

```

USE [Lab_6]
-- TASK1
WITH Clients_CTE AS (
    SELECT C.ClientName, P.Description, E.Status
    FROM Client C
    JOIN Project P ON C.ClientID = P.ClientID
    JOIN EmployeeProjectTask E ON P.ProjectID = E.ProjectID
)
SELECT ClientName, Description, Status
FROM Clients_CTE;

-- TASK2

WITH Departments_CTE AS(
    SELECT D.DepartmentNo,D.DepartmentName FROM
    Department D
    INNER JOIN
    Employee E
    ON E.DepartmentNo = D.DepartmentNo
    GROUP BY D.DepartmentNo,D.DepartmentName
    HAVING COUNT(*) >= 3
)
SELECT DepartmentNo,DepartmentName FROM Departments_CTE;

-- TASK 3
WITH DepartmentSalary AS ( -- select department salary
    SELECT DepartmentNo, AVG(Salary) AS AverageSalary
    FROM Employee
    GROUP BY DepartmentNo
), EmployeesCount AS ( -- employee count at each department
    SELECT DepartmentNo, COUNT(*) AS EmployeeCount
    FROM Employee
    GROUP BY DepartmentNo
), EmployeesAboveAvg AS ( -- employees above average
    SELECT e.DepartmentNo,ds.AverageSalary AS avg_salary,COUNT(*) AS
EmployeesAboveAverage
    FROM Employee e
    JOIN DepartmentSalary ds

        ON e.DepartmentNo = ds.DepartmentNo
    WHERE e.Salary >= ds.AverageSalary

        GROUP BY e.DepartmentNo,ds.AverageSalary
)
SELECT * FROM EmployeesAboveAvg;

-- TASK 4
GO
CREATE VIEW [total_salary] AS
    SELECT DepartmentNo, SUM(Salary) AS total_salary
    FROM Employee
    GROUP BY DepartmentNo
GO

```

```

SELECT * FROM [total_salary]

-- TASK 5
GO
CREATE VIEW[3_chars] AS
    SELECT ClientName,SUBSTRING(ClientName,1,3) AS first_3_chars
    FROM Client
GO

SELECT * FROM [3_chars];

--TASK 6

CREATE PROCEDURE Employee_GetAll
AS
SELECT EmployeeName,Job,Salary FROM
Employee
GO;

EXEC Employee_GetAll

-- TASK 7
CREATE PROCEDURE Employee_Insert
    @EmployeeNo INT,@EmployeeName VARCHAR(30),@Job VARCHAR(50),@Salary INT,@DepartmentNo INT
AS
    INSERT INTO Employee VALUES
    (@EmployeeNo,@EmployeeName,@Job,@Salary,@DepartmentNo);

EXEC Employee_Insert @EmployeeNo = 31,@EmployeeName = 'Geralt of Rivia',
@Job = 'Killing monsters',@Salary = 30000,@DepartmentNo = 1;

SELECT * FROM Employee;

-- TASK 8
CREATE PROCEDURE Employee_Update
@ClientID INT,@ClientName VARCHAR(50)
AS
UPDATE Client
SET ClientName = @ClientName
WHERE ClientID = @ClientID

EXEC Employee_Update @ClientID = 2,@ClientName = 'Small Retailer'

SELECT * FROM Client;

-- TASK 9

CREATE PROCEDURE Employee_Delete
@EmployeeNo INT
AS
    DELETE FROM Employee
    WHERE EmployeeNo = @EmployeeNo

EXEC Employee_Delete @EmployeeNo = 31

SELECT * FROM Employee

```

-- TASK 10

```
ALTER TABLE Department
ADD number_of_employees INT;
```

```
SELECT * FROM Department;
```

```
WITH EmployeesCount AS (
    SELECT DepartmentNo, COUNT(*) AS EmployeeCount
    FROM Employee
    GROUP BY DepartmentNo
)
UPDATE Department
SET number_of_employees = ec.EmployeeCount
FROM Department d
INNER JOIN EmployeesCount ec ON d.DepartmentNo = ec.DepartmentNo;

SELECT * FROM Department;
```

```
CREATE TRIGGER IncreaseNumberOfEmployees
ON Employee
AFTER INSERT
AS
BEGIN
    UPDATE Department
    SET number_of_employees = number_of_employees + 1
    WHERE DepartmentNo IN (SELECT DepartmentNo FROM inserted);
END;
```