------*SQL ASSIGNMENT*------

B1. How to Create an Table student write an SQL Query ?

Ans.

=

```
CREATE TABLE
Student(
Rollno INT,

NAME

VARCHAR(20),

Branch

VARCHAR(30),

PRIMARY

KEY(Rollno)

);
```

B2. How to Create a Exam table with Foreign key on rollno write aSQL Query ?

Ans. =

```
CREATE TABLE
Exam(
Rollno INT,

S_Code VARCHAR(20),

Marks INT,

P_Code VARCHAR(20)

FOREIGN KEY(Rollno) REFERENCES Student(Rollno)
);
```

B3. What is SQL Key Constraints write an Example of SQL KeyConstraints ?

Ans. =

- UNIQUE KEY

- PRIMARY KEY

- FOREIGN KEY

B4. What is SQL View Create a View of Student Table ?

Ans. =

```
SELECT * FROM STUDENT;
```

B5. How to Create a Table user write a SQL

query ?Ans. =

```
CREATE TABLE
USER(
First_Name

VARCHAR(20),

Last_Name

VARCHAR(20),Address

VARCHAR(40), City

VARCHAR(20),

Age INT
);
```

B6. What is SQL and How to Create a table with Forign

Key ?Ans. =

SQL = STRUCTURED QUERY LANGUAGE

```
CREATE TABLE
table_name(
   column1

   datatype,

   column2
```

```
            datatype,column

            3 datatype,

           ....
         );
```

B7. What is store Proceedure write a query of creating storeProceedure ?

Ans. =

```
DELIMITER //

CREATE PROCEDURE EmpData_Salary()BEGIN
SELECT * FROM employee WHERE salary > 700000;END
//

DELIMITER ;


CALL EmpData_Salary;
```

B8. What is save Point How to Create a save Point write a Query ?

Ans. =

```
    START TRANSACTION;

    INSERT INTO student values (4,'John','Computer

                                                    Science')
```

```
;SAVEPOINT a;

UPDATE student SET Name = 'Johnny' WHERE Rollno

= 3;ROLLBACK TO a;
```

B9. What is trigger and how to Create a Trigger in

SQL ?Ans.=

```
CREATE TRIGGER EmpBackup AFTER DELETE
ON employee
FOR EACH
ROW

INSERT INTO empbackup
VALUES(OLD.Employee_ID,
OLD.First_Name,OLD.Last_Name,OLD.Salary,OLD.Joining_Date,O
LD.Department);
```

==
=========
==
=======

I1 -- Get First_Name from employee table using alias name "EmployeeName".

Ans.

```
SELECT  First_Name  AS  Employee_Name  FROM  employee ;
```

I2 -- Get FIRST_NAME, Joining year, Joining Month and Joining Datefrom employee table.

Ans.

```
 SELECT  First_Name  ,  year(Joining_Date) ,
month(Joining_Date),date(Joining_Date)   FROM  employee;
```

I3 -- Get all employee details from the employee table order byFirst_Name Ascending and Salary descending.

Ans.

```
SELECT* FROM employee ORDER BY

First_Name ;SELECT * FROM employee ORDER

BY Salary DESC ;
```

I4 -- Get employee details from employee table whose first namecontains ‘o’.

Ans.

```
SELECT  *  FROM  employee  WHERE  First_Name  LIKE    ' % o
% ' ;
```

I5 -- Get employee details from employee table whose joining monthis “January”.

Ans.

```
SELECT  *  FROM  employee  WHERE       month(Joining_Date)
=  '1'  ;
```

I6 -- Get department, total salary with respect to a department fromemployee table order by total salary descending.

Ans.

```
SELECT  Department  ,  sum(Salary)  FROM  employee
GROUP BYDepartment ORDER BY Salary DESC ;
```

I7 -- Get department wise maximum salary from employee table orderby salary ascending.

Ans.

```
SELECT Department , max(Salary)  FROM  employee GROUP
BYDepartment                          ORDER BY max(Salary)
          ASC ;
```

I8 -- Select first_name, incentive amount from employee and incentives table for those employees who have incentives andincentive amount greater than 3000.

Ans.

```
      SELECT employee.First_Name  ,   incentive.Incentive_Amount
FROM   employee    INNER    JOIN   incentive ON   Employee_ID
=Employee_ref_Id  WHERE       Incentive_Amount  >  3000 ;
```

I9 -- Select 2nd Highest salary from employee table.
Ans.

```
SELECT max(Salary) FROM employee WHERE Salary

 < ( SELECT max(Salary) FROM employee ) ;
```

I10 -- Select first_name, incentive amount from employee

and incentives table for all employees who got incentives using leftjoin.

Ans.

```
SELECT employee.First_Name ,
            incentive.Incentive_AmountFROM employee    L

EFT JOIN incentive  ON  Employee_ID = Employee_ref_Id
WHERE   Incentive_Amount <> 0 OR  Incentive_Amount
<>NULL;
```

I11 -- Create View OF Employee table in which store first name ,lastname and salary only.

Ans.

```
SELECT First_Name , Last_Name , Salary FROM employee ;
```

I12 -- Create Procedure to find out department wise highest salary.

Ans:- SELECT Department , max(Salary) FROM employee GROUP BYDepartment ;

I13 -- Create After Insert trigger on Employee table which insertrecords in view table.

Ans.

```
    # Create Table For BackUp Employee Details #

CREATE TABLE EmpBackup (Employee_ID
INT,First_Name VARCHAR(20),
Last_Name
VARCHAR(20),Salary
INT,
Joining_Date DATETIME,
Department
VARCHAR(20));


CREATE TRIGGER BackUpEmp AFTER DELETE
ON employee
FOR EACH
ROW
INSERT INTO empbackup VALUES
(old.Employee_Id, old.First_Name, old.Last_Name, old.
Salary,old.Joining_Date, old.Department);

DELETE FROM employee WHERE First_Name = 'Jerry';
```

==
=========
==
======

A1. All orders for more than $1000.Ans.

```
SELECT  * FROM  ORDER1  WHERE AMT  > 1000 ;
```

A2. Names and cities of all salespeople in London with commissionabove 0.10.

Ans.

```
SELECT  Sname , City  FROM  salesperson  WHERE  city =
'london'AND  comm > 0.10 ;
```

A8. All salespeople either in Barcelona or in London.Ans.

```
SELECT SName ,  City  FROM  salesperson  WHERE  City
='Barcelona'       OR               City = 'London' ;
```

A9. All salespeople with commission between 0.10 and 0.12.(Boundary values shouldbe excluded).

Ans.

```
SELECT  *  FROM  salesperson WHERE  comm
 BETWEEN                                       0.10AND
 0.12  ;
```

A10. All customers excluding those with rating <= 100 unless they arelocated in Rome.

Ans.

```
SELECT * FROM customer  WHERE  Rating  >=  100 AND    City
```

NOT IN ('Rose','Roe') ;

A11. All orders except those with 0 or NULL value in amt field.Ans.

```
SELECT * FROM order1 WHERE AMT <> 0 OR AMT <>
NULL
;
```

A12. Count the number of salespeople currently listing orders in theorder table.
Ans.

```
SELECT count(DISTINCT Sno)    AS        Salesperson
                              FROM      order1 ;
```

A13. Largest order taken by each salesperson, datewise.Ans.

```
SELECT Sno, max(AMT) AS Maximum_Order,ODE
FROMOrder1 GROUP BY ODE ORDER     BY ODE;
```

A14. Largest order taken by each salesperson with order value morethan $3000.
Ans.

```
SELECT Sno , max(AMT) FROM order1 WHERE AMT >
3000GROUP BY Sno;
```