```
CREATE DATABASE BANK_INFO_358

--LAB 2---

CREATE TABLE DEPOSIT
(
 ACTNO INT,
 CNAME VARCHAR(50),
 BNAME VARCHAR(50),
 AMOUNT DECIMAL(8,2),
 ADATE DATETIME

)

CREATE TABLE CUSTOMERS
(
CNAME
VARCHAR(50),
CITY
VARCHAR(50)
)


CREATE TABLE BRANCH
(
 BNAME VARCHAR(50),
 CITY VARCHAR(50)

)
```

```
CREATE TABLE BORROW
(
LOANNO
INT,
CNAME
VARCHAR(50),
BNAME
VARCHAR(50),
AMOUNT
DECIMAL(8,2)
)

INSERT INTO DEPOSIT
VALUES (101,'ANIL','VRCE',1000.00,'1995-3-1'),
          (102,'SUNIL','AJNI',5000.00,'1996-1-4'),
          (103,'MEHUL','KAROLBAGH',3500.00,'1995-11-17'),
          (104,'MADHURI','CHANDI',1200.00,'1995-12-17'),
          (105,'PRMOD','M.G. ROAD',3000.00,'1996-3-27'),
    (106,'SANDIP','ANDHERI',2000.00,'1996-3-31'),
          (107,'SHIVANI','VIRAR',1000.00,'1995-9-5'),
          (108,'KRANTI','NEHRU PLACE',5000.00,'1995-7-2'),
    (109,'MINU','POWAI',7000.00,'1995-8-10')


INSERT INTO BRANCH
VALUES ('VRCE','NAGPUR'),
          ('AJNI','NAGPUR'),
          ('KAROLBAGH','DELHI'),
    ('CHANDI','DELHI0'),
    ('DHARAMPETH','NAGPUR'),
    ('M.G. ROAD','BANGLORE'),
    ('ANDHERI','BOMBAY'),
    ('VIRAR','BOMBAY'),
    ('NEHRU PLACE','DELHI'),
    ('POWAI','BOMBAY')

INSERT INTO CUSTOMERS
  VALUES('ANIL','CALCUTTA'),
                    ('SUNIL','DELHI'),
                    ('MEHUL',' BARODA'),
                    ('MANDAR',' PATNA '),
      ('MADHURI ','NAGPUR'),
                    ('PRAMOD','NAGPUR '),
                    ('SANDIP',' SURAT '),
                    ('SHIVANI',' BOMBAY'),
                    ('KRANTI',' BOMBAY' ),
                    ('NAREN',' BOMBAY')

INSERT INTO BORROW
  VALUES(201,'ANIL','VRCE',1000.00),
                    (206,'MEHUL','AJNI',5000.00),
                    (311,'SUNIL','DHARAMPETH',3000.00),
                    (321,'MADHURI','ANDHERI',2000.00),
                    (375,'PRMOD','VIRAR',8000.00),
                    (481,'KRANTI','NEHRU PLACE',3000.00)

SELECT*FROM BORROW
SELECT*FROM CUSTOMERS
```

```sql
SELECT*FROM BRANCH

SELECT*FROM DEPOSIT


--LAB 2.1 From the above given tables perform the following queries using SELECT statement:


--1. Retrieve all data from table DEPOSIT.

SELECT * FROM Deposit


--2. Retrieve all data from table BORROW.

SELECT * FROM Borrow


--3. Retrieve all data from table CUSTOMERS.

SELECT * FROM Customers


--4. Display Account No, Customer Name & Amount from DEPOSIT.

SELECT Actno, Cname, Amount FROM Deposit


--5. Display Loan No, Amount from BORROW.

SELECT Loanno, Amount FROM Borrow


--6. Display loan details of all customers who belongs to ‘ANDHERI’ branch.

SELECT * FROM Borrow

WHERE Bname ='ANDHERI'


--7. Give account no and amount of depositor, whose account no is equals to 106.

SELECT Actno, Amount FROM Deposit

WHERE Actno=106


--8. Give name of borrowers having amount greater than 5000.
```

```sql
SELECT Cname FROM Borrow
WHERE Amount>5000

--9. Give name of customers who opened account after date '1-12-96'.
SELECT Cname FROM Deposit
WHERE Adate > '1996-12-01'

--10. Display name of customers whose account no is less than 105.
SELECT Cname FROM Deposit
WHERE Actno<105

--11. Display name of customer who belongs to either ‘NAGPUR’ Or ‘DELHI’. (OR & IN)
--Using OR:
SELECT Cname FROM Customers
WHERE City='nagpur' OR City='delhi'
--Using IN:
SELECT Cname FROM Customers
WHERE City IN('nagpur','delhi')

--12. Display name of customers with branch whose amount is greater than 4000 and account no is less than 105.
SELECT Cname, Bname FROM Deposit
WHERE Amount>4000 AND Actno<105

--13. Find all borrowers whose amount is greater than equals to 3000 & less than equals to 8000.
--(AND & BETWEEN)
--Using AND:
SELECT * FROM Borrow
WHERE Amount>=3000 AND Amount<=8000
```

```sql
--Using BETWEEN:

 SELECT * FROM Borrow

WHERE Amount BETWEEN 3000 AND 8000


--14. Find all depositors who do not belongs to ‘ANDHERI’ branch.

SELECT * FROM Deposit

WHERE Bname <> 'andheri'


--15. Display the name of borrowers whose amount is NULL.

SELECT Cname FROM Borrow

WHERE Amount IS NULL


--16. Display Account No, Customer Name & Amount of such customers who belongs to ‘AJNI’, ‘KAROLBAGH’ Or ‘M.G. ROAD’ and Account No is less than 104.

SELECT Actno, Cname, Amount FROM Deposit

WHERE Bname IN ('ajni','karolbagh','m.g.road') AND Actno<104


--17. Display all the details of first five customers.

SELECT TOP 5 * FROM Customers


--18. Display all the details of first three depositors whose amount is greater than 1000.

SELECT TOP 3 * FROM Deposit

WHERE Amount>1000


--19. Display Loan No, Customer Name of first five borrowers whose branch name does not belongs to ‘ANDHERI’.

SELECT TOP 5 Loanno, Cname FROM Borrow

WHERE Bname <> 'andheri'
```

```sql
--20. Retrieve all unique cities using DISTINCT. (Use Customers Table)
SELECT DISTINCT City FROM Customers


--21. Retrieve all unique branches using DISTINCT. (Use Branch Table)
SELECT DISTINCT Bname FROM Branch
SELECT Bname FROM Branch
select *from BRANCH
--22. Retrieve all the records of customer table as per their city name in ascending order.
SELECT * FROM Customers
ORDER BY City;


--23. Retrieve all the records of deposit table as per their amount column in descending order.
SELECT * FROM Deposit
ORDER BY Amount DESC;


--24. Retrieve all the details of customers in which descending order of their city name.
SELECT * FROM Customers
ORDER BY City DESC;


--25. Show all unique borrowers & label the column UNI_Borrowers. (Display only Names)
--1.
 SELECT DISTINCT Cname [UNI_Borrowers] FROM Borrow;
--2.
 SELECT DISTINCT Cname AS UNI_Borrowers FROM Borrow;



/*LAB 2.2 From the above given tables perform the following queries using UPDATE statement:*/


--1. Update deposit amount of all customers from 3000 to 5000.
```

```
UPDATE Deposit SET Amount=5000

WHERE Amount=3000


--2. Change branch name of ANIL from VRCE to C.G. ROAD. (Use Borrow Table)


UPDATE Borrow SET Bname='c.g.road'

WHERE Cname='anil' AND Bname='VRCE'


--3. Update Account No of SANDIP to 111 & Amount to 5000.


UPDATE Deposit SET Actno=111, Amount=5000

WHERE Cname='sandip'


--4. Give 10% Increment in Loan Amount.


UPDATE Borrow SET Amount=Amount+(Amount*10/100)


--5. Update deposit amount of all depositors to 5000 whose account no between 103 & 107.


UPDATE Deposit SET Amount=5000

WHERE Actno>103 AND Actno<107


--6. Update amount of loan no 321 to NULL.


UPDATE Borrow SET Amount=NULL

WHERE Loanno=321


--7. Change Loan number from 201 to 401 & also change branch name from VRCE to AJNI.
```

```
UPDATE Borrow SET Loanno=401, Bname='AJNI'

WHERE Loanno=201 AND Bname='VRCE'



--8. Modify customer name ANIL to ANIL JAIN.


UPDATE Customers SET Cname = 'ANIL JAIN'

WHERE Cname = 'ANIL'


--9. Change Name to Ramesh, Branch Name VRCE & Amount 5500, Whose Account Number is 102.

UPDATE Deposit SET Cname='Ramesh', BName ='VRCE', Amount=5500

WHERE Actno=102



--10. Make Branch Name & Amount NULL, Whose Loan Number is 481 & Name is KRANTI.


UPDATE Borrow SET Bname = NULL, Amount = NULL

WHERE Loanno = 481 AND Cname = 'KRANTI'





/*LAB 2.3 From the above given tables perform the following queries using DELETE/DROP/TRUNCATE
statement:*/


--1. Delete records of Customer table who belongs to BOMBAY City.


DELETE FROM Customer WHERE City = 'BOMBAY'
```

```
--2. Delete all account numbers whose amount less than equals to 1000.

DELETE FROM Deposit WHERE Amount <=1000

--3. Delete borrowers whose branch name is ‘AJNI’.

DELETE FROM Borrow WHERE Bname = 'AJNI'

--4. Delete all the borrowers whose loan number between 201 to 210.

DELETE FROM Borrow WHERE Loanno>201 AND Loanno<210

--5. Delete customers who opened account after date '1-12-96'. (Use DEPOSIT table)

DELETE FROM Deposit WHERE Adate > '1996-12-01'

--6. Delete all the records of Customers table. (Use TRUNCATE)

TRUNCATE TABLE Customers

--7. Remove all customers whose name is ANIL & Account Number is 101.

DELETE FROM Deposit WHERE Cname = 'ANIL' AND Actno = 101

--8. Delete all the depositors who do not belongs to ‘ANDHERI’ branch.

DELETE FROM Deposit WHERE Bname <> 'ANDHERI'
```

```
--9. Delete all the borrowers whose loan amount is less than 2000 and does not belong to VRCE branch.

DELETE FROM Borrow WHERE Amount < 2000 AND Bname <> 'VRCE'

--10. Remove Branch table. (Use DROP)

DROP TABLE Branch



--LAB 5
CREATE TABLE students
(StuID
INT,
FirstName
VARCHAR(25),
LastName
VARCHAR(25),
Website
VARCHAR(50),
City
VARCHAR(25),
Division
VARCHAR(20),
)
INSERT INTO Students
VALUES (1011,'Keyur','Patel','techonthenet.com','Rajkot','II-BCX'),
                (1022,'Hardik','Shah','digminecraft.com','Ahmedabad','I-BCY'),
                (1033,'Kajal','Trivedi','bigactivities.com','Baroda','IV-DCX'),
                (1044,'Bhoomi','Gajera','checkyourmath.com','Ahmedabad','III-DCW'),
                    (1055,'Harmit','Mitel','NULL','Rajkot','II-BCY'),
                    (1066,'Ashok','Jani','NULL','Baroda','II-BCZ')

SELECT*FROM Students

--1. Display the name of students whose name starts with ‘k’.
  SELECT*from Students
        where FirstName like 'k%'

--2. Display the name of students whose name consists of five characters.
        select FirstName from Students
        where FirstName like '%'

--3. Retrieve the first name & last name of students whose city name ends with a & contains six
characters.
        select * from students
        where FirstName like '_____';
--4. Display all the students whose last name ends with ‘tel’.
        select * from Students
        where LastName like '%tel'

--5. Display all the students whose first name starts with ‘ha’ & ends with ‘t’.
        select * from Students
        where FirstName like 'ha%t'

--6. Display all the students whose first name starts with ‘k’ and third character is ‘y’.
        select * from Students
        where FirstName like 'k_y%'
```

--7. Display the name of students having no website and name consists of five characters.

```
Select * from students

where FirstName like '_____'

and Website is NULL;
```

--8. Display all the students whose last name consist of ‘jer’.

```
SELECT * from Students

where LastName like '%jer%'
```

--9. Display all the students whose city name starts with either ‘r’ or ‘b’.

```
select * from students

where City like 'r%' or City like 'b%';
```

--10. Display all the name students having websites.

```
select FirstName from Students

where website is not NULL;
```

--11. Display all the students whose name starts from alphabet A to H.

```
select * from students

where FirstName like '[a-h]%';
```

--12. Display all the students whose name’s second character is vowel.

```
select * from students

where FirstName like '_[a,e,i,o,u]%';
```

--13. Display student’s name whose city name consist of ‘rod’.

```
select FirstName from students

where City like '%rod%';
```

--14. Retrieve the First & Last Name of students whose website name starts with ‘bi’.

```
select FirstName, LastName from students
where Website like 'bi%';
```

--15. Display student’s city whose last name consists of six characters.

```
select City from students
where LastName like '______';
```

--16. Display all the students whose city name consist of five characters & not starts with ‘ba’.

```
select * from students
where City like '_____' and City not like 'ba';
```

--17. Show all the student’s whose division starts with ‘II’.

```
select * from students
where Division like 'II%';
```

--18. Find out student’s first name whose division contains ‘bc’ anywhere in division name.

```
select FirstName from students
where Division like '%bc%';
```

--19. Show student id and city name in which division consist of six characters and having website name.

```
select Stuid,city from students
where Division like '______' and Website is not NULL;
```

--20. Display all the students whose name’s third character is consonant.

```
select * from students
where FirstName not like '__[a,e,i,o,u]%';
```

--lab 3

```
create table student_1
(enrollment_no varchar(20),
 name varchar(25),
 cpi decimal(5,2),
 birthdate datetime
 );

 /*
        alter command
        syntax 1: to add new column to table
                alter table table_name
                add newcol_name1 datatype(size),

        syntax 2 : to droop existing column
                alter table table_name
                drop column column_name;

        synatx 3 : to change the datatype and size of the columns
                alter table table_name
                alter column column_name new datatype(newsize);
        */
        select *from student

--1.
 alter table student_1
        add city varchar(20),backlog int;

--2.
```

```
 alter table student_1
            alter column name varchar(35);

--3.
alter table student_1
            alter column cpi int;

--4.
sp_rename 'student_1.enrollment_no','eno'

--5.
alter table student_1
            drop column city;

--6.
sp_rename 'student_master','student_1'
            select * from student_master



---7.
alter table student_1
            drop column backlog

--8.
alter table student_1
            alter column name varchar(50) not null

--9.
sp_rename 'student_1.birthdate','bdate'
```

```
--10.
alter table student_1
alter column eno varchar(12)

/* rename commnad

syntax 1: to rename column  name
        sp_rename 'table_name.old_col_name','new_col_name'

synatax 2: to rename table  name
        sp_rename 'old_table_name','new_table_name'

*/

select*from student_1

--ASSINMENET

CREATE TABLE student_assi
(
        StuID decimal(3, 0),
        Name VARCHAR(50),
        EnrollmentNo decimal(12, 0),
        Division VARCHAR(50),
        Sem INT ,
        BirthDate DATETIME,
        Email VARCHAR(50),
        ContactNo decimal(10, 0)
)

INSERT INTO Student_assi
VALUES  (101,'Naimish Patel',090200107051,'BCX-3',3,'1991-12-
06','naimishp49@gmail.com',8866566768),
              (102,'Firoz A. S.',090200107090,'BCY-3',3,'1994-05-
03','Firoz.me@gmail.com',8888999922),
              (103,'Krunal Vyas',090200107101,'BCZ-5',5,'1984-03-
01','Krunal.vyas@gmail.com',9999888877),
              (104,'Vijay Patel',090200107102,'BCX-5',5,'1985-02-
15','Vijay.patel123@gmail.com',8787878787),
              (105,'Vimal Trivedi',090200107103,'BCY-3',3,'1988-01-20','Vimal123@gmail.com',NULL)

SELECT*FROM Student_assi

--1. Display all student details who belongs to ‘bcx-3’ division.

 SELECT * FROM student_assi
 WHERE Division = 'BCX-3';



--2. Display name of student who belongs to either semester 3 or 5. (or & in)

 SELECT Name FROM student_assi
 WHERE Sem = 3
   OR sem = 5;

SELECT Name FROM student_assi
 WHERE Sem IN (3,5);
```

```
--3. Find all students whose name does not starts with ‘v’.

  SELECT Name FROM student_assi
  WHERE Name NOT LIKE 'v%';


--4. Display all the details of first three students.

  SELECT TOP 3 * FROM student_assi;


--5. Display unique divisions.

  SELECT DISTINCT Division FROM student_assi;


--6. Display name, enrollment no of first 30% students who’s contact number ends with 7.

  /*SELECT Name,EnrollmentNo TOP 30% FROM Student_assi
  WHERE ContactNo LIKE '%7';
*/

--7. Find all students in which email contains ‘3@g’ with division bcz-5.

  SELECT * FROM student_assi
  WHERE Email LIKE '3@g'
  AND Division = 'BCZ-5';
```

```
--8. Display student id, name & email address of all students.

  SELECT StuID,Name,Email FROM student_assi;


--9. Find students after born date 01-01-1990.

  SELECT * FROM student_assi
  WHERE BirthDate > 1990-01-01;


--10. Display student name in which student belongs to semester 3 and enrollment number
    --contains ‘107’.

  SELECT Name from student_assi
  WHERE Sem IN (3)
    AND EnrollmentNo LIKE '%107%';


--11. Display student name & division name of all students, also do arrange the list in
    --ascending order.

  SELECT Division,Name FROM student_assi
  ORDER BY Division ASC
  SELECT Division,Name FROM student_assi
   ORDER BY Name ASC;
```

--12. Display all the students in which mobile number is empty.

```
  SELECT * FROM student_assi
  WHERE ContactNo IS NULL;
```

--13. Show all unique semesters & label the column Semester.

```
  SELECT DISTINCT Sem  AS Semester FROM student_assi;

  SELECT DISTINCT Sem [Semester] FROM student_assi;
```

--14. Display name, enrollment no & semester of such students who belongs to division ‘BCX-3’,‘BCZ-5’ Or ‘BCY-3’ and Birthdate after 15-3-86.

```
  SELECT Name,EnrollmentNo,Sem FROM student_assi
  WHERE Division IN ('BCX-3','BCZ-5','BCY-3')
  AND BirthDate > 1986-03-15;
```

--15. Retrieve all the records of student table as per their Birthdate column in descending order

```
  SELECT * FROM student_assi
  ORDER BY BirthDate DESC;
```

```sql
  --lab 6a


   --Math functions

--1. Display the result of 5 multiply by 30.

SELECT 5*30 AS MULTIPLY


--2. Find out the absolute value of -25, 25, -50 and 50.

SELECT ABS(-25),ABS(50),ABS(-50),abs(25)


--3. Find smallest integer value that is greater than or equal to 25.2, 25.7 and -25.2.

SELECT CEILING(25.2),CEILING(25.7),CEILING(-25.2)


--4. Find largest integer value that is smaller than or equal to 25.2, 25.7 and -25.2.

SELECT FLOOR(25.2),FLOOR(-25.2)


--5. Find out remainder of 5 divided 2 and 5 divided by 3.

SELECT 5%2,5%3


--6. Find out value of 3 raised to 2nd power and 4 raised 3rd power.

SELECT POWER(3,2),POWER(4,3)


--7. Find out the square root of 25, 30 and 50.

SELECT SQRT(25),SQRT(30),SQRT(50)


--8. Find out the square of 5, 15, and 25.

SELECT SQUARE(5),SQUARE(15),SQUARE(25)


--9. Find out the value of PI.
```

```
SELECT PI()
--10. Find out round value of 157.732 for 2, 0 and -2 decimal points.
SELECT ROUND(157.732,2), ROUND(157.732,-2), ROUND(157.732,0)

--11. Find out exponential value of 2 and 3.
SELECT EXP(2),EXP(3)

--12. Find out logarithm having base b having value 10 of 5 and 100.
SELECT LOG10(5),LOG10(100),LOG10(10)
SELECT LOG(1),LOG(2)

--13. Find sine, cosine and tangent of 3.1415.
SELECT SIN(3.14),COS(3.14),TAN(3.14)

--14. Find sign of -25, 0 and 25.
SELECT SIGN(0),SIGN(-25),SIGN(25)

--15. Generate random number using function



--lab6b


--1.Find the length of following. (I) NULL (II) ‘ hello ’ (III) Blank
select len('') as length,len(null),len('   demo   ')

--2. Display your name In lower & upper case.
select lower ('NIRAV')

--3. Display first three characters of your name.
```

```sql
select upper ('nirav') as my_name

--4. Display 3rd to 10th character of your name.
select SUBSTRING('nirav kagathara',3,7)

--5. Write a query to convert ‘abc123efg’ to ‘abcXYZefg’ & ‘abcabcabc’ to ‘ab5ab5ab5’ using REPLACE.
select stuff ('abc123efg',4,3,'XYZ')
select stuff ('abcabcabc',3,100,'5ab5ab5')

--6. Write a query to display ASCII code for ‘a’,’A’,’z’,’Z’, 0, 9.
select Ascii('a'),ascii('A'),ascii('z'),ascii('Z'),ascii('0'),ascii('9')

--7. Write a query to display character based on number 97, 65,122,90,48,57.
select char(65),char(97),char(122),char(90),char(48),char(57)

--8. Write a query to remove spaces from left of a given string ‘hello world ‘.
select lTRIM ( '    hello   ')

--9. Write a query to remove spaces from right of a given string ‘ hello world ‘.
select rTRIM ( '    hello   ')

--10. Write a query to display first 4 & Last 5 characters of ‘SQL Server’.
select left ('SQL Server',4)
select right ('SQL Server',5)

--11. Write a query to convert a string ‘1234.56’ to number (Use CAST()).
select cast ('1234.56' as decimal(6,2))

--12. Write a query to convert a float 10.58 to integer (Use CONVERT()).
```

```sql
select convert (int,10.58)

--13. Put 10 space before your name using function.
select space(10)+'nirav'

--14. Combine two strings (Your name & Surname) using + sign as well as CONCAT ().
select 'ANSH' + ' ' + 'PATEL'

select CONCAT('ANSH',' ','PATEL')

--15. Find reverse of “Darshan”.
SELECT REVERSE('Darshan')

--16. Repeat your name 3 times.
select REPLICATE('nirav ',3)

--17. Delete 3 characters from a string, starting in position 1, and then insert "HTML" in position 1. (Use
--STUFF())
select stuff ('12345',1,3,'HTLM')

--18. From Data, returns the first non-null value in a list. (Use COALESCE())
select COALESCE(null,123,0112)

--19. Tests whether the expression is numeric. (Use ISNUMERIC())
select ISNUMERIC('asd')

--20. Search for "t" in string "Customer", and return its position. (Use CHARINDEX())
select CHARINDEX('t','Customer',1)
```

```
------LAB-7b------

CREATE TABLE Student_Marks
(           Sid int not null,
            SName varchar (50) not null,
            Marks int not null
)
INSERT INTO Student_Marks
            VALUES (1, 'John',90),
                          (2 ,'Martin',80),
                          (3, 'Carol', 89),
                          (4 ,'Jack', 99),
                          (5 ,'Rose', 88),
                          (6 ,'Mary', 90)




---LAB-7a------

--1
SELECT GETDATE() AS 'TODAY_DATE'
--2
SELECT GETDATE()+365
--3
SELECT CONVERT(VARCHAR,GETDATE())
--4
SELECT CONVERT(VARCHAR,GETDATE(),106)
```

```
--5
SELECT CONVERT(VARCHAR,GETDATE(),7)
--6
SELECT DATEDIFF(MONTH,'31-DEC-08','31-MAR-09')
--7
SELECT DATEDIFF(YEAR,'25-JAN-12','14-SEP-10')
--8
SELECT DATEDIFF(HOUR,'25-JAN-12 7:00','26-Jan-12 10:30')
--9
SELECT DAY('12-May-16'),MONTH('12-May-16'),YEAR('12-May-16')
--10
SELECT DATEADD(YEAR,5,GETDATE())
--11
SELECT DATEADD(MONTH,-2,GETDATE())
--12
SELECT DATENAME(MONTH,GETDATE())
Select DATEPART(MONTH, GETDATE())
--13
SELECT EOMONTH(GETDATE())
--14
SELECT DATEADD(DAY,30,GETDATE())
SELECT GETDATE()+30
--15
SELECT DATEDIFF(YEAR,'21-MAY-05','22-mar-23')
SELECT DATEDIFF(MONTH,'21-MAY-05','22-MAR-23')



--agrigate function baki che
--7b
```

```
--1. Find total number of students.

Select COUNT(*) AS TotalStudents FROM Student_Marks

--2. Find total of marks scored by all students.

Select SUM(Marks) AS TotalMarks FROM Student_Marks

--3. Find average marks of all students.

Select AVG(Marks) AS AvgMarks FROM Student_Marks

--4. Find minimum marks scored from all students.

Select MIN(Marks) AS Min_Marks FROM Student_Marks

--5. Find maximum marks scored from all students.

Select MAX(Marks) AS Max_Marks FROM Student_Marks



--lab8--


CREATE TABLE Employee1

(

        EID int,

        EName varchar(50),

        Department varchar(50),

        Salary Decimal(8,2),

        JoiningDate datetime,

        City varchar(50)

)


insert into Employee

 values (101, 'Rahul', 'Admin', 56000, '1-Jan-90', 'Rajkot'),

                (102, 'Hardik', 'IT', 18000, '25-Sep-90', 'Ahmedabad'),

                (103, 'Bhavin', 'HR', 25000, '14-May-91', 'Baroda'),

                (104, 'Bhoomi', 'Admin', 39000, '8-Feb-91', 'Rajkot'),
                    (105, 'Rohit', 'IT', 17000, '23-Jul-90', 'Jamnagar'),
                    (106, 'Priya', 'IT', 9000, '18-Oct-90', 'Ahmedabad'),
                    (107, 'Neha', 'HR', 34000, '25-Dec-91', 'Rajkot')

select * from Employee1

--1. Display the Highest, Lowest, Total, and Average salary of all employees & label the columns Maximum,
--Minimum, Total_Sal and Average_Sal, respectively.
select max(salary) as Max_salary, min(salary) as Min_salary,
avg(salary) as Average_salary,sum(salary) as total_salary from Employee1

--2. Find total number of employees of EMPLOYEE table.
select count(EName) as total_employee from employee1

--3. Retrieve maximum salary from IT department.
select max(salary) as Maximum_salary from Employee1
where Department='it'

--4. Count total number of cities of employee without duplication.
select  COUNT(distinct City) as total_city from Employee1

--5. Display city with the total number of employees belonging to each city.
select city, COUNT(City) as total_city from Employee1
group by City

--6. Display city having more than one employee.
select city,COUNT(city) from Employee1
group by City
```

```sql
having count(EName)>1

--7. Give total salary of each department of EMPLOYEE table.
select department,sum(salary) from employee1
group by Department

--8. Give average salary of each department of EMPLOYEE table without displaying the respective
department name.
select avg(salary) from Employee1
group by Department

--9. Give minimum salary of employee who belongs to Ahmedabad.
select min(salary) from Employee1
where City='Ahmedabad'

--10. List the departments having total salaries more than 50000 and located in city Rajkot.
select department,SUM(salary) from Employee1
where City='Rajkot'
group by Department
having sum(salary)>50000

--11. Count the number of employees living in Rajkot.
select COUNT(ename) from Employee1
where City='Rajkot'

--12. Display the difference between the highest and lowest salaries. Label the column name to
SAL_DIFFERENCE.
select max(salary)-min(salary) as SAL_DIFFERENCE from Employee1
```

```
--13. Display the total number of employees hired before 1st January, 1991.

select COUNT(ename) from Employee1

where JoiningDate<'1-jan-91'
```

```
--14. Display total salary of each department with total salary exceeding 35000 and sort the list by total salary.

select department,sum(salary) from Employee1

group by Department

having sum(salary)>35000

order by sum(salary)
```

```
--15. List out department names in which more than two employees.

select Department,COUNT(Ename) from Employee1

group by department

having count(EName)>2
```

```
--16. Display Minimum salary of Rajkot City.

select min(salary) from Employee1

where city='Rajkot'
```

```
--17. Display City wise total employees count.

select city,count(ename) from employee1

group by City
```

```
--18. List all departments with minimum salaries.

select department,min(salary) from Employee1

group by Department
```

```
--19. Give Total salaries of each city without displaying the respective city name.
```

```sql
select distinct city,sum(salary) from Employee1

group by City


--20. Find department wise Minimum, Maximum & Total Salaries.

select department,min(salary) as min_salary ,max(salary) as max_salary,sum(salary) as total_salary from Employee1

group by Department




----LAB-9------set operator set operator


CREATE TABLE COMPUTER

(

        RollNo Int,

        Name varchar (50)

)

CREATE TABLE ELECTRICAL

(

        RollNo Int,

        Name varchar (50)

)

INSERT INTO COMPUTER

       VALUES(101,'Ajay'),

              (109, 'Haresh'),

              (115 ,'Manish')

INSERT INTO ELECTRICAL

       VALUES(105 ,'Ajay'),

              (107 ,'Mahesh'),
                              (115 ,'Manish')


--SELECT NAME FROM COMPUTER
--UNION

--SELECT * FROM COMPUTER

--SELECT * FROM ELECTRICAL

--1.
SELECT NAME FROM COMPUTER
        UNION
        SELECT NAME FROM ELECTRICAL

--2.
 SELECT NAME FROM COMPUTER
        UNION ALL
        SELECT NAME FROM ELECTRICAL

--3.
 SELECT NAME FROM COMPUTER
        INTERSECT
        SELECT NAME FROM ELECTRICAL

--4.
        SELECT NAME FROM COMPUTER
        EXCEPT
        SELECT NAME FROM ELECTRICAL
```

```
--5.

 SELECT NAME FROM ELECTRICAL

            EXCEPT

            SELECT NAME FROM COMPUTER


--6.

 SELECT *FROM COMPUTER

            UNION

            SELECT * FROM ELECTRICAL


--7.

 SELECT * FROM COMPUTER

            INTERSECT

            SELECT * FROM ELECTRICAL



//-- SELECT INTO


---LAB-9 ------


CREATE TABLE CRICKET

( Name varchar (50),

  City varchar (50),

  Age Int

)

INSERT INTO CRICKET

VALUES('Sachin Tendulkar', 'Mumbai', 30),

              ('Rahul Dravid ','Bombay', 35),
     ('M. S. Dhoni' ,'Jharkhand', 31),
     ('Suresh Raina' ,'Gujarat' ,30)

 SELECT * FROM CRICKET



 --1.
  SELECT * INTO WORDCUP FROM CRICKET


 --2.
  SELECT NAME,CITY INTO T20 FROM CRICKET
            WHERE CITY='SURAT';


 --3.
  SELECT * INTO IPL FROM CRICKET
            WHERE 1=2;


 --4.
INSERT INTO IPL SELECT * FROM Cricket WHERE Name LIKE '_A_____%'
 --5.
 TRUNCATE TABLE IPL


 --6.
 DELETE FROM CRICKET
            WHERE CITY='JHARKHAND'


 --7.
 SP_RENAME 'IPL','IPL2018'
```

```
 --8.
 DROP TABLE T20



 --LAB10
CREATE TABLE
Student (Rno int, Name varchar (50), Branch varchar (50))

create table
Result (RNo int, SPI Decimal (4,2))

create table
Employee (EmployeeNo varchar (50), Name varchar (50), ManagerNo varchar (50))

INSERT INTO student
VALUES
(101,'Raju','CE'),
(102,'Amit','CE'),
(103,'Sanjay','ME'),
(104,'Neha','EC'),
(105,'Meera','EE'),
(106,'Mahesh','ME')

insert into Result
values
(101,8.8),
(102,9.2),
(103,7.6),
(104,8.2),
(105,7.0),
(107,8.9)

insert into Employee
values
('E01','Tarun',NULL),
('E02','Rohan','E02'),
('E03','Priya','E01'),
('E04','Milan','E03'),
('E05','Jay','E01'),
('E06','Anjana','E04')


--1. Combine information FROM student and result table using cross join or Cartesian product.
SELECT *
FROM Student
CROSS JOIN Result

SELECT *
FROM Student, Result
--2. Display Rno, Name, Branch and SPI of CE branch’s student only.
SELECT
Student.Rno,
Student.Name,
Student.Branch,
Result.SPI
FROM Student
LEFT OUTER JOIN Result
```

```
ON Student.Rno = Result.Rno
WHERE Student.Branch='CE'
--3. Display Rno, Name, Branch and SPI of other than EC branch’s student only.
SELECT
Student.Rno,
Student.Name,
Student.Branch,
Result.SPI
FROM Student
LEFT OUTER JOIN Result
ON Student.Rno = Result.Rno
WHERE Student.Branch <> 'EC'
--4. Display the average result of each branch.
SELECT
Student.Branch,
AVG(Result.SPI) As Avg_Spi
FROM Student
INNER JOIN Result
ON Student.Rno = Result.Rno
GROUP BY Student.Branch
--5. Display the average result of each branch and sort them in ascending ORDER BY SPI.
SELECT
Student.Branch,
AVG(Result.SPI) As Avg_Spi
FROM Student
INNER JOIN Result
ON Student.Rno = Result.Rno
GROUP BY Student.Branch
ORDER BY AVG(Result.SPI)
```

```
--6. Display average result of CE and ME branch.
SELECT
Student.Branch,
AVG(Result.SPI) As Avg_Spi
FROM Student
INNER JOIN Result
ON Student.Rno = Result.Rno
AND (Student.Branch IN ('CE','me'))
GROUP BY Student.Branch
--7. Perform the left outer join on Student and Result tables.
SELECT
Student.Rno,
Student.Name,
Student.Branch,
Result.SPI
FROM Student
LEFT OUTER JOIN Result
ON Student.Rno = Result.Rno
--8. Perform the right outer join on Student and Result tables.
SELECT
Student.Rno,
Student.Name,
Student.Branch,
Result.SPI
FROM Student
RIGHT OUTER JOIN Result
ON Student.Rno = Result.Rno
--9. Perform the full outer join on Student and Result tables.
SELECT
```

```
Student.Rno,
Student.Name,
Student.Branch,
Result.SPI
FROM Student
FULL OUTER JOIN Result
ON Student.Rno = Result.Rno
--10. Retrieve the names of employees along with their manager’s name FROM the Employee
--table.
SELECT
E.Name As EmployeeName,
M.Name As ManagerName
FROM Employee E
LEFT OUTER JOIN Employee M
ON E.ManagerNo = M.EmployeeNo

create table
City (CityID int, CityName varchar (50), Pincode varchar (6), Remarks varchar (250))

create table
Village (VillageID int, VillageName varchar (50), CityID int)

--1. Display all the villages of Rajkot city.
select V.VillageName
FROM City C
FULL OUTER JOIN Village V
ON C.CityID = V.CityID
WHERE C.CityName='Rajkot'
--2. Display city along with their villages & pin code.
```

```sql
SELECT
C.CityName,
C.Pincode,
V.VillageName
FROM City C
RIGHT OUTER JOIN Village V
ON C.CityID = V.CityID
--3. Display the city having more than one village.
SELECT
City.CityName,
COUNT(Village.CityID) AS NoOfVillages
FROM City
LEFT OUTER JOIN Village
ON City.CityID=Village.CityID
GROUP BY City.CityName
HAVING COUNT(Village.CityID)>1
--4. Display the city having no village.
SELECT
City.CityName
FROM City
LEFT OUTER JOIN Village
ON City.CityID=Village.CityID
GROUP BY City.CityName
HAVING COUNT(Village.CityID) = 0
--5. Count the total number of villages in each city.
SELECT
City.CityName,
COUNT(Village.VillageID) As TotalVillages
FROM City
```

```
LEFT OUTER JOIN Village
ON City.CityID=Village.CityID
GROUP BY City.CityName
--6. Count the number of cities having more than one village.
SELECT COUNT(*)
FROM
(
SELECT City.[CityName],
COUNT(Village.VillageID) AS TOTAL
FROM City
LEFT OUTER JOIN VillageID
ON City.CityID = VillageID.CityID
GROUP BY City. [CityName]
) AS T
WHERE TOTAL > 1



------------lab12
---Views
--1.
--Create  view Personal with all columns.
CREATE VIEW Personal
AS
SELECT * FROM Student_12
--2.
--Create a view Student_Details having columns Name, Branch & SPI.
CREATE VIEW Student_Details
AS
```

```
SELECT Name,Branch,SPI

FROM Student_12

--3.

--Create a view Academic having columns RNo, Name, Branch.

CREATE VIEW Academic

AS

SELECT RNo, Name, Branch

FROM Student_12

--4.

--Create a view Student_Data having all columns but students whose bklog more than 2.

CREATE VIEW Student_Data

AS

SELECT * FROM Student_12

WHERE Backlog > 2

--5.

--Create a view Student_Pattern having RNo, Name & Branch columns in which Name consists of four letters.

CREATE VIEW Student_Pattern

AS

SELECT RNo, Name, Branch

FROM Student_12

WHERE Name Like '____'

--6.

--Insert a new record to Academic view. (107, Meet, ME)

INSERT INTO Academic

SELECT 107, 'Meet', 'ME'

OR

INSERT INTO Academic

Values (107, 'Meet', 'ME')
```

```
--7.
--Update the branch of Amit FROM CE to ME in Student_Details view.
UPDATE Student_Details
SET Branch='ME'
WHERE Name='Amit' And Branch='CE'
--8
--Delete a student whose roll number is 104 FROM Academic view.
DELETE FROM Academic
WHERE RNo=104
--9.
--Create a view that displays information of all students whose spi is above 9.5.
CREATE VIEW [vwSPIAbove9.5]
AS
SELECT * FROM Student_12 WHERE SPI>9.5
--10.
--Create a view that displays 0 backlog students.
CREATE VIEW [vwZeroBklog]
AS
SELECT * FROM Student_12 WHERE Backlog=0
```