From the above given tables perform the following queries:

Part - A

1. Create a stored procedure that takes department name as an input and returns a table with all workers working in that department.

```
CREATE PROCEDURE pr_worker_working_department
        @DepartmentName varchar(100)
AS
BEGIN
SELECT Department.DepartmentName,
         Person.FirstName,
         Person.LastName,
         Person.JoiningDate,
         Person.Salary
FROM Department INNER JOIN Person
ON Person.DepartmentID = Department.DepartmentID
WHERE Department.DepartmentName = @DepartmentName
END

EXEC pr_worker_working_department 'Admin'
```

2. Create procedure that takes department name & designation name as input and returns a table with worker's first name, salary, joining date & department name.

```
CREATE PROCEDURE pr_workers_by_department_and_designation
        @DepartmentName VARCHAR(100),
        @DesignationName VARCHAR(100)
AS
BEGIN
SELECT Person.FirstName,
         Person.Salary,
         Person.JoiningDate,
         Department.DepartmentName
FROM Person
INNER JOIN Department
ON Person.DepartmentID = Department.DepartmentID
INNER JOIN Designation
ON Person.DesignationID = Designation.DesignationID
WHERE Department.DepartmentName = @DepartmentName
        AND
        Designation.DesignationName = @DesignationName;
END;

EXEC pr_workers_by_department_and_designation 'Admin','Welder'
```

3. Create a Procedure that takes the first name as an input parameter and display all the details of the worker with their department & designation name.

```
CREATE PROCEDURE pr_display_all_info
@FirstName varchar(100)
AS
BEGIN
SELECT Person.WorkerID,
       Person.FirstName,
       Person.LastName,
       Person.JoiningDate,
       Person.Salary,
       Department.DepartmentName,
       Designation.DesignationName
FROM Person
INNER JOIN Department
ON Person.DepartmentID = Department.DepartmentID
INNER JOIN Designation
ON Person.DesignationID = Designation.DesignationID
WHERE Person.FirstName = @FirstName
END;

EXEC pr_display_all_info 'Rahul'
```

4. Create Procedure which displays department wise maximum, minimum & total salaries.

```
CREATE PROCEDURE pr_department_wise_salary
AS
BEGIN
SELECT Department.DepartmentName,
       MAX(Person.Salary) AS MaxSalary,
       MIN(Person.Salary) AS MinSalary,
       SUM(Person.Salary) AS TotalSalry
FROM Person
INNER JOIN Department
ON Person.DepartmentID = Department.DepartmentID
GROUP BY Department.DepartmentName
END

EXEC pr_department_wise_salary
```

5. Create Procedure which displays designation wise maximum, minimum & total salaries.

```
CREATE PROCEDURE pr_designation_wise_salary
AS
BEGIN
SELECT Designation.DesignationName,
            MAX(Person.Salary) AS MaxSalary,
            MIN(Person.Salary) AS MinSalary,
            SUM(Person.Salary) AS TotalSalry
FROM Person
INNER JOIN Designation
ON Person.DesignationID = Designation.DesignationID
GROUP BY Designation.DesignationName
END

EXEC pr_designation_wise_salary
```

Part - B

1. Create a Stored Procedure to Calculate Total Salary Expense.

```
CREATE PROCEDURE pr_total_salary_Employees
AS
BEGIN
        SELECT SUM(Emp_Salary) as TOTALSALARY
        FROM Employees
END

EXEC pr_total_salary_Employees
```

2. Create a Stored Procedure to Get Employees with the Longest Tenure.

```
CREATE PROCEDURE pr_earlier_date_of_employees
AS
BEGIN
        DECLARE @EarlierDate Date
        SELECT @EarlierDate =  MIN(Hire_Date) FROM Employees
        SELECT * FROM Employees WHERE Hire_Date = @EarlierDate
END

EXEC pr_earlier_date_of_employees
```

3. Create a Stored Procedure to Calculate the Total Number of Employees in Each Department.

```
CREATE PROCEDURE pr_Total_employee_Eachdepartment
AS
BEGIN
        SELECT Department, SUM(Emp_ID)
        FROM Employees
        GROUP BY Department
END
```

```
EXEC pr_Total_employee_Eachdepartment
```

4. Create a Stored Procedure to Calculate the Average Salary for Each Department.

```
CREATE PROCEDURE pr_averagesalary_employee_Eachdepartment
AS
BEGIN
        SELECT Department, AVG(Emp_Salary)
        FROM Employees
        GROUP BY Department
END

EXEC pr_averagesalary_employee_Eachdepartment
```

Part - C

5. Create a Stored Procedure to Calculate Average Salary in a Department.

```
CREATE PROCEDURE CalculateAverageSalaryByDepartment
@DepartmentName VARCHAR(50)
AS
BEGIN
SELECT Department, AVG(Emp_Salary) AS AverageSalary
FROM Employees
WHERE Department = @DepartmentName
GROUP BY Department
END;

EXEC CalculateAverageSalaryByDepartment 'IT'
```

6. Create a Stored Procedure to Generate Monthly Salary Report.

```
CREATE PROCEDURE GenerateMonthlySalaryReport
AS
BEGIN
  -- Generate the SALARY Report
        SELECT Emp_ID,
                 Emp_Name,
                 Department,
                 Emp_Salary AS Annual_Salary,
                 CAST(Emp_Salary / 12 AS DECIMAL(10,2)) AS Monthly_Salary,
                 Hire_Date
        FROM Employees
        ORDER BY Department, Emp_Name;
END

EXEC GenerateMonthlySalaryReport
```

7. Create a Stored Procedure to Get Highest Paid Employee.

```
CREATE PROCEDURE pr_higest_paid_salary
AS
BEGIN
SELECT * FROM Employees
WHERE Emp_Salary = (SELECT MAX(Emp_Salary) FROM Employees)
END

EXEC pr_higest_paid_salary
```

8. Create a Stored Procedure to Get Employees Hired in a Specific Year.

```
CREATE PROCEDURE pr_employee_hire_specific_year
        @HireYear INT
AS
BEGIN
        SELECT * FROM Employees
        WHERE YEAR(Hire_Date) = @HireYear
END

EXEC pr_employee_hire_specific_year 2021
```