SET :1

1. Create the below tables along with key Constraints and write and Insert Script for insertion of rows with Substitution variables and insert appropriate data.

```
Create table department
(
Dept_no numeric,
Dept-name varchar (10),
Location varchar (10);
);
```

```
1. Insert into department
values (1, 'Sales', 'Delhi'),
       (2, 'IT'., 'Mumbai),
       (3, 'Production', 'Mumbai'),
       (4, 'Marketing', 'Ahmadabad'),
       (5, 'Analysis', 'Surat'),
       (6, 'BCA', 'MP'),
       (7, 'BBA', 'Baroda');
```

2. query:-

1. Display all Department belonging to location 'Surat'.

```
→ Select * From Department
where location like "Surat";
```

2. list all department name Starting with 'A'.

```
→ Select * from Department
where dept_name like "a%";
```

3. list all Department whose number is between 1 and 100.

```
→ Select * From Department
where dept-no between 1 and 100;
```

4. Delete 'TRG' Department.

```
→ Delete From Department
where dept_name = "TRG";
```

5 Change Department name 'BBA' to 'IT'.

```
→ update Department
Set dept-name = "BBA"
where dept-name = "IT";
```

6 update the location whose dept_name Second letter is 'a'

```
→ update Department
Set location = "goa"
where dept_name like "-a%";
```

7 Display whose location is 'Baroda'. 'Surat', and 'Ahemdabad'.

→ Select * From department
where location = "Surat" or location = "baroda",
or location = "Ahemdabad";

8 Display whose are not from "sales" and marketing' department

→ Select * from department
where dept_name not like "Sales"
and dept_name not like "Marketing";

9 list all records of each table in as order

→ Select * From Department order by Department.
Select * from employee order by emp_name;

10. Employee (emp-id, emp-name gender, dept_no, address
designation, salary, experience, email);

→ Create table employee (emp_id numeric, emp-name
varchar (20), designation varchar (20), gender
varchar (10), dept_no int references depertment
(dept_no), address varchar (20), Salary numeric,
experience numeric, email varchar (20);

```
=> Insert into employee values
(1, "Bachi", "Female", 1, "dubai", "ceo", 8000, 5,
 "Bachi@gmail.com"),
(2, "Diya", "female", "us", "worker", 10,000, 6,
 "Diya@gmail.com")
(3, "PanKaj", "male", "us", "worker", 20,000, 7,
 "PanKaj@gmail.Com");
(4, "Parth", "Male", "dubai", "ceo", 50,000, 8,
 "Parth@gmail.Com");
(5, "Rohan", "Male", "Bardoli", "doctor" 70,000, 9,
 "Rohan@gmail.Com");
```

✲ query :-

10. Display female employee list

```
→ Select * from employee
where gender = "female";
```

11. Display all record by order emp-name

```
=> Select * from Employee
order by emp-name. asc;
```

12. Find name of employee who has salary less than 5000 and greater than 2000

```
=> Select emp_name from employee
where salary <5000 and salary >2000;
```

13. Display names and the designation of all Female employee in designation order.

=> Select emp_name, designation from employee
where gender = "Female"
order by emp_no desc;

14. Display name of all the employee who name Start with 'A' ends with 'A';

=> Select emp_name from employee
where emp_name like 'a%'

Or

=> emp_name like '%a';

15 Find the name of employee and salary for these who had obtain maximum salary.

=> Select emp-name, salary from employee
where salary (Select min (salary) from employee)

=> Select emp-name, min (salary) as "lowest salary" from employee;

16. Add 10% raise in (Salary of all employee whoes Department is 'IT'.

```
=> update employee
set Salary = (Salary * 100)/100
where designation = "manger";
```

```
=> update employee set Salary = Salary +
(Salary * 10/100) where dept_no =
(Select dept_name from dept_name = "it");
```

17. list name of employee who are fresher's (less than 1 year of experience)

```
=> Select emp_name from employee
where experience <1;
```

18. list department wise name of employee who has more than 5 year of experiance.

```
=> Select emp_name, experiance from employee
where experireance <5;
```

```
=> Select employee.emp_name from department
inner join employee on
department.dept_no = employee.dept_no
where experience >5
order by dept_name asc;
```

19 list department having no employee

```
=> Select * from department
   where dept_no
   not in (Select dept_no from employee);
```

20 Delete the employee whoes Salary is less than 10000

```
=> Delete from employee
   where Salary <10,000;
```

SET - 2

Create the below three tables along with key Constraints and write an Insert Script for insertions of rows with Substitution variables and insert appropriate data.

STUDENT (rollno, name, class, birthdate)

COURSE (Courseno, Coursename, max_Mark, Pass_Mark)

SC (rollno, Courseno, marks)

```
        Create table students
        (
         rollno int primary key
         name varchar (20),
         class varchar (20),
         Birthdate date);
```

```
insert into Students values
(1, "Rachi" , "Fy" , "1 - Jun - 2003")
(2, "Parth" , "Fy" , "2 - APr - 2003")
(3 , "Pankaj" , "Fy" , "12 - Nov - 2004")
(4, "Diya" , "Sy" , " 5 - Jan - 2004")
(5 , "Mansi" , "Sy" , " 14 - feb - 2004")
(6, "Jhanvi", "Sy" , " 6 - mar 2004")
```

```
Create table Coursec
(
Courseno int primary key
Coursename varchar,
Max Marks int,
pass marks int);

Insert into Course values
(101, "Maths", 100, 35),
(102, "DBMS", 100, 35),
(201, "CN", 100,35),
(202, "CPPM", 100, 35),
(301, "OS", 100, 35),
(302, " PS", 100, 35),
(303, "practical" ,300, 150);

pragma Foreign_Keys =on;
Create table SC2
(
rollno int references Students (rollno),
Courseno int references Course (Courseno),
marks int
);

Insert into SC2 values
(1,101, 60), (1,102, 55);
(2,101, 90), (2,102, 88);
(3,101, 40), (5,102, 34);
(4,201, 72), (4,202, 70);
```

(5, 201, 55), (5, 202, 38);
(6, 201, 88), (6, 202, 74);
(7, 301, 41), (7, 302, 44);
(8, 301, 65), (8, 302, 73);
(9, 301, 50), (9, 302, 61);

1. Display details of Student who takes 'DBMS' Course.

⇒ Select Students, rollno-name, class, birthdate, Coursec(coursename From ((Students inner joint SC2 on Students.rollno = SC2.rollno) inner joint Course on SC2 Courseno = Course.Courseno) where Coursename = "DBMS";

2. Display the name of Student who have Scored more than 70% in Computer Networks and have not failed in any Subject.

⇒ Select name from ((Students inner joint SC2 on Students.rallno = SC2.rollno) inner Joint Coursee on SC2. Courseno = Courcec-Courecena) where Courcee. Coursename = "Computer Network" and SC2.marks >= (70 * 100)/100 and SC2, Marks >= 35;

3. Display the average marks obtained by each Student.

⇒ Select Students. name, avg (SC_2. marks) as "Avg" Marks" from Students inner Joints SC_2 on Students. rollno = SC_2. rollno group by Students.name;

4. Select all Courses where passing marks are more than 30% OF average maximum marks.

⇒ Select Coursename from Coursee where pass.marks = (Select avg (max_ marks)* 30/100 from Coursee);

5. Display all Course name.

⇒ Select Coursename from Coursee;

6 Display the Student detail who have Secure 1st rank in 'Computer Network' Course

⇒ Select Students. rollno, Students.names, Students class, Students. birthdate, max (SC_2. marks) as "max marks" from Students inner Joint SC_2
on Students.
rollno= SC_2 rollno
where SC_2. Courseno = 201;

7. Display all SY Student list along with Course name.

=> Select distinct Students.name, Students.class Coursec.Coursename from ((Students inner joint SC2 on Students.rollno = SC2.rollno) inner joint Coursee on SC2.Courseno = Coursee.Courseno) where Students.class = "SY" group by ~~Students~~ students.name;

8. Display The Average Marks obtained by each student.

=> Select Students.name, avg (SC2.marks) as "Avg Marks" from Students inner Joint SC2 on Students.rollno = SC2.roll no group by Students.name;