```
############################################################
version.py
import sqlite3 as sqlite
import sys

con = None

try:
    con = sqlite.connect('ydb.db')

    cur = con.cursor()
    cur.execute('SELECT SQLITE_VERSION()')

    data = cur.fetchone()[0]

    print(f"SQLite version: {data}")

except sqlite.Error as e:

    print(f"Error {e.args[0]}")
    sys.exit(1)

finally:

    if con:
        con.close()

############################################################
version2.py
import sqlite3 as sqlite

con = sqlite.connect('ydb.db')

with con:

    cur = con.cursor()
    cur.execute('SELECT SQLITE_VERSION()')

    data = cur.fetchone()[0]

    print(f"SQLite version: {data}")
```

```
##############################################
create_table.py
import sqlite3 as sqlite

con = sqlite.connect('ydb.db')

with con:

    cur = con.cursor()

    cur.execute("CREATE TABLE cars(id INT, name TEXT, price INT)")
    cur.execute("INSERT INTO cars VALUES(1,'Audi',52642)")
    cur.execute("INSERT INTO cars VALUES(2,'Mercedes',57127)")
    cur.execute("INSERT INTO cars VALUES(3,'Skoda',9000)")
    cur.execute("INSERT INTO cars VALUES(4,'Volvo',29000)")
    cur.execute("INSERT INTO cars VALUES(5,'Bentley',350000)")
    cur.execute("INSERT INTO cars VALUES(6,'Citroen',21000)")
    cur.execute("INSERT INTO cars VALUES(7,'Hummer',41400)")
    cur.execute("INSERT INTO cars VALUES(8,'Volkswagen',21600)")


##################################################
create_table2.py
import sqlite3 as sqlite

cars = (
    (1, 'Audi', 52642),
    (2, 'Mercedes', 57127),
    (3, 'Skoda', 9000),
    (4, 'Volvo', 29000),
    (5, 'Bentley', 350000),
    (6, 'Hummer', 41400),
    (7, 'Volkswagen', 21600)
)

con = sqlite.connect('ydb.db')

with con:

    cur = con.cursor()
```

```
    cur.execute("DROP TABLE IF EXISTS cars")
    cur.execute("CREATE TABLE cars(id INT, name TEXT, price INT)")
    cur.executemany("INSERT INTO cars VALUES(?, ?, ?)", cars)

#################################################################
create_table3.py
import sqlite3 as sqlite
import sys

con = None

try:
    con = sqlite.connect('ydb.db')

    cur = con.cursor()

    cur.executescript("""
        DROP TABLE IF EXISTS cars;
        CREATE TABLE cars(id INT, name TEXT, price INT);
        INSERT INTO cars VALUES(1,'Audi',52642);
        INSERT INTO cars VALUES(2,'Mercedes',57127);
        INSERT INTO cars VALUES(3,'Skoda',9000);
        INSERT INTO cars VALUES(4,'Volvo',29000);
        INSERT INTO cars VALUES(5,'Bentley',350000);
        INSERT INTO cars VALUES(6,'Citroen',21000);
        INSERT INTO cars VALUES(7,'Hummer',41400);
        INSERT INTO cars VALUES(8,'Volkswagen',21600);
        """)

    con.commit()

except sqlite.Error as e:

    if con:
        con.rollback()

    print(f"Error {e.args[0]}")
    sys.exit(1)

finally:
```

```
    if con:
        con.close()

#####################################################################
lastrowid.py
import sqlite3 as sqlite

con = sqlite.connect(':memory:')

with con:

    cur = con.cursor()
    cur.execute("CREATE TABLE friends(id INTEGER PRIMARY KEY, name TEXT);")
    cur.execute("INSERT INTO friends(name) VALUES ('Tom');")
    cur.execute("INSERT INTO friends(name) VALUES ('Rebecca');")
    cur.execute("INSERT INTO friends(name) VALUES ('Jim');")
    cur.execute("INSERT INTO friends(name) VALUES ('Robert');")

    last_row_id = cur.lastrowid

    print(f"The last Id of the inserted row is {last_row_id}")

#####################################################################
fetch_all.py
import sqlite3 as sqlite

con = sqlite.connect('ydb.db')

with con:

    cur = con.cursor()
    cur.execute("SELECT * FROM cars")

    rows = cur.fetchall()

    for row in rows:
        print(f"{row[0]} {row[1]} {row[2]}")
```

```
#################################################################
fetch_one.py
import sqlite3 as sqlite

con = sqlite.connect('ydb.db')

with con:

    cur = con.cursor()
    cur.execute("SELECT * FROM cars")

    while True:

        row = cur.fetchone()

        if row == None:
            break

        print(f"{row[0]} {row[1]} {row[2]}")

#############################################################
dictionary_cursor.py
import sqlite3 as sqlite

con = sqlite.connect('ydb.db')

with con:

    con.row_factory = sqlite.Row

    cur = con.cursor()
    cur.execute("SELECT * FROM cars")

    rows = cur.fetchall()

    for row in rows:
        print(f"{row['id']} {row['name']} {row['price']}")
```

```
############################################################

parameterized_query.py
import sqlite3 as sqlite

uId = 1
uPrice = 62300

con = sqlite.connect('ydb.db')

with con:

    cur = con.cursor()
    cur.execute("UPDATE cars SET price=? WHERE id=?", (uPrice, uId))

    print(f"Number of rows updated: {cur.rowcount}")

############################################################
named_placeholders.py
import sqlite3 as sqlite

uId = 4

con = sqlite.connect('ydb.db')

with con:

    cur = con.cursor()
    cur.execute("SELECT name, price FROM cars WHERE Id=:Id", {"Id": uId})

    row = cur.fetchone()
    print(f"{row[0]}, {row[1]}")

############################################################
insert_image.py


import sqlite3 as sqlite
import sys

def readImage():
```

```
    fin = None

    try:
        fin = open("sid.jpg", "rb")
        img = fin.read()
        return img

    except IOError as e:

        print(e)
        sys.exit(1)

    finally:

        if fin:
            fin.close()

con = None

try:
    con = sqlite.connect('ydb.db')

    cur = con.cursor()

    data = readImage()
    binary = sqlite.Binary(data)
    cur.execute("INSERT INTO images(data) VALUES (?)", (binary,) )

    con.commit()

except sqlite.Error as e:

    if con:
        con.rollback()

    print(e)
    sys.exit(1)

finally:
```

```
    if con:
        con.close()

#################################################################


read_image.py


import sqlite3 as sqlite
import sys


def writeImage(data):

    fout = None

    try:
        fout = open('sid2.png','wb')
        fout.write(data)

    except IOError as e:

        print(e)
        sys.exit(1)

    finally:

        if fout:
            fout.close()

con = None

try:
    con = sqlite.connect('ydb.db')

    cur = con.cursor()
    cur.execute("SELECT data FROM images LIMIT 1")
    data = cur.fetchone()[0]

    writeImage(data)
```

```
except sqlite.Error as e:

    print(e)
    sys.exit(1)

finally:

    if con:
        con.close()

#############################################################
column_names.py
import sqlite3 as sqlite

con = sqlite.connect('ydb.db')

with con:

    cur = con.cursor()

    cur.execute('PRAGMA table_info(cars)')

    data = cur.fetchall()

    for d in data:
        print(f"{d[0]} {d[1]} {d[2]}")

#############################################################
column_names2.py
import sqlite3 as sqlite

con = sqlite.connect('ydb.db')

with con:

    cur = con.cursor()
    cur.execute('SELECT * FROM cars')

    col_names = [cn[0] for cn in cur.description]
```

```
    rows = cur.fetchall()

    print(f"{col_names[0]:3} {col_names[1]:10} {col_names[2]:7}")

    for row in rows:
        print(f"{row[0]:<3} {row[1]:<10} {row[2]:7}")
################################################
list_tables.py
import sqlite3 as sqlite

con = sqlite.connect('ydb.db')

with con:

    cur = con.cursor()
    cur.execute("SELECT name FROM sqlite_master WHERE type='table'")

    rows = cur.fetchall()

    for row in rows:
        print(row[0])

#####################################################
export_table.py
import sqlite3 as sqlite

cars = (
    (1, 'Audi', 52643),
    (2, 'Mercedes', 57642),
    (3, 'Skoda', 9000),
    (4, 'Volvo', 29000),
    (5, 'Bentley', 350000),
    (6, 'Hummer', 41400),
    (7, 'Volkswagen', 21600)
)

def writeData(data):

    f = open('cars.sql', 'w')
```

```
    with f:
        f.write(data)


con = sqlite.connect(':memory:')

with con:

    cur = con.cursor()

    cur.execute("DROP TABLE IF EXISTS cars")
    cur.execute("CREATE TABLE cars(id INT, name TEXT, price INT)")
    cur.executemany("INSERT INTO cars VALUES(?, ?, ?)", cars)
    cur.execute("DELETE FROM cars WHERE price < 30000")

    data = '\n'.join(con.iterdump())

    writeData(data)

#########################################################
cars.sql
BEGIN TRANSACTION;
CREATE TABLE cars(id INT, name TEXT, price INT);
INSERT INTO "cars" VALUES(1,'Audi',52643);
INSERT INTO "cars" VALUES(2,'Mercedes',57642);
INSERT INTO "cars" VALUES(5,'Bentley',350000);
INSERT INTO "cars" VALUES(6,'Hummer',41400);
COMMIT;

import_table.py
import sqlite3 as sqlite


def readData():

    f = open('cars.sql', 'r')

    with f:

        data = f.read()
```

```
        return data


con = sqlite.connect(':memory:')

with con:

    cur = con.cursor()

    sql = readData()
    cur.executescript(sql)

    cur.execute("SELECT * FROM cars")

    rows = cur.fetchall()

    for row in rows:
        print(row)

################################################
no_commit.py
import sqlite3 as sqlite
import sys

con = None

try:
    con = sqlite.connect('ydb.db')

    cur = con.cursor()
    cur.execute("DROP TABLE IF EXISTS friends")
    cur.execute("CREATE TABLE friends(id INTEGER PRIMARY KEY, name TEXT)")
    cur.execute("INSERT INTO friends(name) VALUES ('Tom')")
    cur.execute("INSERT INTO friends(name) VALUES ('Rebecca')")
    cur.execute("INSERT INTO friends(name) VALUES ('Jim')")
    cur.execute("INSERT INTO friends(name) VALUES ('Robert')")

    #con.commit()

except sqlite.Error as e:
```

```
    if con:
        con.rollback()

    print(e)
    sys.exit(1)

finally:

    if con:
        con.close()

#######################################
autocommit.py
import sqlite3 as sqlite
import sys

con = None

try:
    con = sqlite.connect('ydb.db', isolation_level=None)

    cur = con.cursor()

    cur.execute("DROP TABLE IF EXISTS friends")
    cur.execute("CREATE TABLE friends(id INTEGER PRIMARY KEY, name TEXT)")
    cur.execute("INSERT INTO friends(name) VALUES ('Tom')")
    cur.execute("INSERT INTO friends(name) VALUES ('Rebecca')")
    cur.execute("INSERT INTO friends(name) VALUES ('Jim')")
    cur.execute("INSERT INTO friends(name) VALUES ('Robert')")

except sqlite.Error as e:

    print(e)
    sys.exit(1)

finally:

    if con:
        con.close()
```

column_names.py