

In [2]:

```
#!/usr/bin/python
import sqlite3
conn = sqlite3.connect('test.db')
print ("Opened database successfully");
```

Opened database successfully

In [5]:

```
#!/usr/bin/python
import sqlite3
conn=sqlite3.connect('test.db')
print("data base opened succussfully");
```

data base opened succussfully

In [36]:

```
#!/usr/bin/python
import sqlite3
conn = sqlite3.connect('test.db')
print ("Opened database successfully");
conn.execute('''CREATE TABLE COMPANY
              (ID INT PRIMARY KEY     NOT NULL,
              NAME           TEXT      NOT NULL,
              AGE            INT       NOT NULL,
              ADDRESS        CHAR(50),
              SALARY         REAL);''')
print ("Table created successfully");
conn.close()
```

Opened database successfully

OperationalError Traceback (most recent call last)
C:\Users\ANILKU~1\AppData\Local\Temp\ipykernel_19060\3376885754.py in <module>
e>

```
3 conn = sqlite3.connect('test.db')
4 print ("Opened database successfully");
----> 5 conn.execute('''CREATE TABLE COMPANY
6           (ID INT PRIMARY KEY     NOT NULL,
7           NAME           TEXT      NOT NULL,
```

OperationalError: table COMPANY already exists

In [40]:

```
#!/usr/bin/python
import sqlite3
conn=sqlite3.connect('test.db')
print("database opened successfully");
conn.execute("insert into company(ID,NAME,ADDRESS,SALARY),VALUES(1,'ANIL','ABCD',220000)");
conn.execute("insert into company(ID,NAME,ADDRESS,SALARY),VALUES(2,'AMAR','EFGH',22000)");
conn.execute("insert into company(ID,NAME,ADDRESS,SALARY),VALUES(3,'ANAND','IJKL',13000)");
conn.commit()
print("records inserted successfully...")
conn.close()
```

database opened successfully

```
-----
OperationalError                                Traceback (most recent call last)
C:\Users\ANILKU~1\AppData\Local\Temp\ipykernel_19060\3822007347.py in <modul
e>
```

```
3 conn=sqlite3.connect('test.db')
4 print("database opened successfully");
----> 5 conn.execute("insert into company(ID,NAME,ADDRESS,SALARY),VALUES
(1,'ANIL','ABCD',220000)");
6 conn.execute("insert into company(ID,NAME,ADDRESS,SALARY),VALUES
(2,'AMAR','EFGH',22000)");
7 conn.execute("insert into company(ID,NAME,ADDRESS,SALARY),VALUES
(3,'ANAND','IJKL',13000)");
```

OperationalError: near ",": syntax error

In [44]:

```
#!/usr/bin/python
import sqlite3
conn=sqlite3.connect('test.db')
print("data base opened successfully")
cursor = conn.execute("select id,name,address,salary from company")
for row in cursor :
    print ("ID = ",row[0])
    print ("NAME = ",row[1])
    print ("address = ",row[2])
    print ("salary = ",row[3], "\n")
print("operation done successfully");
conn.close()
```

data base opened successfully
operation done successfully

In []:

```
#!/usr/bin/python
import sqlite3
conn=sqlite3.connect('test.db')
conn.execute(" update company set salary=25000.00 where id=1 ")
conn.commit()
cursor=conn.execute("select id,name,address,salary from company")
for rows in cursor :
    print ("id = ", rows[0])
    print ("name = ", rows[1])
    print ("address = ",rows[2])
    print ("salary = ",rows[3], "\n")
print("operation done successfully")
conn.close()
```

In [48]:

```
conn.execute("select id,name,address,salary from company");
```

In [50]:

```
#!/usr/bin/python
import sqlite3
conn=sqlite3.connect('test.db')
print("sql data base opened successfully")
conn.execute("delete from company where id=2")
conn.commit()
print("total number of rows deleted : ", conn.total_changes)
cursor = conn.execute("select id,name,address,salary from salary")
for rows in cursors :
    print("id = ",row[0])
    print("name = ",row[1])
    print("address = ",rows[2])
    print("salary = ",rows[3], "\n")
print("operation done successfully")
conn.close()
```

sql data base opened successfully

```
-----
OperationalError                                Traceback (most recent call last)
C:\Users\ANILKU~1\AppData\Local\Temp\ipykernel_19060\651787545.py in <module>
>
      3 conn=sqlite3.connect('test.db')
      4 print("sql data base opened successfully")
----> 5 conn.execute("delete from company where id=2")
      6 conn.commit()
      7 print("total number of rows deleted : ", conn.total_changes)
```

OperationalError: database is locked

In []: