Chitramoy Mukherjee

DSC 540-T302

Week-11 and Week-12

Date : 11/09/2023

Activity - 11

Connect to the supplied petsDB, and write a function to check if the connection is done.

Find different age groups in the persons database.

Find the age group that has maximum number of people.

Find the people who do not have a last name.

Find out how many people have more than one pet

Find out how many pets have received treatement.

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

Find how many pets are from city called east port.

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

In [30]:

```python
# Load necessary libraries

import sqlite3

conn = sqlite3.connect("C:\\Users\\14024\\OneDrive\\Desktop\\MS-DSC\DSC-540\Week-12\\p

def connect_to_database(database_path):
    try:
        # Connect to the SQLite database
        connection = sqlite3.connect(database_path)
        print("Connected to the database successfully.")
        return connection
    except sqlite3.Error as e:
        print(f"Error connecting to the database: {e}")
        return None

database_path = 'C:\\Users\\14024\\OneDrive\\Desktop\\MS-DSC\\DSC-540\\Week-12\\petsdb
connection = connect_to_database(database_path)

# Check if the connection is successful
if connection:
# Close the connection when done
    connection.close()
    print("Connection closed.")
else:
    print("Connection failed.")
```

```
Connected to the database successfully.
Connection closed.
```

In [31]:

```
print(is_opened(connection))
```

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

In [45]:

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

for people, age in c.execute("SELECT count(*), age FROM persons GROUP BY age"):
    print("Count of people is {} of age {} ".format(ppl, age))
```

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

In [50]:

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

for people, age in c.execute("SELECT count(*), age FROM persons GROUP BY age ORDER BY
    print("Highest number of people {} came from {} age group".format(people, age))
```

```
Highest number of people 5 came from 73 age group
```

In [52]:

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

cnt = c.execute("SELECT count(*) FROM persons WHERE last_name IS null")
for row in cnt:
    print(row)
```

```
(60,)
```

In [56]:

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

cnt = c.execute("SELECT count(*) FROM (SELECT count(owner_id) FROM pets GROUP BY owner
for row in cnt:
    print("{} People have more than one pet".format(row[0]))
```

```
43 People have more than one pet
```

In [58]:

```
# Find out how many pets have received treatement (treatment_done should have 1)

cnt = c.execute("SELECT count(*) FROM pets WHERE treatment_done=1")
for row in cnt:
    print(row)
```

```
(36,)
```

In [60]:

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

cnt = c.execute("SELECT count(*) FROM pets WHERE treatment_done=1 AND pet_type IS NOT
for row in cnt:
    print(row)
```

```
(16,)
```

In [61]:

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

cnt = c.execute("SELECT count(*) FROM pets JOIN persons ON pets.owner_id = persons.id
for row in cnt:
    print(row)
```

```
(49,)
```

In [62]:

```
# Find out how many pets are from the city called east port and who received a treatem

cnt = c.execute("SELECT count(*) FROM pets JOIN persons ON pets.owner_id = persons.id
for row in cnt:
    print(row)
```

```
(11,)
```