

DavidGrice_Final_CreateTables

December 18, 2019

0.1 Libraries used for creation

```
[ ]: import sqlite3
import os
```

0.2 Connect to Database

```
[ ]: path = './AliensExist.db'
if os.path.exists(path):
    os.remove(path)
conn = sqlite3.connect(path)
curs = conn.cursor()
```

0.3 Functions used for creation

```
[ ]: def createView(name,sql):
    curs.execute("DROP VIEW IF EXISTS " + name)
    curs.execute("CREATE VIEW " + name + " AS " + sql)
    conn.commit()
```

0.4 Tables for UFO data

```
[ ]: curs.execute("DROP TABLE IF EXISTS tTime")
sql = """CREATE TABLE tTime (
        id INTEGER NOT NULL PRIMARY KEY,
        time TEXT);"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tOMonth")
sql = """CREATE TABLE tOMonth (
        id INTEGER NOT NULL PRIMARY KEY,
        month TEXT);"""
curs.execute(sql)
```

```

curs.execute("DROP TABLE IF EXISTS tODay")
sql = """CREATE TABLE tODay (
            id INTEGER NOT NULL PRIMARY KEY,
            day TEXT);"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tOYear")
sql = """CREATE TABLE tOYear (
            id INTEGER NOT NULL PRIMARY KEY,
            year TEXT);"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tUFOOriginalDate")
sql = """CREATE TABLE tUFOOriginalDate(
            sighting_id INTEGER NOT NULL PRIMARY KEY REFERENCES_
↪tUFOSightingInfo(sighting_id),
            time TEXT REFERENCES tUFOSightingInfo(time),
            origMonth TEXT REFERENCES tOMonth(month),
            origDay TEXT REFERENCES tODay(day),
            origYear TEXT REFERENCES tOYear(year));"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tPMonth")
sql = """CREATE TABLE tPMonth (
            id INTEGER NOT NULL PRIMARY KEY,
            month TEXT);"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tPDay")
sql = """CREATE TABLE tPDay (
            id INTEGER NOT NULL PRIMARY KEY,
            day TEXT);"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tPYear")
sql = """CREATE TABLE tPYear (
            id INTEGER NOT NULL PRIMARY KEY,
            year TEXT);"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tUFOPostedDate")
sql = """CREATE TABLE tUFOPostedDate(
            sighting_id INTEGER NOT NULL PRIMARY KEY REFERENCES_
↪tUFOSightingInfo(sighting_id),
            postMonth TEXT REFERENCES tPMonth(month),
            postDay TEXT REFERENCES tPDay(day),
            postYear TEXT REFERENCES tPYear(year));"""

```

```

curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tLongitude")
sql = """CREATE TABLE tLongitude(
        id INTEGER NOT NULL PRIMARY KEY,
        longitude TEXT);"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tLatitude")
sql = """CREATE TABLE tLatitude(
        id INTEGER NOT NULL PRIMARY KEY,
        latitude TEXT);"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tDuration")
sql = """CREATE TABLE tDuration (
        id INTEGER NOT NULL PRIMARY KEY,
        seconds TEXT,
        other TEXT);"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tUFOCOM")
sql = """CREATE TABLE tUFOCOM (
        id INTEGER NOT NULL PRIMARY KEY,
        com TEXT);"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tShape")
sql = """CREATE TABLE tShape(
        id INTEGER NOT NULL PRIMARY KEY,
        shape TEXT);"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tCity")
sql = """CREATE TABLE tCity(
        id INTEGER NOT NULL PRIMARY KEY,
        city TEXT);"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tUFOCityInfo")
sql = """CREATE TABLE tUFOCityInfo(
        sighting_id INTEGER NOT NULL PRIMARY KEY REFERENCES_
        ↳tUFOSightingInfo(sighting_id),
        ufoCity TEXT REFERENCES tCity(city),
        ufoLatitude TEXT REFERENCES tLatitude(latitude),
        ufoLongitude TEXT REFERENCES tLongitude(longitude));"""
curs.execute(sql)

```

```

curs.execute("DROP TABLE IF EXISTS tState")
sql = """CREATE TABLE tState(
            id INTEGER NOT NULL PRIMARY KEY,
            state TEXT);"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tUFOCityStateCountry")
sql = """CREATE TABLE tUFOCityStateCountry(
            sighting_id INTEGER NOT NULL PRIMARY KEY REFERENCES_
↪tUFOsightingInfo(sighting_id),
            city TEXT REFERENCES tUFOCityInfo(ufoCity),
            state TEXT REFERENCES tState(state),
            country TEXT REFERENCES tCountry(country));"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tCountry")
sql = """CREATE TABLE tCountry(
            id INTEGER NOT NULL PRIMARY KEY,
            country TEXT)"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tUFOStateCountry")
sql = """CREATE TABLE tUFOStateCountry(
            sighting_id INTEGER NOT NULL PRIMARY KEY REFERENCES_
↪tUFOsightingInfo(sighting_id),
            state TEXT REFERENCES tUFOCityState(state),
            country TEXT REFERENCES tCountry(country));"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tUFOsightingInfo")
sql = """CREATE TABLE tUFOsightingInfo (
            sighting_id INTEGER NOT NULL PRIMARY KEY,
            time TEXT REFERENCES tTime(time),
            city TEXT REFERENCES tCity(city),
            comments TEXT REFERENCES tUFOCOM(com),
            shape TEXT REFERENCES tShape(shape),
            duration_seconds TEXT REFERENCES tDuration(seconds),
            duration_other TEXT REFERENCES tDuration(other),
            count INTEGER NOT NULL);"""
curs.execute(sql)

conn.commit()

```

0.5 Tables for Flight data

```
[ ]: curs.execute("DROP TABLE IF EXISTS tFlightPassengers")
sql = """CREATE TABLE tFlightPassengers(
        id INTEGER NOT NULL PRIMARY KEY,
        passengers TEXT);"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tFlightSeats")
sql = """CREATE TABLE tFlightSeats(
        id INTEGER NOT NULL PRIMARY KEY,
        seats TEXT);"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tFlightFlights")
sql = """CREATE TABLE tFlightFlights(
        id INTEGER NOT NULL PRIMARY KEY,
        flights TEXT);"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tFlightDistance")
sql = """CREATE TABLE tFlightDistance(
        id INTEGER NOT NULL PRIMARY KEY,
        distance TEXT);"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tFlightPlaneInfo")
sql = """CREATE TABLE tFlightPlaneInfo(
        plane_id INTEGER NOT NULL PRIMARY KEY,
        oAirport TEXT REFERENCES tFlightOrigAirport(orig_airport),
        month TEXT REFERENCES tFlightMonth(month),
        day TEXT REFERENCES tFlightDay(day),
        year TEXT REFERENCES tFlightYear(year),
        dAirport TEXT REFERENCES tFlightDestAirport(dest_airport),
        flights TEXT REFERENCES tFlightFlights(flights));"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tFlightMonth")
sql = """CREATE TABLE tFlightMonth(
        id INTEGER NOT NULL PRIMARY KEY,
        month TEXT)"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tFlightDay")
sql = """CREATE TABLE tFlightDay(
        id INTEGER NOT NULL PRIMARY KEY,
        day TEXT);"""
```

```

curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tFlightYear")
sql = """CREATE TABLE tFlightYear(
            id INTEGER NOT NULL PRIMARY KEY,
            year TEXT);"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tFlightDate")
sql = """CREATE TABLE tFlightDate(
            plane_id INTEGER NOT NULL PRIMARY KEY REFERENCES tFlightPlaneInfo(plane_id),
            month TEXT REFERENCES tFlightMonth(month),
            day TEXT REFERENCES tFlightDay(day),
            year TEXT REFERENCES tFlightYear(year));"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tFlightOrigPopulation")
sql = """CREATE TABLE tFlightOrigPopulation(
            id INTEGER NOT NULL PRIMARY KEY,
            orig_population TEXT);"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tFlightOrigCity")
sql = """CREATE TABLE tFlightOrigCity(
            id INTEGER NOT NULL PRIMARY KEY,
            orig_city TEXT);"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tFlightOrigState")
sql = """CREATE TABLE tFlightOrigState(
            id INTEGER NOT NULL PRIMARY KEY,
            orig_state TEXT);"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tFlightOrigAirport")
sql = """CREATE TABLE tFlightOrigAirport(
            id INTEGER NOT NULL PRIMARY KEY,
            orig_airport TEXT);"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tFlightOrigAirportLat")
sql = """CREATE TABLE tFlightOrigAirportLat(
            id INTEGER NOT NULL PRIMARY KEY,
            orig_airportLat TEXT);"""
curs.execute(sql)

```

```

curs.execute("DROP TABLE IF EXISTS tFlightOrigAirportLong")
sql = """CREATE TABLE tFlightOrigAirportLong(
        id INTEGER NOT NULL PRIMARY KEY,
        orig_airportLong TEXT);"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tFlightOrigAirportInfo")
sql = """CREATE TABLE tFlightOrigAirportInfo(
        plane_id INTEGER NOT NULL PRIMARY KEY REFERENCES_
        ↪tFlightPlanceInfo(plane_id),
        orig_city TEXT REFERENCES tFlightOrigCity(orig_city),
        orig_state TEXT REFERENCES tFlightOrigState(orig_state),
        orig_airport TEXT REFERENCES tFlightOrigAirport(orig_airport),
        orig_airportLat TEXT REFERENCES_
        ↪tFlightOrigAirportLat(orig_airportLat),
        orig_airportLong TEXT REFERENCES_
        ↪tFlightOrigAirportLong(orig_airportLong))"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tFlightDestPopulation")
sql = """CREATE TABLE tFlightDestPopulation(
        id INTEGER NOT NULL PRIMARY KEY,
        dest_population TEXT);"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tFlightDestCity")
sql = """CREATE TABLE tFlightDestCity(
        id INTEGER NOT NULL PRIMARY KEY,
        dest_city TEXT);"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tFlightDestState")
sql = """CREATE TABLE tFlightDestState(
        id INTEGER NOT NULL PRIMARY KEY,
        dest_state TEXT);"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tFlightDestAirport")
sql = """CREATE TABLE tFlightDestAirport(
        id INTEGER NOT NULL PRIMARY KEY,
        dest_airport TEXT);"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tFlightDestAirportLat")
sql = """CREATE TABLE tFlightDestAirportLat(
        id INTEGER NOT NULL PRIMARY KEY,
        dest_airportLat TEXT);"""

```

```

curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tFlightDestAirportLong")
sql = """CREATE TABLE tFlightDestAirportLong(
            id INTEGER NOT NULL PRIMARY KEY,
            dest_airportLong TEXT);"""
curs.execute(sql)

curs.execute("DROP TABLE IF EXISTS tFlightDestAirportInfo")
sql = """CREATE TABLE tFlightDestAirportInfo(
            plane_id INTEGER NOT NULL PRIMARY KEY REFERENCES_
            ↪tFlightPlaneInfo(plane_id),
            dest_city TEXT REFERENCES tFlightDestCity(dest_city),
            dest_state TEXT REFERENCES tFlightDestState(dest_state),
            dest_airport TEXT REFERENCES tFlightDestAirport(dest_airport),
            dest_airportLat TEXT REFERENCES_
            ↪tFlightDestAirportLat(dest_airportLat),
            dest_airportLong TEXT REFERENCES_
            ↪tFlightDestAirportLong(dest_airportLong));"""
curs.execute(sql)

conn.commit()

# Turn on foreign key constraints
curs.execute("PRAGMA foreign_keys = on;")

conn.close()

```

[]: