DBMS LAB END SEM
Roll No 106118107

```
CREATE TABLE Department (
        DeptName VARCHAR(100) ,
        DeptId INT PRIMARY KEY,
        DeptLocation VARCHAR(100)
);

CREATE TABLE Employee (
        EmpId INT PRIMARY KEY,
        Name_ VARCHAR(100),
        DOB DATE,
        Sex VARCHAR(100),
        Salary INT ,
        DeptId INT ,
        FOREIGN KEY(DeptId) REFERENCES Department(DeptId)
);

CREATE TABLE Dependent(
        EmpId INT,
        DependentName VARCHAR(100),
        Sex VARCHAR(100),
        DOB DATE,
        Relationship VARCHAR(100)
);

CREATE TABLE Project (
        Pname VARCHAR(100),
        Pid INT PRIMARY KEY ,
        ProjLocation VARCHAR(100),
        DeptId INT
);

CREATE TABLE WorksOn(
        EmpId INT,
        Pid INT,
        hour INT
);

INSERT INTO Department VALUES
("Sales",1,"Mumbai"),
("IT",2,"DElhi"),
("Marketting",3,"Mumbai"),
("Sale",4,"Delhi");
```

```
INSERT INTO Employee VALUES
(1,"A","2000/02/19","Male",10000,1),
(2,"B","2000/02/09","Female",20000,2),
(3,"C","2000/02/12","Male",10000,1),
(4,"D","2000/02/11","Female",20000,3),
(5,"E","2000/02/14","Male",10000,1),
(6,"F","2000/02/15","Female",340000,4);

INSERT INTO Dependent VALUES
(1,"Z","Male","2000/03/14","Wife"),
(2,"X","Female","2000/02/14","Husband"),
(3,"Y","Male","2000/03/04","Wife"),
(4,"O","Female","2000/05/14","Husband"),
(5,"P","Male","2000/03/15","Wife"),
(6,"Q","Female","2000/01/14","Son");

INSERT INTO Project VALUES
("ABC",1,"Mumbai",1),
("Aas",2,"Delhi",1),
("Afs",3,"Mumbai",2),
("AfdC",4,"Delhi",3),
("ABda",5,"Mumbai",1);

INSERT INTO WorksOn VALUES
(1,1,100),
(1,2,100),
(2,3,100),
(4,4,100),
(6,5,100);


--Q1
SELECT Name_ FROM Employee
WHERE EmpId IN (
        SELECT EmpId FROM WorksOn
        WHERE hour > 10
);

--Q2
DELIMITER //
CREATE FUNCTION abAvgSalary() RETURNS INT DETERMINISTIC
BEGIN
DECLARE avgSalary INT;
DECLARE ans INT;
SELECT AVG(Salary) INTO avgSalary FROM Employee;
SELECT COUNT(*) INTO ans FROM Employee WHERE Salary>avgSalary;
RETURN ans;
END; //
DELIMITER ;

SELECT abAvgSalary();


--Q3
SELECT e.EmpId ,e.Name_,e.DOB,e.Sex,e.Salary,e.DeptId
FROM Employee e
INNER JOIN Department d ON e.DeptId = d.DeptId
WHERE d.DeptLocation='Delhi';


--Q4
DELIMITER //
CREATE PROCEDURE moreThan2Project()
BEGIN
SELECT EmpId FROM WorksOn
GROUP BY EmpId
HAVING COUNT(*) = 2;
END; //
DELIMITER ; //

CALL moreThan2Project();



--Q5
CREATE VIEW Proj AS
SELECT * FROM Project
ORDER BY ProjLocation ASC ;

SELECT * FROM Proj;

--Q6
DELIMITER //
CREATE TRIGGER DeptIdChanged
BEFORE UPDATE ON Department FOR EACH ROW
BEGIN
IF NEW.DeptId != OLD.DeptId THEN
```

```
SIGNAL SQLSTATE "45000" set message_text = "Department ID cant be changed";
END IF;
END //
DELIMITER ;

--Q7
DELIMITER //
CREATE TRIGGER preventRobert
BEFORE INSERT ON Employee FOR EACH ROW
BEGIN
IF NEW.NAME = "Robert"
THEN
SIGNAL SQLSTATE '02000' SET MESSAGE_TEXT = "Name should not be Robert";
END IF;
END //
DELIMITER ;

--Q8
DELIMITER //
CREATE FUNCTION youngestSon()
RETURNS VARCHAR(100)
DETERMINISTIC
BEGIN
DECLARE res VARCHAR(100);
SELECT Name_ INTO res
FROM Employee
WHERE EmpId IN(
        SELECT EmpId
        FROM Dependent
        WHERE DOB IN (
        SELECT MAX(DOB)
        FROM Dependent) );
RETURN res;
END  //
DELIMITER ; //

--Q9
CREATE PROCEDURE DisplayDepartmentDetails(pid numeric (10))
SELECT d.DeptName ,d.DeptId ,d.DeptLocation
FROM Employee e
INNER JOIN Department d ON e.DeptId = d.DeptId
INNER JOIN Project p ON p.DeptId=e.DeptId
WHERE p.Pid=pid;
```

```
--Q10
SELECT * FROM Employee
WHERE EmpId IN (
	SELECT EmpId FROM Dependent
	WHERE Relationship = "Son"
);
```