✅ PART 1– EMPLOYEE TABLE QUESTIONS WITH ANSWERS

Q1. Create the employee table (empid, empname, empdept, job, mgr, sal)

✅ Answer

```
CREATE TABLE employee (
  empid    NUMBER PRIMARY KEY,
  empname  VARCHAR2(30),
  empdept  VARCHAR2(20),
  job      VARCHAR2(20),
  mgr      NUMBER,
  sal      NUMBER
);
```

✅ Insert rows

```
INSERT INTO employee VALUES(1,'pavi','CSE','dev',101,30000);
INSERT INTO employee VALUES(2,'arun','IT','clerk',102,18000);
INSERT INTO employee VALUES(3,'meena','ECE','tester',103,22000);
INSERT INTO employee VALUES(4,'kumar','CSE','analyst',101,25000);
INSERT INTO employee VALUES(5,'sri','MECH','engineer',104,27000);
```

Q2. Display empname in descending order

✅ Answer

```
SELECT empname
FROM employee
ORDER BY empname DESC;
```

Q3. Change the column name (empdept → department) using ALTER

✅ Answer

```
ALTER TABLE employee
RENAME COLUMN empdept TO department;
```

Q4. Insert records using a trigger (empname should be Capitalized automatically)

✅ Create Trigger

```
CREATE OR REPLACE TRIGGER emp_trg
BEFORE INSERT ON employee
FOR EACH ROW
BEGIN
  :NEW.empname := INITCAP(:NEW.empname);
END;
/
```

✅ Insert using trigger (empname becomes 'Pavi' automatically)

```
INSERT INTO employee VALUES(1,'pavi','CSE','dev',101,30000);
```

✅ Output

empname becomes:

```
Pavi
```

Q5. Using GROUP BY display empname and salary for a specific department (e.g., 'CSE')

✅ Answer

```
SELECT empname, sal
FROM employee
WHERE department = 'CSE'
GROUP BY empname, sal;
```

Q6. Display the employee name in each department with minimum salary

✅ Answer

```
SELECT department, empname, sal
FROM employee
WHERE (department, sal) IN
       (SELECT department, MIN(sal)
        FROM employee
        GROUP BY department);
```

✅ PART 2– PRIVILEGES, SAVEPOINT & PROCEDURAL STATEMENT

✅ Q1. Create user and grant CONNECT, RESOURCE privileges

Answer

```
ALTER SESSION SET CONTAINER = XEPDB1;

CREATE USER empuser IDENTIFIED BY emp123;
GRANT CONNECT, RESOURCE TO empuser;
```

✅ Q2. Create the department and employee tables

Answer

```
CREATE TABLE department(
    depno NUMBER PRIMARY KEY,
    depname VARCHAR2(20)
);

CREATE TABLE employee(
    empid NUMBER PRIMARY KEY,
    empname VARCHAR2(20),
    dept VARCHAR2(10),
    job VARCHAR2(10),
    mgr NUMBER,
    sal NUMBER
);
```

✅ Q3. Develop a query to grant some privileges to employees on the departments table

Answer

```
GRANT SELECT, UPDATE ON department TO empuser;
```

✅ Q4. Develop a query to revoke ALL privileges from employees on the departments table

Answer

```
REVOKE ALL PRIVILEGES ON department FROM empuser;
```

✅ Q5. Develop a query to revoke SOME privileges from employees on the departments table

(Here UPDATE privilege is revoked, SELECT privilege remains.)

Answer

```
GRANT SELECT, UPDATE ON department TO empuser;
REVOKE UPDATE ON department FROM empuser;
```

✅ Q6. Implement SAVEPOINT in SQL

Answer

```
INSERT INTO employee VALUES (10, 'Arun', 'IT', 'Clerk', 200, 15000);

SAVEPOINT sp1;

UPDATE employee
SET sal = sal + 2000
WHERE empid = 10;

ROLLBACK TO sp1;

COMMIT;
```

✅ Q7. Demonstrate a procedural statement

Answer

```
SET SERVEROUTPUT ON;

DECLARE
    v_count NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_count FROM employee;
    DBMS_OUTPUT.PUT_LINE('Total Employees = ' || v_count);
END;
/.
```

✅ PART 3 – EMPLOYEE TABLE OPERATIONS

✅ Q1. Create a table employee (sno, name, designation, branch)

Answer

```
CREATE TABLE employee2 (
    sno         NUMBER(5),
    name        VARCHAR2(30),
    designation VARCHAR2(20),
    branch      VARCHAR2(20)
);
```

✅ Insert sample rows

```
INSERT INTO employee2 VALUES (1, 'Arun', 'Manager', 'IT');
INSERT INTO employee2 VALUES (2, 'Priya', 'Clerk',   'Finance');
INSERT INTO employee2 VALUES (3, 'Kumar', 'HR',      'Admin');
```

✅ Q2. Add a column `salary` to the table

Answer

```
ALTER TABLE employee2 ADD salary NUMBER(10);
```

✅ Q3. Delete the 2nd row from the table

Answer

```
DELETE FROM employee2 WHERE sno = 2;
```

✅ Q4. Create a copy of the table and drop the original table

Answer

✅ Create copy of the table

```
CREATE TABLE employee2_copy AS
SELECT * FROM employee2;
```

✅ Drop the original table

```
DROP TABLE employee2;
```

✅ Q5. Demonstrate a trigger for automatic updation (default salary)

Whenever a row is inserted **without a salary**, the trigger automatically sets salary to **10000**.

Answer: Create Trigger

```
CREATE OR REPLACE TRIGGER trg_auto_salary
BEFORE INSERT ON employee2_copy
FOR EACH ROW
BEGIN
    IF :NEW.salary IS NULL THEN
        :NEW.salary := 10000;   -- default salary
    END IF;
END;
/
```

✅ Insert using trigger (salary will auto-update)

```
INSERT INTO employee2_copy (sno, name, designation, branch)
VALUES (4, 'Meena', 'Analyst', 'IT');

select * from employee2_copy;
```

✅ Resulting row inserted:

sno	name	designation	branch	salary
4	Meena	Analyst	IT	10000
