Question 1:

Write the SQL commands to create the EMP table with the following structure:

- empno as a number datatype with up to 4 digits
- ename as a variable character datatype up to 10 characters
- job as a variable character up to 9 characters
- mgr as a number datatype with up to 4 digits
- hiredate as a date
- sal as a number with up to 7 digits, including 2 decimal places
- comm as a number with up to 7 digits, including 2 decimal places
- deptno as a number with up to 2 digits

Answer:

```
CREATE TABLE EMP (
   empno     INT(4),
   ename     VARCHAR(10),
   job       VARCHAR(9),
   mgr       INT(4),
   hiredate  DATE,
   sal       DECIMAL(7, 2),
   comm      DECIMAL(7, 2),
   deptno    INT(2)
);
```

Question 2:

Write the SQL commands to create the DEPT table with the following structure:

- deptno as a number with up to 2 digits
- dname as a variable character datatype up to 10 characters
- loc as a variable character up to 20 characters

Answer:

```
CREATE TABLE DEPT (
   deptno INT(2),
   dname  VARCHAR(10),
   loc    VARCHAR(20)
```

);

Question 3:

Create tables EMPLOYEE and DEPARTMENT with the following schema by applying Primary and Foreign key: Emp(empno as primary key, empname, salary, phoneno) Dept(deptno primary key, empno foreign key, deptname, location).

Answer:

```
CREATE TABLE EMPLOYEE (
   empno    INT PRIMARY KEY,
   empname  VARCHAR(50),
   salary   DECIMAL(10, 2),
   phoneno  VARCHAR(15)
);


CREATE TABLE DEPARTMENT (
   deptno   INT PRIMARY KEY,
   empno    INT,
   deptname VARCHAR(50),
   location VARCHAR(100),
   FOREIGN KEY (empno) REFERENCES EMPLOYEE(empno)
);
```

Question 4:

Insert the following rows of data into the EMP table using INSERT command: EMPNO: 7876, ENAME: ADAMS, DNAME: RESEARCH, JOB: CLERK, HIREDATE: 23-MAY-87, LOC: DALLAS EMPNO: 7499, ENAME: ALLEN, DNAME: SALES, JOB: SALESMAN, HIREDATE: 20-FEB-81, LOC: CHICAGO EMPNO: 7698, ENAME: SMITH, DNAME: SALES, JOB: MANAGER, HIREDATE: 01-MAY-81, LOC: CHICAGO EMPNO: 7782, ENAME: CLARK, DNAME: ACCOUNTING, JOB: MANAGER, HIREDATE: 09-JUN-81, LOC: NEW YORK Delete the record for the employee SMITH from the EMP table.

Answer:

```
INSERT INTO EMP (empno, ename, job, hiredate, loc) VALUES
   (7876, 'ADAMS', 'CLERK', '1987-05-23', 'DALLAS'),
   (7499, 'ALLEN', 'SALESMAN', '1981-02-20', 'CHICAGO'),
   (7698, 'SMITH', 'MANAGER', '1981-05-01', 'CHICAGO'),
   (7782, 'CLARK', 'MANAGER', '1981-06-09', 'NEW YORK');
```

DELETE FROM EMP WHERE ename = 'SMITH';

Question 5:

Insert the following multiple records into the EMP table using a single INSERT command: EMPNO: 7902, ENAME: FORD, DNAME: RESEARCH, JOB: ANALYST, HIREDATE: 03-DEC-81, LOC: DALLAS EMPNO: 7900, ENAME: JAMES, DNAME: SALES, JOB: CLERK, HIREDATE: 03-DEC-81, LOC: CHICAGO EMPNO: 7566, ENAME: JONES, DNAME: RESEARCH, JOB: MANAGER, HIREDATE: 02-APR-81, LOC: DALLAS EMPNO: 7839, ENAME: KING, DNAME: ACCOUNTING, JOB: PRESIDENT, HIREDATE: 17-NOV-81, LOC: NEW YORK Update the job title of the employee ADAMS to "MANAGER".

Answer:

INSERT INTO EMP (EMPNO, ENAME, DNAME, JOB, HIREDATE, LOC) VALUES

(7902, 'FORD', 'RESEARCH', 'ANALYST', '1981-12-03', 'DALLAS'),

(7900, 'JAMES', 'SALES', 'CLERK', '1981-12-03', 'CHICAGO'),

(7566, 'JONES', 'RESEARCH', 'MANAGER', '1981-04-02', 'DALLAS'),

(7839, 'KING', 'ACCOUNTING', 'PRESIDENT', '1981-11-17', 'NEW YORK');

UPDATE EMP SET JOB = 'MANAGER' WHERE ENAME = 'ADAMS';

Question 6:

Create table EMPLOYEE and DEPARTMENT with the following schema: EMP (empno, empname, salary, phno)
Dept (deptno, empno, deptname, location, jobtype)
Create user Jay identified by any admin;

- Grant create table, create view to Jay;
- Grant select, insert, update on Emp to Jay;
- Grant select, update (deptno, empno) on Dept to Jay;

Answer:

CREATE TABLE EMP (

empno INT PRIMARY KEY,

empname VARCHAR(50),

salary DECIMAL(10, 2),

phno VARCHAR(15)

);

CREATE TABLE DEPT (

```
    deptno INT PRIMARY KEY,
    empno INT,
    deptname VARCHAR(50),
    location VARCHAR(100),
    jobtype VARCHAR(30),
    FOREIGN KEY (empno) REFERENCES EMP(empno)
);

CREATE USER 'Jay'@'localhost' IDENTIFIED BY '123';

GRANT CREATE, CREATE VIEW ON mysql.* TO 'Jay'@'localhost';

GRANT SELECT, INSERT, UPDATE ON EMP TO 'Jay'@'localhost';

GRANT SELECT, UPDATE(deptno, empno) ON DEPT TO 'Jay'@'localhost';

FLUSH PRIVILEGES;
```

Question 7:

Create table EMPLOYEE and DEPARTMENT with the following schema: EMP (empno, empname, salary, phno)
Dept (deptno, empno, deptname, location, jobtype)
Create user Jay identified by any admin;
Alter user Jay identified by admin;
Revoke create table, create views from Jay;
Revoke select, insert, update on Emp from Jay;

Answer:

```
CREATE TABLE EMP (
    empno INT PRIMARY KEY,
    empname VARCHAR(50),
    salary DECIMAL(10, 2),
    phno VARCHAR(15)
);
```

```
CREATE TABLE DEPT (
   deptno INT PRIMARY KEY,
   empno INT,
   deptname VARCHAR(50),
   location VARCHAR(100),
   jobtype VARCHAR(30),
   FOREIGN KEY (empno) REFERENCES EMP(empno)
);

CREATE USER 'Jay'@'localhost' IDENTIFIED BY '123';

ALTER USER 'Jay'@'localhost' IDENTIFIED BY 'admin';

REVOKE CREATE, CREATE VIEW ON mysql.* FROM 'Jay'@'localhost';

REVOKE SELECT, INSERT, UPDATE ON EMP FROM 'Jay'@'localhost';

FLUSH PRIVILEGES;
```

Question 8:

Create table EMPLOYEE and DEPARTMENT with the following schema: EMP (empno, empname, salary, phno)
Dept (deptno, empno, deptname, location, jobtype)
Create user Jay identified by any admin;
Create role emp_pvr;
Grant create table, create views to emp_pvr;
Grant emp_pvr to Jay, John;

Answer:

```
CREATE TABLE EMP (
   empno INT PRIMARY KEY,
   empname VARCHAR(50),
   salary DECIMAL(10, 2),
   phno VARCHAR(15)
```

```
);

CREATE TABLE DEPT (
   deptno INT PRIMARY KEY,
   empno INT,
   deptname VARCHAR(50),
   location VARCHAR(100),
   jobtype VARCHAR(30),
   FOREIGN KEY (empno) REFERENCES EMP(empno)
);

CREATE USER 'Jay'@'localhost' IDENTIFIED BY '123';

CREATE USER 'John'@'localhost' IDENTIFIED BY '123';

CREATE ROLE 'emp_pvr';

GRANT CREATE, CREATE VIEW ON mysql.* TO 'emp_pvr';

GRANT 'emp_pvr' TO 'Jay'@'localhost';

GRANT 'emp_pvr' TO 'John'@'localhost';

FLUSH PRIVILEGES;
```

Question 9:

Create table Orders(cust_id, order_id, items, amount) insert required values and write queries for the following:

- Display new column named total_amount which is 200 added to the amount field.
- Display new column named offer_price which is 100 subtracted from the amount field.

Answer:

```
CREATE TABLE Orders (
```

```
    cust_id INT,
    order_id INT,
    items VARCHAR(255),
    amount DECIMAL(10, 2)
);

INSERT INTO Orders (cust_id, order_id, items, amount)
VALUES
    (1, 101, 'Laptop', 1200.00),
    (2, 102, 'Smartphone', 800.00),
    (3, 103, 'Tablet', 450.00),
    (4, 104, 'Monitor', 300.00);

SELECT
    cust_id,
    order_id,
    items,
    amount,
    (amount + 200) AS total_amount
FROM Orders;

SELECT
    cust_id,
    order_id,
    items,
    amount,
    (amount - 100) AS offer_price
FROM Orders;
```

Question 10:

Create table Orders(cust_id, order_id, items, amount) insert required values and write queries for the following:

- Display new column named revised_amount which is multiplied by 5 times the amount field.
- Display new column named half_amount which is divided by 2 to the amount field.

Answer:

```
CREATE TABLE Orders (
    cust_id INT,
    order_id INT,
    items VARCHAR(255),
    amount DECIMAL(10, 2)
);


INSERT INTO Orders (cust_id, order_id, items, amount)
VALUES
    (1, 101, 'Laptop', 1200.00),
    (2, 102, 'Smartphone', 800.00),
    (3, 103, 'Tablet', 450.00),
    (4, 104, 'Monitor', 300.00);


SELECT
    cust_id,
    order_id,
    items,
    amount,
    (amount * 5) AS revised_amount
FROM Orders;


SELECT
    cust_id,
    order_id,
    items,
```

```
    amount,
    (amount / 2) AS half_amount
FROM Orders;
```

Question 11:

Create table Emp(empno,ename,job,mgr,hiredate,sal,comm,deptno) insert required values and write queries for the following:

- Display employees whose city is ‘Mumbai’ and earns more than 50000.
- Display employees whose job is Clerk or commission is 500.

Answer:

```
CREATE TABLE Emp (
    empno INT PRIMARY KEY,
    ename VARCHAR(50),
    job VARCHAR(50),
    mgr INT,
    hiredate DATE,
    sal DECIMAL(10, 2),
    comm DECIMAL(10, 2),
    deptno INT,
    city VARCHAR(50)
);

INSERT INTO Emp (empno, ename, job, mgr, hiredate, sal, comm, deptno, city)
VALUES
    (1, 'John', 'Manager', NULL, '2020-01-15', 60000, 1000, 10, 'Mumbai'),
    (2, 'Jane', 'Clerk', 1, '2021-07-23', 30000, 500, 20, 'Pune'),
    (3, 'Alice', 'Developer', 1, '2019-05-01', 70000, 0, 10, 'Delhi'),
    (4, 'Bob', 'Clerk', 1, '2022-03-11', 25000, 500, 30, 'Mumbai');

SELECT
    empno,
```

```
    ename,
    job,
    sal,
    city
FROM Emp
WHERE city = 'Mumbai' AND sal > 50000;


SELECT
    empno,
    ename,
    job,
    comm
FROM Emp
WHERE job = 'Clerk' OR comm = 500;
```

Question 12:

Create table Emp(empno,ename,job,mgr,hiredate,sal,comm,deptno) insert required values and write queries for the following:

- Display details of employees whose salary is between 20000 and 50000.
- Display details of employees who stay at Mumbai, Pune, Nashik or Nagpur.

Answer:

```
CREATE TABLE Emp (
    empno INT PRIMARY KEY,
    ename VARCHAR(50),
    job VARCHAR(50),
    mgr INT,
    hiredate DATE,
    sal DECIMAL(10, 2),
    comm DECIMAL(10, 2),
    deptno INT,
    city VARCHAR(50)
);
```

```
INSERT INTO Emp (empno, ename, job, mgr, hiredate, sal, comm, deptno, city)
VALUES
  (1, 'John', 'Manager', NULL, '2020-01-15', 45000, 1500, 10, 'Mumbai'),
  (2, 'Jane', 'Clerk', 1, '2021-07-23', 28000, 500, 20, 'Pune'),
  (3, 'Alice', 'Developer', 1, '2019-05-01', 75000, 0, 10, 'Nashik'),
  (4, 'Bob', 'Clerk', 1, '2022-03-11', 25000, 0, 30, 'Nagpur'),
  (5, 'Charlie', 'Salesman', 1, '2021-08-15', 55000, 1000, 20, 'Mumbai');

SELECT *
FROM Emp
WHERE sal BETWEEN 20000 AND 50000;

SELECT *
FROM Emp
WHERE city IN ('Mumbai', 'Pune', 'Nashik', 'Nagpur');
```

Question 13:

Create table Student (stu_name, course_id, Roll_no, percentage) insert required values and write queries for the following:

- Select stu_name, course_id, from Student WHERE percentage is >=60 and <=100;
- Select details of students whose Roll numbers are above 15;
- Select stu_id, Roll_no from Student WHERE course_id != 121;

Answer:

```
CREATE TABLE Student (
  stu_name VARCHAR(100),
  course_id INT,
  Roll_no INT PRIMARY KEY,
  percentage DECIMAL(5, 2)
);
```

```
INSERT INTO Student (stu_name, course_id, Roll_no, percentage)
VALUES
  ('Alice', 101, 12, 85.50),
  ('Bob', 102, 16, 75.00),
  ('Charlie', 103, 18, 62.00),
  ('David', 104, 14, 50.00),
  ('Eve', 105, 22, 90.00),
  ('Frank', 101, 11, 45.50),
  ('Grace', 106, 19, 95.00);

SELECT
  stu_name,
  course_id
FROM Student
WHERE percentage >= 60 AND percentage <= 100;

SELECT *
FROM Student
WHERE Roll_no > 15;

SELECT
  stu_name,
  Roll_no
FROM Student
WHERE course_id != 121;
```

Question 14:

Create table emp1(empno,ename,deptno)
emp2(empno,ename,deptno) insert required values and write SQL commands for the following:

- Display the names of employees including duplicate employee names.
- Display the names of employees excluding duplicate employee names.

Answer:

```
CREATE TABLE emp1 (
   empno INT,
   ename VARCHAR(50),
   deptno INT
);

CREATE TABLE emp2 (
   empno INT,
   ename VARCHAR(50),
   deptno INT
);

INSERT INTO emp1 (empno, ename, deptno)
VALUES
   (1, 'John', 10),
   (2, 'Alice', 20),
   (3, 'Bob', 30),
   (4, 'John', 40);

INSERT INTO emp2 (empno, ename, deptno)
VALUES
   (5, 'Jane', 10),
   (6, 'Alice', 20),
   (7, 'Charlie', 30),
   (8, 'Bob', 40);

SELECT ename
FROM emp1
UNION ALL
SELECT ename
```

```
FROM emp2;


SELECT ename
FROM emp1
UNION
SELECT ename
FROM emp2;
```

Question 15:

Create table emp1(empno,ename,deptno)
emp2(empno,ename,deptno) insert required values and write SQL commands for the following:

- Display the common employee names from both the tables.
- List employees who are not assigned to any department?

Answer:

```
CREATE TABLE emp1 (
   empno INT,
   ename VARCHAR(50),
   deptno INT
);


CREATE TABLE emp2 (
   empno INT,
   ename VARCHAR(50),
   deptno INT
);


INSERT INTO emp1 (empno, ename, deptno)
VALUES
   (1, 'John', 10),
   (2, 'Alice', 20),
   (3, 'Bob', NULL),
```

```
    (4, 'David', 30);

INSERT INTO emp2 (empno, ename, deptno)
VALUES
    (5, 'Alice', 20),
    (6, 'Bob', NULL),
    (7, 'Charlie', 30),
    (8, 'David', 30);

-- Select names from emp1 that are also in emp2
SELECT ename
FROM emp1
WHERE ename IN (SELECT ename FROM emp2);

-- Select names from emp1 and emp2 where deptno is NULL
SELECT ename
FROM emp1
WHERE deptno IS NULL
UNION
SELECT ename
FROM emp2
WHERE deptno IS NULL;
```

Question 16:

Write following queries:

1. Select concat (‘Jay’ ‘IITB’) from Dual;
2. Select ltrim (‘Shreya’,‘s’) from Dual;
3. Select upper(‘raj’) from Dual;
4. Select rpad (‘HR’, 10, ‘*’) from Dual;

Answer: SELECT CONCAT('Jay', ' IITB') FROM DUAL;

```
SELECT LTRIM('Shreya', 's') FROM DUAL;


SELECT UPPER('raj') FROM DUAL;


SELECT RPAD('HR', 10, '*') FROM DUAL;
```

Question 17:

Write a query to calculate the total price of a shopping cart by multiplying the quantity of an item by its price (e.g., Quantity * Price). Then, calculate the discounted price using subtraction (Price - Discount).

Answer:

```
SELECT
   quantity * price AS total_price,
   price - discount AS discounted_price
FROM
   shopping_cart;
```

Question 18:

Create Two Tables Emp and Dept, Consider the following schema: Emp (Emp_no as primary key, E_name, Dept_no, Dept_name, name, Job_id, Salary)
Dept (Dept_no as primary key, emp_no foreign key, deptname, location)

1. Display the information of the tables using SELECT command.
2. Execute the SQL queries using SELECT, WHERE, GROUP BY clause.

Answer:

```
CREATE TABLE Emp (
   Emp_no INT PRIMARY KEY,
   E_name VARCHAR(50),
   Dept_no INT,
   Dept_name VARCHAR(50),
   Job_id VARCHAR(50),
   Salary DECIMAL(10, 2)
);
```

```
CREATE TABLE Dept (
   Dept_no INT PRIMARY KEY,
   Emp_no INT,
   Dept_name VARCHAR(50),
   Location VARCHAR(50),
   FOREIGN KEY (Emp_no) REFERENCES Emp(Emp_no)
);


SELECT * FROM Emp;


SELECT * FROM Dept;


SELECT Dept_name, COUNT(*) AS num_employees
FROM Emp
GROUP BY Dept_name
HAVING COUNT(*) > 1;
```

Question 19:

Create Two Tables Emp and Dept, Consider the following schema: Emp (Emp_no as primary key, E_name, Dept_no, Dept_name, name, Job_id, Salary)
Dept (Dept_no as primary key, emp_no foreign key, deptname, location)
Write SQL queries to retrieve and manipulate data from the Emp and Dept tables using the following clauses:

- SELECT: To choose the columns you want to display from the tables.
- WHERE: To filter the rows based on specific conditions, such as salary ranges, employee names, or department locations.

Answer:

```
CREATE TABLE Emp (
   Emp_no INT PRIMARY KEY,
   E_name VARCHAR(50),
   Dept_no INT,
   Dept_name VARCHAR(50),
   Job_id VARCHAR(50),
```

```
    Salary DECIMAL(10, 2)
);

CREATE TABLE Dept (
    Dept_no INT PRIMARY KEY,
    Emp_no INT,
    Dept_name VARCHAR(50),
    Location VARCHAR(50),
    FOREIGN KEY (Emp_no) REFERENCES Emp(Emp_no)
);

INSERT INTO Emp (Emp_no, E_name, Dept_no, Dept_name, Job_id, Salary) VALUES
(101, 'John', 10, 'HR', 'Manager', 50000),
(102, 'Alice', 20, 'Sales', 'Salesman', 40000),
(103, 'Bob', 10, 'HR', 'Clerk', 30000);

INSERT INTO Dept (Dept_no, Emp_no, Dept_name, Location) VALUES
(10, 101, 'HR', 'New York'),
(20, 102, 'Sales', 'London');

SELECT Emp_no, E_name, Dept_name, Salary FROM Emp;

SELECT * FROM Emp WHERE Salary > 40000;

SELECT * FROM Dept WHERE Location = 'London';
```

Question 20:

Write SQL queries to retrieve and manipulate data from the Emp and Dept tables using the following clauses:

- GROUP BY: To group the data by one or more columns, such as Dept_name or Job_id.
- HAVING: To filter the results of the GROUP BY clause, for example, showing only departments with an average salary above a certain value.

Answer:

```
CREATE TABLE Emp (
   Emp_no INT PRIMARY KEY,
   E_name VARCHAR(50),
   Dept_no INT,
   Dept_name VARCHAR(50),
   Job_id VARCHAR(50),
   Salary DECIMAL(10, 2)
);

CREATE TABLE Dept (
   Dept_no INT PRIMARY KEY,
   Emp_no INT,
   Dept_name VARCHAR(50),
   Location VARCHAR(50),
   FOREIGN KEY (Emp_no) REFERENCES Emp(Emp_no)
);

INSERT INTO Emp (Emp_no, E_name, Dept_no, Dept_name, Job_id, Salary) VALUES
(101, 'John', 10, 'HR', 'Manager', 50000),
(102, 'Alice', 20, 'Sales', 'Salesman', 40000),
(103, 'Bob', 10, 'HR', 'Clerk', 30000);

INSERT INTO Dept (Dept_no, Emp_no, Dept_name, Location) VALUES
(10, 101, 'HR', 'New York'),
(20, 102, 'Sales', 'London');

SELECT Dept_name, AVG(Salary) AS avg_salary FROM Emp GROUP BY Dept_name;

SELECT Dept_name, AVG(Salary) AS avg_salary FROM Emp GROUP BY Dept_name HAVING
AVG(Salary) > 35000;
```

Question 21:

Create Two Tables Emp and Dept, Consider the following schema: Emp (Emp_no as primary key, E_name, Dept_no, Dept_name, name, Job_id, Salary)
Dept (Dept_no as primary key, emp_no foreign key, deptname, location)
Write SQL queries to retrieve and sort data from the Emp and Dept tables using the following clauses:

- SELECT: To display specific columns such as employee names, salaries, department names, etc.
- WHERE: To filter rows based on conditions like employee job roles or department locations.

Answer:

```
CREATE TABLE Emp (
   Emp_no INT PRIMARY KEY,
   E_name VARCHAR(50),
   Dept_no INT,
   Dept_name VARCHAR(50),
   Job_id VARCHAR(50),
   Salary DECIMAL(10, 2)
);

CREATE TABLE Dept (
   Dept_no INT PRIMARY KEY,
   Emp_no INT,
   Dept_name VARCHAR(50),
   Location VARCHAR(50),
   FOREIGN KEY (Emp_no) REFERENCES Emp(Emp_no)
);

INSERT INTO Emp (Emp_no, E_name, Dept_no, Dept_name, Job_id, Salary) VALUES
(101, 'John', 10, 'HR', 'Manager', 50000),
(102, 'Alice', 20, 'Sales', 'Salesman', 40000),
(103, 'Bob', 10, 'HR', 'Clerk', 30000);
```

```
INSERT INTO Dept (Dept_no, Emp_no, Dept_name, Location) VALUES
(10, 101, 'HR', 'New York'),
(20, 102, 'Sales', 'London');

SELECT E_name, Salary, Dept_name FROM Emp WHERE Job_id = 'Manager';

SELECT * FROM Dept WHERE Location = 'New York';
```

Question 22:

Create Two Tables Emp and Dept, Consider the following schema: Emp (Emp_no as primary key, E_name, Dept_no, Dept_name, name, Job_id, Salary)
Dept (Dept_no as primary key, emp_no foreign key, deptname, location)
Write SQL queries to retrieve and sort data from the Emp and Dept tables using the following clauses:

- ORDER BY: To sort the results by one or more columns, such as sorting employees by salary in ascending or descending order or arranging departments alphabetically by name.

Answer:

```
CREATE TABLE Emp (
    Emp_no INT PRIMARY KEY,
    E_name VARCHAR(50),
    Dept_no INT,
    Dept_name VARCHAR(50),
    Job_id VARCHAR(50),
    Salary DECIMAL(10, 2)
);

CREATE TABLE Dept (
    Dept_no INT PRIMARY KEY,
    Emp_no INT,
    Dept_name VARCHAR(50),
    Location VARCHAR(50),
    FOREIGN KEY (Emp_no) REFERENCES Emp(Emp_no)
);
```

```
INSERT INTO Emp (Emp_no, E_name, Dept_no, Dept_name, Job_id, Salary) VALUES
(101, 'John', 10, 'HR', 'Manager', 50000),
(102, 'Alice', 20, 'Sales', 'Salesman', 40000),
(103, 'Bob', 10, 'HR', 'Clerk', 30000);

INSERT INTO Dept (Dept_no, Emp_no, Dept_name, Location) VALUES
(10, 101, 'HR', 'New York'),
(20, 102, 'Sales', 'London');

SELECT E_name, Salary, Dept_name FROM Emp ORDER BY Salary ASC;

SELECT * FROM Dept ORDER BY Dept_name;
```

Question 23:

Create Two Tables Emp and Dept, Consider the following schema: Emp (Emp_no as primary key, E_name, Dept_no, Dept_name, name, Job_id, Salary)
Dept (Dept_no as primary key, emp_no foreign key, deptname, location)

1. Display employee Nikhil's employee number, name, department number, and department location.
2. Display the list of employees who work in the sales department.

Answer:

```
CREATE TABLE Emp (
    Emp_no INT PRIMARY KEY,
    E_name VARCHAR(50),
    Dept_no INT,
    Dept_name VARCHAR(50),
    Job_id VARCHAR(50),
    Salary DECIMAL(10, 2)
);

CREATE TABLE Dept (
```

```
    Dept_no INT PRIMARY KEY,
    Emp_no INT,
    Dept_name VARCHAR(50),
    Location VARCHAR(50),
    FOREIGN KEY (Emp_no) REFERENCES Emp(Emp_no)
);

INSERT INTO Emp (Emp_no, E_name, Dept_no, Dept_name, Job_id, Salary) VALUES
(101, 'John', 10, 'HR', 'Manager', 50000),
(102, 'Alice', 20, 'Sales', 'Salesman', 40000),
(103, 'Bob', 10, 'HR', 'Clerk', 30000);

INSERT INTO Dept (Dept_no, Emp_no, Dept_name, Location) VALUES
(10, 101, 'HR', 'New York'),
(20, 102, 'Sales', 'London');

SELECT Emp_no, E_name, Dept_no, Location FROM Emp JOIN Dept ON Emp.Dept_no =
Dept.Dept_no WHERE E_name = 'Nikhil';

SELECT E_name, Dept_name FROM Emp WHERE Dept_name = 'Sales';
```

Question 24:

Create Two Tables Emp and Dept, Consider the following schema: Emp (Emp_no as primary key, E_name, Dept_no, Dept_name, name, Job_id, Salary)
Dept (Dept_no as primary key, emp_no foreign key, deptname, location)

1. Display the list of employees who do not work in the sales department.
2. Display the employee names and salary of all employees who report to Sumit Patil.

Answer:

```
CREATE TABLE Emp (
    Emp_no INT PRIMARY KEY,
    E_name VARCHAR(50),
    Dept_no INT,
```

```
    Dept_name VARCHAR(50),
    Job_id VARCHAR(50),
    Salary DECIMAL(10, 2)
);

CREATE TABLE Dept (
    Dept_no INT PRIMARY KEY,
    Emp_no INT,
    Dept_name VARCHAR(50),
    Location VARCHAR(50),
    FOREIGN KEY (Emp_no) REFERENCES Emp(Emp_no)
);

INSERT INTO Emp (Emp_no, E_name, Dept_no, Dept_name, Job_id, Salary) VALUES
(101, 'John', 10, 'HR', 'Manager', 50000),
(102, 'Alice', 20, 'Sales', 'Salesman', 40000),
(103, 'Bob', 10, 'HR', 'Clerk', 30000);

INSERT INTO Dept (Dept_no, Emp_no, Dept_name, Location) VALUES
(10, 101, 'HR', 'New York'),
(20, 102, 'Sales', 'London');

SELECT E_name, Dept_name FROM Emp WHERE Dept_name != 'Sales';

SELECT E_name, Salary FROM Emp WHERE Mgr = (SELECT Emp_no FROM Emp WHERE E_name =
'Sumit Patil');
```

Question 25:

Create Table Emp Consider the following schema: Emp (Emp_no as primary key, E_name, Dept_no, Dept_name, name, Job_id, Salary)

1. Create view emp_view as select emp_no, enema, salary from emp;
2. Update emp_view set e_name='Jay' where emp_no=101;

3. Delete from emp_view where emp_no= 105;
4. Drop view emp_view;

Answer:

```
CREATE TABLE Emp (
   Emp_no INT PRIMARY KEY,
   E_name VARCHAR(50),
   Dept_no INT,
   Dept_name VARCHAR(50),
   Job_id VARCHAR(50),
   Salary DECIMAL(10, 2)
);

INSERT INTO Emp (Emp_no, E_name, Dept_no, Dept_name, Job_id, Salary) VALUES
(101, 'John', 10, 'HR', 'Manager', 50000),
(102, 'Alice', 20, 'Sales', 'Salesman', 40000),
(103, 'Bob', 10, 'HR', 'Clerk', 30000);

CREATE VIEW emp_view AS SELECT Emp_no, E_name, Salary FROM Emp;

UPDATE emp_view SET E_name = 'Jay' WHERE Emp_no = 101;

DELETE FROM emp_view WHERE Emp_no = 105;

DROP VIEW emp_view;
```

Question 26:

Create Two Tables Emp and Dept, Consider the following schema: Emp (Emp_no as primary key, E_name, Dept_no, Dept_name, name, Job_id, Salary)
Dept (Dept_no as primary key, Emp_no foreign key, Dept_name, Location)
Create view emp_view as select emp_no, enema, salary from emp;

- Update emp_view set e_name='Jay' where emp_no=101;
- Delete from emp_view where emp_no= 105;

- Drop view emp_view;

Answer:

```
CREATE TABLE Emp (
   Emp_no INT PRIMARY KEY,
   E_name VARCHAR(50),
   Dept_no INT,
   Dept_name VARCHAR(50),
   Job_id VARCHAR(50),
   Salary DECIMAL(10, 2)
);

CREATE TABLE Dept (
   Dept_no INT PRIMARY KEY,
   Emp_no INT,
   Dept_name VARCHAR(50),
   Location VARCHAR(50),
   FOREIGN KEY (Emp_no) REFERENCES Emp(Emp_no)
);

INSERT INTO Emp (Emp_no, E_name, Dept_no, Dept_name, Job_id, Salary) VALUES
(101, 'John', 10, 'HR', 'Manager', 50000),
(102, 'Alice', 20, 'Sales', 'Salesman', 40000),
(103, 'Bob', 10, 'HR', 'Clerk', 30000);

INSERT INTO Dept (Dept_no, Emp_no, Dept_name, Location) VALUES
(10, 101, 'HR', 'New York'),
(20, 102, 'Sales', 'London');

CREATE VIEW emp_view AS SELECT Emp_no, E_name, Salary FROM Emp;

UPDATE emp_view SET E_name = 'Jay' WHERE Emp_no = 101;
```

```
DELETE FROM emp_view WHERE Emp_no = 105;


DROP VIEW emp_view;
```

Question 27:

Write a PL/SQL program that asks the user for their age and then prints “You can vote" if they are over 18, and "You cannot vote" otherwise.

Answer:

```
DECLARE
   age NUMBER;
BEGIN
   -- Ask for age
   age := &age; -- This prompts the user for input


   IF age > 18 THEN
      DBMS_OUTPUT.PUT_LINE('You can vote');
   ELSE
      DBMS_OUTPUT.PUT_LINE('You cannot vote');
   END IF;
END;
```

Question 28:

Write a PL/SQL program that checks if a given number is positive, and if it is, prints "Number is positive".

Answer:

```
SET SERVEROUTPUT ON; -- Ensure that output is displayed in SQL*Plus or SQL Developer


DECLARE
   num NUMBER;
BEGIN
   -- Ask for the number
```

```
    num := &num; -- This prompts the user for input

    IF num > 0 THEN
        DBMS_OUTPUT.PUT_LINE('Number is positive');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Number is not positive');
    END IF;
END;
/
```

Question 29:

Write a PL/SQL program that asks the user for percentage and then assigns grades based on the following conditions:

- Distinction (>=75%)
- First Class (>=60 and <75)
- Second Class (>=45 and <60)
- Pass Class (>=40 and <45)
- Fail (<40)

Answer:

```
SET SERVEROUTPUT ON; -- Ensure that output is displayed in SQL*Plus or SQL Developer

DECLARE
    percentage NUMBER;
BEGIN
    -- Ask for percentage
    percentage := &percentage; -- This prompts the user for input

    IF percentage >= 75 THEN
        DBMS_OUTPUT.PUT_LINE('Distinction');
    ELSIF percentage >= 60 AND percentage < 75 THEN
        DBMS_OUTPUT.PUT_LINE('First Class');
```

```
    ELSIF percentage >= 45 AND percentage < 60 THEN
        DBMS_OUTPUT.PUT_LINE('Second Class');
    ELSIF percentage >= 40 AND percentage < 45 THEN
        DBMS_OUTPUT.PUT_LINE('Pass Class');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Fail');
    END IF;
END;
/
```

Question 30:

Write a PL/SQL program to display multiplication table of 5 using FOR loop.

Answer:

```
SET SERVEROUTPUT ON; -- Ensure that output is displayed in SQL*Plus or SQL Developer

BEGIN
    FOR i IN 1..10 LOOP
        DBMS_OUTPUT.PUT_LINE('5 * ' || i || ' = ' || (5 * i));
    END LOOP;
END;
/
```

Question 31:

Write a PL/SQL program to calculate factorial of 10 by using PL/SQL WHILE LOOP statement.

Answer:

```
SET SERVEROUTPUT ON; -- Ensure that output is displayed in SQL*Plus or SQL Developer

DECLARE
    num NUMBER := 10;
    fact NUMBER := 1;
BEGIN
```

```
   WHILE num > 1 LOOP
      fact := fact * num;
      num := num - 1;
   END LOOP;
   DBMS_OUTPUT.PUT_LINE('Factorial of 10 is: ' || fact);
END;
/
```

Question 32:

Write a PL/SQL program to calculate the prime numbers between 1 to 50.

Answer:

```
SET SERVEROUTPUT ON; -- Ensure that output is displayed in SQL*Plus or SQL Developer

DECLARE
   num NUMBER;
   i NUMBER;
   is_prime BOOLEAN;
BEGIN
   FOR num IN 2..50 LOOP
      is_prime := TRUE; -- Assume num is prime
      FOR i IN 2..num-1 LOOP
         IF num MOD i = 0 THEN
            is_prime := FALSE; -- num is not prime
            EXIT; -- Exit the inner loop
         END IF;
      END LOOP;

      IF is_prime THEN
         DBMS_OUTPUT.PUT_LINE(num || ' is prime');
      END IF;
   END LOOP;
```

```
END;
/
```

Question 33:

Write a PL/SQL program for displaying details of students studying in computer department using cursors.

Answer:

```
SET SERVEROUTPUT ON; -- Ensure that output is displayed in SQL*Plus or SQL Developer

DECLARE
  CURSOR student_cursor IS
    SELECT student_name
    FROM students
    WHERE department = 'Computer';

  student_record student_cursor%ROWTYPE;
BEGIN
  OPEN student_cursor;
  LOOP
    FETCH student_cursor INTO student_record;
    EXIT WHEN student_cursor%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE(student_record.student_name);
  END LOOP;
  CLOSE student_cursor;
END;
/
```

Question 34:

Write a PL/SQL program to print even number of records stored in the student table.

Answer:

```
SET SERVEROUTPUT ON; -- Ensure that output is displayed in SQL*Plus or SQL Developer
```

```
DECLARE
   CURSOR student_cursor IS
      SELECT * FROM students;

   student_record student_cursor%ROWTYPE;
   row_count NUMBER := 0;
BEGIN
   OPEN student_cursor;
   LOOP
      FETCH student_cursor INTO student_record;
      EXIT WHEN student_cursor%NOTFOUND;
      row_count := row_count + 1;
      IF MOD(row_count, 2) = 0 THEN
         DBMS_OUTPUT.PUT_LINE('Even Record: ' || student_record.student_name);
      END IF;
   END LOOP;
   CLOSE student_cursor;
END;
/
```

Question 35:

Write a PL/SQL program to display the number of items having price more than 10000 in store table using cursors.

Answer:

```
SET SERVEROUTPUT ON; -- Ensure that output is displayed in SQL*Plus or SQL Developer

DECLARE
   item_count NUMBER;
BEGIN
   SELECT COUNT(*)
```

```
    INTO item_count
    FROM store
    WHERE price > 10000;

    DBMS_OUTPUT.PUT_LINE('Number of items with price > 10000: ' || item_count);
END;
/
```

Question 36:

Write a PL/SQL program that asks the user to input two numbers and divide the first number by the second. Handle the predefined exception for division by zero and display an appropriate message if it occurs.

Answer:

```
SET SERVEROUTPUT ON; -- Ensure that output is displayed in SQL*Plus or SQL Developer

DECLARE
    num1 NUMBER := &num1;
    num2 NUMBER := &num2;
    result NUMBER;
BEGIN
    result := num1 / num2;
    DBMS_OUTPUT.PUT_LINE('Result: ' || result);
EXCEPTION
    WHEN ZERO_DIVIDE THEN
        DBMS_OUTPUT.PUT_LINE('Cannot divide by zero');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('An unexpected error occurred: ' || SQLERRM);
END;
/
```

Question 37:

Write a PL/SQL program that retrieves the salary of an employee based on their employee ID (emp_id). If the employee ID does not exist in the database, handle the NO_DATA_FOUND exception and print a message saying, "Employee ID not found."

Answer:

```
SET SERVEROUTPUT ON; -- Ensure that output is displayed in SQL*Plus or SQL Developer


DECLARE
   emp_id NUMBER := &emp_id; -- Prompt for employee ID
   emp_salary NUMBER;
BEGIN
   -- Retrieve the salary of the employee with the given ID
   SELECT salary
   INTO emp_salary
   FROM employees
   WHERE employee_id = emp_id;


   -- Output the salary
   DBMS_OUTPUT.PUT_LINE('Salary: ' || emp_salary);


EXCEPTION
   WHEN NO_DATA_FOUND THEN
      DBMS_OUTPUT.PUT_LINE('Employee ID not found');
   WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('An unexpected error occurred: ' || SQLERRM);
END;
/
```

Question 38:

Write a PL/SQL program that asks for customer Id, when user enters invalid Id, the exception Invalid-Id is raised.

Answer:

```
SET SERVEROUTPUT ON; -- Ensure that output is displayed in SQL*Plus or SQL Developer
```

```
DECLARE
   customer_id NUMBER := &customer_id; -- Prompt for customer ID
BEGIN
   -- Check if customer_id is valid
   IF customer_id <= 0 THEN
      RAISE_APPLICATION_ERROR(-20001, 'Invalid Customer ID: Must be greater than 0');
   END IF;

   -- If the ID is valid, print a message
   DBMS_OUTPUT.PUT_LINE('Valid Customer ID: ' || customer_id);

EXCEPTION
   WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('An unexpected error occurred: ' || SQLERRM);
END;
/
```

Question 39:

Write a procedure emp_count() to count number of employees in department, use dept_no as input parameter.

Answer:

```
CREATE OR REPLACE PROCEDURE emp_count(dept_no IN NUMBER) AS
   emp_count NUMBER;
BEGIN
   -- Count the number of employees in the specified department
   SELECT COUNT(*)
   INTO emp_count
   FROM employees
   WHERE department_id = dept_no;
```

```
   -- Output the result
   DBMS_OUTPUT.PUT_LINE('Number of employees in department ' || dept_no || ': ' ||
emp_count);

EXCEPTION
   WHEN NO_DATA_FOUND THEN
      DBMS_OUTPUT.PUT_LINE('No employees found for department ID: ' || dept_no);
   WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('An unexpected error occurred: ' || SQLERRM);
END emp_count;
/
```

Question 40:

Create a stored procedure to accept name and greet user with name.

Answer:

```
CREATE OR REPLACE PROCEDURE greet_user(name IN VARCHAR2) AS
BEGIN
   DBMS_OUTPUT.PUT_LINE('Hello, ' || name);
END greet_user;
/
```

Question 41:

Write a PL/SQL procedure to insert any three records in EMP table.

Answer:

```
BEGIN
   -- Insert records into the emp table
   INSERT INTO emp (emp_id, emp_name, salary) VALUES (1, 'John', 5000);
   INSERT INTO emp (emp_id, emp_name, salary) VALUES (2, 'Alice', 6000);
   INSERT INTO emp (emp_id, emp_name, salary) VALUES (3, 'Bob', 7000);

   -- Commit the transaction
```

```
    COMMIT;

    -- Optional: Output a success message
    DBMS_OUTPUT.PUT_LINE('Records inserted successfully.');

EXCEPTION
    WHEN OTHERS THEN
        -- Rollback in case of an error
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('Error occurred: ' || SQLERRM);
END;
/
```

Question 42:

Write PL/SQL function which will compute and return the maximum of two values.

Answer:

```
CREATE OR REPLACE FUNCTION max_of_two(val1 IN NUMBER, val2 IN NUMBER)
RETURN NUMBER IS
BEGIN
    IF val1 > val2 THEN
        RETURN val1;
    ELSE
        RETURN val2;
    END IF;
END max_of_two;
/
```

Question 43:

Write PL/SQL function to calculate the factorial of given no.

Answer:

```
CREATE OR REPLACE FUNCTION factorial(n IN NUMBER) RETURN NUMBER IS
```

```
    fact NUMBER := 1;
BEGIN
    IF n < 0 THEN
        RETURN NULL; -- Factorial is not defined for negative numbers
    ELSIF n = 0 THEN
        RETURN 1; -- Factorial of 0 is 1
    ELSE
        FOR i IN 1..n LOOP
            fact := fact * i;
        END LOOP;
    END IF;
    RETURN fact;
END factorial;
/
```

Question 44:

Create a trigger on EMP table which is invoked when salary is below 5000.

Answer:

```
CREATE OR REPLACE TRIGGER salary_check
BEFORE INSERT OR UPDATE ON emp
FOR EACH ROW
BEGIN
    IF :NEW.salary < 5000 THEN
        RAISE_APPLICATION_ERROR(-20001, 'Salary must be at least 5000');
    END IF;
END salary_check;
/
```

Question 45:

Create a trigger which invokes on updation of record in Department table.

Answer:

```
CREATE OR REPLACE TRIGGER dept_update
AFTER UPDATE ON department
FOR EACH ROW
BEGIN
   DBMS_OUTPUT.PUT_LINE('Department updated: ' || :NEW.department_name);
END dept_update;
/
```