ASSIGNMENT-1

- **CREATE A TABLES:-**
  Department
  Employee
  Salarylog
  Emplog

```
.open database1.db
CREATE TABLE Department (
   Dept_id text PRIMARY KEY,
   Dept_name TEXT NOT NULL
);
CREATE TABLE Employee (
   Emp_id text PRIMARY KEY,
   Dept_id INTEGER NOT NULL,
   Emp_name TEXT NOT NULL,
   Mobile TEXT NOT NULL,
   Email TEXT NOT NULL,
   Age INTEGER NOT NULL CHECK (Age > 18 AND Age < 100),
   City TEXT NOT NULL,
   Salary REAL CHECK (Salary > 10000),
   FOREIGN KEY (Dept_id) REFERENCES Department(Dept_id)
);
CREATE TABLE Salarylog (
   Emp_id INTEGER,
   Old_salary REAL,
   New_salary REAL,
   Date TEXT,
   FOREIGN KEY (Emp_id) REFERENCES Employee(Emp_id)
);
CREATE TABLE Employ (
   Emp_id INTEGER,
```

```
    Emp_name TEXT,
    Dept_id INTEGER,
    Salary REAL,
    Date TEXT,
    FOREIGN KEY (Emp_id) REFERENCES Employee(Emp_id),
    FOREIGN KEY (Dept_id) REFERENCES Department(Dept_id)
);
.table
```

OUTPUT:-

```
sqlite> .table
Department  Employ      Employee    Salarylog
sqlite>
```

1. TRIGGER

1. **Create trigger before insert on table department to check if the deot_id starts with ‘D’ or not. If it not starts with ‘d’ then abort the insert.**

```
CREATE TRIGGER trg_check_deptid
BEFORE INSERT ON Department
FOR EACH ROW
BEGIN
  SELECT
    CASE
     WHEN SUBSTR(NEW.Dept_id, 1, 1) != 'D' THEN
       RAISE(ABORT, 'Dept_id must start with D')
    END;
END;
```

2. **Create trigger before insert on table employee to check if the emp_id starts with ‘E’ or not. If it not starts with ‘e’ then abort the insert.**

```
CREATE TRIGGER trg_check_empid
BEFORE INSERT ON Employee
FOR EACH ROW
BEGIN
  SELECT
    CASE
     WHEN SUBSTR(NEW.Emp_id, 1, 1) != 'E' THEN
       RAISE(ABORT, 'Emp_id must start with E')
    END;
END;
```

3. **Create a trigger on employee table which track record of salary change of each employee. For salary logs use table salarylog.**

```
CREATE TRIGGER trg_salary_change
AFTER UPDATE OF Salary ON Employee
FOR EACH ROW
WHEN OLD.Salary != NEW.Salary
BEGIN
  INSERT INTO Salarylog (Emp_id, Old_salary, New_salary, Date)
  VALUES (OLD.Emp_id, OLD.Salary, NEW.Salary, DATE('now'));
END;
```

4. **Create a trigger on employee table which has back up of all the employees who are removed from the table. Use table emplog for the same.**

```
CREATE TRIGGER trg_backup_employee
BEFORE DELETE ON Employee
FOR EACH ROW
BEGIN
  INSERT INTO Employ (Emp_id, Emp_name, Dept_id, Salary, Date)
  VALUES (OLD.Emp_id, OLD.Emp_name, OLD.Dept_id, OLD.Salary, DATE('now'));
END;
```

2. QUERIES

1. Insert appropriate records in department and employee tables.

INSERT INTO Department VALUES ('D1', 'HR');

INSERT INTO Department VALUES ('D2', 'Account');

INSERT INTO Department VALUES ('D3', 'IT');

INSERT INTO Department VALUES ('D4', 'Marketing');

INSERT INTO Department VALUES ('D5', 'Finance');

INSERT INTO Department VALUES ('D6', 'Customer Service');

.mode table

select*from Department;

```
sqlite> .mode table
sqlite> select*from Department;
+---------+------------------+
| Dept_id |    Dept_name     |
+---------+------------------+
| D1      | HR               |
| D2      | Account          |
| D3      | IT               |
| D4      | Marketing        |
| D5      | Finance          |
| D6      | Customer Service |
+---------+------------------+
sqlite>
```

INSERT INTO Employee VALUES ('E01', 'D1', 'Radha', '9876543210', 'radha@gmail.com', 25, 'Surat', 120000);

INSERT INTO Employee VALUES ('E02', 'D2', 'Rama', '9865321470', 'rama@gmail.com', 29, 'Ahmedabad', 90000);

INSERT INTO Employee VALUES ('E03', 'D1', 'Mahira', '9999888877', 'mahira@gmail.com', 30, 'Baroda', 130000);

INSERT INTO Employee VALUES ('E04', 'D3', 'Ravi', '9988776655', 'ravi@gmail.com', 35, 'Rajkot', 85000);

INSERT INTO Employee VALUES ('E05', 'D4', 'Simran', '9988771122', 'simran@gmail.com', 28, 'Jamnagar', 65000);

INSERT INTO Employee VALUES ('E06', 'D2', 'Anjali', '9876123450', 'anjali@gmail.com', 26, 'Bhavnagar', 72000);

INSERT INTO Employee VALUES ('E07', 'D5', 'Yash', '9845123460', 'yash@gmail.com', 40, 'Surat', 99000);

INSERT INTO Employee VALUES ('E08', 'D1', 'Kavita', '9781234567', 'kavita@gmail.com', 38, 'Vadodara', 78000);

INSERT INTO Employee VALUES ('E09', 'D4', 'Meera', '9876543100', 'meera@gmail.com', 32, 'Nadiad', 94000);

INSERT INTO Employee VALUES ('E10', 'D3', 'Vikram', '9966554433', 'vikram@gmail.com', 34, 'Anand', 102000);

INSERT INTO Employee VALUES ('E11', 'D2', 'Sita', '9990001112', 'sita@gmail.com', 22, 'Bharuch', 56000);

INSERT INTO Employee VALUES ('E12', 'D5', 'Arjun', '9811122233', 'arjun@gmail.com', 45, 'Surat', 88000);

INSERT INTO Employee VALUES ('E13', 'D1', 'Naina', '9933445566', 'naina@gmail.com', 31, 'Valsad', 87000);

INSERT INTO Employee VALUES ('E14', 'D3', 'Amit', '9776655443', 'amit@gmail.com', 50, 'Navsari', 110000);

INSERT INTO Employee VALUES ('E15', 'D2', 'Mona', '9988773322', 'mona@gmail.com', 24, 'Surat', 61000);

INSERT INTO Employee VALUES ('E16', 'D4', 'Komal', '9765432190', 'komal@gmail.com', 29, 'Ahmedabad', 99000);

INSERT INTO Employee VALUES ('E17', 'D5', 'Raj', '9988123456', 'raj@gmail.com', 60, 'Surat', 115000);

INSERT INTO Employee VALUES ('E18', 'D1', 'Neha', '9887766554', 'neha@gmail.com', 33, 'Junagadh', 70000);

INSERT INTO Employee VALUES ('E19', 'D3', 'Divya', '9856231458', 'divya@gmail.com', 21, 'Palanpur', 67000);

INSERT INTO Employee VALUES ('E20', '6', 'Isha', '9767894321', 'isha@gmail.com', 27, 'Ahmedabad', 75000);

select*from Employee;

```
sqlite> select*from Employee;
+--------+---------+----------+------------+------------------+-----+-----------+----------+
| Emp_id | Dept_id | Emp_name |   Mobile   |      Email       | Age |   City    |  Salary  |
+--------+---------+----------+------------+------------------+-----+-----------+----------+
| E01    | D1      | Radha    | 9876543210 | radha@gmail.com  | 25  | Surat     | 120000.0 |
| E02    | D2      | Rama     | 9865321470 | rama@gmail.com   | 29  | Ahmedabad | 90000.0  |
| E03    | D1      | Mahira   | 9999888877 | mahira@gmail.com | 30  | Baroda    | 130000.0 |
| E04    | D3      | Ravi     | 9988776655 | ravi@gmail.com   | 35  | Rajkot    | 85000.0  |
| E05    | D4      | Simran   | 9988771122 | simran@gmail.com | 28  | Jamnagar  | 65000.0  |
| E06    | D2      | Anjali   | 9876123450 | anjali@gmail.com | 26  | Bhavnagar | 72000.0  |
| E07    | D5      | Yash     | 9845123460 | yash@gmail.com   | 40  | Surat     | 99000.0  |
| E08    | D1      | Kavita   | 9781234567 | kavita@gmail.com | 38  | Vadodara  | 78000.0  |
| E09    | D4      | Meera    | 9876543100 | meera@gmail.com  | 32  | Nadiad    | 94000.0  |
| E10    | D3      | Vikram   | 9966554433 | vikram@gmail.com | 34  | Anand     | 102000.0 |
| E11    | D2      | Sita     | 9990001112 | sita@gmail.com   | 22  | Bharuch   | 56000.0  |
| E12    | D5      | Arjun    | 9811122233 | arjun@gmail.com  | 45  | Surat     | 88000.0  |
| E13    | D1      | Naina    | 9933445566 | naina@gmail.com  | 31  | Valsad    | 87000.0  |
| E14    | D3      | Amit     | 9776655443 | amit@gmail.com   | 50  | Navsari   | 110000.0 |
| E15    | D2      | Mona     | 9988773322 | mona@gmail.com   | 24  | Surat     | 61000.0  |
| E16    | D4      | Komal    | 9765432190 | komal@gmail.com  | 29  | Ahmedabad | 99000.0  |
| E17    | D5      | Raj      | 9988123456 | raj@gmail.com    | 60  | Surat     | 115000.0 |
| E18    | D1      | Neha     | 9887766554 | neha@gmail.com   | 33  | Junagadh  | 70000.0  |
| E19    | D3      | Divya    | 9856231458 | divya@gmail.com  | 21  | Palanpur  | 67000.0  |
| E20    | 6       | Isha     | 9767894321 | isha@gmail.com   | 27  | Ahmedabad | 75000.0  |
+--------+---------+----------+------------+------------------+-----+-----------+----------+
sqlite>
```

INSERT INTO salarylog (Emp_id, Old_salary, New_salary, Date) VALUES

('E01', 15000, 18000, '2025-07-01'),

('E12', 20000, 23000, '2025-07-05'),

('E03', 18000, 20000, '2025-07-10'),

('E14', 25000, 27000, '2025-07-15'),

('E05', 22000, 25000, '2025-07-20');

select * from Salarylog;

```
sqlite> select * from Salarylog;
+--------+------------+------------+------------+
| Emp_id | Old_salary | New_salary |    Date    |
+--------+------------+------------+------------+
| E01    | 15000.0    | 18000.0    | 2025-07-01 |
| E12    | 20000.0    | 23000.0    | 2025-07-05 |
| E03    | 18000.0    | 20000.0    | 2025-07-10 |
| E14    | 25000.0    | 27000.0    | 2025-07-15 |
| E05    | 22000.0    | 25000.0    | 2025-07-20 |
+--------+------------+------------+------------+
sqlite>
```

INSERT INTO employ (Emp_id, Emp_name, Dept_id, Salary, Date) VALUES

('E11', 'Rahul Mehta', 'D1', 18000, '2025-07-01'),

('E02', 'Priya Sharma', 'D1', 23000, '2025-07-05'),

('E13', 'Amit Patel', 'D1', 20000, '2025-07-10'),

('E04', 'Neha Joshi', 'D1', 27000, '2025-07-15'),

('E15', 'Vikas Singh', 'D1', 25000, '2025-07-20');

select * from Employ;

```
sqlite> select * from Employ;
+--------+--------------+---------+---------+------------+
| Emp_id |   Emp_name   | Dept_id | Salary  |    Date    |
+--------+--------------+---------+---------+------------+
| E11    | Rahul Mehta  | D1      | 18000.0 | 2025-07-01 |
| E02    | Priya Sharma | D1      | 23000.0 | 2025-07-05 |
| E13    | Amit Patel   | D1      | 20000.0 | 2025-07-10 |
| E04    | Neha Joshi   | D1      | 27000.0 | 2025-07-15 |
| E15    | Vikas Singh  | D1      | 25000.0 | 2025-07-20 |
+--------+--------------+---------+---------+------------+
sqlite>
```

2. **Find the employees name who works in “HR” department.**

SELECT Emp_name FROM Employee

WHERE Dept_id = (SELECT Dept_id FROM Department WHERE Dept_name = 'HR');

```
sqlite> SELECT Emp_name FROM Employee
   ...> WHERE Dept_id = (SELECT Dept_id FROM Department WHERE Dept_name = 'HR');
+----------+
| Emp_name |
+----------+
| Radha    |
| Mahira   |
| Kavita   |
| Naina    |
| Neha     |
+----------+
sqlite>
```

3. **Find the employee who has maximum salary.**

SELECT * FROM Employee

ORDER BY Salary DESC LIMIT 1;

```
sqlite> SELECT * FROM Employee
   ...> ORDER BY Salary DESC LIMIT 1;
+--------+---------+----------+------------+------------------+-----+--------+----------+
| Emp_id | Dept_id | Emp_name |   Mobile   |      Email       | Age |  City  |  Salary  |
+--------+---------+----------+------------+------------------+-----+--------+----------+
| E03    | D1      | Mahira   | 9999888877 | mahira@gmail.com | 30  | Baroda | 130000.0 |
+--------+---------+----------+------------+------------------+-----+--------+----------+
sqlite>
```

4. **Find all the details of employees whose name’s second and last letter is ‘a’. Ex. Rama, Radha, Mahira...**

SELECT * FROM Employee

WHERE SUBSTR(Emp_name, 2, 1) = 'a' AND SUBSTR(Emp_name, -1) = 'a';

```
sqlite> SELECT * FROM Employee
   ...> WHERE SUBSTR(Emp_name, 2, 1) = 'a' AND SUBSTR(Emp_name, -1) = 'a';
+--------+---------+----------+------------+------------------+-----+-----------+----------+
| Emp_id | Dept_id | Emp_name |   Mobile   |      Email       | Age |   City    |  Salary  |
+--------+---------+----------+------------+------------------+-----+-----------+----------+
| E01    | D1      | Radha    | 9876543210 | radha@gmail.com  | 25  | Surat     | 120000.0 |
| E02    | D2      | Rama     | 9865321470 | rama@gmail.com   | 29  | Ahmedabad | 90000.0  |
| E03    | D1      | Mahira   | 9999888877 | mahira@gmail.com | 30  | Baroda    | 130000.0 |
| E08    | D1      | Kavita   | 9781234567 | kavita@gmail.com | 38  | Vadodara  | 78000.0  |
| E13    | D1      | Naina    | 9933445566 | naina@gmail.com  | 31  | Valsad    | 87000.0  |
+--------+---------+----------+------------+------------------+-----+-----------+----------+
sqlite>
```

5. **Display all the employees whose age is less than 30 and working in Account department.**

SELECT * FROM Employee

WHERE Age < 30 AND Dept_id = (SELECT Dept_id FROM Department WHERE Dept_name = 'Account');

```
sqlite> SELECT * FROM Employee
   ...> WHERE Age < 30 AND Dept_id = (SELECT Dept_id FROM Department WHERE Dept_name = 'Account');
+--------+---------+----------+------------+------------------+-----+-----------+---------+
| Emp_id | Dept_id | Emp_name |   Mobile   |      Email       | Age |   City    | Salary  |
+--------+---------+----------+------------+------------------+-----+-----------+---------+
| E02    | D2      | Rama     | 9865321470 | rama@gmail.com   | 29  | Ahmedabad | 90000.0 |
| E06    | D2      | Anjali   | 9876123450 | anjali@gmail.com | 26  | Bhavnagar | 72000.0 |
| E11    | D2      | Sita     | 9990001112 | sita@gmail.com   | 22  | Bharuch   | 56000.0 |
| E15    | D2      | Mona     | 9988773322 | mona@gmail.com   | 24  | Surat     | 61000.0 |
+--------+---------+----------+------------+------------------+-----+-----------+---------+
```

6. **Display customer name and city who have salary more than 100000.**

SELECT Emp_name, City FROM Employee

WHERE Salary > 100000;

```
sqlite> SELECT Emp_name, City FROM Employee
   ...> WHERE Salary > 100000;
+----------+---------+
| Emp_name |  City   |
+----------+---------+
| Radha    | Surat   |
| Mahira   | Baroda  |
| Vikram   | Anand   |
| Amit     | Navsari |
| Raj      | Surat   |
+----------+---------+
```

7. **Display total number of employees working in each department.**

SELECT Dept_id, COUNT(*) AS Total_Employees

FROM Employee

GROUP BY Dept_id;

```
sqlite> SELECT Dept_id, COUNT(*) AS Total_Employees
   ...> FROM Employee
   ...> GROUP BY Dept_id;
+---------+-----------------+
| Dept_id | Total_Employees |
+---------+-----------------+
| 6       | 1               |
| D1      | 5               |
| D2      | 4               |
| D3      | 4               |
| D4      | 3               |
| D5      | 3               |
+---------+-----------------+
```

8. Count total salary for each department.

SELECT Dept_id, SUM(Salary) AS Total_Salary

FROM Employee

GROUP BY Dept_id;

```
sqlite> SELECT Dept_id, SUM(Salary) AS Total_Salary
   ...> FROM Employee
   ...> GROUP BY Dept_id;
+---------+--------------+
| Dept_id | Total_Salary |
+---------+--------------+
| 6       | 75000.0      |
| D1      | 485000.0     |
| D2      | 279000.0     |
| D3      | 364000.0     |
| D4      | 258000.0     |
| D5      | 302000.0     |
+---------+--------------+
```

9. Display all the employees in descending order of their age.

SELECT * FROM Employee

ORDER BY Age DESC;

```
sqlite> SELECT * FROM Employee
   ...> ORDER BY Age DESC;
```

Emp_id	Dept_id	Emp_name	Mobile	Email	Age	City	Salary
E17	D5	Raj	9988123456	raj@gmail.com	60	Surat	115000.0
E14	D3	Amit	9776655443	amit@gmail.com	50	Navsari	110000.0
E12	D5	Arjun	9811122233	arjun@gmail.com	45	Surat	88000.0
E07	D5	Yash	9845123460	yash@gmail.com	40	Surat	99000.0
E08	D1	Kavita	9781234567	kavita@gmail.com	38	Vadodara	78000.0
E04	D3	Ravi	9988776655	ravi@gmail.com	35	Rajkot	85000.0
E10	D3	Vikram	9966554433	vikram@gmail.com	34	Anand	102000.0
E18	D1	Neha	9887766554	neha@gmail.com	33	Junagadh	70000.0
E09	D4	Meera	9876543100	meera@gmail.com	32	Nadiad	94000.0
E13	D1	Naina	9933445566	naina@gmail.com	31	Valsad	87000.0
E03	D1	Mahira	9999888877	mahira@gmail.com	30	Baroda	130000.0
E02	D2	Rama	9865321470	rama@gmail.com	29	Ahmedabad	90000.0
E16	D4	Komal	9765432190	komal@gmail.com	29	Ahmedabad	99000.0
E05	D4	Simran	9988771122	simran@gmail.com	28	Jamnagar	65000.0
E20	6	Isha	9767894321	isha@gmail.com	27	Ahmedabad	75000.0
E06	D2	Anjali	9876123450	anjali@gmail.com	26	Bhavnagar	72000.0
E01	D1	Radha	9876543210	radha@gmail.com	25	Surat	120000.0
E15	D2	Mona	9988773322	mona@gmail.com	24	Surat	61000.0
E11	D2	Sita	9990001112	sita@gmail.com	22	Bharuch	56000.0
E19	D3	Divya	9856231458	divya@gmail.com	21	Palanpur	67000.0

10. Display the employee from each department who is having maximum salary.

```
SELECT * FROM Employee e
WHERE Salary = (
  SELECT MAX(Salary)
  FROM Employee
  WHERE Dept_id = e.Dept_id
);
```

```
sqlite> SELECT * FROM Employee e
   ...> WHERE Salary = (
(x1...>   SELECT MAX(Salary)
(x1...>   FROM Employee
(x1...>   WHERE Dept_id = e.Dept_id
(x1...> );
```

Emp_id	Dept_id	Emp_name	Mobile	Email	Age	City	Salary
E02	D2	Rama	9865321470	rama@gmail.com	29	Ahmedabad	90000.0
E03	D1	Mahira	9999888877	mahira@gmail.com	30	Baroda	130000.0
E14	D3	Amit	9776655443	amit@gmail.com	50	Navsari	110000.0
E16	D4	Komal	9765432190	komal@gmail.com	29	Ahmedabad	99000.0
E17	D5	Raj	9988123456	raj@gmail.com	60	Surat	115000.0
E20	6	Isha	9767894321	isha@gmail.com	27	Ahmedabad	75000.0

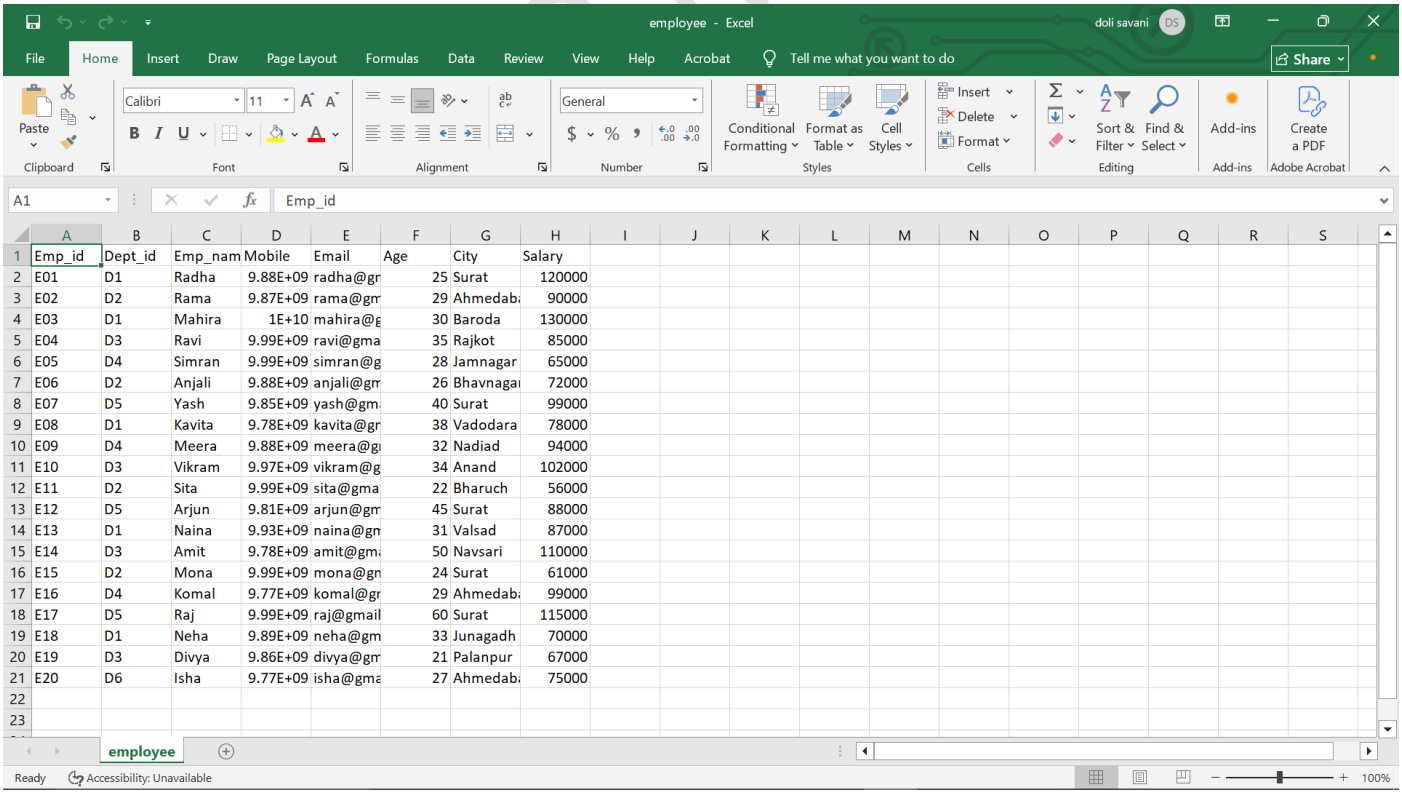
11. Count total salary increment given to all employees in July month of 2024.

```
SELECT SUM(New_salary - Old_salary) AS Total_Increment
FROM Salarylog
WHERE strftime('%m', Date) = '07' AND strftime('%Y', Date) = '2024';
```

```
sqlite> SELECT SUM(New_salary - Old_salary) AS Total_Increment
   ...> FROM Salarylog
   ...> WHERE strftime('%m', Date) = '07' AND strftime('%Y', Date) = '2024';
+-----------------+
| Total_Increment |
+-----------------+
|                 |
+-----------------+
```

12. Export employee table into employee.csv file.

```
.mode csv
.headers on
.output employee.csv
SELECT * FROM Employee;
.output stdout
```

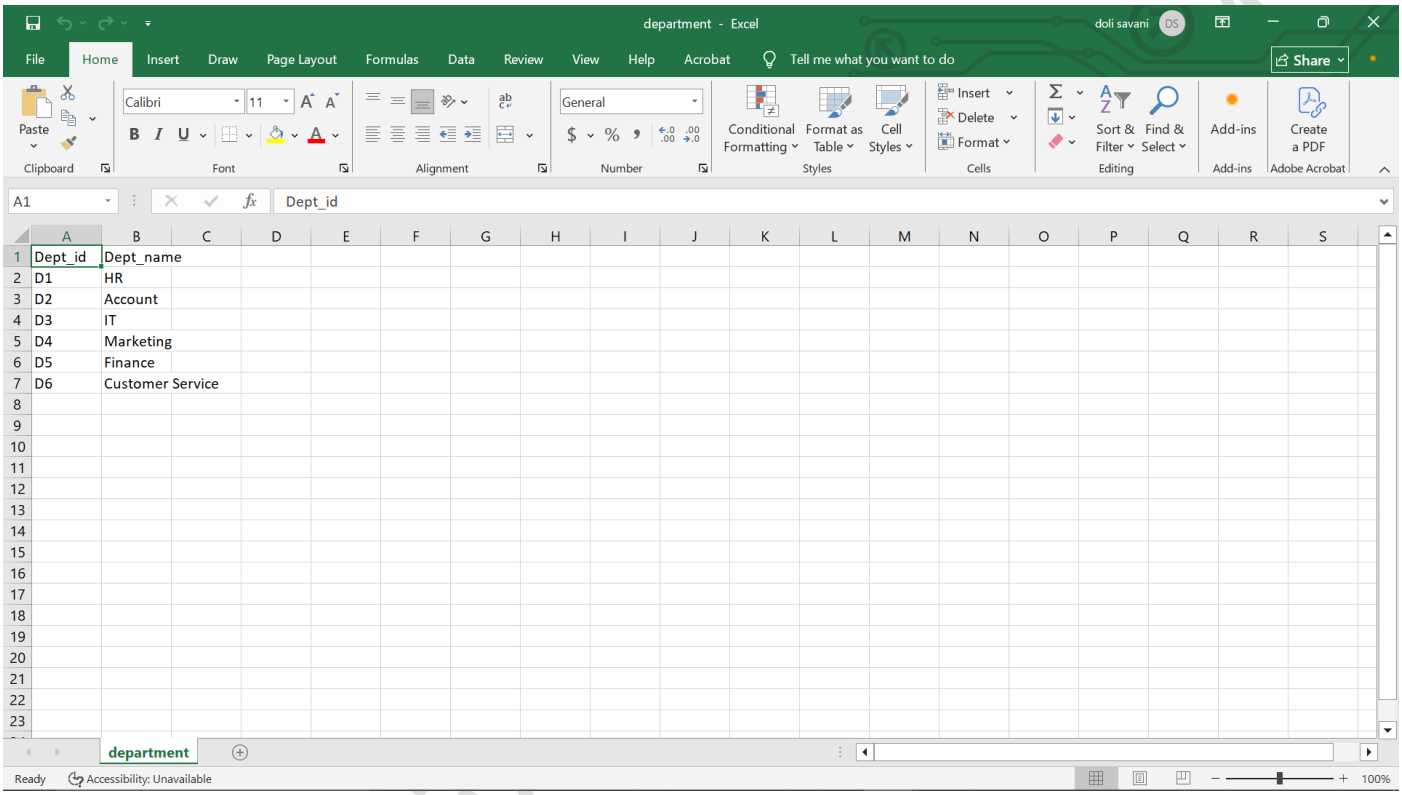


employee - Excel

Emp_id	Dept_id	Emp_nam	Mobile	Email	Age	City	Salary
E01	D1	Radha	9.88E+09	radha@gr	25	Surat	120000
E02	D2	Rama	9.87E+09	rama@gm	29	Ahmedab	90000
E03	D1	Mahira	1E+10	mahira@g	30	Baroda	130000
E04	D3	Ravi	9.99E+09	ravi@gma	35	Rajkot	85000
E05	D4	Simran	9.99E+09	simran@g	28	Jamnagar	65000
E06	D2	Anjali	9.88E+09	anjali@gm	26	Bhavnaga	72000
E07	D5	Yash	9.85E+09	yash@gm	40	Surat	99000
E08	D1	Kavita	9.78E+09	kavita@gr	38	Vadodara	78000
E09	D4	Meera	9.88E+09	meera@g	32	Nadiad	94000
E10	D3	Vikram	9.97E+09	vikram@g	34	Anand	102000
E11	D2	Sita	9.99E+09	sita@gma	22	Bharuch	56000
E12	D5	Arjun	9.81E+09	arjun@gm	45	Surat	88000
E13	D1	Naina	9.93E+09	naina@gn	31	Valsad	87000
E14	D3	Amit	9.78E+09	amit@gma	50	Navsari	110000
E15	D2	Mona	9.99E+09	mona@gn	24	Surat	61000
E16	D4	Komal	9.77E+09	komal@gr	29	Ahmedab	99000
E17	D5	Raj	9.99E+09	raj@gmail	60	Surat	115000
E18	D1	Neha	9.89E+09	neha@gm	33	Junagadh	70000
E19	D3	Divya	9.86E+09	divya@gm	21	Palanpur	67000
E20	D6	Isha	9.77E+09	isha@gma	27	Ahmedab	75000

13. Export department table data into department.csv file.

```
.mode csv
.headers on
.output department.csv
SELECT * FROM Department;
.output stdout
```



14. Take backup of whole database in “mycompany” file.

```
.backup mycompany_backup.db
```

```
sqlite>
sqlite> .backup mycompany_backup.db
sqlite>
```

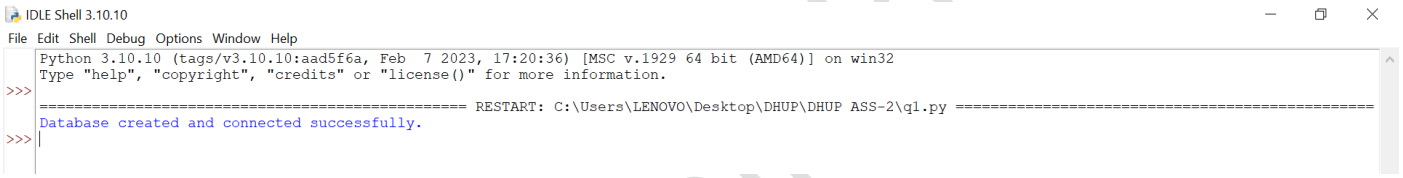
Back_up Database:-

```
sqlite> .open mycompany_backup.db
sqlite> .table
Department  Employ      Employee    Salarylog
sqlite> .mode table
sqlite> select * from Employee
   ...> ;
+--------+---------+----------+------------+------------------+-----+-----------+----------+
| Emp_id | Dept_id | Emp_name |   Mobile   |      Email       | Age |   City    |  Salary  |
+--------+---------+----------+------------+------------------+-----+-----------+----------+
| E01    | D1      | Radha    | 9876543210 | radha@gmail.com  | 25  | Surat     | 120000.0 |
| E02    | D2      | Rama     | 9865321470 | rama@gmail.com   | 29  | Ahmedabad | 90000.0  |
| E03    | D1      | Mahira   | 9999888877 | mahira@gmail.com | 30  | Baroda    | 130000.0 |
| E04    | D3      | Ravi     | 9988776655 | ravi@gmail.com   | 35  | Rajkot    | 85000.0  |
| E05    | D4      | Simran   | 9988771122 | simran@gmail.com | 28  | Jamnagar  | 65000.0  |
| E06    | D2      | Anjali   | 9876123450 | anjali@gmail.com | 26  | Bhavnagar | 72000.0  |
| E07    | D5      | Yash     | 9845123460 | yash@gmail.com   | 40  | Surat     | 99000.0  |
| E08    | D1      | Kavita   | 9781234567 | kavita@gmail.com | 38  | Vadodara  | 78000.0  |
| E09    | D4      | Meera    | 9876543100 | meera@gmail.com  | 32  | Nadiad    | 94000.0  |
| E10    | D3      | Vikram   | 9966554433 | vikram@gmail.com | 34  | Anand     | 102000.0 |
| E11    | D2      | Sita     | 9990001112 | sita@gmail.com   | 22  | Bharuch   | 56000.0  |
| E12    | D5      | Arjun    | 9811122233 | arjun@gmail.com  | 45  | Surat     | 88000.0  |
| E13    | D1      | Naina    | 9933445566 | naina@gmail.com  | 31  | Valsad    | 87000.0  |
| E14    | D3      | Amit     | 9776655443 | amit@gmail.com   | 50  | Navsari   | 110000.0 |
| E15    | D2      | Mona     | 9988773322 | mona@gmail.com   | 24  | Surat     | 61000.0  |
| E16    | D4      | Komal    | 9765432190 | komal@gmail.com  | 29  | Ahmedabad | 99000.0  |
| E17    | D5      | Raj      | 9988123456 | raj@gmail.com    | 60  | Surat     | 115000.0 |
| E18    | D1      | Neha     | 9887766554 | neha@gmail.com   | 33  | Junagadh  | 70000.0  |
| E19    | D3      | Divya    | 9856231458 | divya@gmail.com  | 21  | Palanpur  | 67000.0  |
| E20    | 6       | Isha     | 9767894321 | isha@gmail.com   | 27  | Ahmedabad | 75000.0  |
+--------+---------+----------+------------+------------------+-----+-----------+----------+
sqlite>
```

ASSIGNMENT - 2

1. Write Python code to Create database 'Student_Information.db'

```
import sqlite3
try:
    conn = sqlite3.connect("Student_Information.db")
    print("Database created and connected successfully.")
except Exception as e:
    print("Error creating database:", e)
finally:
    conn.close()
```



2. Write python code to Create table Student with following constrains.

a. RollNumber text primary key

b. Name text Not null

c. PYTHON int not null marks must greater than 0 and less than 100

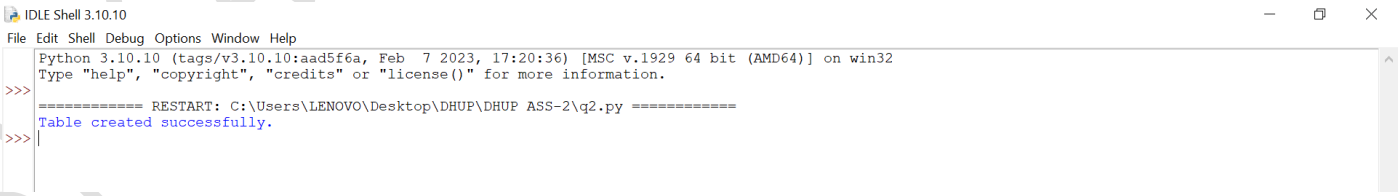
d. OOPS int not null marks must greater than 0 and less than 100

e. WEB int not null marks must greater than 0 and less than 100

f. MIL int not null marks must greater than 0 and less than 100

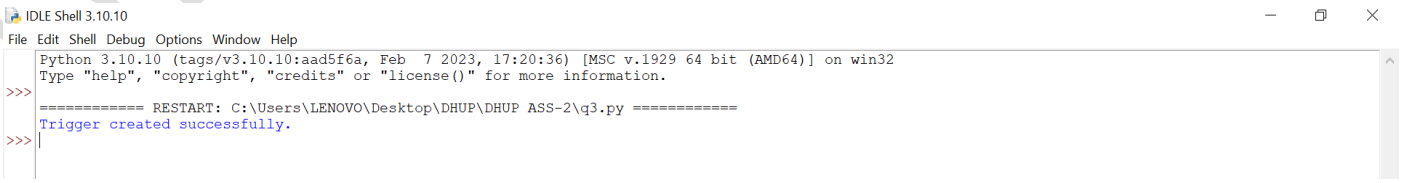
g. STATE int not null marks must greater than 0 and less than 100

```
import sqlite3


try:
    conn = sqlite3.connect("Student_Information.db")
    cur = conn.cursor()
    cur.execute("""
    CREATE TABLE IF NOT EXISTS Student (
        RollNumber TEXT PRIMARY KEY,
        Name TEXT NOT NULL,
        PYTHON INTEGER NOT NULL CHECK(PYTHON > 0 AND PYTHON < 100),
        OOPS INTEGER NOT NULL CHECK(OOPS > 0 AND OOPS < 100),
        WEB INTEGER NOT NULL CHECK(WEB > 0 AND WEB < 100),
        MIL INTEGER NOT NULL CHECK(MIL > 0 AND MIL < 100),
        STATE INTEGER NOT NULL CHECK(STATE > 0 AND STATE < 100)
    )
    """)
    print("Table created successfully.")
except Exception as e:
    print("Error creating table:", e)
finally:
    conn.commit()
    conn.close()
```



3. Write python code to create trigger named as 'rollnumbercheck' which check for the RollNumber must starts with 'R/r'.

```
import sqlite3


try:
    conn = sqlite3.connect("Student_Information.db")
    cur = conn.cursor()
    cur.execute("""
    CREATE TRIGGER IF NOT EXISTS rollnumbercheck
    BEFORE INSERT ON Student
    FOR EACH ROW
    BEGIN
       SELECT CASE
          WHEN NEW.RollNumber NOT LIKE 'R%' AND NEW.RollNumber NOT LIKE 'r%'
          THEN RAISE(ABORT, 'RollNumber must start with R or r')
       END;
    END;
    """)
    print("Trigger created successfully.")
except Exception as e:
    print("Error creating trigger:", e)
finally:
    conn.commit()
    conn.close()
```

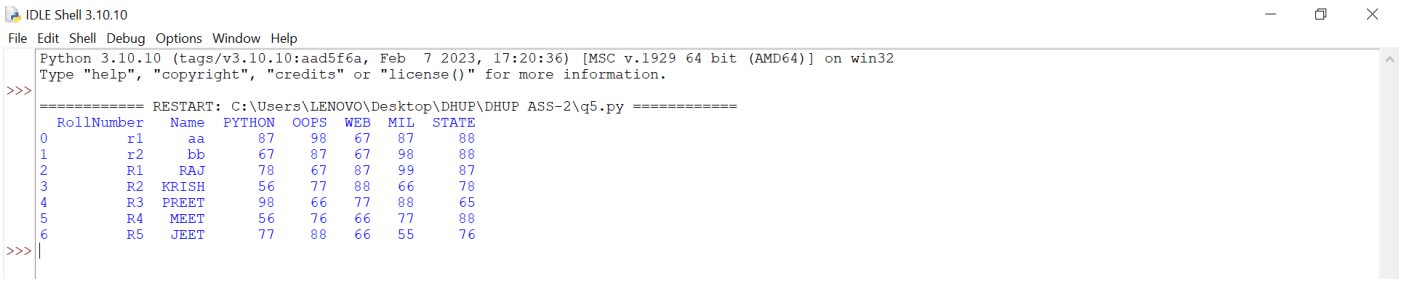


4. Write python code to insert user inputted data into the table by taking appropriate inputs from the user. (Ask user to take number of students to be inserted)

```
import sqlite3


try:
    conn = sqlite3.connect("Student_Information.db")
    cur = conn.cursor()
    n = int(input("Enter number of students: "))
    for _ in range(n):
        roll = input("Enter Roll Number: ")
        name = input("Enter Name: ")
        python_m = int(input("PYTHON marks: "))
        oops_m = int(input("OOPS marks: "))
        web_m = int(input("WEB marks: "))
        mil_m = int(input("MIL marks: "))
        state_m = int(input("STATE marks: "))
        cur.execute("INSERT INTO Student VALUES (?, ?, ?, ?, ?, ?, ?)",
                    (roll, name, python_m, oops_m, web_m, mil_m, state_m))
    print("Data inserted successfully.")
except Exception as e:
    print("Error inserting data:", e)
finally:
    conn.commit()
    conn.close()
```

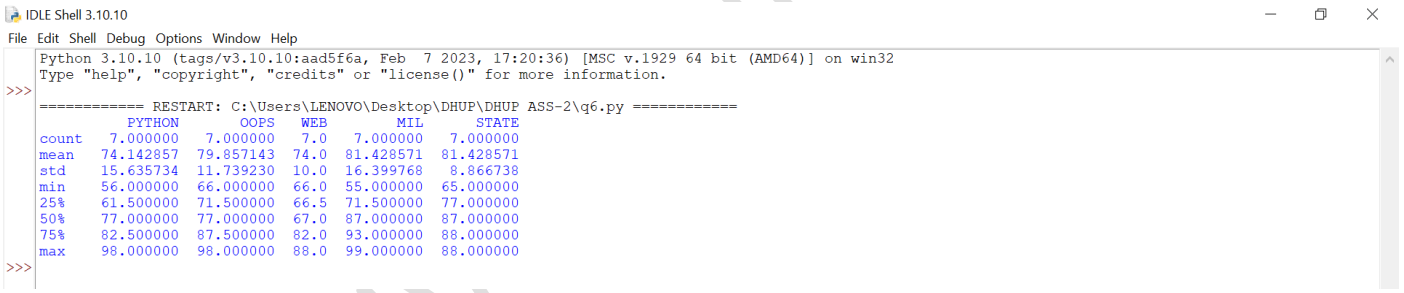
5. Write a python code to fetch all the data from the table and display it in appropriate table.

```
import sqlite3
import pandas as pd

try:
    conn = sqlite3.connect("Student_Information.db")
    df = pd.read_sql_query("SELECT * FROM Student", conn)
    print(df)
except Exception as e:
    print("Error fetching data:", e)
finally:
    conn.close()
```



6. Write a python code to fetch all the records in data frame. Use the appropriate method to describe all the data.

```
import sqlite3
import pandas as pd

try:
    conn = sqlite3.connect("Student_Information.db")
    df = pd.read_sql_query("SELECT * FROM Student", conn)
    print(df.describe())
except Exception as e:
    print("Error describing data:", e)
finally:
    conn.close()
```



7. Write a python code to add following columns to data frame. Use appropriate methods of pandas module.

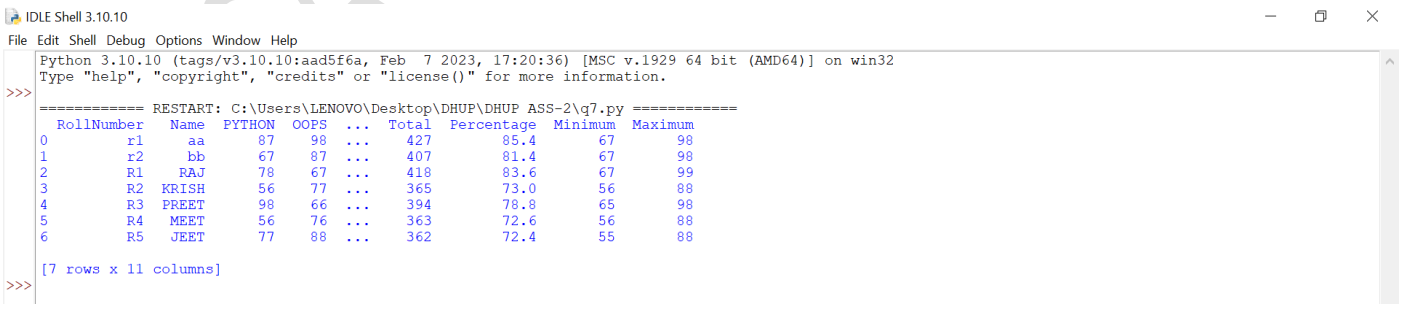
a. Total

b. Percentage

c. Minimum

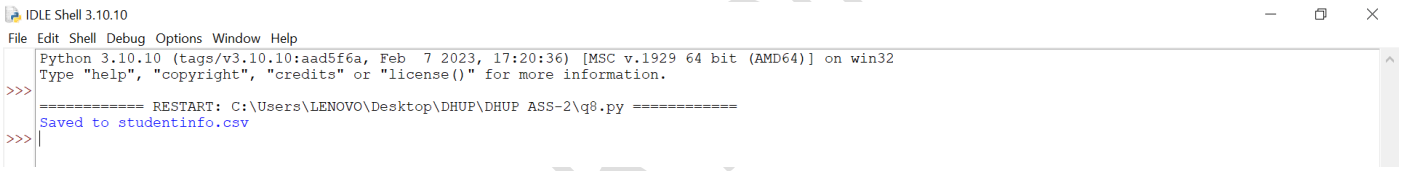
d. Maximum

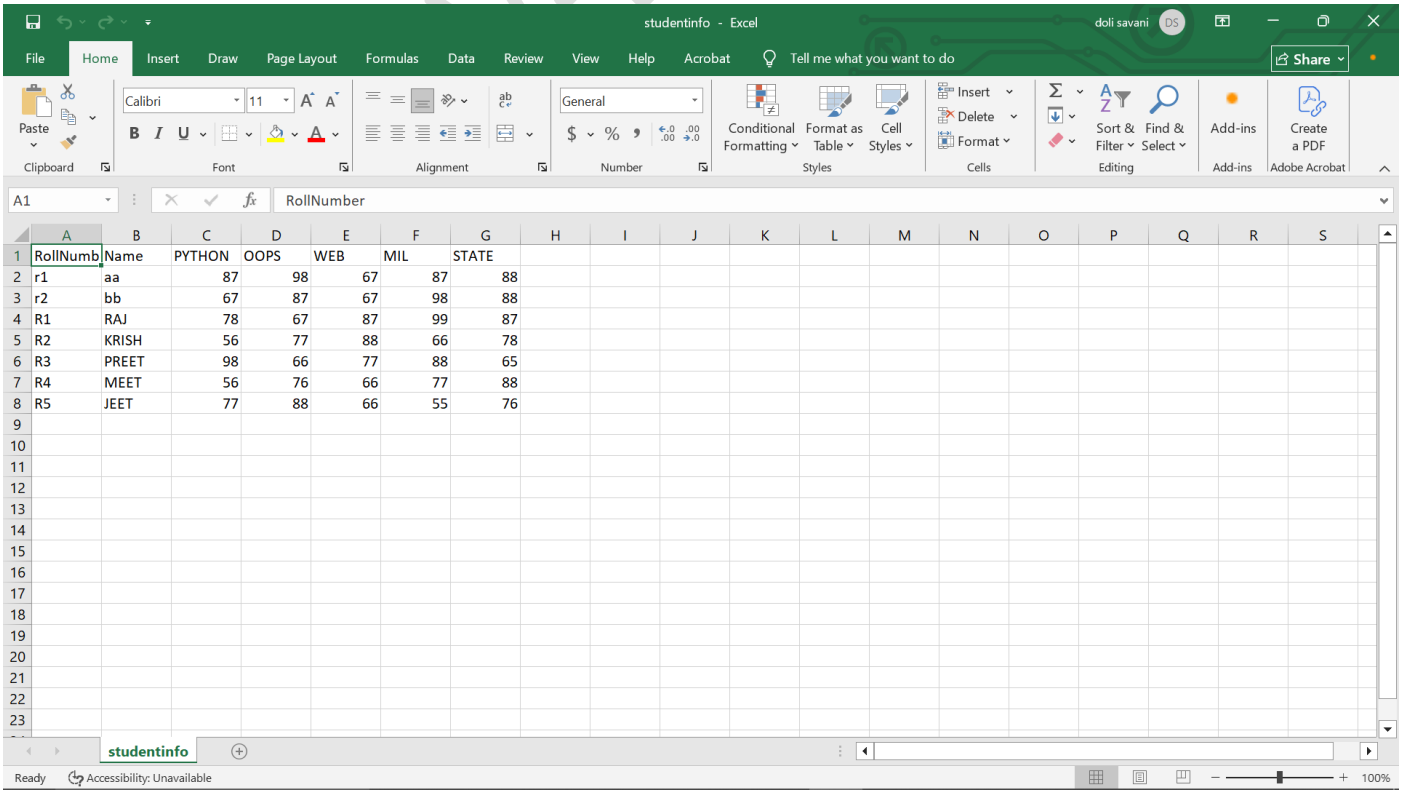
```
import sqlite3
import pandas as pd


try:
    conn = sqlite3.connect("Student_Information.db")
    df = pd.read_sql_query("SELECT * FROM Student", conn)
    df['Total'] = df[['PYTHON','OOPS','WEB','MIL','STATE']].sum(axis=1)
    df['Percentage'] = df['Total'] / 5
    df['Minimum'] = df[['PYTHON','OOPS','WEB','MIL','STATE']].min(axis=1)
    df['Maximum'] = df[['PYTHON','OOPS','WEB','MIL','STATE']].max(axis=1)
    print(df)
except Exception as e:
    print("Error adding columns:", e)
finally:
    conn.close()
```



8. Write a python code to write the data frame in the csv file. Name csv file as “studentinfo.csv”

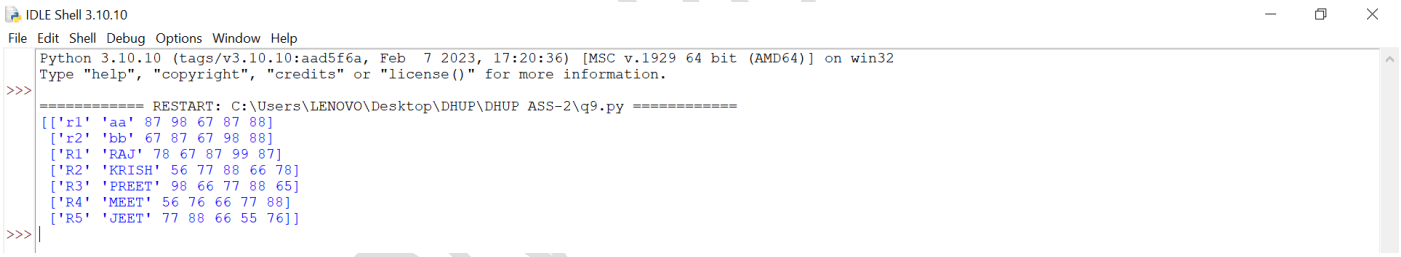
```
import sqlite3
import pandas as pd

try:
    conn = sqlite3.connect("Student_Information.db")
    df = pd.read_sql_query("SELECT * FROM Student", conn)
    df.to_csv("studentinfo.csv", index=False)
    print("Saved to studentinfo.csv")
except Exception as e:
    print("Error saving CSV:", e)
finally:
    conn.close()
```





9. Convert the data frame into the numpy ndarray and display it.

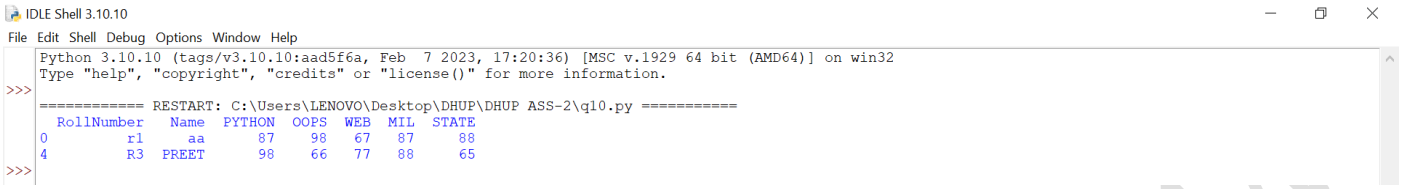
```
import sqlite3
import pandas as pd
import numpy as np

try:
    conn = sqlite3.connect("Student_Information.db")
    df = pd.read_sql_query("SELECT * FROM Student", conn)
    arr = df.to_numpy()
    print(arr)
except Exception as e:
    print("Error converting to NumPy:", e)
finally:
    conn.close()
```



10. Display all details of student who get more than 85 marks in Python.

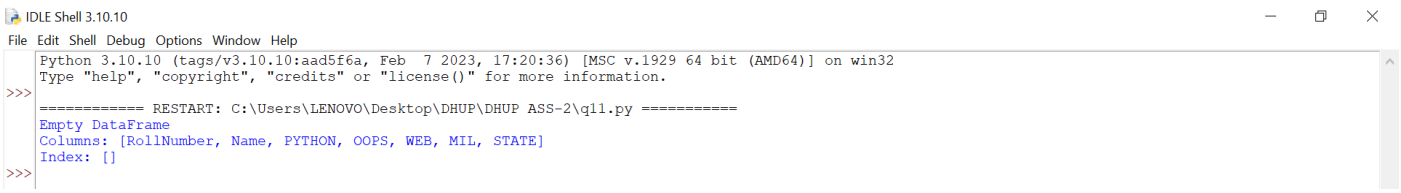
```
import sqlite3
import pandas as pd

try:
    conn = sqlite3.connect("Student_Information.db")
    df = pd.read_sql_query("SELECT * FROM Student", conn)
    print(df[df['PYTHON'] > 85])
except Exception as e:
```

```
    print("Error filtering data:", e)
finally:
    conn.close()
```



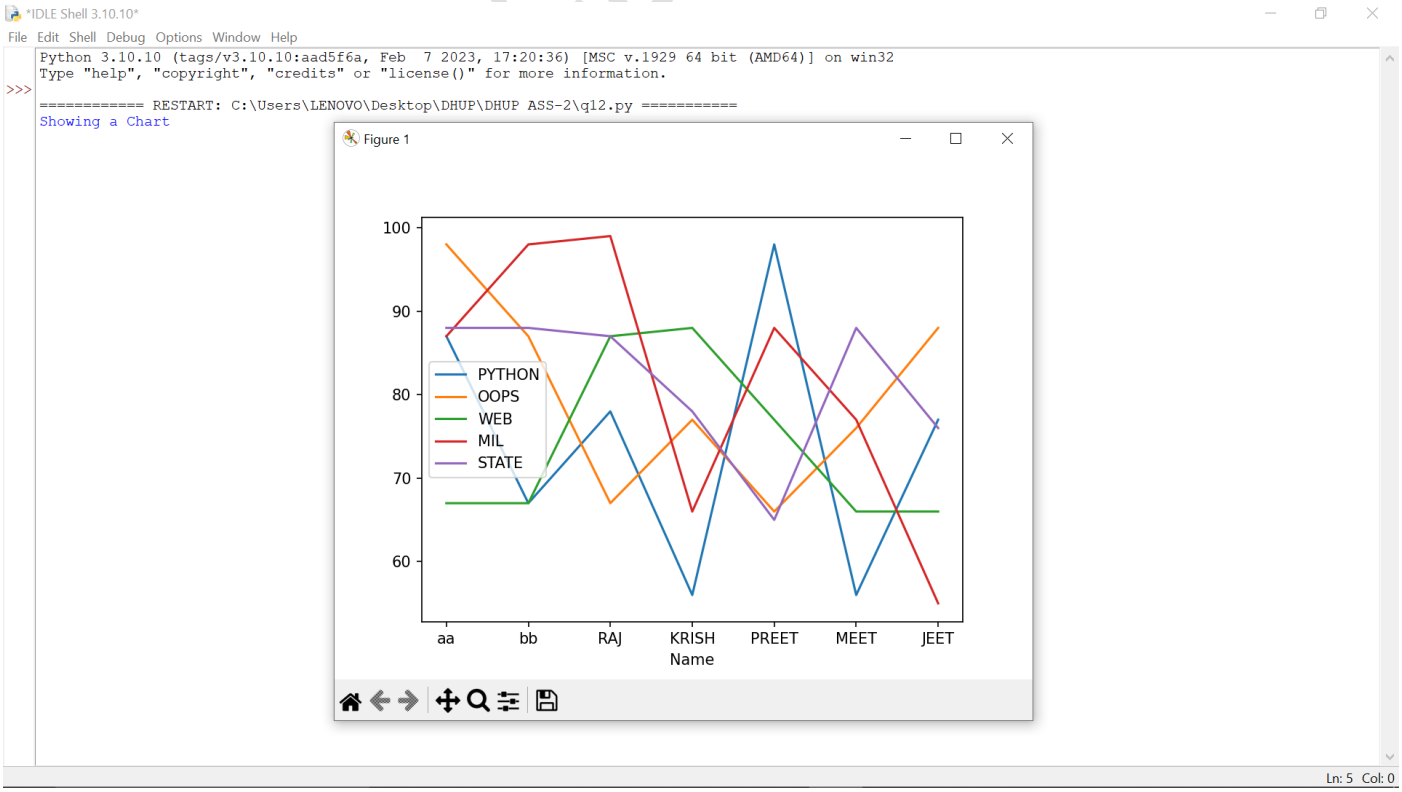
11. Write a python code to display all the students who get less than 36 Marks in any subject.

```
import sqlite3
import pandas as pd

try:
    conn = sqlite3.connect("Student_Information.db")
    df = pd.read_sql_query("SELECT * FROM Student", conn)
    low = df[(df['PYTHON'] < 36) | (df['OOPS'] < 36) | (df['WEB'] < 36) | (df['MIL'] < 36) | (df['STATE'] < 36)]
    print(low)
except Exception as e:
    print("Error filtering data:", e)
finally:
    conn.close()
```



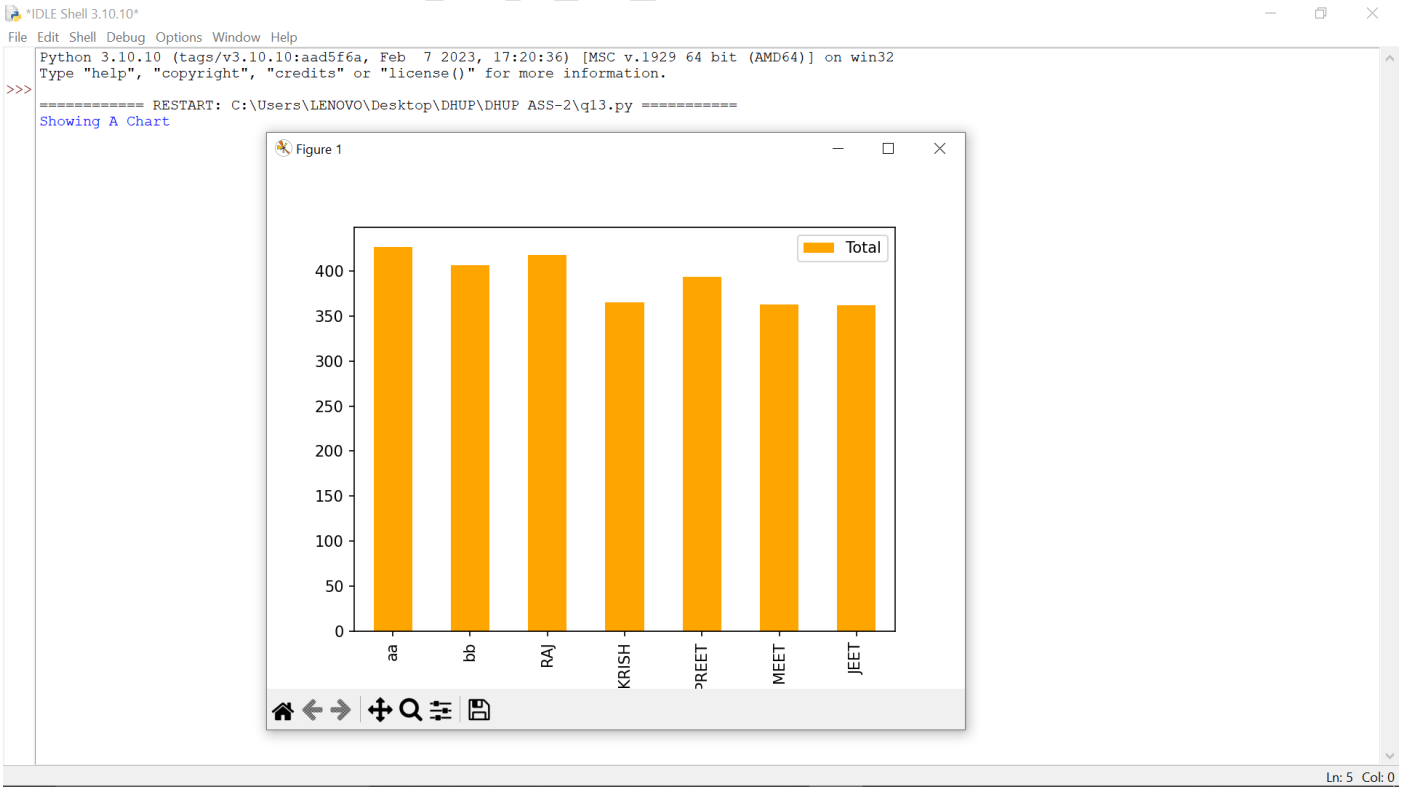
12. Write a python code to plot all the marks of students in line chart. Use appropriate legend to describe the lines.

```
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt

try:
    conn = sqlite3.connect("Student_Information.db")
    df = pd.read_sql_query("SELECT * FROM Student", conn)
    df.plot(x='Name', y=['PYTHON','OOPS','WEB','MIL','STATE'], kind='line')
    print("Showing a Chart")
    plt.show()
    print("Chart Created Successfuly")
except Exception as e:
    print("Error plotting chart:", e)
finally:
    conn.close()
```



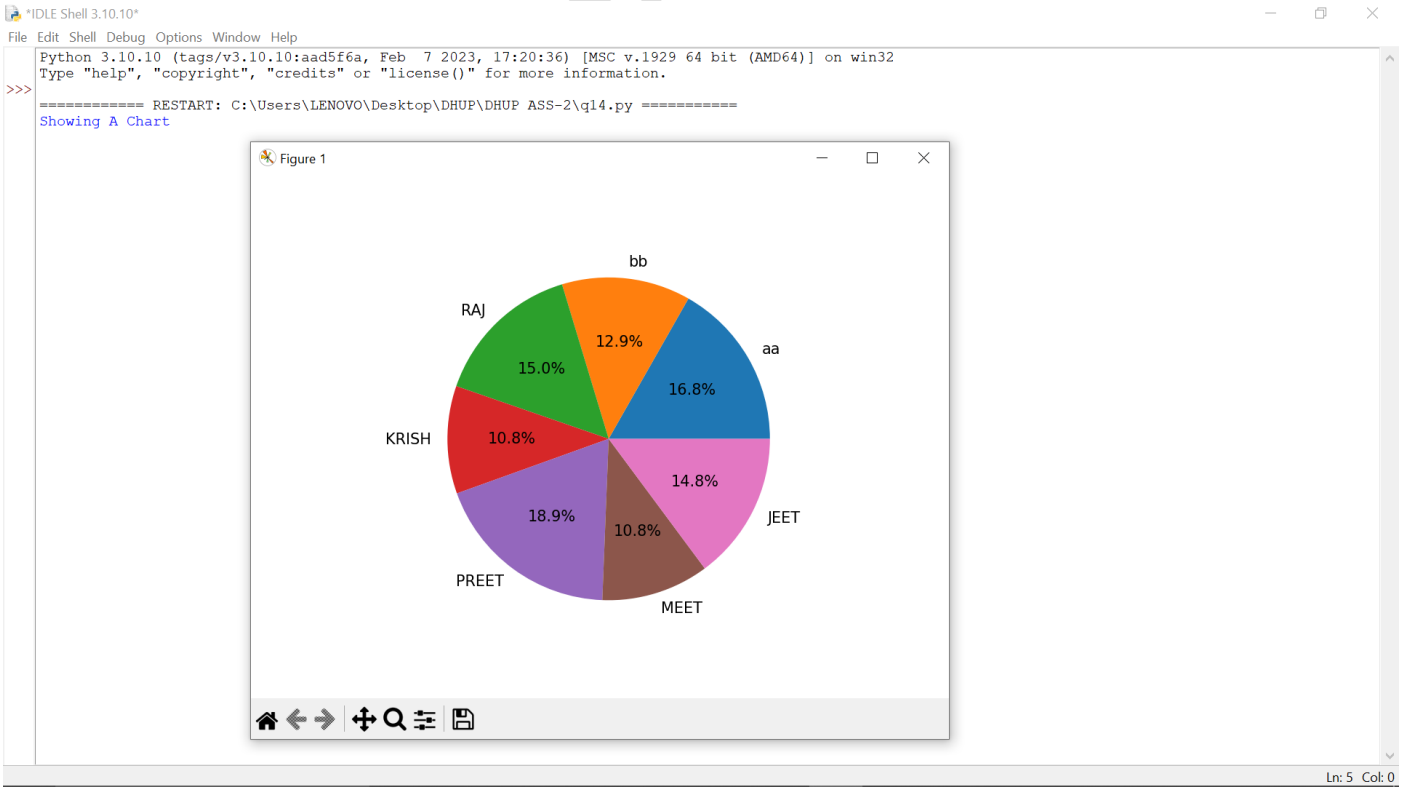
13. Create a bar chart of the student's Total marks.

```
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt

try:
    conn = sqlite3.connect("Student_Information.db")
    df = pd.read_sql_query("SELECT * FROM Student", conn)
    df['Total'] = df[['PYTHON','OOPS','WEB','MIL','STATE']].sum(axis=1)
    df.plot(x='Name', y='Total', kind='bar', color='orange')
    print("Showing A Chart")
    plt.show()
    print("Chart Created Successfuly")
except Exception as e:
    print("Error plotting bar chart:", e)
finally:
    conn.close()
```



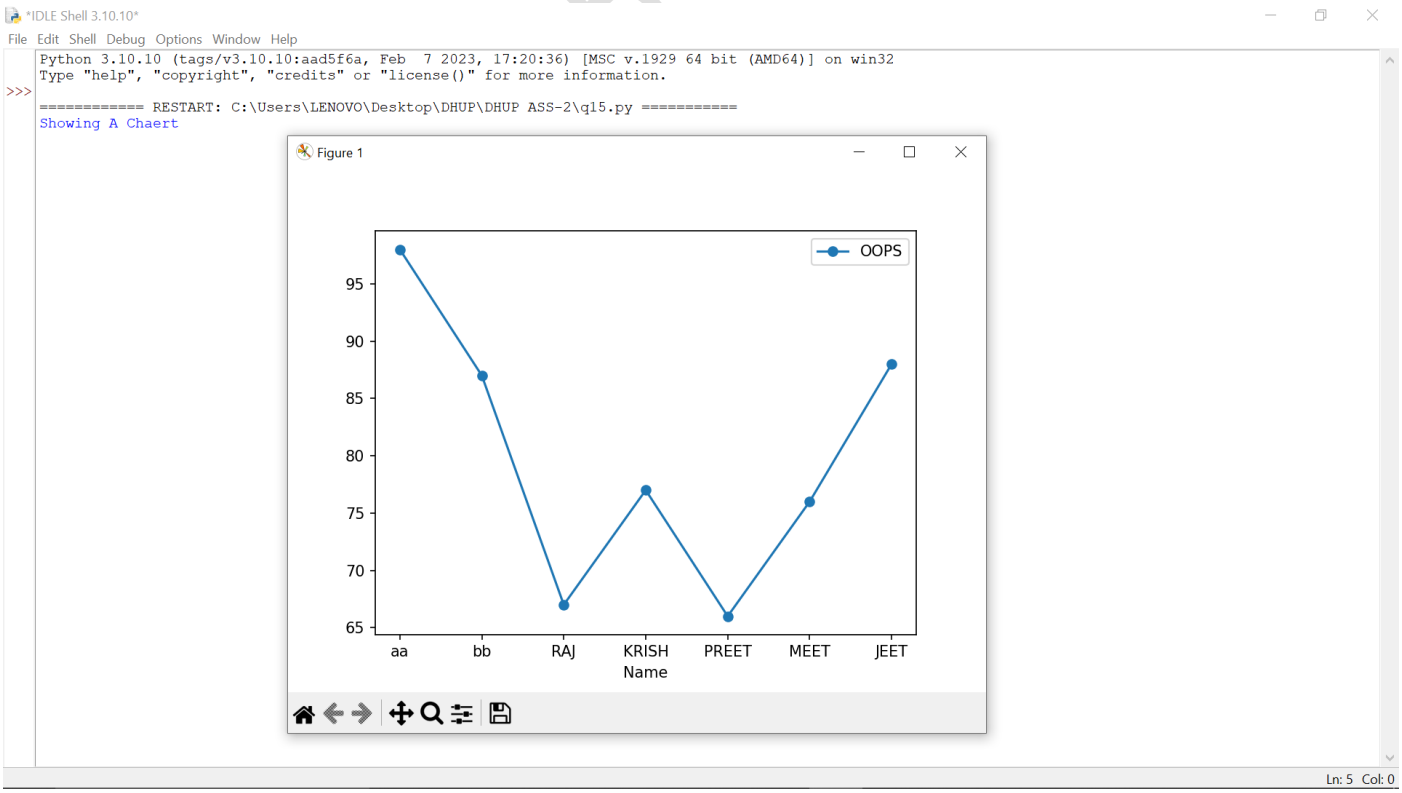
14. Create pie charts for the PYTHON subject.

```
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt

try:
    conn = sqlite3.connect("Student_Information.db")
    df = pd.read_sql_query("SELECT * FROM Student", conn)
    plt.pie(df['PYTHON'], labels=df['Name'], autopct='%1.1f%%')
    print("Showing A Chart")
    plt.show()
    print("Chart Created Successfuly")
except Exception as e:
    print("Error plotting pie chart:", e)
finally:
    conn.close()
```

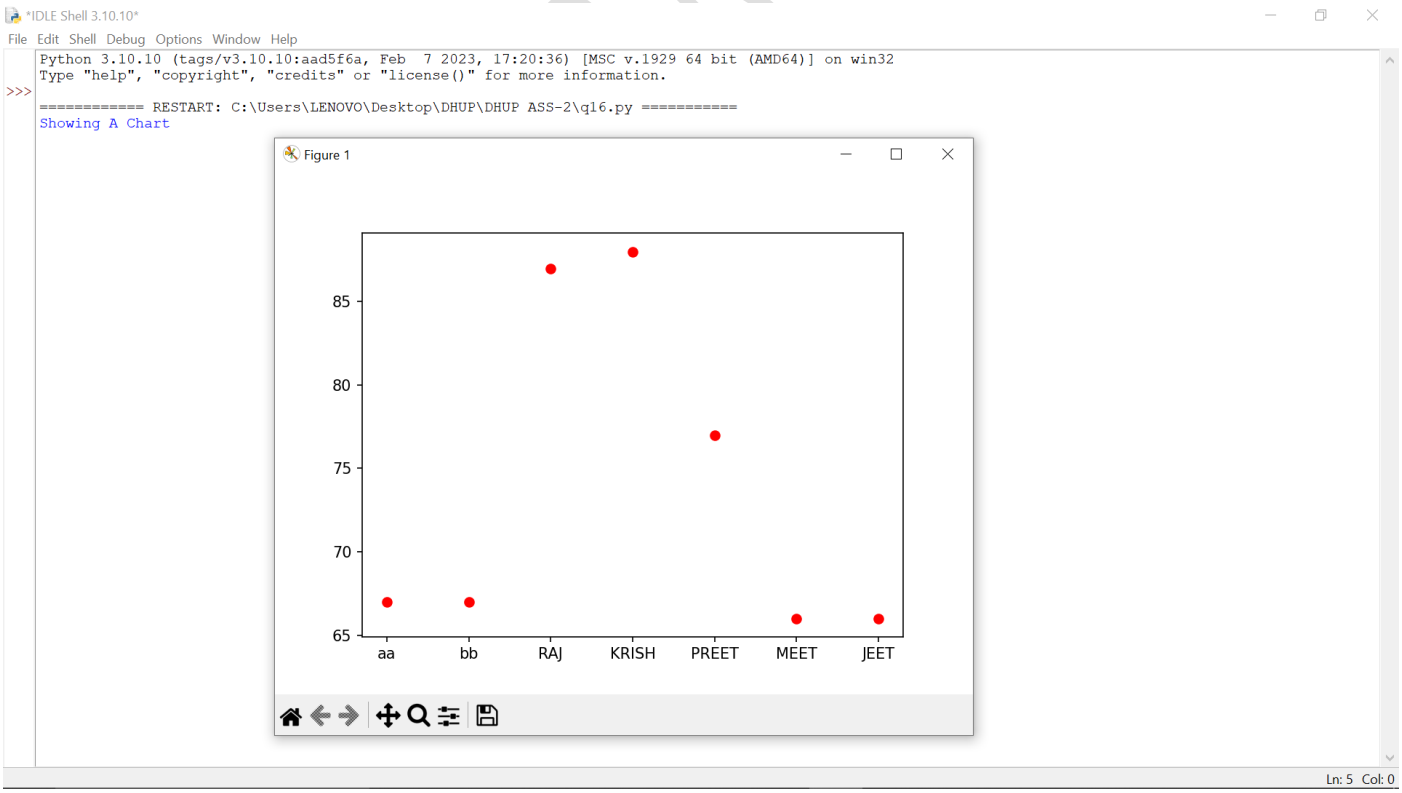
15. Create line chart for the OOPS marks.

```
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt

try:
    conn = sqlite3.connect("Student_Information.db")
    df = pd.read_sql_query("SELECT * FROM Student", conn)
    df.plot(x='Name', y='OOPS', kind='line', marker='o')
    print("Showing A Chaert")
    plt.show()
    print("Chart Created Successfuly")
except Exception as e:
    print("Error plotting OOPS chart:", e)
finally:
    conn.close()
```

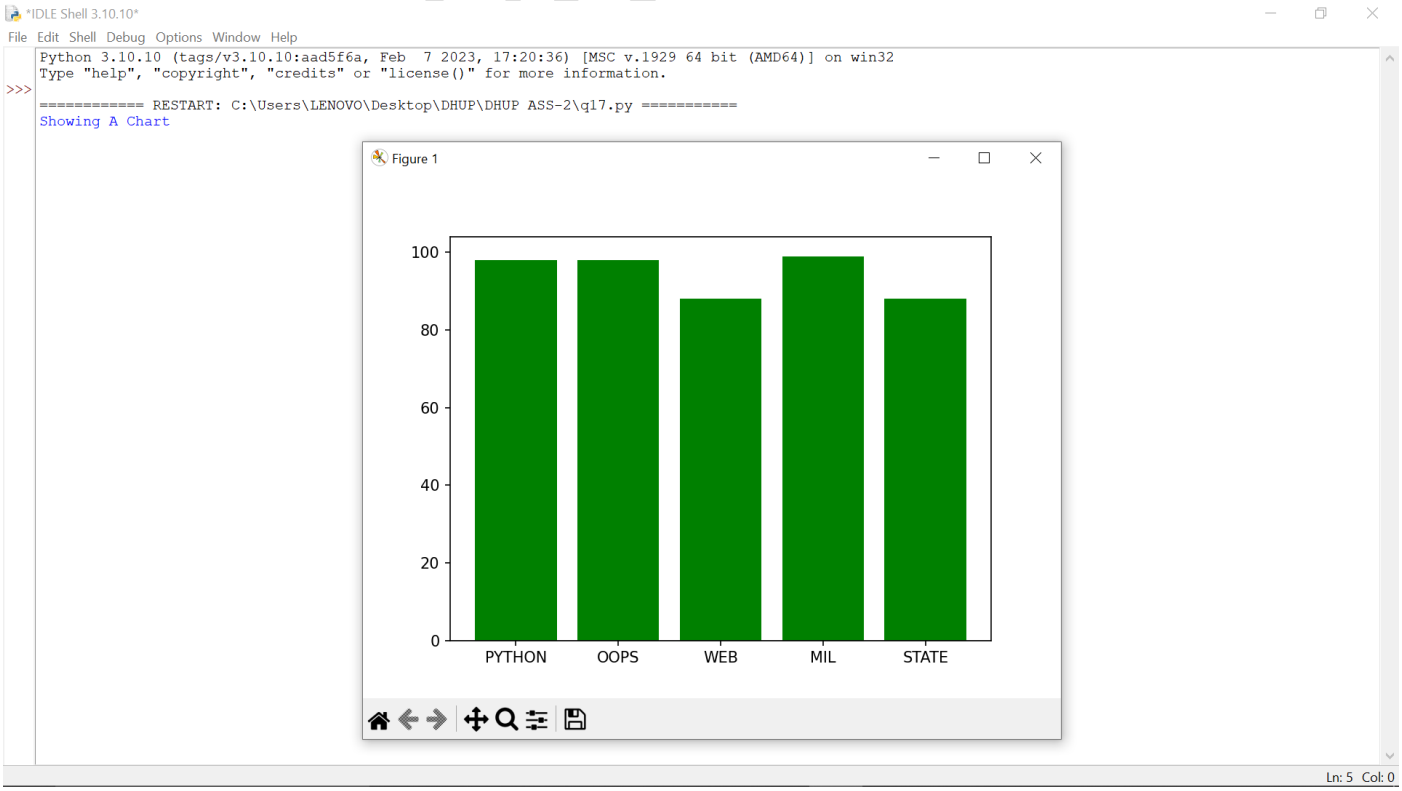
16. Create a scatter plot chart for the web marks.

```
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt
try:
    conn = sqlite3.connect("Student_Information.db")
    df = pd.read_sql_query("SELECT * FROM Student", conn)
    plt.scatter(df['Name'], df['WEB'], color='red')
    print("Showing A Chart")
    plt.show()
    print("Chart Created Successfuly")
except Exception as e:
    print("Error plotting scatter:", e)
finally:
    conn.close()
```



17. Find highest marks in each subject and make a bar chart of the same.

```
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt
try:
    conn = sqlite3.connect("Student_Information.db")
    df = pd.read_sql_query("SELECT * FROM Student", conn)
    subjects = ['PYTHON','OOPS','WEB','MIL','STATE']
    max_marks = [df[s].max() for s in subjects]
    plt.bar(subjects, max_marks, color='green')
    print("Showing A Chart")
    plt.show()
    print("Chart Created Successfuly")
except Exception as e:
    print("Error plotting highest marks:", e)
finally:
    conn.close()
```



18. Create a pie chart of the student's Total marks and save this marks in a file named as "result.png"

```
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt
try:
    conn = sqlite3.connect("Student_Information.db")
    df = pd.read_sql_query("SELECT * FROM Student", conn)
    df['Total'] = df[['PYTHON','OOPS','WEB','MIL','STATE']].sum(axis=1)
    plt.pie(df['Total'], labels=df['Name'], autopct='%1.1f%%')
    plt.savefig("result.png")
    print("Pie chart saved as result.png In Your Folder")
except Exception as e:
    print("Error saving pie chart:", e)
finally:
    conn.close()
```