**Database = dogs.db**

**Created table = license Test**

**Put Dogsload csv file into Python directory (it’s the same as the Dogs file in the Dogdb folder on my desktop)**

**Load csv file into license – 24Nov asked Ron, Jerrin, Andew for help**

**Table 2 = breedrank I think I need to do this so the GUI pulls the rank for each breed?**

**Table 3 = namerank I think I need to do this so the GUI pulls the rank for each name?**

**Homework #16, Judy Minichelli**

1Dec2017 Data 520 Introduction to Programming

**Gries 17.10 1.a-j, Chinese Zodiac**

**1a.**

>>> import sqlite3

>>> con = sqlite3.connect('census.db')

**1b.**

>>> cur = con.cursor()

>>> cur.execute('CREATE TABLE Density(Province\_Territory TEXT, Population INTEGER, Land\_Area REAL)')

<sqlite3.Cursor object at 0x031B5720>

**1c.**

>>> prov = '''

insert into Density (Province\_Territory, Population, Land\_Area) values

('Newfoundland and Labrador', 512930, 370501.69), ('Prince Edward Island', 135294, 5684.39),

('Nova Scotia', 908007, 52917.43), ('New Brunswick', 729498, 71355.67), ('Quebec', 7237479, 1357743.08),

('Ontario', 11410046, 907655.59), ('Manitoba', 1119583, 551937.87), ('Saskatchewan', 978933, 586561.35),

('Alberta', 2974807, 639987.12), ('British Columbia', 3907738, 926492.48), ('Yukon Territory', 28674, 474706.97),

('Northwest Territories', 37360, 1141108.37), ('Nunavut', 26745, 1925460.18)'''

>>> cur.execute(prov)

<sqlite3.Cursor object at 0x031B5720>

>>> con.commit()

**1d.**

>>> cur.execute('SELECT \* FROM Density')

<sqlite3.Cursor object at 0x031B5720>

>>> cur.fetchall()

[('Newfoundland and Labrador', 512930, 370501.69), ('Prince Edward Island', 135294, 5684.39), ('Nova Scotia', 908007, 52917.43), ('New Brunswick', 729498, 71355.67), ('Quebec', 7237479, 1357743.08), ('Ontario', 11410046, 907655.59), ('Manitoba', 1119583, 551937.87), ('Saskatchewan', 978933, 586561.35), ('Alberta', 2974807, 639987.12), ('British Columbia', 3907738, 926492.48), ('Yukon Territory', 28674, 474706.97), ('Northwest Territories', 37360, 1141108.37), ('Nunavut', 26745, 1925460.18)]

**1e.**

>>> cur.execute('SELECT Population FROM Density')

<sqlite3.Cursor object at 0x031B5720>

>>> cur.fetchall()

[(512930,), (135294,), (908007,), (729498,), (7237479,), (11410046,), (1119583,), (978933,), (2974807,), (3907738,), (28674,), (37360,), (26745,)]

**1f.**

>>> cur.execute('''SELECT Province\_Territory FROM Density WHERE Population < 1000000''')

<sqlite3.Cursor object at 0x031B5720>

>>> cur.fetchall()

[('Newfoundland and Labrador',), ('Prince Edward Island',), ('Nova Scotia',), ('New Brunswick',), ('Saskatchewan',), ('Yukon Territory',), ('Northwest Territories',), ('Nunavut',)]

**1g.**

>>> cur.execute('''SELECT Province\_Territory FROM Density WHERE Population < 1000000 OR Population > 5000000''')

<sqlite3.Cursor object at 0x031B5720>

>>> cur.fetchall()

[('Newfoundland and Labrador',), ('Prince Edward Island',), ('Nova Scotia',), ('New Brunswick',), ('Quebec',), ('Ontario',), ('Saskatchewan',), ('Yukon Territory',), ('Northwest Territories',), ('Nunavut',)]

**1h.**

>>> cur.execute('''SELECT Province\_Territory FROM Density WHERE NOT (Population < 1000000 OR Population > 5000000)''')

<sqlite3.Cursor object at 0x031B5720>

>>> cur.fetchall()

[('Manitoba',), ('Alberta',), ('British Columbia',)]

**1i.**

>>> cur.execute('''SELECT Population FROM Density WHERE Land\_Area > 200000''')

<sqlite3.Cursor object at 0x031B5720>

>>> cur.fetchall()

[(512930,), (7237479,), (11410046,), (1119583,), (978933,), (2974807,), (3907738,), (28674,), (37360,), (26745,)]

**1j.**

>>> cur.execute('''SELECT Province\_Territory, round(Population/Land\_Area,2) FROM Density''')

<sqlite3.Cursor object at 0x031B5720>

>>> cur.fetchall()

[('Newfoundland and Labrador', 1.38), ('Prince Edward Island', 23.8), ('Nova Scotia', 17.16), ('New Brunswick', 10.22), ('Quebec', 5.33), ('Ontario', 12.57), ('Manitoba', 2.03), ('Saskatchewan', 1.67), ('Alberta', 4.65), ('British Columbia', 4.22), ('Yukon Territory', 0.06), ('Northwest Territories', 0.03), ('Nunavut', 0.01)]

**Chinese Zodiac**

>>> import sqlite3

>>> con = sqlite3.connect('Chinese\_zodiac.db')

>>> cur = con.cursor()

>>> cur.execute('CREATE TABLE Anim\_charact(Animal TEXT, Charact TEXT)')

<sqlite3.Cursor object at 0x031B6720>

>>> data = ''' insert into Anim\_charact(Animal, Charact) values ('rat', 'forthright, industrious, sensitive, intellectual, sociable'), ('ox', 'dependable, methodical, modest, born leader, patient'), ('tiger', 'unpredictable, rebellious, passionate, daring, impulsive'), ('rabbit', 'good friend, kind, soft-spoken, cautious, artistic'), ('dragon', 'strong, self-assured, proud, decisive, loyal'), ('snake', 'deep thinker, creative, responsible, calm, purposeful'), ('horse', 'cheerful, quick-witted, perceptive, talkative, open-minded'), ('goat', 'sincere, sympathetic, shy, generous, mothering'),('monkey', 'motivator, inquisitive, flexible, innovative, problem solver'), ('rooster' ,'organized, self-assured, decisive, perfectionist, zealous'), ('dog', 'honest, unpretentious, idealistic, moralistic, easy going'), ('pig', 'peace-loving, hard-working , trusting, understanding, thoughtful')'''

>>> cur.execute(data)

<sqlite3.Cursor object at 0x031B6720>

>>> con.commit()

>>> cur.execute('SELECT \* FROM Anim\_charact')

<sqlite3.Cursor object at 0x031B6720>

>>> cur.fetchall()

[('rat', 'forthright, industrious, sensitive, intellectual, sociable'), ('ox', 'dependable, methodical, modest, born leader, patient'), ('tiger', 'unpredictable, rebellious, passionate, daring, impulsive'), ('rabbit', 'good friend, kind, soft-spoken, cautious, artistic'), ('dragon', 'strong, self-assured, proud, decisive, loyal'), ('snake', 'deep thinker, creative, responsible, calm, purposeful'), ('horse', 'cheerful, quick-witted, perceptive, talkative, open-minded'), ('goat', 'sincere, sympathetic, shy, generous, mothering'), ('monkey', 'motivator, inquisitive, flexible, innovative, problem solver'), ('rooster', 'organized, self-assured, decisive, perfectionist, zealous'), ('dog', 'honest, unpretentious, idealistic, moralistic, easy going'), ('pig', 'peace-loving, hard-working , trusting, understanding, thoughtful')]

**TEST 1: Passed**

>>> cur.execute('SELECT Charact FROM Anim\_charact WHERE Animal = "goat"')

<sqlite3.Cursor object at 0x031B6720>

>>> cur.fetchall()

[('sincere, sympathetic, shy, generous, mothering',)]

**TEST 2: Passed**

>>> cur.execute('SELECT Animal FROM Anim\_charact WHERE Charact = "forthright, industrious, sensitive, intellectual, sociable"')

<sqlite3.Cursor object at 0x031C6720>

>>> cur.fetchall()

[('rat',)]

**TEST 3: Passed**

>>> cur.execute('SELECT Animal FROM Anim\_charact WHERE Charact LIKE "%decisive%"')

<sqlite3.Cursor object at 0x031B6720>

>>> cur.fetchall()

[('dragon',), ('rooster',)]