“Homework17” Justin Minsk

1.

from sqlite3 import \*  
  
  
def make\_database():  
 dataframe = connect('census.db')  
 return dataframe  
  
  
def make\_db\_table():  
 dataframe = connect('census.db')  
 df = dataframe.cursor()  
 df.execute('CREATE TABLE Density(Province TEXT, Population INTEGER, Area REAL)')  
 dataframe.commit()  
 return df  
  
  
def add\_entries():  
 dataframe = connect('census.db')  
 df = dataframe.cursor()  
 df.execute('CREATE TABLE Density(Province TEXT, Population INTEGER, Area REAL)')  
 dataframe.commit()  
 table = [  
 ('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),  
 ]  
 for line in table:  
 df.execute('INSERT INTO Density VALUES (?, ?, ?)', line)  
 dataframe.commit()  
  
  
  
def get\_content():  
 dataframe = connect('census.db')  
 df = dataframe.cursor()  
 df.execute('CREATE TABLE Density(Province TEXT, Population INTEGER, Area REAL)')  
 dataframe.commit()  
 table = [  
 ('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),  
 ]  
 for line in table:  
 df.execute('INSERT INTO Density VALUES (?, ?, ?)', line)  
 dataframe.commit()  
 df.execute('SELECT \* FROM Density')  
 for line in df.fetchall():  
 print(line)  
  
  
def get\_pop():  
 dataframe = connect('census.db')  
 df = dataframe.cursor()  
 df.execute('SELECT Population FROM Density')  
 for line in df.fetchall():  
 print(line)  
  
  
def get\_prov\_lt10mill():  
 dataframe = connect('census.db')  
 df = dataframe.cursor()  
 df.execute('SELECT Province FROM Density WHERE Population < 1000000')  
 for line in df.fetchall():  
 print(line)  
  
  
def get\_prov\_lt10mill\_gt5mill():  
 dataframe = connect('census.db')  
 df = dataframe.cursor()  
 df.execute('SELECT Province FROM Density WHERE (Population < 1000000 or Population > 5000000)')  
 for line in df.fetchall():  
 print(line)  
  
  
def get\_prov\_nlt10mill\_ngt5mill():  
 dataframe = connect('census.db')  
 df = dataframe.cursor()  
 df.execute('SELECT Province FROM Density WHERE NOT(Population < 1000000 or Population > 5000000)')  
 for line in df.fetchall():  
 print(line)  
  
  
def get\_prov\_landgt200th():  
 dataframe = connect('census.db')  
 df = dataframe.cursor()  
 df.execute('SELECT Province FROM Density WHERE Area > 200000')  
 for line in df.fetchall():  
 print(line)  
  
  
def get\_popden():  
 dataframe = connect('census.db')  
 df = dataframe.cursor()  
 df.execute('SELECT Province, Population / Area FROM Density')  
 for line in df.fetchall():  
 print(line)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 get\_popden()

2.

from sqlite3 import \*  
  
  
def Chinese\_Zodiac\_SQL():  
 dataframe = connect('zodiac.db')  
 df = dataframe.cursor()  
 df.execute('CREATE TABLE Zodiac(Cycle INTERGER, Animal TEXT, Characteristics TEXT)')  
 dataframe.commit()  
 table = [  
 (0, 'Rat', 'Forthright , industrious, sensitive, intellectual, sociable'),  
 (1, 'Ox', 'Dependable , methodical , modest , born leader, patient'),  
 (2, 'Tiger', 'Unpredictable , rebellious, passionate, daring, impulsive'),  
 (3, 'Rabbit', 'Good friend, kind, soft-spoken , cautious , artistic'),  
 (4, 'Dragon', 'Strong, self-assured, proud, decisive, loyal'),  
 (5, 'Snake', 'Deep thinker, creative, responsible, calm, purposeful'),  
 (6, 'Horse', 'Cheerful, quick-witted, perceptive, talkative, open-minded'),  
 (7, 'Goat', 'Sincere, sympathetic, shy, generous, mothering'),  
 (8, 'Monkey', 'Motivator, inquisitive, flexible, innovative, problem solver'),  
 (9, 'Rooster', 'Organized, self-assured, decisive, perfectionist, zealous'),  
 (10, 'Dog', 'Honest , unpretentious, idealistic, moralistic, easy going'),  
 (11, 'Pig', 'Peace-loving, hard-working , trusting, understanding, thoughtful'),  
 ]  
 for line in table:  
 df.execute('INSERT INTO Zodiac VALUES (?, ?, ?)', line)  
 dataframe.commit()  
  
 terminate = False  
  
 # program greeting  
 print('This program will display your Chinese zodiac sign and associated')  
 print('personal characteristics . \n')  
  
 # get current year from module datetime  
 current\_yr = datetime.date.today().year  
  
 while not terminate:  
 # get year of birth  
 birth\_year = int(input('Enter your year of birth (yyyy) :'))  
  
 # trap birth year errors - notice I can comment to the far left but it hurts the readability  
 while birth\_year < 1900 or birth\_year > current\_yr:  
 print('Invalid year . Please re- enter\n')  
 birth\_year = int(input('Enter your year of birth (yyyy) :'))  
  
 # get index of animal based on birth year  
 cycle\_num = (birth\_year - 1900) % 12  
 cycle = (cycle\_num,)  
 # output results  
 df.execute('SELECT Animal FROM Zodiac WHERE Cycle =?', cycle)  
 data = df.fetchone()  
 print('Your Chinese zodiac sign is the', data[0], '\n')  
 print('Your personal characteristics ...')  
 df.execute('SELECT Characteristics FROM Zodiac WHERE Cycle =?', cycle)  
 data = df.fetchone()  
 print(data[0])  
  
 # continue? / classify another birth year?  
 response = input('\nWould you like to enter another year? (y/n) :')  
 while response != 'y' and response != 'n':  
 response = input("Please enter'y'or'n': ")  
 if response == 'n':  
 terminate = True  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 Chinese\_Zodiac\_SQL()