Name: Bhavik Agrawal Div : A Roll No:41

Q-1: -

Create the following tables and enter at least 10 records with appropriate constraints:

Employees(employee_id, first_name, last_name, age, email, phone_number, hire_date, job_id, salary, commission_pct, manager_id, department_id)

Departments(Department_Id, Department_Name, Manager_Id, Location_Id)

Locations(location_id, street_address, postal_code city, state_province, country_id)

Countries(country_id, country_name, region_id)

(A) Write a query to display the names (first_name, last_name) using alias name "First Name", "Last Name"

```
C:\Users\pratham sarang\AppData\Local\Temp\Temp6_sqlite-tools-win32-x86-3350500.zip\sqlite-tools-win32
sqlite> select first_name as 'First Name',last_name as 'Last Name' from employees;
```

First Name	Last Name
Rishit	Sarang
Ali	Akbar
Pruthvi	solanki
Yash	Makwana
Bhavik	Tripathi
Rahul	Solanki
Ritesh	Sharma
Riya	Sharma
Priya	Matter
Sonu	Sarang

```
sqlite>
```

Name: Bhavik Agrawal Div : A Roll No:41

(B) Write a query to select first 10 records from a table

C:\Users\pratham sarang\AppData\Local\Temp\Temp6_sqlite-tools-win32-x86-3350500.zip\sqlite-tools-win32-x86-3350500\sqlite3.exe

```
sqlite> select *from employees;
```

emp_id	first_name	last_name	age	email	phone_number	hire_date	job_id	job_title	salary	commision_pct	manager_id	dept_id
501	Rishit	Sarang	20	rishit@gmail.com	9106641816	01-02-2021	601	assistant_manger	30000	10%	401	301
502	Ali	Akbar	22	ali@gmail.com	9123456799	01-05-2022	602	project_manger	31000	11%	402	302
503	Pruthvi	solanki	21	pruthvi@gmail.com	9874506799	01-02-2022	603	president	35000	05%	403	303
504	Yash	Makwana	24	yash@gmail.com	9564913054	01-10-2019	604	clerk	20000	07%	404	304
505	Bhavik	Tripathi	22	bhavik@gmail.com	9876543218	01-01-2018	605	officer	18000	03%	405	305
506	Rahul	Solanki	20	rahul@gmail.com	9408978119	15-08-2020	606	sales_manager	21000	05%	406	306
507	Ritesh	Sharma	25	ritesh@gmail.com	9638460271	15-10-2021	607	department_executive	36000	10%	407	307
508	Riya	Sharma	23	riya@gmail.com	9867524624	15-02-2020	608	assistant_director	26000	15%	408	308
509	Priya	Matter	23	priya@gmail.com	9455652644	15-07-2021	609	head_salesman	30000	10%	409	309
510	Sonu	Sarang	22	sonu@gmail.com	8555652454	15-06-2017	610	project_executive	32000	09%	410	310

```
sqlite> select*from departments;
```

dept_id	dept_name	manager_id	location_id
301	production	401	201
302	sales	402	202
303	IT	403	203
304	production	404	204
305	sales	405	205
306	IT	406	206
307	production	407	207
308	sales	408	208
309	IT	409	209
310	production	410	210

```
sqlite>
```

C:\Users\pratham sarang\AppData\Local\Temp\Temp6_sqlite-tools-win32-x86-3350500.zip\sqlite-tools-win32-x86-3350500\sqlite3.exe

```
sqlite> select*from locations;
```

location_id	street_address	postal_code	city	state_province	country_id
202	Dineper River	3451236	Kyiv	oblast	101
201	Dhaka Tower St	4567856	Madian	Dhaka	102
203	Andrew Street	9485343	Bruges	west Flanders	103
204	Football St	9423567	Braflia	Central west	104
205	Marshall St	4567865	Sofia	Sofia Province	105
206	Cario St	3452532	Toronto	Ontario	106
207	Andheri	3567935	Mumbai	Maharastra	107
208	NYC St	5644325	New York City	Queens	108
209	Mohammed-ul-arham St	5364521	Kuwait City	Kuwait	109
210	Link Road	4536754	Addu	Maldives	110

```
sqlite> select*from countries;
```

country_id	country_name	region_id
101	ukraine	1
102	Bangladesh	2
103	Belgium	3
104	Bulgaria	4
105	Canada	5
106	Brazil	6
107	India	7
108	United States	8
109	Kuwait	9
110	Maldives	10

```
sqlite>
```

(C) Write a query to display the last names of employees whose names have exactly 6 characters

C:\Users\pratham sarang\AppData\Local\Temp\Temp6_sqlite-tools-win32-x86-3350500.zip\sqlite-tools-win32-

```
sqlite> select first_name,last_name from employees where last_name like '______';
```

first_name	last_name
Rishit	Sarang
Ritesh	Sharma
Riya	Sharma
Priya	Matter
Sonu	Sarang

```
sqlite>
```

(D) Write a query to get the department ID and the total salary payable in each department

```
C:\Users\pratham sarang\AppData\Local\Temp\Temp6_sqlite-tools-win32-x86-3350500.zip\sqlite-to
sqlite> select dept_id,sum(salary) salary from employees group by dept_id;
```

dept_id	salary
301	30000
302	31000
303	35000
304	20000
305	18000
306	21000
307	36000
308	26000
309	30000
310	32000

```
sqlite>
```

(E) Write a query to find the names (first_name, last_name) of the employees who have a manager who works for a departmentbased in the United States

```
C:\Users\pratham sarang\AppData\Local\Temp\Temp6_sqlite-tools-win32-x86-3350500
sqlite> select first_name,last_name from employees
   ...> where manager_id=(select manager_id from departments
   ...> where location_id=(select location_id from locations
   ...> where country_id=(select country_id from countries
   ...> where country_name='United States'
   ...> )));
```

first_name	last_name
Riya	Sharma

```
sqlite>
```

(F) Write a query to find the names (first_name, last_name), the salary of the employees who earn more than the average salary and who works in any of the IT departments

```
C:\Users\pratham sarang\AppData\Local\Temp\Temp6_sqlite-tools-win32-x86-3350500.zip\sqlite-
sqlite> select first_name,last_name,salary from employees
   ...> where dept_id
   ...> in(select dept_id from Departments where dept_name like 'IT')
   ...> and salary>(select avg(salary) from Employees);
```

first_name	last_name	salary
Pruthvi	solanki	35000
Priya	Matter	30000

```
sqlite>
```

(G) Write a query to find the employee id, name (last_name) along with their manager_id, manager name (last_name).

```
C:\Users\pratham sarang\AppData\Local\Temp\Temp6_sqlite-tools-win32-x86-3350500.zip\sqlite-tools-win32-x86-3350500
sqlite> select emp_id,first_name,last_name as 'Manager Last Name',manager_id from employees;
```

emp_id	first_name	Manager Last Name	manager_id
501	Rishit	Sarang	401
502	Ali	Akbar	402
503	Pruthvi	solanki	403
504	Yash	Makwana	404
505	Bhavik	Tripathi	405
506	Rahul	Solanki	406
507	Ritesh	Sharma	407
508	Riya	Sharma	408
509	Priya	Matter	409
510	Sonu	Sarang	410

```
sqlite>
```

(H) Write a query to display the job title and average salary of employees.

```
sqlite> SELECT job_title, AVG(salary)
   ...> FROM employees
   ...> NATURAL JOIN jobs
   ...> GROUP BY job_title;
```

job_title	AVG(salary)
assistant_director	31000.0
assistant_manager	30000.0
clerk	33000.0
departmental_executive	36000.0
head_salesman	30000.0
officer	34000.0
president	40000.0
project_executive	33000.0
project_manager	31000.0
sales_manager	35000.0

```
sqlite>
```

(I) Write a query to find the addresses (location_id, street_address, city, state_province, country_name) of all the departments

```
C:\Users\pratham sarang\AppData\Local\Temp\Temp6_sqlite-tools-win32-x86-3350500.zip\sqlite-tools-win32-x86-3350500\sqlite3.exe
sqlite> select location_id,street_address,postal_code,city,state_province,country_name
   ...> from locations
   ...> natural join countries;
```

location_id	street_address	postal_code	city	state_province	country_name
202	Dineper River	3451236	Kyiv	oblast	ukraine
201	Dhaka Tower St	4567856	Madian	Dhaka	Bangladesh
203	Andrew Street	9485343	Bruges	west Flanders	Belgium
204	Football St	9423567	Braflia	Central west	Bulgaria
205	Marshall St	4567865	Sofia	Sofia Province	Canada
206	Cario St	3452532	Toronto	Ontario	Brazil
207	Andheri	3567935	Mumbai	Maharastra	India
208	NYC St	5644325	New York City	Queens	United States
209	Mohammed-ul-arham St	5364521	Kuwait City	Kuwait	Kuwait
210	Link Road	4536754	Addu	Maldives	Maldives

```
sqlite>
```

Q-2:-

Write a command to Dump entire database with proper file name and tables structure into a file named as your rollno.

```
SQLite version 3.35.5 2021-04-19 18:32:05
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .open temp.db
sqlite> .ouput 41_SYBCA_A.sql
Error: unknown command or invalid arguments:  "ouput". Enter ".help" for help
sqlite> .output 41_SYBCA_A.sql
sqlite> .schema
sqlite> .dump
sqlite>
```

Name: Bhavik Agrawal Div : A Roll No:41

Q-3:-

Write a trigger called AGECHECK on table employees that don't allow the insertion or update of any record that has an age less than 18.

INSERT TRIGGER

Command Prompt - sqlite3

```
sqlite>
sqlite> create trigger AGECHECK before insert on employees
   ...> BEGIN
   ...> SELECT
   ...> CASE
   ...> WHEN NEW.age<18
   ...> THEN RAISE (Abort,'invalid age')
   ...> END;
   ...> END;
sqlite> insert into employees values(511,'Ali','Akbar',21,'ali@gmail.com',988422475,'23-07-1999',601,30000,'10%',401,301);
sqlite> select * from employees;
```

emp_id	first_name	last_name	age	email	phone_number	hire_date	job_id	salary	commision_pct	manager_id	dept_id
501	Ali	Akbar	24	mukesh@gmail.com	988422475	23-07-1999	601	30000	10%	401	301
502	Yash	Makwana	21	yash@gmail.com	928442341	28-08-1998	602	31000	11%	402	302
503	Pruthvi	Solanki	22	pruthvi@gmail.com	948426462	29-09-1996	603	40000	12%	403	303
504	Akshay	Dhanecha	24	akshay@gmail.com	958426543	21-01-1997	604	33000	13%	404	304
505	Mukesh	Sharma	26	mukesh@gmail.com	968427764	22-02-1998	605	34000	14%	405	305
506	Firoz	Khan	25	firoz@gmail.com	988426435	23-03-1999	606	35000	10%	406	306
507	Jash	Mori	24	jash@gmail.com	998425666	24-04-1994	607	36000	11%	407	307
508	Rakesh	Yadav	24	rakesh@gmail.com	968424367	25-05-1995	608	31000	12%	408	308
509	Amit	Gupta	23	amit@gmail.com	948427648	26-06-1992	609	30000	13%	409	303
510	Rahul	Tripathi	29	rahul@gmail.com	938424869	10-10-1991	610	33000	14%	410	310
511	Ali	Akbar	21	ali@gmail.com	988422475	23-07-1999	601	30000	10%	401	301

```
sqlite>
sqlite> insert into employees values(512,'Ali','Akbar',17,'ali@gmail.com',988422475,'23-07-1999',601,30000,'10%',401,301);
Runtime error: invalid age (19)
sqlite>
```

UPDATE TRIGGER

Command Prompt - sqlite3

```
sqlite>
sqlite> CREATE TRIGGER AGE_CHECK AFTER UPDATE ON employees
   ...> BEGIN
   ...> SELECT CASE
   ...> WHEN (SELECT age FROM employees WHERE age<18)
   ...> THEN RAISE(ABORT, 'INVALID AGE')
   ...> END;
   ...> END;
sqlite> update employees set age = 28 where emp_id=501;
sqlite> select * from employees;
```

emp_id	first_name	last_name	age	email	phone_number	hire_date	job_id	salary	commision_pct	manager_id	dept_id
501	Ali	Akbar	28	mukesh@gmail.com	988422475	23-07-1999	601	30000	10%	401	301
502	Yash	Makwana	21	yash@gmail.com	928442341	28-08-1998	602	31000	11%	402	302
503	Pruthvi	Solanki	22	pruthvi@gmail.com	948426462	29-09-1996	603	40000	12%	403	303
504	Akshay	Dhanecha	24	akshay@gmail.com	958426543	21-01-1997	604	33000	13%	404	304
505	Mukesh	Sharma	26	mukesh@gmail.com	968427764	22-02-1998	605	34000	14%	405	305
506	Firoz	Khan	25	firoz@gmail.com	988426435	23-03-1999	606	35000	10%	406	306
507	Jash	Mori	24	jash@gmail.com	998425666	24-04-1994	607	36000	11%	407	307
508	Rakesh	Yadav	24	rakesh@gmail.com	968424367	25-05-1995	608	31000	12%	408	308
509	Amit	Gupta	23	amit@gmail.com	948427648	26-06-1992	609	30000	13%	409	303
510	Rahul	Tripathi	29	rahul@gmail.com	938424869	10-10-1991	610	33000	14%	410	310

```
sqlite> update employees set age = 17 where emp_id=501;
Runtime error: INVALID AGE (19)
sqlite>
```

Q-4:-

Write a python program to find mean, median, mode from set of numbers in a list.

main.py

```python
print("THE MEAN, MEDIAN AND MODE OF [1, 1, 2, 4, 7]\n")
# CALCULATE MEAN
numbers = [1, 1, 2, 4, 7]
n = len(numbers)
get_sum = sum(numbers)
mean = get_sum / n
print("Mean is: " + str(mean))

# CALCULATE MEDIAN
numbers = [1, 1, 2, 4, 7]
n = len(numbers)
numbers.sort()

if n % 2 == 0:
    median1 = numbers[n//2]
    median2 = numbers[n//2 - 1]
    median = (median1 + median2)/2
else:
    median = numbers[n//2]
print("Median is: " + str(median))

# CALCULATE MODE
numbers=[1, 1, 2, 4, 7]
mode=max (numbers, key=numbers.count)
print("Mode is :"+str(mode))
```

OUTPUT:

```
THE MEAN, MEDIAN AND MODE OF [1, 1, 2, 4, 7]

Mean is: 3.0
Median is: 2
Mode is :1
>
```

Name: Bhavik Agrawal Div : A Roll No:41

Q-5

Write a python program to retrieve all rows from employee table and display the column values in tabular format.

```python
import sqlite3
import pandas as pd
conn=sqlite3.connect('ali_db1.db')
print("connection established...")

cursor=conn.cursor()


sql="CREATE TABLE employees(emp_id primary key,first_name,last_name,age,email,phone_number,hire_date,job_id,salary,commision_pct,manager_id,dept_id)"
cursor.execute(sql)
print("table is created...")

cursor.execute("INSERT INTO employees VALUES(501,'Ali','Akbar',28,'mukesh@gmail.com',988422475,'23-07-1999',601,30000,'10%',401,301)")
cursor.execute("INSERT INTO employees VALUES(502,'Yash','Makwana',21,'yash@gmail.com',928442341,'28-08-1998',602,31000,'11%',402,302)")
cursor.execute("INSERT INTO employees VALUES(503,'Pruthvi','Solanki',22,'pruthvi@gmail.com',948426462,'29-09-1996',603,40000,'12%',403,303)")
cursor.execute("INSERT INTO employees VALUES(504,'Akshay','Dhanecha',24,'akshay@gmail.com',958426543,'21-01-1997',604,33000,'13%',404,304)")
cursor.execute("INSERT INTO employees VALUES(505,'Mukesh','Sharma',26,'mukesh@gmail.com',968427764,'22-02-1998',605,34000,'14%',405,305)")
cursor.execute("INSERT INTO employees VALUES(506,'Firoz','Khan',25,'firoz@gmail.com',988426435,'23-03-1999',606,35000,'10%',406,306)")
cursor.execute("INSERT INTO employees VALUES(507,'Jash','Mori',24,'jash@gmail.com',998425666,'24-04-1994',607,36000,'11%',407,307)")
cursor.execute("INSERT INTO employees VALUES(508,'Rakesh','Yadav',24,'rakesh@gmail.com',968424367,'25-05-1995',608,31000,'12%',408,308)")
cursor.execute("INSERT INTO employees VALUES(509,'Amit','Gupta',23,'amit@gmail.com',948427648,'26-06-1992',609,30000,'13%',409,'303')")
cursor.execute("INSERT INTO employees VALUES(510,'Rahul','Tripathi',29,'rahul@gmail.com',938424869,'10-10-1991',610,33000,'14%',410,310)")
print("record inserted...")

cursor.execute("select * from employees")
result=cursor.fetchall()

df=pd.DataFrame(result,columns=
["emp_id","first_name","last_name","age","email","phone_number","hire_date","job_id","salary","commision_pct","manager_id","dept_id"])
print(df)
conn.commit()
conn.close()
```

OUTPUT:

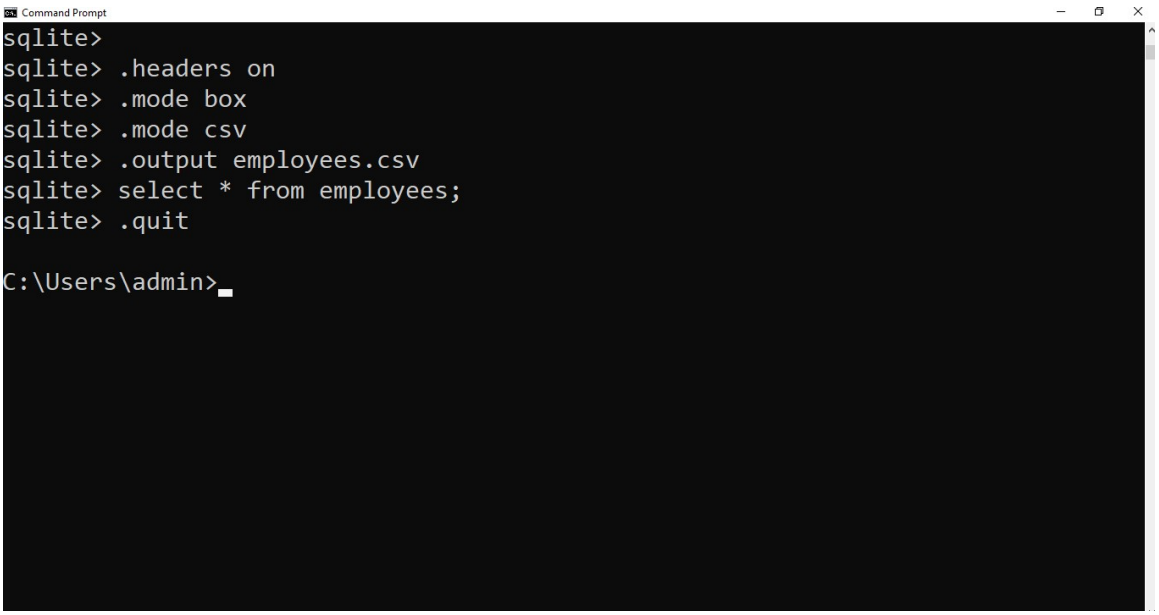
```
connection established...
table is created...
record inserted...
   emp_id first_name last_name  age              email  phone_number  \
0     501        Ali     Akbar   28   mukesh@gmail.com     988422475
1     502       Yash   Makwana   21      yash@gmail.com     928442341
2     503    Pruthvi   Solanki   22  pruthvi@gmail.com     948426462
3     504     Akshay  Dhanecha   24   akshay@gmail.com     958426543
4     505     Mukesh    Sharma   26   mukesh@gmail.com     968427764
5     506      Firoz      Khan   25    firoz@gmail.com     988426435
6     507       Jash      Mori   24     jash@gmail.com     998425666
7     508     Rakesh     Yadav   24   rakesh@gmail.com     968424367
8     509       Amit     Gupta   23     amit@gmail.com     948427648
9     510      Rahul  Tripathi   29    rahul@gmail.com     938424869

    hire_date  job_id  salary commision_pct  manager_id dept_id
0  23-07-1999     601   30000          10%         401     301
1  28-08-1998     602   31000          11%         402     302
2  29-09-1996     603   40000          12%         403     303
3  21-01-1997     604   33000          13%         404     304
4  22-02-1998     605   34000          14%         405     305
5  23-03-1999     606   35000          10%         406     306
6  24-04-1994     607   36000          11%         407     307
7  25-05-1995     608   31000          12%         408     308
8  26-06-1992     609   30000          13%         409     303
9  10-10-1991     610   33000          14%         410     310
```

Name: Bhavik Agrawal Div : A Roll No:41

Q-6:-

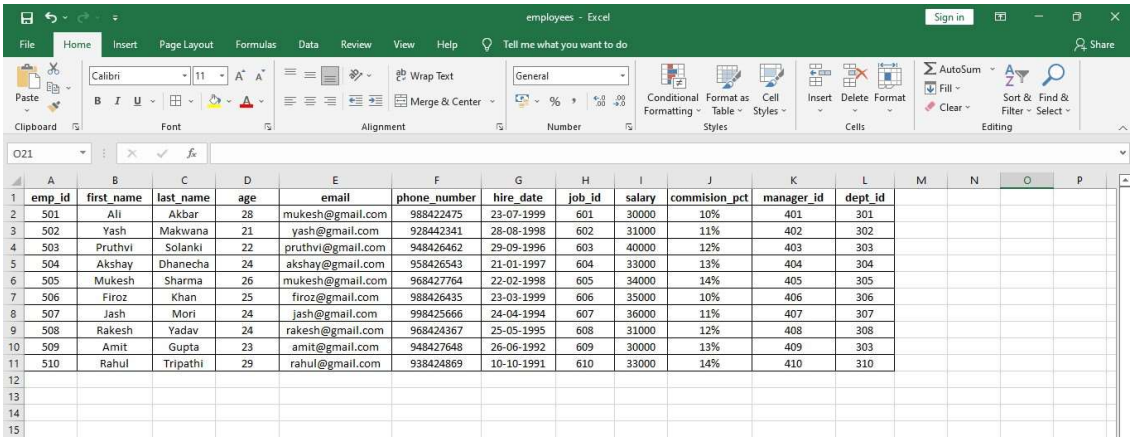
Write a python program to read CSV file and upload data into table

EXPORTING CSV FILE FROM SQLITE DATABASE:



```
Command Prompt
sqlite>
sqlite> .headers on
sqlite> .mode box
sqlite> .mode csv
sqlite> .output employees.csv
sqlite> select * from employees;
sqlite> .quit

C:\Users\admin>
```

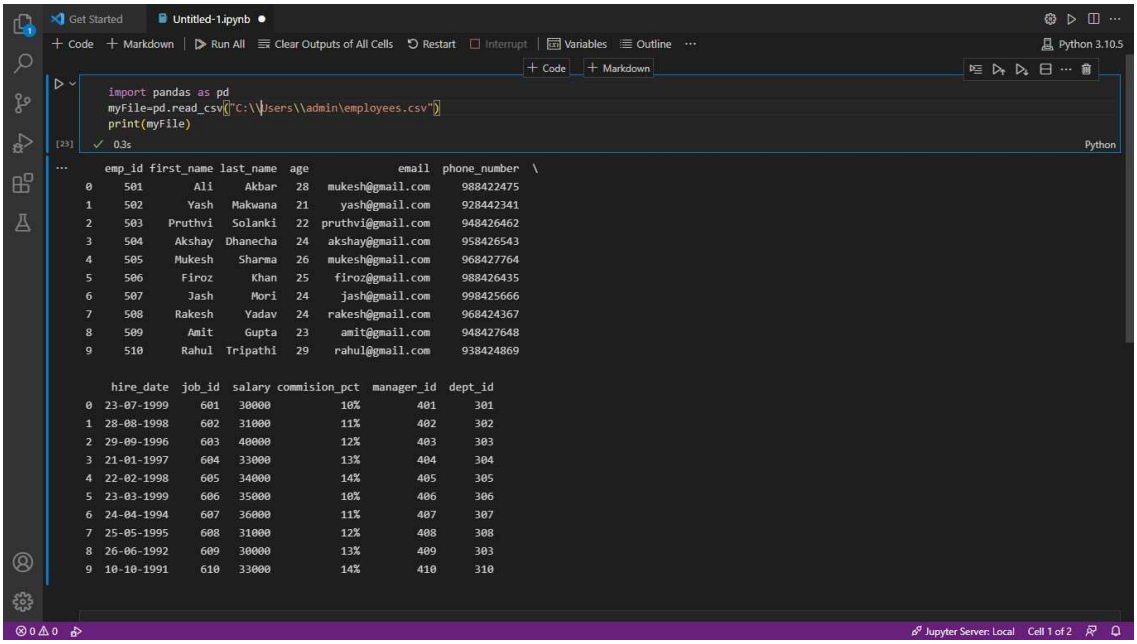
OUTPUT OF CSV FILE:



emp_id	first_name	last_name	age	email	phone_number	hire_date	job_id	salary	commision_pct	manager_id	dept_id
501	Ali	Akbar	28	mukesh@gmail.com	988422475	23-07-1999	601	30000	10%	401	301
502	Yash	Makwana	21	yash@gmail.com	928442341	28-08-1998	602	31000	11%	402	302
503	Pruthvi	Solanki	22	pruthvi@gmail.com	948426462	29-09-1996	603	40000	12%	403	303
504	Akshay	Dhanecha	24	akshay@gmail.com	958426543	21-01-1997	604	33000	13%	404	304
505	Mukesh	Sharma	26	mukesh@gmail.com	968427764	22-02-1998	605	34000	14%	405	305
506	Firoz	Khan	25	firoz@gmail.com	988426435	23-03-1999	606	35000	10%	406	306
507	Jash	Mori	24	jash@gmail.com	998425666	24-04-1994	607	36000	11%	407	307
508	Rakesh	Yadav	24	rakesh@gmail.com	968424367	25-05-1995	608	31000	12%	408	308
509	Amit	Gupta	23	amit@gmail.com	948427648	26-06-1992	609	30000	13%	409	303
510	Rahul	Tripathi	29	rahul@gmail.com	938424869	10-10-1991	610	33000	14%	410	310

Name: Bhavik Agrawal Div : A Roll No:41

READING THAT CSV FILE USING PYTHON:



Q-7:-

Write a python program to retrieve all rows from employee table and dump into ‘employee_details.csv’ CSV file.

```
Id	Name	Salary

1	vivek	30000

2	samrat	40000

3	smit	30000

4	hardik	50000

5	krish	60000

6	meet	60000

7	urvish	60000

8	kenil	60000

9	harsh	60000
```

```
In [2]: import sqlite3 as sql
        import os
        import csv
        from sqlite3 import Error

        try:

          # Connect to database
          conn=sql.connect('emp.db')

          # Create Table Into database
          conn.execute('''CREATE TABLE IF NOT EXISTS Employee(Id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\
                       Name TEXT NOT NULL, Salary INT NOT NULL
                      );''')
          # Insert some values to database
          conn.execute('''INSERT INTO Employee(Name, Salary) VALUES('vivek', 30000);''')
          conn.execute('''INSERT INTO Employee(Name, Salary) VALUES('samrat', 40000);''')
          conn.execute('''INSERT INTO Employee(Name, Salary) VALUES('smit', 30000);''')
          conn.execute('''INSERT INTO Employee(Name, Salary) VALUES('hardik', 50000);''')
          conn.execute('''INSERT INTO Employee(Name, Salary) VALUES('krish', 60000);''')
          conn.execute('''INSERT INTO Employee(Name, Salary) VALUES('meet', 60000);''')
          conn.execute('''INSERT INTO Employee(Name, Salary) VALUES('urvish', 60000);''')
          conn.execute('''INSERT INTO Employee(Name, Salary) VALUES('kenil', 60000);''')
          conn.execute('''INSERT INTO Employee(Name, Salary) VALUES('harsh', 60000);''')
          conn.commit()

         # To view table data in table format
          print ("******Employee Table Data*******")
          cur = conn.cursor()
          cur.execute('''SELECT * FROM Employee''')
          rows = cur.fetchall()

          for row in rows:
              print(row)

         # Export data into CSV file
          print ("Exporting data into CSV.............")
          cursor = conn.cursor()
          cursor.execute("select * from Employee")
          with open("myfile.csv", "w") as csv_file:
              csv_writer = csv.writer(csv_file, delimiter="\t")
              csv_writer.writerow([i[0] for i in cursor.description])
              csv_writer.writerows(cursor)

          dirpath = os.getcwd() + "/myfile.csv"
          print ("Data exported Successfully into {}".format(dirpath))

        except Error as e:
          print(e)

        # Close database connection
        finally:
          conn.close()

        ******Employee Table Data*******
        (1, 'vivek', 30000)
        (2, 'samrat', 40000)
        (3, 'smit', 30000)
        (4, 'hardik', 50000)
        (5, 'krish', 60000)
        (6, 'meet', 60000)
        (7, 'urvish', 60000)
        (8, 'kenil', 60000)
        (9, 'harsh', 60000)
        Exporting data into CSV.............
        Data exported Successfully into C:\Users\UCC\Documents\samrat/myFile.csv
```

Name: Bhavik Agrawal Div : A Roll No:41

Q-8:-

Write a program to implement DML operations using sqlite3.

CREATION OF TABLE:-

```
import sqlite3
conn = sqlite3.connect('ali.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
Table created successfully
```

1. INSERTION:-

```
import sqlite3
conn = sqlite3.connect('ali.db')
print ("Opened database successfully")
conn.execute("INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) \
 VALUES (1, 'Ali', 32, 'Surat', 20000.00 )")
conn.execute("INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) \
 VALUES (2, 'Yash', 25, 'Bhopal', 15000.00 )")
conn.execute("INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) \
 VALUES (3, 'PruthviRaj', 23, 'Mumbai', 20000.00 )")
conn.execute("INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) \
 VALUES (4, 'Dhruv', 25, 'Mirzapur', 65000.00 )")
conn.commit()
print ("Records created successfully")
conn.close()
```

```
Opened database successfully
Records created successfully
```

Name: Bhavik Agrawal Div : A Roll No:41

2. SELECTION

```python
import sqlite3
conn = sqlite3.connect('ali.db')
print ("Opened database 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])
    print("------------------------")
```

```
Opened database successfully
ID =  1
NAME =  Ali
ADDRESS =  Surat
SALARY =  20000.0
------------------------
ID =  2
NAME =  Yash
ADDRESS =  Bhopal
SALARY =  15000.0
------------------------
ID =  3
NAME =  PruthviRaj
ADDRESS =  Mumbai
SALARY =  20000.0
------------------------
ID =  4
NAME =  Dhruv
ADDRESS =  Mirzapur
SALARY =  65000.0
```

3. UPDATION

```python
import sqlite3
conn = sqlite3.connect('ali.db')
print ("Opened database successfully")
conn.execute("UPDATE COMPANY set SALARY = 25000.00 where ID = 1")
conn.commit()
print ("Total number of rows updated :", conn.total_changes)
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])
    print("------------------------")
```

```
Opened database successfully
Total number of rows updated : 1
ID =  1
NAME =  Ali
ADDRESS =  Surat
SALARY =  25000.0
------------------------
ID =  2
NAME =  Yash
ADDRESS =  Bhopal
SALARY =  15000.0
------------------------
ID =  3
NAME =  PruthviRaj
ADDRESS =  Mumbai
SALARY =  20000.0
------------------------
ID =  4
NAME =  Dhruv
ADDRESS =  Mirzapur
SALARY =  65000.0
```

Name: Bhavik Agrawal Div : A Roll No:41

4. DELETE

```python
import sqlite3
conn = sqlite3.connect('ali.db')
print ("Opened database successfully")
conn.execute("DELETE from COMPANY where ID = 4;")
conn.commit()
print ("Total number of rows deleted :", conn.total_changes)
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])
 print("-------------------------")
print("RECORD DELETED SUCCESSFULLY")
```

```
Opened database successfully
Total number of rows deleted : 0
ID =  1
NAME =  Ali
ADDRESS =  Surat
SALARY =  25000.0
-------------------------
ID =  2
NAME =  Yash
ADDRESS =  Bhopal
SALARY =  15000.0
-------------------------
ID =  3
NAME =  PruthviRaj
ADDRESS =  Mumbai
SALARY =  20000.0
-------------------------
RECORD DELETED SUCCESSFULLY
```

Q-9:-

Get total salary from employees table and show line plot with the following Style properties Generated line plot must include following Style properties: -

- Line Style dotted and Line-color should be red
- Show legend at the lower right location.
- X label name = salary .
- Y label name = Employee name
- Add a circle marker.
- Line marker color as red
- Line width should be 3

```
E_name,T_salary
vivek,55000
krish,45000
meet,50000
smit,60000
hardik,25500
sem,75000
urvish,45000
kenil,30000
harsh,20000
rahul,70000
```

In [14]:
```
import csv
import pandas as pd
import matplotlib.pyplot as plt

with open('employee_details.csv','r') as File:
    reader = csv.reader(File, delimiter=',')
    for row in reader:
        print(row)

df = pd.read_csv("employee_details.csv")
employeeList = df ['E_name'].tolist()
salaryList  = df ['T_salary'].tolist()

plt.plot(employeeList,salaryList, label = 'Salary data of last year',
      color='r', marker='o', markerfacecolor='k',
      linestyle='--', linewidth=3)

plt.xlabel('EMPLOYEE NAME')
plt.ylabel('SALARY OF EMPLOYEE')
plt.legend(loc='lower right')
plt.title('Employee Salary data of last year')
plt.xticks(employeeList)
plt.yticks([10000, 20000, 30000, 40000, 50000,60000,70000,80000])
plt.show()
```

```
['E_name', 'T_salary']
['vivek', '55000']
['krish', '45000']
['meet', '50000']
['smit', '60000']
['hardik', '25500']
['sem', '75000']
['urvish', '45000']
['kenil', '30000']
['harsh', '20000']
['rahul', '70000']
```

Q-10:-

Use employee_details.csv file and read salary and commission_pct data and show it using the bar chart The bar chart should display the number of units for each employee. Add a separate bar for each first name in the same chart.

```
month_number,salary,commission
1,14000,7000
2,15000,5000
3,7000,6000
4,12000,4000
5,8000,5500
6,17500,6500
7,14000,8000
8,16000,2000
9,10000,3000
10,12000,9000
11,13000,4500
12,19000,7500
```

```
In [19]: import pandas as pd
         import matplotlib.pyplot as plt

         df = pd.read_csv(" employee_details.csv")
         monthList  = df ['month_number'].tolist()
         salaryData   = df ['salary'].tolist()
         commissionData   = df ['commission'].tolist()

         plt.bar([a-0.25 for a in monthList], salaryData, width= 0.20,color='r', label = 'salary data', align='edge')
         plt.bar([a+0.25 for a in monthList], commissionData, width= -0.20, label = 'commission data', align='edge')
         plt.xlabel('Month Number')
         plt.ylabel('salary and commission in number')
         plt.legend(loc='upper left')
         plt.title(' Sales data')

         plt.xticks(monthList)
         plt.grid(True, linewidth= 1, linestyle="--")
         plt.title('salary and commission data')
         plt.show()
```