SDJ INTERNATIONAL COLLEGE

Name :- VINAYAK PUROHIT
Class :- SYBCA (sem 3)
Roll No :- G-772

1 . Create Student Table with appropriate constraints. STUDENT(sno number primary key, sname text(20), age number, total_marks number) write python programs to perform following task: 1) store the table data into a dataframe and display the dataframe. 2) List out top three records from the dataframe 3) Display all records from dataframe whose age is not less than 18. 4) Display age of student whose sno is 5. (use loc() and iloc() function)

< SQLite Code >

```
sqlite> create table STUDENT(
(x1...> sno number primary key,
(x1...> sname text(20) not null,
(x1...> age number,
(x1...> total_marks number);

sqlite> insert into STUDENT values(101,'vinayak',18,570);
sqlite> insert into STUDENT values(102,'john',19,490);
sqlite> insert into STUDENT values(103,'alex',17,530);
sqlite> insert into STUDENT values(104,'micheal',19,550);

sqlite> select * from Student;
```

sno	sname	age	total_marks
101	vinayak	18	570
102	john	19	490
103	alex	17	530
104	micheal	19	550
5	sirdharth	12	580

Name :- VINAYAK PUROHIT
Class :- SYBCA (sem 3)
Roll No :- G-772

<PYTHON CODE>

```
import sqlite3

import pandas as pd


try:

    db = sqlite3.connect('jounralsolution')

    db.cursor()

    df = pd.read_sql_query('SELECT * FROM STUDENT', db)


    print("Printing entire dataframe.....:")

    print(df)


    print("\nTask 2: Top three records:")

    top_three=df.head(3)

    print(top_three)


    print("\nTask 3: Records with age not less than 18:")

    above_18=df[df['age']>=18]

    print(above_18)


    print("\nTask 4: Displaying age of student whose sno is 5: ")

    print("\nage of student with sno 5 (loc):",df.loc[df['sno']==5 ,'age'].values[0])

    print("\nage of student with sno 5
(iloc):",df.iloc[df.index[df['sno']==5],df.columns.get_loc('age')].values[0])

    db.commit()


except:

    print('ERROR 402: Something went wrong.....')

    db.rollback()
```

Name :- VINAYAK PUROHIT
Class :- SYBCA (sem 3)
Roll No :- G-772

```
= RESTART: C:\sqlite\Q1.py
Printing entire dataframe........:
   sno       sname  age  total_marks
0  101     vinayak   18          570
1  102        john   19          490
2  103        alex   17          530
3  104     micheal   19          550
4    5  sirdharth   12          580

Task 2: Top three records:
   sno     sname  age  total_marks
0  101   vinayak   18          570
1  102      john   19          490
2  103      alex   17          530

Task 3: Records with age not less than 18:
   sno     sname  age  total_marks
0  101   vinayak   18          570
1  102      john   19          490
3  104   micheal   19          550

Task 4: Displaying age of student whose sno is 5:

age of student with sno 5 (loc): 12

age of student with sno 5 (iloc): 12
```

Name :- VINAYAK PUROHIT
Class :- SYBCA (sem 3)
Roll No :- G-772

Q2. Create following table and store any five records: Employee(eno number primary key, Ename text(20),designation text(10),basic number , da number, gross_salary number) write python programs to perform following tasks:

1) Store the table data into dataframe and display the dataframe. 2) Sort the dataframe based used on gross salary and List out bottom two record from the dataframe. 3) Display all records from dataframe whose gross Display gross salary is more than 25000 . 4) Display gross salary of employee whose eno is 4.

<SQLite Code>

```
sqlite> CREATE TABLE IF NOT EXISTS Employee (
(x1...>     eno INTEGER PRIMARY KEY,
(x1...>     Ename TEXT(20),
(x1...>     designation TEXT(10),
(x1...>     basic REAL,
(x1...>     da REAL,
(x1...>     gross_salary REAL
(x1...> );
sqlite> insert into Employee values (1, 'Vinayak', 'Manager', 50000, 10000, 60000);
sqlite> insert into Employee values (2, 'ketan, 'Engineer', 45000, 9000, 54000);
sqlite> insert into Employee values (3, 'shailesh', 'Developer', 55000, 11000, 66000);
sqlite> insert into Employee values (4, 'siddharth', 'Analyst', 48000, 9600, 57600);
sqlite> insert into Employee values (5, 'Eva Brown', 'Designer', 42000, 8400, 50400);
sqlite> select * from Employee;
```

```
sqlite> select * from Employee;
```

eno	Ename	designation	basic	da	gross_salary
1	Vinayak	Manager	50000.0	10000.0	60000.0
2	ketan	Engineer	45000.0	9000.0	54000.0
3	shailesh	Developer	55000.0	11000.0	66000.0
4	siddharth	Analyst	48000.0	9600.0	57600.0
5	Eva Brown	Designer	42000.0	8400.0	50400.0

Name :- VINAYAK PUROHIT
Class :- SYBCA (sem 3)
Roll No :- G-772

< PYTHON CODE >

```
import sqlite3

import pandas as pd


try:

    db = sqlite3.connect('jounralsolution')

    db.cursor()

    df = pd.read_sql_query('SELECT * FROM Employee', db)


    print("\nPrinting entire dataframe.....:")

    print(df)


    print("\nTask 2:Sorting the dataframe based used on gross salary....")

    df_sorted = df.sort_values(by='gross_salary')

    print(df_sorted)


    print("\nprinting bottom two record from the dataframe.....")

    bottom = df.tail(2)

    print(bottom)


    print("\nTask 3:Displaying all records from dataframe whose gross Display gross salary is more
than 25000")

    sorted_data = df[df['gross_salary']>25000]

    print(sorted_data)

    print("Task 4: Displaying gross salary of employee whose eno is 4...")

    print("\ngross salary of employee with sno 4 (loc):",df.loc[df['eno']==4 ,'gross_salary'].values[0])

    print("\ngross salary of employee with sno 4
(iloc):",df.iloc[df.index[df['eno']==4],df.columns.get_loc('gross_salary')].values[0])

except:
```

```
print('Something went wrong')
```

```
Printing entire dataframe.....:
   eno      Ename designation    basic       da  gross_salary
0    1    Vinayak     Manager  50000.0  10000.0       60000.0
1    2      ketan    Engineer  45000.0   9000.0       54000.0
2    3   shailesh   Developer  55000.0  11000.0       66000.0
3    4  siddharth     Analyst  48000.0   9600.0       57600.0
4    5  Eva Brown    Designer  42000.0   8400.0       50400.0

Task 2:Sorting the dataframe based used on gross salary....
   eno      Ename designation    basic       da  gross_salary
4    5  Eva Brown    Designer  42000.0   8400.0       50400.0
1    2      ketan    Engineer  45000.0   9000.0       54000.0
3    4  siddharth     Analyst  48000.0   9600.0       57600.0
0    1    Vinayak     Manager  50000.0  10000.0       60000.0
2    3   shailesh   Developer  55000.0  11000.0       66000.0

printing bottom two record from the dataframe.....
   eno      Ename designation    basic      da  gross_salary
3    4  siddharth     Analyst  48000.0  9600.0       57600.0
4    5  Eva Brown    Designer  42000.0  8400.0       50400.0

Task 3:Displaying all records from dataframe whose gross Display gross salary is more than 25000
   eno      Ename designation    basic       da  gross_salary
0    1    Vinayak     Manager  50000.0  10000.0       60000.0
1    2      ketan    Engineer  45000.0   9000.0       54000.0
2    3   shailesh   Developer  55000.0  11000.0       66000.0
3    4  siddharth     Analyst  48000.0   9600.0       57600.0
4    5  Eva Brown    Designer  42000.0   8400.0       50400.0
Task 4: Displaying gross salary of employee whose eno is 4...

gross salary of employee with sno 4 (loc): 57600.0

gross salary of employee with sno 4 (iloc): 57600.0
```

Name :- VINAYAK PUROHIT
Class :- SYBCA (sem 3)
Roll No :- G-772

Q3. Create CSV file for product selling for 6 months and add only 5 record for 5 different product

Prod_name JAN FEB MAR APR MAY JUN

Create Python script for following program:

1.Read data into DataFrame 2) Add columns and calculate total_sell, average_sell 3) Plot Total sell and average sell together on Line chart with proper legends, Titles and Lables. 4) Save the DataFrame to CSV named 'sell_analysis.csv'

< SQLiteCode >

```
sqlite> CREATE TABLE IF NOT EXISTS sales (
(x1...>     sid INTEGER PRIMARY KEY AUTOINCREMENT,
(x1...>     year INTEGER NOT NULL,
(x1...>     totalsales REAL,
(x1...>     CONSTRAINT year_check CHECK (year >= 0),
(x1...>     CONSTRAINT totalsales_check CHECK (totalsales >= 0)
(x1...> );
sqlite> insert into sales values(1,2018,53000.33);
sqlite> insert into sales values(2,2019,24000.56);
sqlite> insert into sales values(3,2020,34200.42);
sqlite> insert into sales values(4,2021,31220.23);
sqlite> insert into sales values(5,2022,50220.87);
sqlite> .mode box
sqlite> select * from sales;
```

```
sqlite> select * from Sales;
┌─────┬──────┬─────────────┐
│ sid │ year │ totalsales  │
├─────┼──────┼─────────────┤
│ 1   │ 2018 │ 53000.0     │
│ 2   │ 2019 │ 24000.56    │
│ 3   │ 2020 │ 34200.42    │
│ 4   │ 2021 │ 31220.23    │
│ 5   │ 2022 │ 50220.87    │
└─────┴──────┴─────────────┘
```

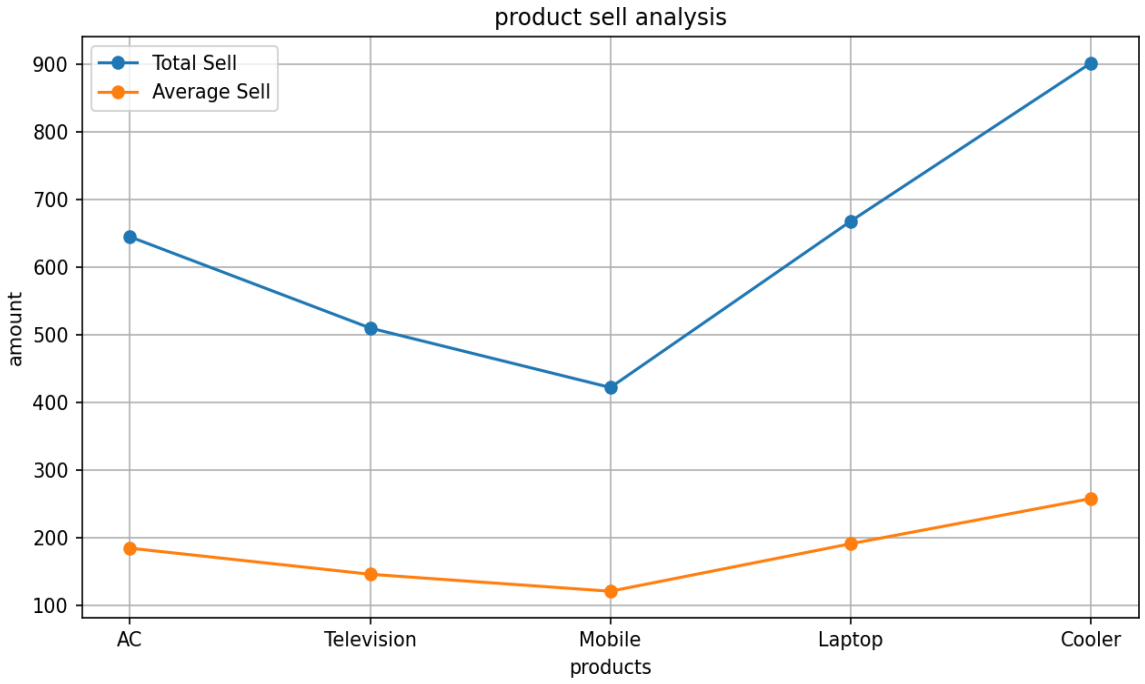
Name :- VINAYAK PUROHIT
Class :- SYBCA (sem 3)
Roll No :- G-772

< PYTHON CODE >

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv('data.csv')
print(df)

print("\nTask 2: AddING columns and calculate total_sell and average_sell...")
df['total_sell']=df.iloc[:,1:].sum(axis=1)
df['average_sell']=df.iloc[:,1:].mean(axis=1)
print(df)

print("\nTask 3: printting linediagram.....")
plt.figure(figsize=(10,6))
plt.plot(df['prod_name'],df['total_sell'],label='Total Sell',marker='o')
plt.plot(df['prod_name'],df['average_sell'],label='Average Sell',marker='o')
plt.xlabel('products')
plt.ylabel('amount')
plt.title('product sell analysis')
plt.legend()
plt.grid(True)
plt.show()

print("\nTask 4: DataFrame to CSV")
df.to_csv('sell_analysis.csv',index=False)
print("CSV file created successfully.....")
```

Name :- VINAYAK PUROHIT
Class :- SYBCA (sem 3)
Roll No :- G-772

```
= RESTART: C:\sqlite\Q3.py
    prod_name   JAN   FEB   MAR   APR   MAY   JUN
0          AC   100   120    90   110   130    95
1  Television    80    85    75    90    88    92
2      Mobile    60    70    65    75    72    80
3      Laptop   110   105   100   115   120   118
4      Cooler   150   140   160   145   155   152

Task 2: AddING columns and calculate total_sell and average_sell...
    prod_name   JAN   FEB   MAR   APR   MAY   JUN  total_sell  average_sell
0          AC   100   120    90   110   130    95         645    184.285714
1  Television    80    85    75    90    88    92         510    145.714286
2      Mobile    60    70    65    75    72    80         422    120.571429
3      Laptop   110   105   100   115   120   118         668    190.857143
4      Cooler   150   140   160   145   155   152         902    257.714286

Task 3: printting linediagram......

Task 4: DataFrame to CSV
CSV file created successfully......
```



Name :- VINAYAK PUROHIT
Class :- SYBCA (sem 3)
Roll No :- G-772

Q4. Write a phython script to do following on student (Rollno, Name, Sub 1, Sub 2, Sub 3, total) table:

1) Insert atleast 5 to 10 records. 2) Update the specific record value. 3) Delete the record specific record. 4) Display student detail who got highest total marks

---

< PYTHON CODE >

```
import sqlite3
db = sqlite3.connect('jounralsolution')
try:
   cur=db.cursor()
   cur.execute("drop table if exists student1;")
   cur.execute("""CREATE TABLE IF NOT EXISTS Student1 (
               Rollno INTEGER PRIMARY KEY,
               Name TEXT,
               Sub1 INTEGER,
               Sub2 INTEGER,
               Sub3 INTEGER,
               Total INTEGER
               );""")
   print('Table created successfully.....')
```

```
======================== RESTART: C:\sqlite\Q4 - Copy.py ========================
Table created successfully......
```

```
   data = [(1, 'Vinayak', 85, 90, 78, 253),
      (2, 'Virat', 92, 88, 76, 256),
      (3, 'Mahendra', 78, 85, 80, 243),
      (4, 'Rohit', 90, 92, 85, 267),
```

```
    (5, 'Aakash', 88, 86, 94, 0)]

cur.executemany("""
    INSERT INTO Student1 (Rollno, Name, Sub1, Sub2, Sub3, Total)
    VALUES (?, ?, ?, ?, ?, ?)
""", data)

db.commit()
print("Records successfully inserted.....")
```

```
======================== RESTART: C:\sqlite\Q4 - Copy.py ========================
Records successfully inserted.....
```

```
sqlite> select * from Student1;
```

Rollno	Name	Sub1	Sub2	Sub3	Total
1	Vinayak	85	90	78	253
2	Virat	92	88	76	256
3	Mahendra	78	85	80	243
4	Rohit	90	92	85	267
5	Aakash	88	86	94	0

```
print("\nTask 3: Updating the specific record value")
cur.execute("UPDATE Student1 SET Total = ? WHERE Rollno = ?", ((88 + 86 + 94), 5))
print("successfully updated...")
db.commit()
```

```
= RESTART: C:\sqlite\Q4 - Copy.py

Task 3: Updating the specific record value
successfully updated...
```

```
sqlite> select * from Student1;
```

Rollno	Name	Sub1	Sub2	Sub3	Total
1	Vinayak	85	90	78	253
2	Virat	92	88	76	256
3	Mahendra	78	85	80	243
4	Rohit	90	92	85	267
5	Aakash	88	86	94	268

```
print("\nTask 4: Delete the record specific record")
rno=int(input("Enter the rollno to be deleted: "))
cur.execute("delete from student1 where Rollno = ?",(rno,))
print("record deleted successfully....")
db.commit()
```

```
= RESTART: C:\sqlite\Q4 - Copy.py

Task 4: Delete the record specific record
Enter the rollno to be deleted: 5
record deleted successfully....
```

```
sqlite> select * from Student1;
```

Rollno	Name	Sub1	Sub2	Sub3	Total
1	Vinayak	85	90	78	253
2	Virat	92	88	76	256
3	Mahendra	78	85	80	243
4	Rohit	90	92	85	267

```
    print("\nTask 5: Printing higest numbet std Details.....")
    cur.execute("SELECT * FROM Student1 WHERE Total = (SELECT MAX(Total) FROM
Student1);")
    maxnum=cur.fetchone()
    print(maxnum)
except:
    print("Something went wrong.....")
```

```
= RESTART: C:\sqlite\Q4 - Copy.py

Task 5: Printing higest numbet std Details.....
(4, 'Rohit', 90, 92, 85, 267)
```

Name :- VINAYAK PUROHIT
Class :- SYBCA (sem 3)
Roll No :- G-772

Q5. Write Python Script to do followings on item.csv (Item_no, Item_name, Price, Qty, total)

1) Write item's detail in the item.csv file. Calculate total = price * Qty 2) Using data frame display item name and price whose price is between 1000 to 5000. 3) Display alternate records from item.csv file. 4) Display items whose price is minimum, maximum. 5) Sort the data according to item name wise. 6) Display items rows between 3th to 7th row. 7) Display last 6 rows

---

< PYTHON CODE >

```
import pandas as pd

print("\nTask 1: Write item's detail in the item.csv file and calculate total")

data = {

   'Item_no': [1, 2, 3, 4, 5],

   'Item_name': ['Laptop', 'Blender', 'Scooter', 'TV', 'Freedge'],

   'Price': [1500, 3000, 2500, 4500, 2000],

   'Qty': [2, 3, 4, 1, 5]

   }

df = pd.DataFrame(data)

df['Total'] = df['Price'] * df['Qty']

df.to_csv('item.csv', index=False)
```

	A	B	C	D	E
1	Item_no	Item_nam	Price	Qty	Total
2	1	Laptop	1500	2	3000
3	2	Blender	3000	3	9000
4	3	Scooter	2500	4	10000
5	4	TV	4500	1	4500
6	5	Freedge	2000	5	10000
7					

```
print("\nTask 2: Display item name and price whose price is between 1000 to 5000")

filtered_df = df[(df['Price'] >= 1000) & (df['Price'] <= 5000)]

print("Items with price between 1000 and 5000:")

print(filtered_df[['Item_name', 'Price']])
```

```
Task 1: Write item's detail in the item.csv file and calculate total

Task 2: Display item name and price whose price is between 1000 to 5000
Items with price between 1000 and 5000:
  Item_name  Price
0    Laptop   1500
1   Blender   3000
2   Scooter   2500
3        TV   4500
4   Freedge   2000
```

```
print("\nTask 3: Display alternate records from item.csv file")

print("\nAlternate records:")

print(df[::2])
```

```
= RESTART: C:\sqlite\Q5.py

Task 3: Display alternate records from item.csv file

Alternate records:
   Item_no Item_name  Price  Qty  Total
0        1    Laptop   1500    2   3000
2        3   Scooter   2500    4  10000
4        5   Freedge   2000    5  10000
```

```
print("\nTask 4: Display items whose price is minimum and maximum")

min_price_item = df[df['Price'] == df['Price'].min()]

max_price_item = df[df['Price'] == df['Price'].max()]

print("\nItems with minimum price:")
```

```
print(min_price_item)

print("\nItems with maximum price:")

print(max_price_item)


print("\nTask 5: Sort the data according to item name wise...")

sorted_df = df.sort_values(by='Item_name')

print(sorted_df)


print("\nTask 6: Display items rows between 3th to 7th row...")

print(df.iloc[2:7])


print("\nTask 7: Display last 6 rows...")

print(df.tail(6))
```

Name :- VINAYAK PUROHIT
Class :- SYBCA (sem 3)
Roll No :- G-772

```
= RESTART: C:\sqlite\Q5.py

Task 4: Display items whose price is minimum and maximum


Items with minimum price:
   Item_no Item_name  Price  Qty  Total
0        1    Laptop   1500    2   3000


Items with maximum price:
   Item_no Item_name  Price  Qty  Total
3        4         TV   4500    1   4500


Task 5: Sort the data according to item name wise...

   Item_no Item_name  Price  Qty  Total
1        2   Blender   3000    3   9000
4        5   Freedge   2000    5  10000
0        1    Laptop   1500    2   3000
2        3   Scooter   2500    4  10000
3        4         TV   4500    1   4500


Task 6: Display items rows between 3th to 7th row...

   Item_no Item_name  Price  Qty  Total
2        3   Scooter   2500    4  10000
3        4         TV   4500    1   4500
4        5   Freedge   2000    5  10000


Task 7: Display last 6 rows...

   Item_no Item_name  Price  Qty  Total
0        1    Laptop   1500    2   3000
1        2   Blender   3000    3   9000
2        3   Scooter   2500    4  10000
3        4         TV   4500    1   4500
4        5   Freedge   2000    5  10000
```

Name :- VINAYAK PUROHIT
Class :- SYBCA (sem 3)
Roll No :- G-772

Q6. Sales (sid, year, totalsales) Create above table into a SQLite database with appropriate constraints. 1) Insert at least 5-10 records into the sales table 2) Export sales table data into sales.csv file. 3) Write a python scripts that read the sales.csv file and plot a bar chart that shows totalsales of the year. Also decorate the chart with appropriate title, lables, colours etc.

< SQLite Code >

```
sqlite> CREATE TABLE sales (
(x1...> sid INTEGER PRIMARY KEY AUTOINCREMENT,
(x1...> year INTEGER NOT NULL,
(x1...> totalsales REAL);


sqlite> INSERT INTO Employee VALUES(1,'Vinayak','Manager',50000.0,10000.0,60000.0);
sqlite> INSERT INTO Employee VALUES(2,'Jane Smith','Engineer',45000.0,9000.0,54000.0);
sqlite> INSERT INTO Employee VALUES(3,'shailesh','Developer',55000.0,11000.0,66000.0);
sqlite> INSERT INTO Employee VALUES(4,'siddharth','Analyst',48000.0,9600.0,57600.0);
sqlite> INSERT INTO Employee VALUES(5,'Eva Brown','Designer',42000.0,8400.0,50400.0);


sqlite> .header on
sqlite> .mode csv
sqlite> .output sales.csv
sqlite> select * from sales;
```

	A	B	C	D
1	sid	year	totalsales	
2	1	2018	53000	
3	2	2019	24000.56	
4	3	2020	34200.42	
5	4	2021	31220.23	
6	5	2022	50220.87	
7				

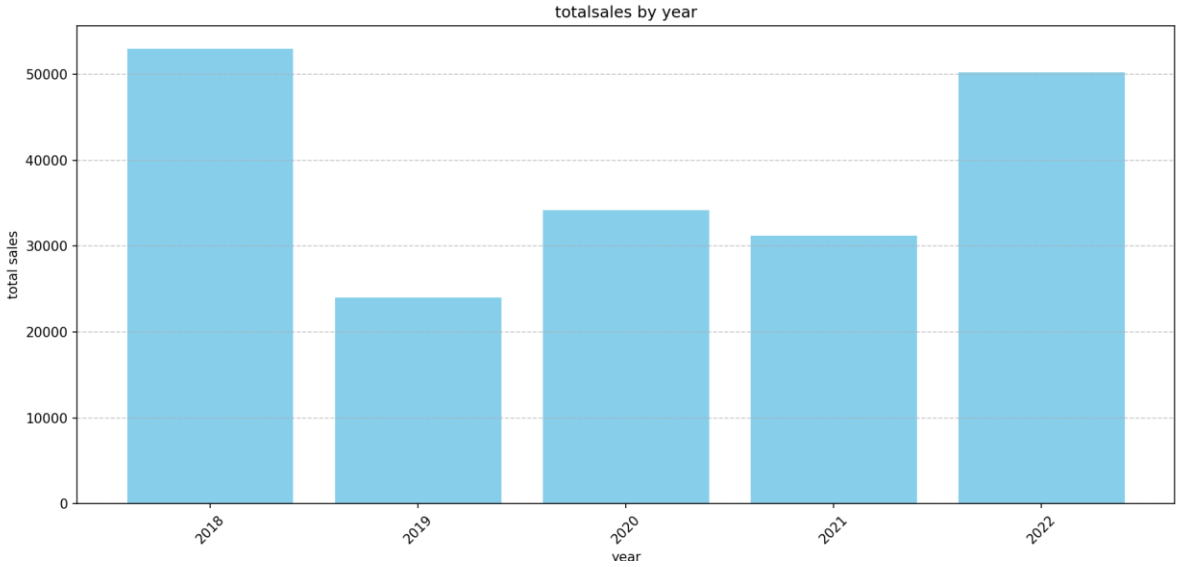
Name :- VINAYAK PUROHIT
Class :- SYBCA (sem 3)
Roll No :- G-772

< PYTHON CODE >

```
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv('sales.csv')
print(df)


plt.figure(figsize=(10,6))
plt.bar(df['year'],df['totalsales'],color='skyblue')
plt.xlabel('year')
plt.ylabel('total sales')
plt.title('totalsales by year')
plt.xticks(rotation=45)
plt.grid(axis='y',linestyle='--',alpha=0.7)
plt.tight_layout()
plt.show()
```

```
= RESTART: C:\sqlite\Q6.py
   sid  year  totalsales
0    1  2018    53000.00
1    2  2019    24000.56
2    3  2020    34200.42
3    4  2021    31220.23
4    5  2022    50220.87
|
```

Name :- VINAYAK PUROHIT
Class :- SYBCA (sem 3)
Roll No :- G-772



Name :- VINAYAK PUROHIT
Class :- SYBCA (sem 3)
Roll No :- G-772

Q7. Create following table with appropriate constraints in Collage Database: Employee (E_ID, Name, Dob, Designation, Salary) a) Dump Employee table structure and data in Emp.csv file. b) Dump whole Database named College in Emp1.csv file.

```
sqlite> CREATE TABLE Employee (
(x1...> E_ID INTEGER PRIMARY KEY,
(x1...> Name TEXT NOT NULL,
(x1...> Dob DATE,
(x1...> Designation TEXT NOT NULL,
(x1...> Salary REAL CHECK (Salary >= 0)
(x1...> );
sqlite> INSERT INTO Employee VALUES(1,'Amrit','1990-01-
15','Manager',60000.0);
sqlite> INSERT INTO Employee VALUES(2,'Suresh','1995-03-
22','Engineer',55000.0);
sqlite> INSERT INTO Employee VALUES(3,'kartik','1987-07-
10','Analyst',50000.0);
sqlite> INSERT INTO Employee VALUES(4,'Ajay','1992-09-
05','Designer',52000.0);
sqlite> INSERT INTO Employee VALUES(5,'Ram','1998-12-
30','Developer',58000.0);
sqlite> INSERT INTO Employee VALUES(6,'Laxman','1994-06-
18','Tester',53000.0);
sqlite> .mode box
sqlite> select * from Employee;
```

```
sqlite> select * from employee;
```

E_ID	Name	Dob	Designation	Salary
1	Amrit	1990-01-15	Manager	60000.0
2	Suresh	1995-03-22	Engineer	55000.0
3	kartik	1987-07-10	Analyst	50000.0
4	Ajay	1992-09-05	Designer	52000.0
5	Ram	1998-12-30	Developer	58000.0
6	Laxman	1994-06-18	Tester	53000.0

sqlite> .mode csv

sqlite> .output Emp.csv

sqlite> .schema Employee

	A	B	C	D	E
1	CREATE TABLE Employee (				
2	E_ID INTEGER PRIMARY KEY				
3	Name TEXT NOT NULL				
4	Dob DATE				
5	Designation TEXT NOT NULL				
6	Salary REAL CHECK (Salary >= 0)				
7	);				
8	1	Amrit	15-01-1990	Manager	60000
9	2	Suresh	22-03-1995	Engineer	55000
10	3	kartik	10-07-1987	Analyst	50000
11	4	Ajay	05-09-1992	Designer	52000
12	5	Ram	30-12-1998	Developer	58000
13	6	Laxman	18-06-1994	Tester	53000
14					

Name :- VINAYAK PUROHIT
Class :- SYBCA (sem 3)
Roll No :- G-772

sqlite> select * from Employee;

sqlite> .output Emp1.csv

sqlite> .dump

	A	B	C	D	E
1	PRAGMA foreign_keys=OFF;				
2	BEGIN TRANSACTION;				
3	CREATE TABLE Employee (				
4	E_ID INTEGER PRIMARY KEY				
5	Name TEXT NOT NULL				
6	Dob DATE				
7	Designation TEXT NOT NULL				
8	Salary REAL CHECK (Salary >= 0)				
9	);				
10	INSERT IN	'Amrit'	'1990-01-1	'Manager'	60000.0);
11	INSERT IN	'Suresh'	'1995-03-2	'Engineer'	55000.0);
12	INSERT IN	'kartik'	'1987-07-1	'Analyst'	50000.0);
13	INSERT IN	'Ajay'	'1992-09-0	'Designer'	52000.0);
14	INSERT IN	'Ram'	'1998-12-3	'Develope	58000.0);
15	INSERT IN	'Laxman'	'1994-06-1	'Tester'	53000.0);
16					
17	COMMIT;				
18					

Name :- VINAYAK PUROHIT
Class :- SYBCA (sem 3)
Roll No :- G-772

8. Create following table with appropriate Constraints: Product (prod_id , prod_name , price, qty,total_amount) 1) Import Product.csv file data into Product table. 2) Export Product table data into prod.csv file.

< SQLite Code >

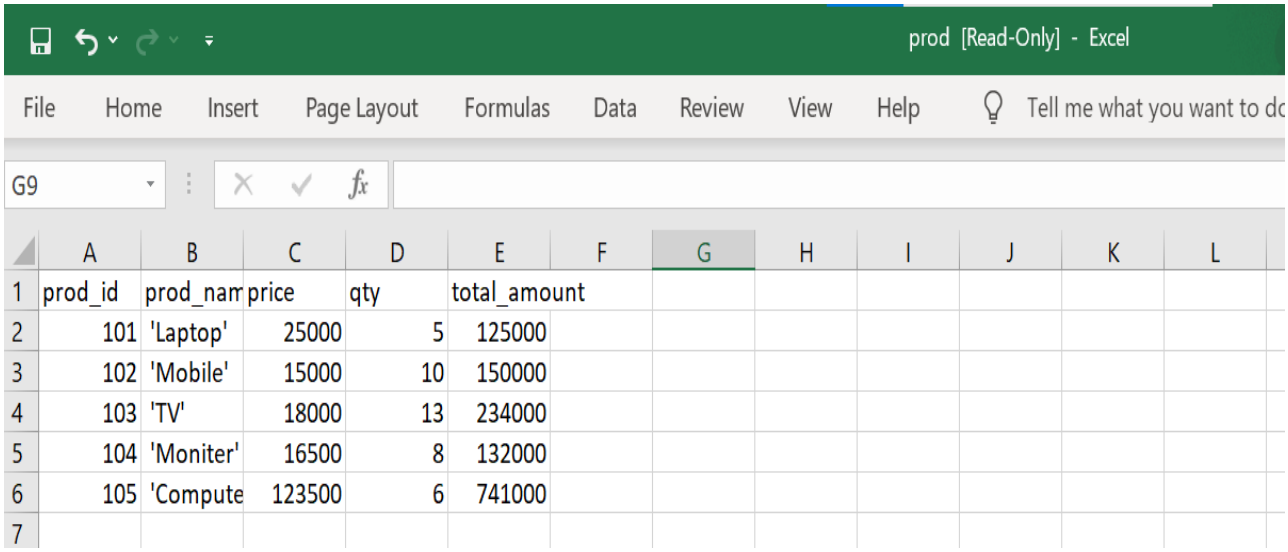
```
sqlite> CREATE TABLE Product (
(x1...> prod_id TEXT PRIMARY KEY,
(x1...> prod_name TEXT NOT NULL,
(x1...> price real NOT NULL,
(x1...> qty INTEGER not null,
(x1...> total_amount REAL);


sqlite> .mode csv
sqlite> .import product.csv Product
sqlite> .mode box
sqlite> select * from Product;
```

	A	B	C	D	E	F	G
1	prod_nam	JAN	FEB	MAR	APR	MAY	JUN
2	AC	100	120	90	110	130	95
3	Television	80	85	75	90	88	92
4	Mobile	60	70	65	75	72	80
5	Laptop	110	105	100	115	120	118
6	Cooler	150	140	160	145	155	152
7							

```
sqlite> select * from Product;
```

prod_id	prod_name	price	qty	total_amount
101	'Laptop'	25000.0	5	125000.0
102	'Mobile'	15000.0	10	150000.0
103	'TV'	18000.0	13	234000.0
104	'Moniter'	16500.0	8	132000.0
105	'Computer'	123500.0	6	741000.0

```
sqlite> .mode csv
sqlite> .output prod.csv
sqlite> select * from Product;
```



	A	B	C	D	E	F	G	H	I	J	K	L
1	prod_id	prod_nam	price	qty	total_amount							
2	101	'Laptop'	25000	5	125000							
3	102	'Mobile'	15000	10	150000							
4	103	'TV'	18000	13	234000							
5	104	'Moniter'	16500	8	132000							
6	105	'Compute	123500	6	741000							
7												

Name :- VINAYAK PUROHIT
Class :- SYBCA (sem 3)
Roll No :- G-772

Q9. Employee(Eno number ,Ename text ,Desg text ,Salary number ,City text ,Email text) Write a SQL trigger named emp_trigger that is designed to execute before inserting records into the emp table. The trigger should perform the following action: 1) Check if the 'email' field in the newly inserted record follows a specific email address pattern. (example : abc@gmail.com)

< SQLite Code >

```
sqlite> CREATE TABLE Employees (
(x1...> Eno INTEGER PRIMARY KEY,
(x1...> Ename TEXT NOT NULL,
(x1...> Desg TEXT,
(x1...> Salary REAL,
(x1...> City TEXT,
(x1...> Email TEXT
(x1...> );
sqlite> create trigger check_email
  ...> before
  ...> insert on Employees
  ...> begin
  ...>       select
  ...>             case
  ...>                   when new.email not like '%_@_%._%'
  ...>             then
  ...>                   raise(abort,'Invail Email Address')
  ...>       end;
  ...> end;


sqlite> .mode box
```

Name :- VINAYAK PUROHIT
Class :- SYBCA (sem 3)
Roll No :- G-772

sqlite> select * from Employees;

```
sqlite> insert into Employees values(1,'Ram prasad','Account',150000,'Surat','@ram.gmail.com');
Runtime error: Invail Email Address (19)
sqlite> insert into Employees values(1,'Ram prasad','Account',150000,'Surat','ramprasad@gmail.com');
sqlite> insert into Employees values(2,'Vinayak Purohit','Cyber security',230000,'Surat','purohitvinayak48@gmail.com');
sqlite> insert into Employees values(3,'Vishal','CA',124000,'Surat','@gmail.vishal.in');
Runtime error: Invail Email Address (19)
sqlite> select * from Employees;
```

Eno	Ename	Desg	Salary	City	Email
1	Ram prasad	Account	150000.0	Surat	ramprasad@gmail.com
2	Vinayak Purohit	Cyber security	230000.0	Surat	purohitvinayak48@gmail.com