment_done = 1 and pet_type column value as null and query the pets table.

In [8]:
```
# construct the query to fetch pet count for those rows that has treatment_done = 1 an
query6 = "SELECT count(*) FROM pets WHERE treatment_done = 1 AND pet_type is NOT null"

# Execute the query pet count from query 6
rows = cursr.execute(query6)

for row in rows:
    print(row[0] , " pets has received treatment with their type is known")
```

```
16  pets has received treatment with their type is known
```

7 How many pets are from the city called east port?

Join two tables on owner_id to find the mergdata that has city column value with 'east port' in persons table and count the records

In [9]:
```
# construct the query to fetch number of pets whose owner's city value is 'east port'
query7 = "SELECT count(*) FROM pets JOIN persons ON pets.owner_id = persons.id WHERE p

# Execute the query and print count of pets from query 7
rows = cursr.execute(query7)

for row in rows:
    print(row[0] , " pets are from the city called east port")
```

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

8 How many pets are from the city called east port, and who received treatment?

Join two tables on owner_id to find the mergdata that has city column value with 'east port' in persons table , 'treatment_done' value 1 in pets table and count the records

In [10]:
```
# construct the query to fetch number of pets whose owner's city value is 'east port'
query8 = "SELECT count(*) FROM pets \
          JOIN persons ON pets.owner_id = persons.id \
          WHERE persons.city='east port' AND \
          pets.treatment_done = 1"

# Execute the query and print count of pets from query 8
rows = cursr.execute(query8)

for row in rows:
    print(row[0] , " pets are from the city called east port, and who received treatme
```

```
11  pets are from the city called east port, and who received treatment
```

In [11]:

```
# Close the connection
conn.close()
```