Sql codes

```
Create database proj;
use proj;

#all amounts are in lacs
#salary of employees is lac per annum(net salary)


CREATE TABLE Projects (
   PID VARCHAR(10) PRIMARY KEY,
   Pname VARCHAR(50),
   timeperiod FLOAT,
   status VARCHAR(20),
   expenses FLOAT,
   description VARCHAR(255),
   NoOfEmp INT
);

INSERT INTO Projects (PID, Pname, timeperiod, status, expenses, description, NoOfEmp)
VALUES
         ('P101', 'QuantumByte', 1, 'Completed', 17, 'Quantum computing software development.
Cutting-edge technology for advanced computations.', 3),
   ('P124', 'CyberVortex', 2, 'Pending', 18, 'Cybersecurity framework development. Protecting
digital assets from threats and breaches.', 3),
   ('P156', 'HyperByte', 2, 'Pending', 13, 'Innovative digital platform creation. Revolutionizing
user experiences with intuitive interfaces.', 2),
   ('P189', 'CodeSphere', 3.5, 'In process', 15, 'Agile software development. Rapid iteration for
scalable and robust applications.', 3),
   ('P235', 'DataNebula', 1.25, 'In process', 13, 'Big data analytics platform. Extracting insights
from massive datasets for informed decisions.', 2),
   ('P304', 'PixelPulse', 2.5, 'Completed', 12, 'Digital media production. Creative content
generation for engaging audience experiences.', 2),
   ('P456', 'NanoGrid', 1.5, 'Completed', 14, 'Nanotechnology research and development.
Pioneering solutions for efficient energy systems.', 3),
   ('P477', 'ByteFusion', 2.5, 'Completed', 20, ' Integration software development. Seamlessly
connecting disparate systems for streamlined operations.', 4),
   ('P560', 'QuantumMesh', 3, 'In process', 15, 'Quantum communication network. Secure and
efficient data transmission using quantum principles.', 4),
   ('P587', 'HyperLoopIT', 3.25, 'Pending', 13, 'Transportation infrastructure innovation.
High-speed transit systems for sustainable urban mobility.', 3);

CREATE TABLE Employees (
   EmployeeID INT PRIMARY KEY,
   EmployeeName VARCHAR(50),
   EmployeeRole VARCHAR(50),
   PhoneNumber CHAR(10),
   Salary DECIMAL(10, 2),
   Username VARCHAR(50) Unique key,
   Password VARCHAR(50) Unique key
);

INSERT INTO Employees (EmployeeID, EmployeeName, EmployeeRole, PhoneNumber,
Salary, Username, Password)
VALUES
   (101, 'Shraddha tripathi', 'Software Engineer', '9876543210', 10, 'shraddha_tripathi',
'shr@ddha21'),
   (102, 'Joy Shetty', 'Project Manager', '8654321098', 10, 'joy_shetty', 'shettyj123'),
   (113, 'Rajesh Mishra', 'Data Analyst', '7543210987', 9, 'rajesh_mishra', 'rajeshm45'),
   (121, 'Meena Singh', 'Web Developer', '9432109876', 10, 'meena_singh', 'meena@90'),
   (122, 'Rupali Shah', 'UI/UX Designer', '8765432109', 9, 'rupali_shah', 'shahruapali122'),
   (133, 'Veena Jain', 'Network Engineer', '9321098765', 7, 'veena_jain', 'jainv780'),
   (134, 'Rishi Kumar', 'Finance Manager', '8210987654', 12, 'rishi_kumar', 'rishik@133'),
   (141, 'Sheetal Mehra', 'Systems Analyst', '8109876543', 9, 'sheetal_mehra', 'sheetal@m'),
   (152, 'Rose D’souza', 'Admin', '9098765432', 10, 'admin_rose', 'admin_123'),
   (163, 'Tanuj Mehta', 'Software Developer', '7987654321', 12, 'tanuj_mehta', 'tanujmehta09'),
   (174, 'Usha Negi', 'Business Analyst', '9870223309', 10, 'usha_negi', 'usha@433'),
   (175, 'Asha Jaiswal', 'Frontend Developer', '9321654870', 10, 'asha_jaiswal', 'ashaj123'),
   (177, 'Manpreet Sahni', 'Security Specialist', '9924490058', 10, 'manpreet_sahni',
'manpreet268'),
   (189, 'Anshul Sharma', 'DevOps Engineer', '7834672981', 10, 'anshul_sharma',
'anshul@sha122'),
   (190, 'Viraj Singh', 'Backend Developer', '3426791765', 12, 'viraj_singh', 'virajsingh33');

CREATE TABLE ProjectEmployees (
   ProjectID VARCHAR(10),
   EmployeeID INT,
   FOREIGN KEY (ProjectID) REFERENCES Projects(PID),
   FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID),
   PRIMARY KEY (ProjectID, EmployeeID)
);

INSERT INTO ProjectEmployees (ProjectID, EmployeeID) VALUES
   ('P124', 101), ('P124', 113), ('P124', 121),
   ('P156', 102), ('P156', 121),
   ('P189', 121), ('P189', 163), ('P189', 174),
   ('P235', 101), ('P235', 141),
   ('P304', 113), ('P304', 163),
   ('P456', 122), ('P456', 177), ('P456', 190),
   ('P477', 101), ('P477', 102), ('P477', 174), ('P477', 175),
   ('P560', 133), ('P560', 134), ('P560', 175), ('P560', 189),
   ('P587', 177), ('P587', 189), ('P587', 190);

CREATE TABLE Marketing (
   MarketingID VARCHAR(10) PRIMARY KEY,
   Name VARCHAR(100),
   Amount INT,
   NoOfProj_m INT
);

INSERT INTO Marketing (MarketingID, Name, Amount, NoOfProj_m ) VALUES
   ('M1', 'Email Newsletter', 10, 2),
   ('M2', 'Webinar Series', 12, 3),
   ('M3', 'Google AdsCampaign', 13, 2),
   ('M4', 'Content Marketing', 15, 3);

CREATE TABLE ProjectMarketing (
   MarketingID VARCHAR(10),
   PID VARCHAR(10),
   FOREIGN KEY (PID) REFERENCES Projects(PID),
   FOREIGN KEY (MarketingID) REFERENCES Marketing(MarketingID),
   PRIMARY KEY (MarketingID, PID)
);

INSERT INTO ProjectMarketing (MarketingID, PID) VALUES
   ('M1', 'P124'), ('M1', 'P156'),
   ('M2', 'P124'), ('M2', 'P189'), ('M2', 'P235'),
   ('M3', 'P156'), ('M3', 'P189'),
   ('M4', 'P101'), ('M4', 'P189'), ('M4', 'P477');
```

```
CREATE TABLE ExtraExpenses (
    ExpenseID VARCHAR(10) PRIMARY KEY,
    Purpose VARCHAR(100),
    Amount INT
);

INSERT INTO ExtraExpenses (ExpenseID, Purpose, Amount)
VALUES
    ('E1', 'Office Supplies', 20),
    ('E2', 'Utilities', 15),
    ('E3', 'Software Licenses', 25),
    ('E4', 'Professional Services', 30),
    ('E5', 'Insurance', 10);

CREATE TABLE Sales (
    SaleID VARCHAR(10) PRIMARY KEY,
    ProjectID VARCHAR(10),
    Profit INT,
    FOREIGN KEY (ProjectID) REFERENCES Projects(PID)
);

INSERT INTO Sales (SaleID ,ProjectID, Profit) VALUES
('SL1', 'P101', 30),
('SL2', 'P124', 10),
('SL3', 'P156', 25),
('SL4', 'P189', 20),
('SL5', 'P235', 15),
('SL6', 'P304', 5),
('SL7', 'P456', 35),
('SL8', 'P477', 40),
('SL9', 'P560', 35),
('SL10', 'P587', 30);

CREATE TABLE Funds (
    TransactionID VARCHAR(10) PRIMARY KEY,
    InvestorName VARCHAR(100),
    Amount INT
);
```

```
INSERT INTO Funds (TransactionID, InvestorName, Amount) VALUES
('INV001', 'XYZ Ventures', 45),
('INV002', 'Growth Equity Fund', 50),
('INV003', 'Tech Investment Group', 55),
('INV004', 'Innovation Capital', 70),
('INV005', 'Seed Fund Inc.', 80);

CREATE TABLE Funds_noOfProj (
   FundID VARCHAR(10),
   NoOfProj_f INT,
   FOREIGN KEY (FundID) REFERENCES Funds(TransactionID)
);

INSERT INTO Funds_noOfProj (FundID, NoOfProj_f )
VALUES
   ('INV001', 1),
   ('INV002', 2),
   ('INV003', 3),
   ('INV004', 2),
   ('INV005', 2);

CREATE TABLE ProjectFunds (
         FundID VARCHAR(10),
   ProjectID VARCHAR(10),
   FOREIGN KEY (ProjectID) REFERENCES Projects(PID),
   FOREIGN KEY (FundID) REFERENCES Funds(TransactionID),
   PRIMARY KEY (FundID, ProjectID)
);

INSERT INTO ProjectFunds (FundID, ProjectID) VALUES
   ('INV001', 'P101'),
   ('INV002', 'P235'), ('INV002', 'P456'),
   ('INV003', 'P587'), ('INV003', 'P560'), ('INV003', 'P156'),
   ('INV004', 'P304'), ('INV004', 'P477'),
   ('INV005', 'P124'), ('INV005', 'P189');

select * from employees;
select * from projects;
select * from projectemployees;
select * from projectfunds;
select * from funds;
select * from Funds_noOfProj;
select * from marketing;
select * from projectmarketing;
select * from ExtraExpenses;
select * from sales;

show tables;

SELECT ProjectID, GROUP_CONCAT(EmployeeID) AS Employees
FROM ProjectEmployees
GROUP BY ProjectID
ORDER BY ProjectID;


#1 procedure

DELIMITER //
CREATE PROCEDURE CalculateTotalAmount(IN tableName VARCHAR(50))
BEGIN
   DECLARE total DECIMAL(10,2);
   IF tableName = 'Employees' THEN
      SELECT SUM(Salary) INTO total FROM Employees;
   ELSEIF tableName = 'Marketing' THEN
      SELECT SUM(Amount) INTO total FROM Marketing;
   ELSEIF tableName = 'Funds' THEN
      SELECT SUM(Amount) INTO total FROM Funds;
   ELSEIF tableName = 'ExtraExpenses' THEN
      SELECT SUM(Amount) INTO total FROM ExtraExpenses;
   ELSEIF tableName = 'Sales' THEN
      SELECT SUM(Profit) INTO total FROM Sales;
   ELSE
      SET total = NULL;
   END IF;
   SELECT total AS TotalAmount_in_lacs;
END//
DELIMITER ;

CALL CalculateTotalAmount('Employees');
CALL CalculateTotalAmount('Marketing');
CALL CalculateTotalAmount('Funds');
CALL CalculateTotalAmount('ExtraExpenses');
CALL CalculateTotalAmount('Sales');

#2 procedure

DELIMITER //
CREATE PROCEDURE GetProjectsByStatus(IN projectStatus VARCHAR(20))
BEGIN
   SELECT *
   FROM Projects
   WHERE status = projectStatus;
END//
DELIMITER ;

call project.GetProjectsByStatus('Completed');
call project.GetProjectsByStatus('Pending');
call project.GetProjectsByStatus('In process');

#3 procedure

DELIMITER //

CREATE PROCEDURE GetEmployeeProjectDetails(IN emp_username VARCHAR(50))
BEGIN
   DECLARE emp_id INT;

   -- Get the EmployeeID based on the username
   SELECT EmployeeID INTO emp_id FROM Employees WHERE Username =
emp_username;

   -- Get the project details for the employee
   SELECT
      p.PID AS ProjectID,
      p.Pname AS ProjectName,
      p.timeperiod AS TimePeriod,
      p.status AS Status,
      p.expenses AS Expenses,
      p.description AS Description,
        GROUP_CONCAT(CONCAT(e.EmployeeName, ' - ', e.EmployeeRole, ' - ',
e.PhoneNumber)) AS TeamMembersDetails
    FROM Projects p
    INNER JOIN ProjectEmployees pe ON p.PID = pe.ProjectID
    INNER JOIN Employees e ON pe.EmployeeID = e.EmployeeID
    WHERE pe.ProjectID IN (
        SELECT ProjectID FROM ProjectEmployees WHERE EmployeeID = emp_id
    )
    GROUP BY p.PID;
END //

DELIMITER ;

CALL GetEmployeeProjectDetails('shraddha_tripathi');

#4 procedure

DELIMITER //

CREATE PROCEDURE CalculateSalarySplitByUsername(IN emp_username VARCHAR(50))
BEGIN
    DECLARE emp_id INT;
    DECLARE emp_salary DECIMAL(10, 2);
    DECLARE bs DECIMAL(10, 2);
    DECLARE hra DECIMAL(10, 2);
    DECLARE sa DECIMAL(10, 2);
    DECLARE bonus DECIMAL(10, 2);
    DECLARE pf DECIMAL(10, 2);
    DECLARE tax DECIMAL(10, 2);
    DECLARE net_salary DECIMAL(10, 2);

    SELECT EmployeeID, Salary INTO emp_id, emp_salary FROM Employees WHERE
Username = emp_username;

    -- Calculating Basic Salary Annually
    SET net_salary = emp_salary;

    -- Calculating House Rent Allowance (HRA)
    SET hra = net_salary * 0.2;
    -- Calculating Special Allowance (SA)
    SET sa = net_salary * 0.36;

    -- Calculating Bonus
    SET bonus = net_salary * 0.04;

    -- Calculating Provident Fund (PF)
    SET pf = net_salary * 0.048;

    -- Calculating Income Tax
    IF net_salary <= 3 THEN
        SET tax = 0;
    ELSEIF net_salary <= 6 THEN
        SET tax = (net_salary - 3) * 0.02;
    ELSEIF net_salary <= 9 THEN
        SET tax = 0.15 + (net_salary - 6) * 0.04;
    ELSEIF net_salary <= 12 THEN
        SET tax = 0.45 + (net_salary - 9) * 0.06;
    ELSEIF net_salary <= 15 THEN
        SET tax = 0.9 + (net_salary - 12) * 0.08;
    ELSE
        SET tax = 1.95 + (net_salary - 15) * 0.12;
    END IF;

    -- Calculating Base Salary
    SET bs = net_salary - (hra + sa + bonus - tax - pf);

    -- Returning the result
    SELECT
        emp_id,
        emp_username AS Username,
        bs AS BasicSalary,
        hra AS HouseRentAllowance,
        sa AS SpecialAllowance,
        bonus AS Bonus,
        tax AS IncomeTax,
        pf AS ProvidentFund,
        net_salary AS NetSalary;
END //
```

```
DELIMITER ;

CALL CalculateSalarySplitByUsername('shraddha_tripathi'); #10
CALL CalculateSalarySplitByUsername('rajesh_mishra'); #9
CALL CalculateSalarySplitByUsername('veena_jain'); #7
CALL CalculateSalarySplitByUsername('rishi_kumar'); #12


#5 procedure
DELIMITER //
CREATE PROCEDURE UpdateEmployee(
    IN empID INT,
    IN empRole VARCHAR(50),
    IN phone CHAR(10),
    IN salary DECIMAL(10, 2)
)
BEGIN
    UPDATE Employees
    SET EmployeeRole = empRole,
        PhoneNumber = phone,
        Salary = salary
    WHERE EmployeeID = empID;
END //

DELIMITER ;
CALL UpdateEmployee(102, 'Senior Project Manager', '9876543210', 12);
CALL UpdateEmployee(102, 'Project Manager', '8654321098', 10);
select * from employees;



#1 function

DELIMITER //
CREATE FUNCTION `CalculateNetProfitLoss`() RETURNS VARCHAR(50)
DETERMINISTIC
BEGIN
    DECLARE total_expenses DECIMAL(10,2);
    DECLARE total_income DECIMAL(10,2);
    DECLARE net_profit_loss DECIMAL(10,2);
    -- Subtract total income from projects
    SELECT SUM(expenses) INTO total_expenses FROM Projects;

    -- Add total income from funds
    SELECT SUM(Amount) INTO total_income FROM Funds;

    -- Add total profit from sales
    SELECT SUM(Profit) INTO total_income FROM Sales;

    -- Subtract total employee salaries
    SELECT SUM(Salary) INTO total_expenses FROM Employees;

    -- Subtract total marketing expenses
    SELECT SUM(Amount) INTO total_expenses FROM Marketing;

    -- Subtract total extra expenditures
    SELECT SUM(Amount) INTO total_expenses FROM ExtraExpenses;

    -- Calculate net profit or loss
    SET net_profit_loss = total_income - total_expenses;

    -- Return result with sign
    RETURN CONCAT(
        CASE
            WHEN net_profit_loss >= 0 THEN '+'
            ELSE '-'
        END,
        ABS(net_profit_loss)
    );
END//
DELIMITER ;

SELECT CalculateNetProfitLoss() AS 'Net Profit/Loss';

#2 function

DELIMITER //
```

```
CREATE FUNCTION UpdateProjectStatus(projectID VARCHAR(10), newStatus
VARCHAR(20)) RETURNS VARCHAR(100)
deterministic
BEGIN
   DECLARE rowsAffected INT;
   UPDATE Projects
   SET status = newStatus
   WHERE PID = projectID;
   SET rowsAffected = ROW_COUNT();
   IF rowsAffected > 0 THEN
      RETURN CONCAT('Status of project ', projectID, ' updated successfully.');
   ELSE
      RETURN CONCAT('Project ', projectID, ' not found. Status update failed.');
   END IF;
END //

DELIMITER ;

SELECT UpdateProjectStatus('P101', 'Pending');
Select * from projects;
SELECT UpdateProjectStatus('P101', 'Completed');

#3 Functions
DELIMITER //

CREATE FUNCTION subtotal(category VARCHAR(20))
RETURNS DECIMAL(10,2)
DETERMINISTIC
BEGIN
   DECLARE total DECIMAL(10,2);

   IF category = 'employee' THEN
      SELECT SUM(Salary) INTO total FROM employees;
   ELSEIF category = 'project' THEN
      SELECT SUM(expenses) INTO total FROM projects;
   ELSEIF category = 'extra' THEN
      SELECT SUM(Amount) INTO total FROM extraexpenses;
   ELSEIF category = 'sales' THEN
      SELECT SUM(profit) INTO total FROM sales;
   ELSEIF category = 'marketing' THEN
```

```
        SELECT SUM(Amount) INTO total FROM marketing;
    ELSEIF category = 'funds' THEN
        SELECT SUM(Amount) INTO total FROM funds;
    ELSE
        SET total = 0; -- Or any other value you wish to return for unknown category
    END IF;

    RETURN total;
END//

DELIMITER ;

SELECT subtotal('project');
SELECT subtotal('extra');
SELECT subtotal('sales');
SELECT subtotal('marketing');
SELECT subtotal('funds');
```

Trigger 1

```
DELIMITER $$

CREATE TRIGGER GenerateUniqueRandomPID
BEFORE INSERT ON Projects
FOR EACH ROW
BEGIN
    DECLARE random_string VARCHAR(4); -- Since "P" and three digits
    DECLARE existing_count INT;
    SET @chars = '1234567890'; -- Digits only

    REPEAT
        SET random_string = 'P'; -- Start with "P"
        SET @i = 1;
        WHILE @i <= 3 DO
```

```
      SET random_string = CONCAT(random_string, SUBSTRING(@chars, FLOOR(1 +
RAND() * 10), 1));
      SET @i = @i + 1;
    END WHILE;
    SET existing_count = (SELECT COUNT(*) FROM Projects WHERE PID =
random_string);
  UNTIL existing_count = 0 END REPEAT;
  SET NEW.PID = random_string;
END$$
DELIMITER ;

#trigger 2
DELIMITER //

CREATE TRIGGER calculate_no_of_employees AFTER INSERT ON ProjectEmployees
FOR EACH ROW
BEGIN
  DECLARE project_id VARCHAR(10);
  DECLARE emp_count INT;

  -- Get the project ID for the newly inserted row
  SET project_id = NEW.ProjectID;

  -- Count the number of employees for the project
  SELECT COUNT(*) INTO emp_count FROM ProjectEmployees WHERE ProjectID =
project_id;

  -- Update the NoOfEmp column in the Projects table with the calculated count
  UPDATE Projects SET NoOfEmp = emp_count WHERE PID = project_id;
END;
//

DELIMITER ;


#trigger 3
DELIMITER $$
CREATE TRIGGER extraexpenses_id_trigger BEFORE INSERT ON extraexpenses
FOR EACH ROW
BEGIN
```

```
    DECLARE max_id INT;
    SELECT COALESCE(MAX(CAST(SUBSTRING(ExpenseID, 2) AS UNSIGNED)), 0)
INTO max_id FROM extraexpenses;
    SET NEW.ExpenseID = CONCAT('E', max_id + 1);
END$$
DELIMITER ;
```