DBMS ANS

SLIP 1 :-

Q1)

```
CREATE TABLE EMPLOYEE_DATA (
   Empid NUMBER PRIMARY KEY,
   Empname VARCHAR2(15),
   Manager_id NUMBER,
   Dept_id NUMBER,
   Salary NUMBER
);

-- Insert 5 meaningful records
INSERT INTO EMPLOYEE_DATA (Empid, Empname, Manager_id, Dept_id, Salary) VALUES
(101, 'John', 201, 10, 50000),
(102, 'Sara', 202, 20, 60000),
(103, 'Sam', 201, 10, 55000),
(104, 'Sophia', 203, 30, 60000),
(105, 'Steve', 202, 20, 50000);
```

Q2)

A
```
SELECT Empname, Dept_name
FROM EMPLOYEE_DATA E
JOIN DEPARTMENT D ON E.Dept_id = D.Dept_id;
```

B
```
SELECT Empname
FROM EMPLOYEE_DATA
WHERE Empname LIKE 'S%';
```

C
```
ALTER TABLE EMPLOYEE_DATA
ADD Phone VARCHAR2(15);
```

D SELECT Empname, Salary

FROM EMPLOYEE_DATA

WHERE Salary IN (

SELECT Salary

FROM EMPLOYEE_DATA

GROUP BY Salary

HAVING COUNT(*) > 1

);

SLIP 2 :-

Q1)

```
CREATE TABLE STUDENT (
   Rollno NUMBER PRIMARY KEY,
   Fname VARCHAR2(15),
   Lname VARCHAR2(15),
   Course_name VARCHAR2(15) NOT NULL,
   Major VARCHAR2(15),
   Email_id VARCHAR2(15)
);

-- Insert 5 meaningful records
INSERT INTO STUDENT (Rollno, Fname, Lname, Course_name, Major, Email_id) VALUES
(1, 'John', 'Doe', 'BSCIT', 'Computer Science', 'john.doe@gmail.com'),
(2, 'Sara', 'Smith', 'BSC', 'Physics', 'sara.smith@yahoo.co.in'),
(3, 'Steve', 'Rogers', 'BSCIT', 'Information Technology', 'steve.rogers@gmail.com'),
(4, 'Sophia', 'Williams', 'BSC', 'Mathematics', 'sophia.williams@yahoo.co.in'),
(5, 'Emily', 'Davis', 'BSCIT', 'Computer Science', 'emily.davis@hotmail.com');
```

Q2)

A
```
SELECT COUNT(*) AS Total_Students_BSCIT
FROM STUDENT
WHERE Course_name = 'BSCIT';
```

B
```
SELECT Course_name, COUNT(*) AS Total_Students
FROM STUDENT
GROUP BY Course_name;
```

C
```
SELECT DISTINCT Course_name
FROM STUDENT
WHERE Course_name <> 'BSC';
```

D
```
SELECT Fname, Lname, Email_id
FROM STUDENT
WHERE Email_id LIKE '%@yahoo.co.in';
```

SLIP 3 :-

Q1)

```
CREATE TABLE CUSTOMER (
   Custid NUMBER(4) PRIMARY KEY,
   Lname VARCHAR2(15),
   Fname VARCHAR2(15),
   Area VARCHAR2(10),
   Phone NUMBER(8)
);

-- Insert 5 meaningful records into CUSTOMER table
INSERT INTO CUSTOMER (Custid, Lname, Fname, Area, Phone) VALUES
(1001, 'Smith', 'John', 'NYC', 12345678),
(1002, 'Doe', 'Jane', 'LA', 87654321),
```

```
(1003, 'Brown', 'Emily', 'SF', 12348765),
(1004, 'Wilson', 'Alex', 'TX', 87651234),
(1005, 'Davis', 'Sophia', 'NYC', 34567812);

CREATE TABLE MOVIE (
   Mvno NUMBER(2) PRIMARY KEY,
   Title VARCHAR2(25),
   Type VARCHAR2(10),
   Star VARCHAR2(25),
   Price NUMBER(8,2)
);

-- Insert 5 meaningful records into MOVIE table
INSERT INTO MOVIE (Mvno, Title, Type, Star, Price) VALUES
(1, 'Inception', 'Sci-Fi', 'Leonardo DiCaprio', 15.50),
(2, 'Titanic', 'Drama', 'Kate Winslet', 12.00),
(3, 'The Avengers', 'Action', 'Robert Downey Jr.', 18.00),
(4, 'The Godfather', 'Drama', 'Marlon Brando', 10.00),
(5, 'Star Wars', 'Sci-Fi', 'Harrison Ford', 14.00);

Q2)
A       SELECT C.Fname, C.Lname
        FROM CUSTOMER C
        JOIN MOVIE M ON C.Custid = M.Mvno
        WHERE M.Type = 'Drama';
B       SELECT C.Fname, C.Lname, M.Title
        FROM CUSTOMER C
        JOIN MOVIE M ON C.Custid = M.Mvno;
C       SELECT Fname, Lname
        FROM CUSTOMER
        WHERE Fname LIKE 'A%';
```

D ALTER TABLE CUSTOMER

ADD Age NUMBER(3);

SLIP 4 :-

Q1)

```
CREATE TABLE EMPLOYEE_DATA (
   Empid NUMBER PRIMARY KEY,
   Empname VARCHAR2(15),
   Manager_id NUMBER,
   Dept_id NUMBER,
   Salary NUMBER
);

INSERT INTO EMPLOYEE_DATA (Empid, Empname, Manager_id, Dept_id, Salary) VALUES
(1, 'John Doe', 2, 101, 50000),
(2, 'Jane Smith', NULL, 102, 60000),
(3, 'Sam Brown', 2, 101, 55000),
(4, 'Sue Green', 1, 103, 70000),
(5, 'Tom White', 3, 102, 65000);

CREATE TABLE DEPT (
   Dept_id VARCHAR2(15) PRIMARY KEY,
   Dept_name VARCHAR2(15)
);

INSERT INTO DEPT (Dept_id, Dept_name) VALUES
('101', 'HR'),
('102', 'Finance'),
('103', 'IT'),
('104', 'Marketing'),
('105', 'Sales');
```

Q2)

A
```
SELECT e1.Empname AS Employee, e2.Empname AS Manager
FROM EMPLOYEE_DATA e1
LEFT JOIN EMPLOYEE_DATA e2 ON e1.Manager_id = e2.Empid;
```

B
```
SELECT Empname
FROM EMPLOYEE_DATA
WHERE Empname LIKE 'S%';
```

C
```
ALTER TABLE EMPLOYEE_DATA
ADD Hiredate DATE;
```

D
```
SELECT Empname
FROM EMPLOYEE_DATA
WHERE (Dept_id, Salary) IN (
 SELECT Dept_id, MAX(Salary)
 FROM EMPLOYEE_DATA
GROUP BY Dept_id
);
```

SLIP 5 :-

Q1)

```
CREATE TABLE STUDENT (
   Rollno NUMBER PRIMARY KEY,
   Fname VARCHAR2(15),
   Lname VARCHAR2(15),
   Course_Id VARCHAR2(15),
   Major VARCHAR2(15),
   Age NUMBER
);

INSERT INTO STUDENT (Rollno, Fname, Lname, Course_Id, Major, Age) VALUES
(1, 'John', 'Doe', 'C101', 'CS', 20),
(2, 'Jane', 'Smith', 'C102', 'Math', 22),
(3, 'Sam', 'Brown', 'C103', 'Physics', 21),
```

```
(4, 'Sue', 'Green', 'C104', 'Chemistry', 23),
(5, 'Tom', 'White', 'C105', 'Biology', 24);

CREATE TABLE COURSES (
   Course_id VARCHAR2(15) PRIMARY KEY,
   Course_name VARCHAR2(15),
   Archive VARCHAR2(15)
);

INSERT INTO COURSES (Course_id, Course_name, Archive) VALUES
('C101', 'Computer Science', 'No'),
('C102', 'Mathematics', 'No'),
('C103', 'Physics', 'No'),
('C104', 'Chemistry', 'No'),
('C105', 'Biology', 'No');
```

Q2)

A
```
SELECT c.Course_name, s.Fname, s.Rollno
FROM STUDENT s
JOIN COURSES c ON s.Course_Id = c.Course_id
WHERE s.Age > 25;
```

B
```
SELECT c.Course_name, COUNT(s.Rollno) AS Total_Students
FROM STUDENT s
JOIN COURSES c ON s.Course_Id = c.Course_id
WHERE c.Course_name != 'BSCIT'
GROUP BY c.Course_name;
```

C
```
SELECT s.*
FROM STUDENT s
JOIN COURSES c ON s.Course_Id = c.Course_id
WHERE c.Course_name LIKE 'B%';
```

SLIP 6 :-

Q1)

```
CREATE TABLE CUSTOMER (
   Custid NUMBER(4) PRIMARY KEY,
   Lname VARCHAR2(15),
   Fname VARCHAR2(15),
   Area VARCHAR2(10),
   Phone NUMBER(8)
);
INSERT INTO CUSTOMER (Custid, Lname, Fname, Area, Phone) VALUES
(1, 'Doe', 'John', 'Downtown', 12345678),
(2, 'Smith', 'Jane', 'Uptown', 23456789),
(3, 'Brown', 'Sam', 'Midtown', 34567890),
(4, 'Green', 'Sue', 'Suburb', 45678901),
(5, 'White', 'Tom', 'Riverside', 56789012);

CREATE TABLE MOVIE (
   Mvno NUMBER(2) PRIMARY KEY,
   Title VARCHAR2(25),
   Type VARCHAR2(10),
   Star VARCHAR2(25),
   Price NUMBER(8,2)
);
INSERT INTO MOVIE (Mvno, Title, Type, Star, Price) VALUES
(1, 'Inception', 'Sci-Fi', 'Leonardo DiCaprio', 150.00),
(2, 'Titanic', 'Romance', 'Leonardo DiCaprio', 200.00),
(3, 'The Matrix', 'Action', 'Keanu Reeves', 180.00),
(4, 'Interstellar', 'Sci-Fi', 'Matthew McConaughey', 170.00),
(5, 'The Shining', 'Horror', 'Jack Nicholson', 160.00);
```

```
CREATE TABLE INVOICE (
   Invno NUMBER(4) PRIMARY KEY,
   Mvno NUMBER(2),
   Custid NUMBER(4),
   Issuedate DATE
);
INSERT INTO INVOICE (Invno, Mvno, Custid, Issuedate) VALUES
(1, 1, 1, TO_DATE('2024-01-01', 'YYYY-MM-DD')),
(2, 2, 2, TO_DATE('2024-02-01', 'YYYY-MM-DD')),
(3, 3, 3, TO_DATE('2024-03-01', 'YYYY-MM-DD')),
(4, 4, 4, TO_DATE('2024-04-01', 'YYYY-MM-DD')),
(5, 5, 5, TO_DATE('2024-05-01', 'YYYY-MM-DD'));
```

Q2)

A
```
SELECT c.Fname, c.Lname
FROM CUSTOMER c
JOIN INVOICE i ON c.Custid = i.Custid
GROUP BY c.Fname, c.Lname
HAVING COUNT(i.Invno) > 5;
```

B
```
SELECT c.Fname, c.Lname, m.Star
FROM CUSTOMER c
JOIN INVOICE i ON c.Custid = i.Custid
JOIN MOVIE m ON i.Mvno = m.Mvno
WHERE m.Title = 'Mr.';
```

C
```
SELECT m.Mvno, m.Title, m.Type
FROM MOVIE m
WHERE m.Star LIKE 'H%';
```

D
```
SELECT Title
FROM MOVIE
WHERE Type != 'Horror';
```

SLIP 7 :-

Q1)

```
CREATE TABLE EMP (
   Empno NUMBER PRIMARY KEY,
   Ename VARCHAR2(15),
   HireDate DATE,
   Deptno NUMBER NOT NULL,
   Gender VARCHAR2(10),
   Salary NUMBER,
   Commission NUMBER
);

INSERT INTO EMP (Empno, Ename, HireDate, Deptno, Gender, Salary, Commission) VALUES
(1, 'John Doe', TO_DATE('2022-01-01', 'YYYY-MM-DD'), 101, 'M', 30000, 5000),
(2, 'Jane Smith', TO_DATE('2022-02-01', 'YYYY-MM-DD'), 102, 'F', 35000, 7000),
(3, 'Sam Brown', TO_DATE('2022-03-01', 'YYYY-MM-DD'), 103, 'M', 25000, 3000),
(4, 'Sue Green', TO_DATE('2022-04-01', 'YYYY-MM-DD'), 104, 'F', 40000, 8000),
(5, 'Tom White', TO_DATE('2022-05-01', 'YYYY-MM-DD'), 105, 'M', 45000, 9000),
(6, 'Alice Blue', TO_DATE('2022-06-01', 'YYYY-MM-DD'), 101, 'F', 32000, 6000),
(7, 'Bob Black', TO_DATE('2022-07-01', 'YYYY-MM-DD'), 102, 'M', 37000, 7500),
(8, 'Charlie Red', TO_DATE('2022-08-01', 'YYYY-MM-DD'), 103, 'M', 27000, 3500),
(9, 'Diana Yellow', TO_DATE('2022-09-01', 'YYYY-MM-DD'), 104, 'F', 42000, 8500),
(10, 'Eve Purple', TO_DATE('2022-10-01', 'YYYY-MM-DD'), 105, 'F', 46000, 9500),
(11, 'Frank Orange', TO_DATE('2022-11-01', 'YYYY-MM-DD'), 101, 'M', 33000, 6500),
(12, 'Grace Pink', TO_DATE('2022-12-01', 'YYYY-MM-DD'), 102, 'F', 38000, 8000),
(13, 'Hank Gray', TO_DATE('2023-01-01', 'YYYY-MM-DD'), 103, 'M', 28000, 4000),
(14, 'Ivy Cyan', TO_DATE('2023-02-01', 'YYYY-MM-DD'), 104, 'F', 43000, 9000),
(15, 'Jack Brown', TO_DATE('2023-03-01', 'YYYY-MM-DD'), 105, 'M', 47000, 10000);
```

Q2)

A
```
SELECT COUNT(DISTINCT Deptno) AS UniqueDepartments
FROM EMP;
```

B
```
SELECT Deptno, SUM(Salary) AS TotalSalary
FROM EMP
GROUP BY Deptno
HAVING SUM(Salary) > 20000;
```

C
```
SELECT Ename
FROM EMP
WHERE Salary NOT BETWEEN 10000 AND 20000;
```

SLIP 8 :-

Q1)

```
CREATE TABLE Emp (
   Empno NUMBER PRIMARY KEY,
   Ename VARCHAR2(15),
   HireDate DATE,
   DOB DATE,
   Manager_id NUMBER,
   Deptno NUMBER NOT NULL,
   Gender VARCHAR2(10),
   Salary NUMBER,
   Commission NUMBER
);
```

```
INSERT INTO Emp (Empno, Ename, HireDate, DOB, Manager_id, Deptno, Gender, Salary, Commission)
VALUES
(1, 'John Doe', TO_DATE('2015-01-01', 'YYYY-MM-DD'), TO_DATE('1985-01-01', 'YYYY-MM-DD'), 101, 10, 'M', 50000, 5000),
(2, 'Jane Smith', TO_DATE('2016-02-01', 'YYYY-MM-DD'), TO_DATE('1986-02-01', 'YYYY-MM-DD'), 102, 20, 'F', 60000, 6000),
```

(3, 'Sam Brown', TO_DATE('2017-03-01', 'YYYY-MM-DD'), TO_DATE('1987-03-01', 'YYYY-MM-DD'), 103, 30, 'M', 55000, 5500),

(4, 'Sue Green', TO_DATE('2018-04-01', 'YYYY-MM-DD'), TO_DATE('1988-04-01', 'YYYY-MM-DD'), 104, 40, 'F', 70000, 7000),

(5, 'Tom White', TO_DATE('2019-05-01', 'YYYY-MM-DD'), TO_DATE('1989-05-01', 'YYYY-MM-DD'), 105, 50, 'M', 65000, 6500);

Q2)

A
```
SELECT Manager_id, COUNT(Empno) AS TotalEmployees
FROM Emp
GROUP BY Manager_id;
```

B
```
SELECT Ename, FLOOR(MONTHS_BETWEEN(SYSDATE, DOB) / 12) AS Age
FROM Emp;
```

C
```
SELECT Ename
FROM Emp
WHERE HireDate > TO_DATE('2010-01-01', 'YYYY-MM-DD');
```

SLIP 9 :-

Q1)

```
CREATE TABLE Customer (
   Product_id NUMBER PRIMARY KEY,
   Product_name VARCHAR2(25),
   Company_name VARCHAR2(25),
   Unit_price NUMBER,
   Quantity NUMBER
);
```

```
INSERT INTO Customer (Product_id, Product_name, Company_name, Unit_price, Quantity) VALUES
(1, 'Laptop', 'TechCorp', 1000, 50),
(2, 'Smartphone', 'MobileInc', 500, 200),
(3, 'Tablet', 'GadgetCo', 300, 150),
(4, 'Headphones', 'SoundTech', 100, 300),
```

```
(5, 'Smartwatch', 'WearableTech', 200, 100);

CREATE TABLE Order_product (
   Order_id NUMBER PRIMARY KEY,
   Product_id NUMBER,
   Total_units NUMBER,
   Total_cost NUMBER,
   Customer_name VARCHAR2(25)
);

INSERT INTO Order_product (Order_id, Product_id, Total_units, Total_cost, Customer_name) VALUES
(1, 1, 5, 5000, 'Alice'),
(2, 2, 10, 5000, 'Bob'),
(3, 3, 7, 2100, 'Charlie'),
(4, 4, 15, 1500, 'Diana'),
(5, 5, 3, 600, 'Eve');
```

Q2)

A
```
SELECT *
FROM Customer
WHERE Product_id NOT IN (SELECT Product_id FROM Order_product);
```

B
```
SELECT Product_name
FROM Customer
WHERE Unit_price = (SELECT MIN(Unit_price) FROM Customer);
```

C
```
SELECT *
FROM Customer
WHERE Unit_price = (SELECT MAX(Unit_price) FROM Customer);
```

D
```
SELECT Customer_name
FROM Order_product
WHERE Product_id IN (SELECT Product_id FROM Order_product GROUP BY Product_id
HAVING COUNT(*) > 1);
```

SLIP 10 :-

Q1)

```
CREATE TABLE EMPLOYEE (
   ID NUMBER PRIMARY KEY,
   LAST_NAME VARCHAR2(25),
   FIRST_NAME VARCHAR2(25),
   userid VARCHAR2(25),
   SALARY NUMBER
);

INSERT INTO EMPLOYEE (ID, LAST_NAME, FIRST_NAME, userid, SALARY) VALUES
(1, 'Patel', 'Ralph', 'rpatel', 895),
(2, 'Barcis', 'Ben', 'bbarcis', 860),
(3, 'Dini', 'Betty', 'bdini', 1100),
(4, 'Newman', 'Chad', 'cnewman', 750),
(5, ‘Ropeburn’,’ Audrey’,’ aropebur’,1550);
```

Q2)

A
```
UPDATE EMPLOYEE
SET LAST_NAME = 'Dreper'
WHERE ID = 3;
```

B
```
DELETE FROM EMPLOYEE
WHERE FIRST_NAME = 'Betty' AND LAST_NAME = 'Dancs';
```

C
```
UPDATE EMPLOYEE
SET SALARY = 1000
WHERE SALARY < 900;
```

D
```
CREATE TABLE EMPLOYEES2 AS
SELECT ID AS EMPLOYEE_ID, FIRST_NAME, LAST_NAME, SALARY, DEPARTMENT_ID
FROM EMPLOYEE;
```

E ALTER TABLE EMPLOYEE

DROP COLUMN FIRST_NAME;

DESC EMPLOYEE;

SLIP 11 :-

Q1)

```
CREATE TABLE Customer (
   custid NUMBER PRIMARY KEY,
   cname VARCHAR2(25),
   ccity VARCHAR2(25),
   cphone VARCHAR2(15),
   title VARCHAR2(25)
);

INSERT INTO Customer (custid, cname, ccity, cphone, title) VALUES
(1, 'Alice', 'New York', '1234567890', 'Ms.'),
(2, 'Bob', 'Los Angeles', '2345678901', 'Mr.'),
(3, 'Charlie', 'Chicago', '3456789012', 'Mr.'),
(4, 'Diana', 'Houston', '4567890123', 'Ms.'),
(5, 'Eve', 'Phoenix', '5678901234', 'Ms.'),
(6, 'Frank', 'Philadelphia', '6789012345', 'Mr.'),
(7, 'Grace', 'San Antonio', '7890123456', 'Ms.'),
(8, 'Hank', 'San Diego', '8901234567', 'Mr.'),
(9, 'Ivy', 'Dallas', '9012345678', 'Ms.'),
(10, 'Jack', 'San Jose', '0123456789', 'Mr.');

CREATE TABLE Movie (
   movieno NUMBER PRIMARY KEY,
   movietype VARCHAR2(25),
   actor VARCHAR2(25),
```

```
    director VARCHAR2(25)
);

INSERT INTO Movie (movieno, movietype, actor, director) VALUES
(1, 'Action', 'Tom Cruise', 'Christopher McQuarrie'),
(2, 'Drama', 'Leonardo DiCaprio', 'Martin Scorsese'),
(3, 'Comedy', 'Jim Carrey', 'Peter Farrelly'),
(4, 'Horror', 'Jamie Lee Curtis', 'John Carpenter'),
(5, 'Sci-Fi', 'Keanu Reeves', 'Lana Wachowski'),
(6, 'Romance', 'Ryan Gosling', 'Damien Chazelle'),
(7, 'Thriller', 'Jake Gyllenhaal', 'Denis Villeneuve'),
(8, 'Fantasy', 'Daniel Radcliffe', 'David Yates'),
(9, 'Adventure', 'Harrison Ford', 'Steven Spielberg'),
(10, 'Animation', 'Tom Hanks', 'John Lasseter');

CREATE TABLE Invoice (
    custid NUMBER,
    movieno NUMBER,
    returndate DATE,
    FOREIGN KEY (custid) REFERENCES Customer(custid),
    FOREIGN KEY (movieno) REFERENCES Movie(movieno)
);

INSERT INTO Invoice (custid, movieno, returndate) VALUES
(1, 1, TO_DATE('2024-01-01', 'YYYY-MM-DD')),
(2, 2, TO_DATE('2024-02-01', 'YYYY-MM-DD')),
(3, 3, TO_DATE('2024-03-01', 'YYYY-MM-DD')),
(4, 4, TO_DATE('2024-04-01', 'YYYY-MM-DD')),
(5, 5, TO_DATE('2024-05-01', 'YYYY-MM-DD')),
(6, 6, TO_DATE('2024-06-01', 'YYYY-MM-DD')),
(7, 7, TO_DATE('2024-07-01', 'YYYY-MM-DD')),
```

```
(8, 8, TO_DATE('2024-08-01', 'YYYY-MM-DD')),
(9, 9, TO_DATE('2024-09-01', 'YYYY-MM-DD')),
(10, 10, TO_DATE('2024-10-01', 'YYYY-MM-DD'));
1)      SELECT DISTINCT movietype
        FROM Movie;
2)      SELECT MAX(Unit_price) AS max_price, MIN(Unit_price) AS min_price
        FROM Movie;
3)      SELECT title
        FROM Movie
        WHERE actor LIKE 'M%';
4)      SELECT title
        FROM Movie
        WHERE Unit_price > 150;
5)      SELECT movietype, COUNT(*) AS number_of_movies
        FROM Movie
        GROUP BY movietype;

Q2)
DECLARE
   num1 NUMBER := 10;
   num2 NUMBER := 20;
   sum NUMBER;
BEGIN
   sum := num1 + num2;
   DBMS_OUTPUT.PUT_LINE('The sum of ' || num1 || ' and ' || num2 || ' is: ' || sum);
END;
/
```

SLIP 12:-

```
CREATE TABLE Emp (
   EmpId INT PRIMARY KEY,
   EName VARCHAR(50),
   CompId INT,
   Salary DECIMAL(10, 2),
   JoinDate DATE
);
INSERT INTO Emp (EmpId, EName, CompId, Salary, JoinDate) VALUES
(1, 'Raj', 101, 50000, '2004-01-15'),
(2, 'Radha', 101, 60000, '2004-01-20'),
(3, 'Gupta', 102, 55000, '2003-12-10'),
(4, 'Amit', 103, 45000, '2005-02-25'),
(5, 'Sita', 104, 70000, '2004-01-05');

CREATE TABLE Company (
   CompId INT PRIMARY KEY,
   CompName VARCHAR(50),
   Year INT,
   City VARCHAR(50)
);
INSERT INTO Company (CompId, CompName, Year, City) VALUES
(101, 'TechCorp', 2000, 'Mumbai'),
(102, 'InnovateLtd', 2001, 'Delhi'),
(103, 'FutureTech', 2002, 'Bangalore'),
(104, 'NextGen', 2003, 'Hyderabad');

CREATE TABLE Department (
   did INT PRIMARY KEY,
   dname VARCHAR(50),
   darea VARCHAR(50)
```

);

INSERT INTO Department (did, dname, darea) VALUES

(1, 'HR', 'Recruitment'),

(2, 'IT', 'Development'),

(3, 'Finance', 'Accounting'),

(4, 'Marketing', 'Sales');

1) UPDATE Emp

SET Salary = (SELECT Salary FROM Emp WHERE EName = 'Radha' AND CompId = (SELECT CompId FROM Emp WHERE EName = 'Raj'))

WHERE EName = 'Raj';

2) SELECT Gender, COUNT(*) AS Count

FROM Emp

WHERE JoinDate BETWEEN '2004-01-01' AND '2004-01-31'

GROUP BY Gender;

3) SELECT City, COUNT(*) AS TotalCompanies

FROM Company

GROUP BY City;

4) SELECT EName, JoinDate, Salary

FROM Emp

WHERE Salary = (SELECT Salary FROM Emp WHERE EName = 'Gupta');

5) SELECT EName, Salary

FROM Emp

ORDER BY Salary DESC

OFFSET 2 ROWS FETCH NEXT 1 ROW ONLY;

Q2)

```
DECLARE
   num NUMBER := &input_number;
BEGIN
   IF num > 0 THEN
      DBMS_OUTPUT.PUT_LINE('The number ' || num || ' is positive.');
   ELSIF num < 0 THEN
      DBMS_OUTPUT.PUT_LINE('The number ' || num || ' is negative.');
   ELSE
      DBMS_OUTPUT.PUT_LINE('The number ' || num || ' is zero.');
   END IF;
END;
/
```

SLIP 13:-

Q1)

```
CREATE TABLE Employee (
   EmpId INT PRIMARY KEY,
   EName VARCHAR(50) NOT NULL,
   CompId INT,
   Salary DECIMAL(10, 2) CHECK (Salary > 0),
   JoinDate DATE,
   Gender VARCHAR(10) CHECK (Gender IN ('Male', 'Female')),
   DeptNo INT,
   UNIQUE (EName, CompId)
);


INSERT INTO Employee (EmpId, EName, CompId, Salary, JoinDate, Gender, DeptNo) VALUES
(1, 'John Doe', 101, 50000, '2020-01-15', 'Male', 1),
(2, 'Jane Smith', 102, 60000, '2020-02-20', 'Female', 2),
(3, 'Alice Johnson', 103, 55000, '2020-03-25', 'Female', 3),
(4, 'Bob Brown', 104, 45000, '2020-04-30', 'Male', 4),
```

```
(5, 'Charlie Davis', 105, 70000, '2020-05-05', 'Male', 5),
(6, 'Diana Evans', 106, 80000, '2020-06-10', 'Female', 6),
(7, 'Eve Foster', 107, 75000, '2020-07-15', 'Female', 7),
(8, 'Frank Green', 108, 65000, '2020-08-20', 'Male', 8),
(9, 'Grace Harris', 109, 60000, '2020-09-25', 'Female', 9),
(10, 'Hank Irving', 110, 55000, '2020-10-30', 'Male', 10),
(11, 'Ivy Johnson', 111, 50000, '2020-11-05', 'Female', 11),
(12, 'Jack King', 112, 45000, '2020-12-10', 'Male', 12),
(13, 'Karen Lee', 113, 40000, '2021-01-15', 'Female', 13),
(14, 'Leo Martin', 114, 35000, '2021-02-20', 'Male', 14),
(15, 'Mona Nelson', 115, 30000, '2021-03-25', 'Female', 15);
```

Q2)

```
DECLARE
   score NUMBER := &input_score;
   result VARCHAR2(20);
BEGIN
   result := CASE
      WHEN score >= 90 THEN 'Excellent'
      WHEN score >= 75 THEN 'Very Good'
      WHEN score >= 60 THEN 'Good'
      WHEN score >= 50 THEN 'Pass'
      ELSE 'Fail'
   END;
   DBMS_OUTPUT.PUT_LINE('The result is: ' || result);
END;
/
```

SLIP 14 :-

Q1)

```
CREATE TABLE product_details (
   product_id INT PRIMARY KEY,
   product_name VARCHAR(50),
   quantity INT,
   price DECIMAL(10, 2)
);

INSERT INTO product_details (product_id, product_name, quantity, price) VALUES
(1001, 'book', 50, 900),
(1002, 'pen drive', 130, 900),
(1003, 'headphone', 100, 2000),
(1004, 'DVD', 20, 300),
(1005, 'speaker', 60, 2400);

CREATE TABLE sale_details (
   sale_no INT PRIMARY KEY,
   product_id INT,
   quantity INT,
   price DECIMAL(10, 2),
   customer_name VARCHAR(50),
   FOREIGN KEY (product_id) REFERENCES product_details(product_id)
);

INSERT INTO sale_details (sale_no, product_id, quantity, price, customer_name) VALUES
(2001, 1001, 50, 900, 'savni'),
(2002, 1004, 10, 300, 'savni'),
(2003, 1003, 120, 2000, 'savni'),
(2004, 1005, 420, 2400, 'harsh'),
(2005, 1002, 40, 900, 'Akash');
```

Inner join

```
SELECT p.product_id, p.product_name, s.quantity, s.price, s.customer_name
FROM product_details p
INNER JOIN sale_details s ON p.product_id = s.product_id;
```

Outer join

```
SELECT p.product_id, p.product_name, s.quantity, s.price, s.customer_name
FROM product_details p
LEFT JOIN sale_details s ON p.product_id = s.product_id
UNION
SELECT p.product_id, p.product_name, s.quantity, s.price, s.customer_name
FROM product_details p
RIGHT JOIN sale_details s ON p.product_id = s.product_id;
```

Q2)

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

SLIP 15 :-

Q1)

```
CREATE TABLE Employee (
   Emp_no INT PRIMARY KEY,
   First_name VARCHAR(50),
   Last_name VARCHAR(50),
   City VARCHAR(50),
   Salary DECIMAL(10, 2)
);
```

```
INSERT INTO Employee (Emp_no, First_name, Last_name, City, Salary) VALUES
(1001, 'Vasant', 'Powar', 'Pune', 15000),
(1002, 'Seema', 'Kharat', 'Mumbai', 25000),
(1003, 'Nitin', 'Pawar', 'Pune', 20000),
(1004, 'Swati', 'Jadhav', 'Mumbai', 25000),
(1005, 'Swaraj', 'Sawant', 'Nagpur', 28000);
```

AGGREGATE FUN

```
SELECT City, AVG(Salary) AS Avg_Salary, SUM(Salary) AS Total_Salary, MAX(Salary) AS Max_Salary,
MIN(Salary) AS Min_Salary
FROM Employee
GROUP BY City;
```

Q2)

```
CREATE TABLE product_details (
   product_id INT PRIMARY KEY,
   product_name VARCHAR(50),
   quantity INT,
   price DECIMAL(10, 2)
);

INSERT INTO product_details (product_id, product_name, quantity, price) VALUES
(1001, 'pendrive', 100, 900),
(1002, 'harddisk', 200, 4000),
(1003, 'headphone', 1000, 15000),
(1004, 'DVD', 20, 1000),
(1005, 'speaker', 600, 2400);

CREATE TABLE sales (
   sale_no INT PRIMARY KEY,
   product_id INT,
```

```
    quantity INT,
    price DECIMAL(10, 2),
    customer_name VARCHAR(50),
    FOREIGN KEY (product_id) REFERENCES product_details(product_id)
);
INSERT INTO sales (sale_no, product_id, quantity, price, customer_name) VALUES
(2001, 1001, 50, 900, 'savni'),
(2002, 1004, 10, 1000, 'savni'),
(2003, 1003, 120, 15000, 'savni'),
(2004, 1005, 420, 2400, 'harsh'),
(2005, 1002, 40, 4000, 'Akash');
```

AGGREGATE FUN USING CLAUSE

```
SELECT product_id, SUM(quantity) AS Total_Quantity, AVG(price) AS Avg_Price
FROM sales
GROUP BY product_id;
```