**Question 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. Display age of student whose sno is 5. (use loc() and iloc() function)**

**Create Table**

create table STUDENT(sno number primary key, sname text(20), age number, total\_marks number);

**Solution**

import pandas as p

#Read Data Into DataFrame

student\_data={'sno':[1,2,3,4,5],

'sname':['Parth','Jay','Vivek','Maan','Dev'],

'age':[18,17,18,20,21],

'total\_marks':[85,86,82,84,90]}

df=p.DataFrame(student\_data,index=student\_data['sno'])

print("Table Data into Dataframe :- ")

print(df)

#top three data from DataFrame

print('Top Three Records From DataFrame :- ')

print(df.head(3))

#Displaying Records Whose age is not less than 18

print('Records Whose Age is not Less Than 18 :- ')

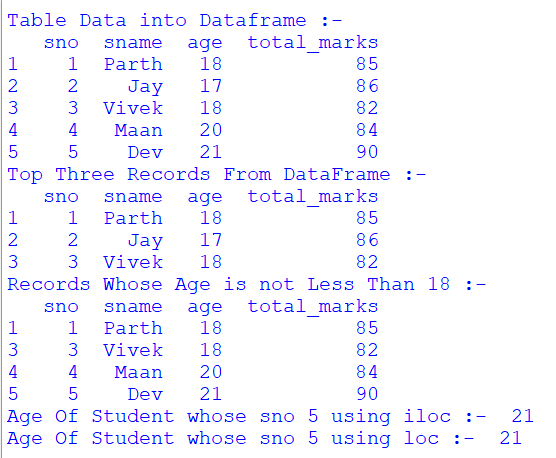
print(df[df['age']>=18])

#Displaying Age Of Student whose sno is 5 using loc & iloc

print('Age Of Student whose sno 5 using iloc :- ',df.iloc[4]['age'])

print('Age Of Student whose sno 5 using loc :- ',df.loc[df['sno']==5,'age'].values[0])

**Output**



**Question 2**

**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.**

**Create Table**

create table Employee(eno number primary key, Ename text(20),designation text(10),basic number , da number, gross\_salary number);

**Solution**

import pandas as p

#Read Data Into DataFrame

employee\_data={

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

'ename':['Parth','Jay','Vivek','Maan','Dev'],

'designation':['Manager','Developer','DBA','Designner','Head'],

'basic':[20000,23000,22500,21500,23500],

'da':[2000,2300,2250,2150,2350],

'gross\_salary':[22000,25300,24750,23650,25850]

}

emp=p.DataFrame(employee\_data)

print("DataFrame :- ")

print(emp)

#Sorting DataFrame By Gross Salary & Listing Out Bottom Two Records

sorted=emp.sort\_values('gross\_salary')

print("Sorted Values :-")

print(sorted)

print("Bottom Two Records :-\n",sorted.tail(2))

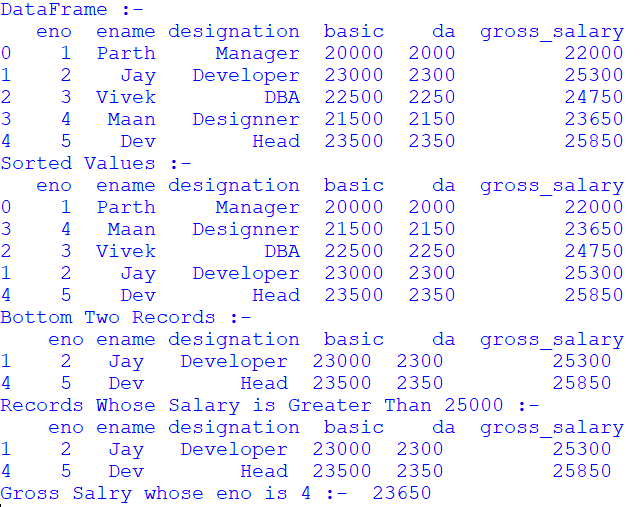
#Displaying Records Whose Salary is Greater Than 25000

print('Records Whose Salary is Greater Than 25000 :-\n',emp[emp['gross\_salary']>=25000])

#Displaying Gross Salary Whose eno is 4

print('Gross Salry whose eno is 4 :- ',emp.loc[emp['eno']==4,'gross\_salary'].values[0])

**Output**

****

**Question 3**

**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'**

**Solution**

import pandas as p

import matplotlib.pyplot as plt

#Read Data into DataFrame

data={

'Prod\_name':['Smartphone','Refrigerator','Air Conditioner','Washing Machine','Laptop'],

'JAN':[80,34,43,20,46],

'FEB':[90,32,45,11,34],

'MAR':[75,40,56,36,65],

'APR':[95,34,54,23,45],

'MAY':[56,43,56,67,43],

'JUN':[87,54,32,9,76]

}

df=p.DataFrame(data)

print(df)

#Calculate Total Sell And Average Sell

df['total\_sell']=df.iloc[:,1:7].sum(axis=1)

df['average\_sell']=df.iloc[:,1:7].mean(axis=1)

print(df)

#Line Chart

plt.figure(figsize=(10,6))

plt.plot(df['Prod\_name'],df['total\_sell'],label='total\_sell')

plt.plot(df['Prod\_name'],df['average\_sell'],label='average\_sell')

plt.xlabel('Products')

plt.ylabel('Amount')

plt.title('Sales Analysis')

plt.xticks(rotation=45)

plt.legend()

#DataFrame To CSV

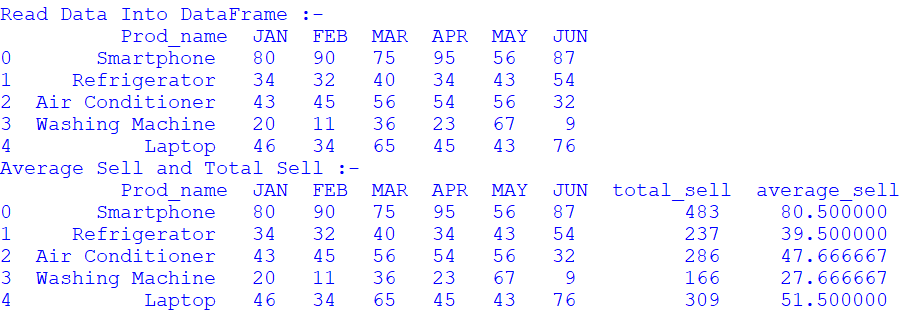
df.to\_csv('sell\_analysis.csv',index=False)

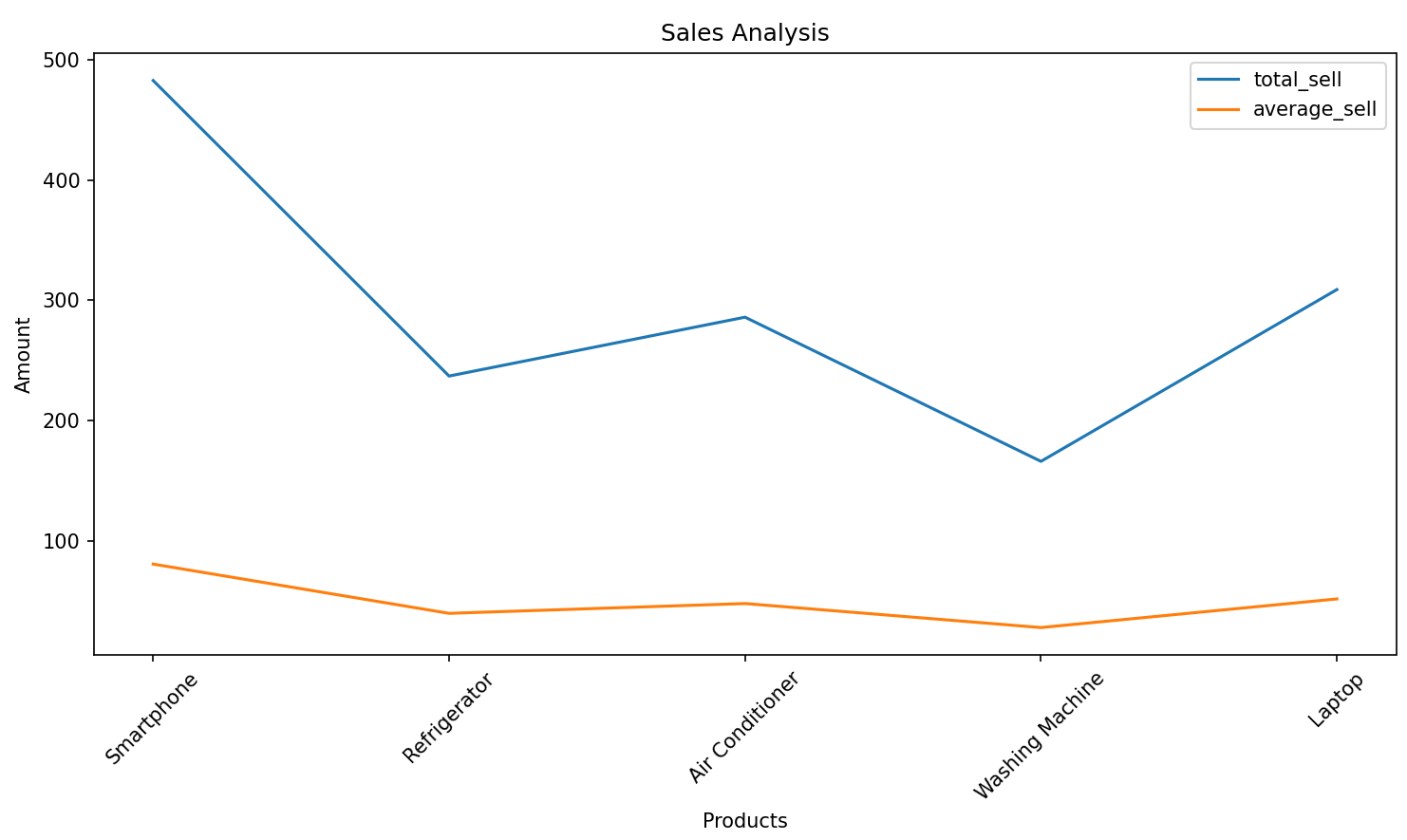
#Show Chart

plt.tight\_layout()

plt.show()

**Output**

****

****

sell\_analysis.csv

Prod\_name,JAN,FEB,MAR,APR,MAY,JUN,total\_sell,average\_sell

Smartphone,80,90,75,95,56,87,483,80.5

Refrigerator,34,32,40,34,43,54,237,39.5

Air Conditioner,43,45,56,54,56,32,286,47.666666666666664

Washing Machine,20,11,36,23,67,9,166,27.666666666666668

Laptop,46,34,65,45,43,76,309,51.5

**Question 4**

**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**

**Solution**

import sqlite3

def insert():

db=sqlite3.connect('student.db')

cur=db.cursor()

rollno=int(input("Enter RollNo :- "))

name=input("Enter Name :- ")

sub1=int(input("Enter Marks1 :- "))

sub2=int(input("Enter Marks2 :- "))

sub3=int(input("Enter Marks3 :- "))

total=sub1+sub2+sub3

cur.execute("insert into student values(?,?,?,?,?,?)",(rollno,name,sub1,sub2,sub3,total))

db.commit()

cur.close()

db.close()

def update():

db=sqlite3.connect('student.db')

cur=db.cursor()

up=int(input("Enter RollNo To Update :- "))

name=input("Enter New Name :- ")

sub1=int(input("Enter New Marks1 :- "))

sub2=int(input("Enter New Marks2 :- "))

sub3=int(input("Enter New Marks3 :- "))

total=sub1+sub2+sub3

cur.execute("update student set name=?,sub1=?,sub2=?,sub3=?,total=? where rollno=?",(name,sub1,sub2,sub3,total,up))

db.commit()

cur.close()

db.close()

def delete():

db=sqlite3.connect('student.db')

cur=db.cursor()

d=input("Enter RollNo To Delete :- ")

cur.execute("delete from student where rollno=?",(d))

db.commit()

cur.close()

db.close()

def fetch():

db=sqlite3.connect('student.db')

cur=db.cursor()

cur.execute("select \* from student where total=(select max(total)from student)")

d=cur.fetchone()

print(d)

db.commit()

cur.close()

db.close()

def main():

ch=0

while ch!=5:

ch=int(input("\n1.Insert\n2.Update\n3.Delete\n4.Highest Marks\n5.Exit\nEnter Choice:- "))

if(ch==1):

insert()

elif(ch==2):

update()

elif(ch==3):

delete()

elif(ch==4):

fetch()

if \_\_name\_\_=="\_\_main\_\_":

main()

**Output**

1.Insert

2.Update

3.Delete

4.Highest Marks

5.Exit

Enter Choice:- 1

Enter RollNo :- 1

Enter Name :- Jay

Enter Marks1 :- 87

Enter Marks2 :- 78

Enter Marks3 :- 90

1.Insert

2.Update

3.Delete

4.Highest Marks

5.Exit

Enter Choice:- 1

Enter RollNo :- 2

Enter Name :- Vivek

Enter Marks1 :- 99

Enter Marks2 :- 97

Enter Marks3 :- 97

1.Insert

2.Update

3.Delete

4.Highest Marks

5.Exit

Enter Choice:- 1

Enter RollNo :- 3

Enter Name :- Parth

Enter Marks1 :- 67

Enter Marks2 :- 87

Enter Marks3 :- 77

1.Insert

2.Update

3.Delete

4.Highest Marks

5.Exit

Enter Choice:- 2

Enter RollNo To Update :- 3

Enter New Name :- Parth1

Enter New Marks1 :- 65

Enter New Marks2 :- 65

Enter New Marks3 :- 65

1.Insert

2.Update

3.Delete

4.Highest Marks

5.Exit

Enter Choice:- 3

Enter RollNo To Delete :- 1

1.Insert

2.Update

3.Delete

4.Highest Marks

5.Exit

Enter Choice:- 4

(2, 'Vivek', 99, 97, 97, 293)

1.Insert

2.Update

3.Delete

4.Highest Marks

5.Exit

Enter Choice:- 5

**Question 5**

**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**

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

**Solution**

import pandas as p

data=p.read\_csv('item.csv')

#Write CSV file

new\_item={'Item\_no':9,'Item\_name':'Mouse','Price':800,'Qty':500}

new\_item['total']=new\_item['Price']\*new\_item['Qty']

data=data.\_append(new\_item,ignore\_index=True)

data.to\_csv('item.csv',index=False)

#Display Item name and price whose price is between 1000 to 5000

filtered=data[(data['Price']>=1000) & (data['Price']<=5000)]

print(filtered[['Item\_name','Price']])

#Alternate Records

print("Alternate Records :-\n",data.iloc[::2])

#Minimum & Maximum Price

min=data[data['Price']==data['Price'].min()]

max=data[data['Price']==data['Price'].max()]

print("Minimum :-\n",min)

print("Maximum :-\n",max)

#Sort

sorted=data.sort\_values(by='Item\_name')

print("Sorted :-\n",sorted)

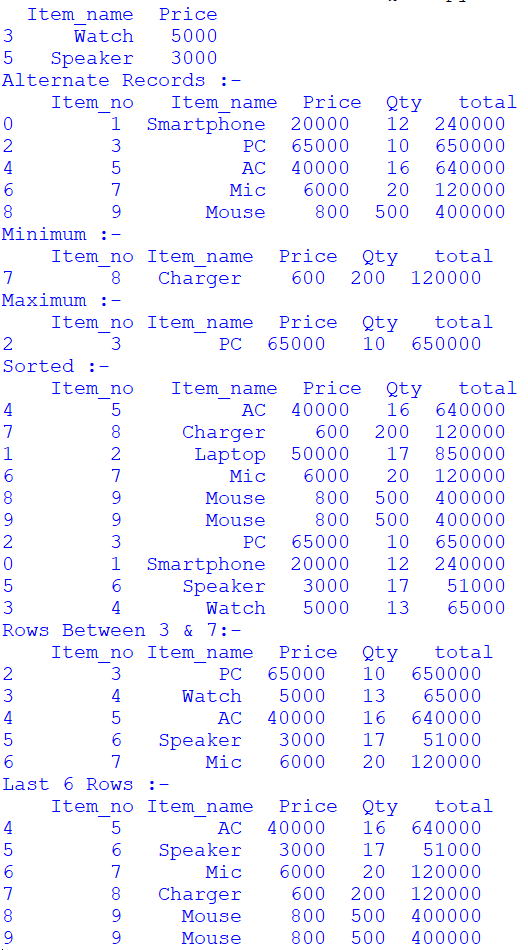
#Display Rows Between 3 & 7

print("Rows Between 3 & 7:-\n",data.iloc[2:7])

#Last 6 Rows

print("Last 6 Rows :-\n",data.tail(6))

**Output**



**Question 6**

**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.**

**Solution**

**SQLite**

sqlite> .open sales.db

sqlite> create table Sales(sid number primary key,year number,totalsales number);

sqlite> insert into Sales values(1,2010,30000),(2,2012,34000),(3,2012,32000),(4,2014,37000),(5,2016,45000),(6,2010,33000);

sqlite> .mode csv

sqlite> .output sales.csv

sqlite> .header on

sqlite> select \* from Sales;

**Python**

import pandas as p

import matplotlib.pyplot as plt

df=p.read\_csv('sales.csv')

yearly\_sales=df.groupby('year')['totalsales'].sum()

plt.figure(figsize=(10,6))

yearly\_sales.plot(kind='bar')

plt.title('Total Sales By Year')

plt.xlabel('Year')

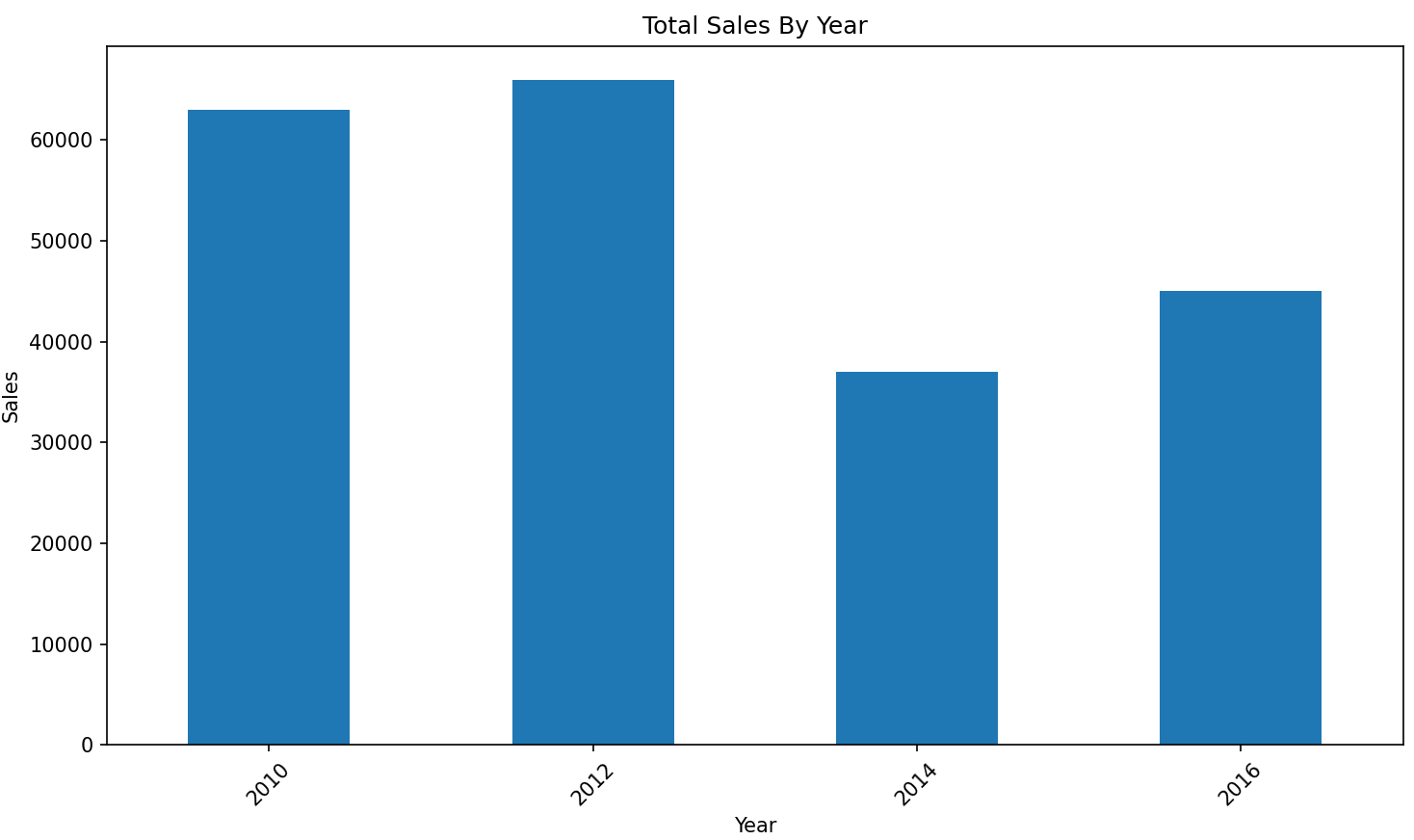
plt.ylabel('Sales')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()

**Output**



**Question 7**

**Create following table with appropriate constraints in Collage Database:**

**Employee (E\_ID, Name, Dob, Designation, Salary )**

* 1. **Dump Employee table structure and data in Emp.csv file.**
  2. **Dump whole Database named College in Emp1.csv file.**

**Solution**

sqlite>.open Collage.db

sqlite>CREATE TABLE Employee (E\_ID number primary key,Name text,Dob date,Designation text, Salary real);

sqlite>INSERT INTO Employee VALUES

(1,'Jay','25-Jun-2005','Developer',20000.0),

(2,'Vivek','15-Apr-2005','Designer',23000.0),

(3,'Parth','18-Aug-2004','Head',25000.0);

sqlite>.output emp.csv

sqlite>.dump Employee

sqlite>.output emp1.csv

sqlite>.dump

**Output**

emp.csv

PRAGMA foreign\_keys=OFF;

BEGIN TRANSACTION;

CREATE TABLE Employee (E\_ID number primary key,Name text,Dob date,Designation text, Salary real);

INSERT INTO Employee VALUES(1,'Jay','25-Jun-2005','Developer',20000.0);

INSERT INTO Employee VALUES(2,'Vivek','15-Apr-2005','Designer',23000.0);

INSERT INTO Employee VALUES(3,'Parth','18-Aug-2004','Head',25000.0);

COMMIT;

emp1.csv

PRAGMA foreign\_keys=OFF;

BEGIN TRANSACTION;

CREATE TABLE Employee (E\_ID number primary key,Name text,Dob date,Designation text, Salary real);

INSERT INTO Employee VALUES(1,'Jay','25-Jun-2005','Developer',20000.0);

INSERT INTO Employee VALUES(2,'Vivek','15-Apr-2005','Designer',23000.0);

INSERT INTO Employee VALUES(3,'Parth','18-Aug-2004','Head',25000.0);

COMMIT;

**Question 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**

**Solution**

sqlite> create table Product (prod\_id number primary key, prod\_name text, price real, qty number,total\_amount real);

sqlite> .mode csv

sqlite> .import product.csv Product

sqlite> .output prod.csv

sqlite> select \* from Product;

**Output**

prod.csv

prod\_id,prod\_name,price,qty,total\_amount

1,Smartphone,20000.0,12,240000.0

2,Laptop,50000.0,17,850000.0

3,PC,65000.0,10,650000.0

4,Watch,5000.0,13,65000.0

5,AC,40000.0,16,640000.0

6,Speaker,3000.0,17,51000.0

7,Mic,6000.0,20,120000.0

8,Charger,600.0,200,120000.0

9,Mouse,800.0,500,400000.0

**Question 9**

**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)**

**Solution**

sqlite> create table Employee(Eno number primary key,Ename text,Desg text,Salary number,City text,Email text);

sqlite> create trigger emp\_trigger before insert on Employee

...> begin

...> select

...> case

...> when new.Email not like '%@gmail.com' then

...> raise(abort,'Invalid Email')

...> end;

...> end;

**Output**

sqlite> insert into Employee values (1,'Jay','Head',20000,'Surat','123');

Runtime error: Invalid Email (19)

sqlite> insert into Employee values (1,'Jay','Head',20000,'Surat','123@gmail.com');

1 Record Inserted;