In [2]:

```
pip install mysql-connector-python
```

```
Requirement already satisfied: mysql-connector-python in c:\users\v3it\an
aconda3\lib\site-packages (8.0.32)
Requirement already satisfied: protobuf<=3.20.3,>=3.11.0 in c:\users\v3it
\anaconda3\lib\site-packages (from mysql-connector-python) (3.19.1)
Note: you may need to restart the kernel to use updated packages.
```

In [3]:

```
#Importing the liabraries

import mysql.connector
from mysql.connector import Error
import pandas as pd
```

In [4]:

```
def create_server_connection(host_name,user_name,user_password):

    global connection
    connection = None

    try:
        connection = mysql.connector.connect(
        host = host_name,
        user = user_name,
            password = user_password
        )
        print("MySQL connection Successfull")

    except Error as er:
        print(f"Error: {er}")

    return connection

ps = "12345"  #exact password of server

db = "Internship"

create_server_connection("localhost","root",ps) # root is username
```

```
MySQL connection Successfull
```

Out[4]:

```
<mysql.connector.connection_cext.CMySQLConnection at 0x1d3aa1ae070>
```

In [8]:

```
#create database Internship

def create_database(connection,query):
    cursor = connection.cursor()

    try:
        cursor.execute(query)
        print("Database Created Successfully")

    except Error as er:
        print(f"Error: {er}")

create_database_query = "Create Database Internship"
create_database(connection,create_database_query)
#create_database(connection,"create database Internship")
```

```
Error: 1007 (HY000): Can't create database 'internship'; database exists
```

In [10]:

```
# connect to database

def create_db_connection(host_name,user_name,user_password,db_name):
    connection = None

    try:
        connection = mysql.connector.connect(
        host = host_name,
        user = user_name,
            password = user_password,
            database = db_name
        )
        print("MySQL Database Connection Successfully")

    except Error as er:
        print(f"Error: {er}")

    return connection
```

In [11]:

```
# Execute SQL Queries

global cursor

def execute_query(connection,query):
    cursor = connection.cursor()

    try:
        cursor.execute(query) #cursor is middle wear between
        connection.commit() #save the last action use commit
        print("Query was Successful")

    except Error as er:
            print(f"Error: {er}")
```

In [13]:

```
#multiple line use"""
create_order_table = """
Create table orders(
order_id int primary key,
customer_name varchar(30) not null,
product_name varchar(30) not null,
date_ordered date,
quantity int,
unit_price float,
phone_number varchar(30));
"""

# connect to the database

connection = create_db_connection("localhost", "root", ps, db)
execute_query(connection,create_order_table)
```

```
MySQL Database Connection Successfully
Error: 1050 (42S01): Table 'orders' already exists
```

In [14]:

```
# insert data

data_orders = """
insert into orders values
(101, 'Steve', 'Laptop', '2018-06-12', 2, 800, '6293730802'),
(102, 'Jos', 'Books', '2019-02-10', 10, 12, '8367489124'),
(103, 'Stacy', 'Trousers', '2019-12-25', 5, 50, '8976123645'),
(104, 'Nancy', 'T-Shirts', '2018-07-14', 7, 30, '7368145099'),
(105, 'Maria', 'Headphones', '2019-05-30', 6, 48, '8865316698'),
(106, 'Danny', 'Smart TV', '2018-08-20', 10, 300, '7720130449');
"""
connection = create_db_connection("localhost", "root",ps, db)
execute_query(connection,data_orders)
```

```
MySQL Database Connection Successfully
Error: 1062 (23000): Duplicate entry '101' for key 'orders.PRIMARY'
```

In [17]:

```
def read_query(connection,query):
    cursor = connection.cursor()
    results = None

    try:
        cursor.execute(query)
        result = cursor.fetchall()
        return result

    except Error as er:
            print(f"Error: {er}")
```

In [18]:

```
# Using Select statement

Q1 = """
Select * from orders;
"""
connection = create_db_connection("localhost", "root", ps, db)
results = read_query(connection,Q1)

for result in results:
    print(result)
```

```
MySQL Database Connection Successfully
(101, 'Steve', 'Laptop', datetime.date(2018, 6, 12), 2, 800.0, '629373080
2')
(102, 'Joe', 'Books', datetime.date(2019, 2, 10), 10, 12.0, '8367489124')
(104, 'Nancy', 'T-Shirts', datetime.date(2018, 7, 14), 7, 30.0, '73681450
99')
(105, 'Maria', 'Headphones', datetime.date(2019, 5, 30), 6, 48.0, '886531
6698')
(106, 'Danny', 'Smart TV', datetime.date(2018, 8, 20), 10, 300.0, '772013
0449')
```

In [49]:

```
#create dataframe

from_db = []

for result in results:
    result = list(result)
    from_db.append(result)

columns = ["order_id","customer_name",
"product_name",
"date_ordered",
"quantity",
"unit_price",
"phone_number"]

df = pd.DataFrame(from_db,columns = columns)
display(df)
```

	order_id	customer_name	product_name	date_ordered	quantity	unit_price	phone_numb
0	101	Steve	Laptop	2018-06-12	2	800.0	62937308
1	102	Jos	Books	2019-02-10	10	12.0	83674891
2	103	Stacy	Trousers	2019-12-25	5	50.0	89761236
3	104	Nancy	T-Shirts	2018-07-14	7	30.0	73681450
4	105	Maria	Headphones	2019-05-30	6	48.0	88653166
5	106	Danny	Smart TV	2018-08-20	10	300.0	77201304

In [73]:

```
Q2 = """
SELECT DISTINCT year(date_ordered) FROM orders;
Select * from orders;
"""
connection = create_db_connection("localhost", "root", ps, db)
results = read_query(connection,Q2)

for result in results:
    print(result)
```

```
MySQL Database Connection Successfully
(2018,)
(2019,)
```

Query Tasks -

1- delete a row 2 - update a record 3 - fetch some condition(unit price is greater or smaller) 4 - query with order by 5 - select * from orders;

In [19]:

```
Query1 = """
delete from orders
where order_id = 102;
"""
connection = create_db_connection("localhost", "root", ps, db)
result = execute_query(connection,Query1)

for result in results:
    print(result)
```

```
MySQL Database Connection Successfully
Query was Successful
(101, 'Steve', 'Laptop', datetime.date(2018, 6, 12), 2, 800.0, '629373080
2')
(102, 'Joe', 'Books', datetime.date(2019, 2, 10), 10, 12.0, '8367489124')
(104, 'Nancy', 'T-Shirts', datetime.date(2018, 7, 14), 7, 30.0, '73681450
99')
(105, 'Maria', 'Headphones', datetime.date(2019, 5, 30), 6, 48.0, '886531
6698')
(106, 'Danny', 'Smart TV', datetime.date(2018, 8, 20), 10, 300.0, '772013
0449')
```

In [23]:

```
Query2 = """
UPDATE orders
SET customer_name = 'Joe'
where order_id = 102;
"""
connection = create_db_connection("localhost", "root", ps, db)
results = execute_query(connection,Query2)
for result in results:
    print(result)
```

```
MySQL Database Connection Successfully
Query was Successful
```

In [25]:

```
Query3 = """
SELECT * from orders where unit_price > 50;
Select * from orders;
"""
connection = create_db_connection("localhost", "root", ps, db)
results = read_query(connection,Query3)

for result in results:
    print(result)
```

```
MySQL Database Connection Successfully
(101, 'Steve', 'Laptop', datetime.date(2018, 6, 12), 2, 800.0, '629373080
2')
(106, 'Danny', 'Smart TV', datetime.date(2018, 8, 20), 10, 300.0, '772013
0449')
```

In [122]:

```
Query4 = """
SELECT * FROM orders ORDER BY customer_name = "DESC"
"""
connection = create_db_connection("localhost", "root", ps, db)
results = read_query(connection,Query4)

for result in results:
    print(result)
```

```
MySQL Database Connection Successfully
(101, 'Steve', 'Laptop', datetime.date(2018, 6, 12), 2, 800.0, '629373080
2')
(102, 'Joe', 'Books', datetime.date(2019, 2, 10), 10, 12.0, '8367489124')
(104, 'Nancy', 'T-Shirts', datetime.date(2018, 7, 14), 7, 30.0, '73681450
99')
(105, 'Maria', 'Headphones', datetime.date(2019, 5, 30), 6, 48.0, '886531
6698')
(106, 'Danny', 'Smart TV', datetime.date(2018, 8, 20), 10, 300.0, '772013
0449')
```

In [28]:

```
Query5 = """
Select * from orders;
"""
connection = create_db_connection("localhost", "root", ps, db)
results = read_query(connection,Query5)

for result in results:
    print(result)
```

```
MySQL Database Connection Successfully
(101, 'Steve', 'Laptop', datetime.date(2018, 6, 12), 2, 800.0, '629373080
2')
(104, 'Nancy', 'T-Shirts', datetime.date(2018, 7, 14), 7, 30.0, '73681450
99')
(105, 'Maria', 'Headphones', datetime.date(2019, 5, 30), 6, 48.0, '886531
6698')
(106, 'Danny', 'Smart TV', datetime.date(2018, 8, 20), 10, 300.0, '772013
0449')
```

In [ ]: