# SQL SERVER PRACTICAL EXAM (20 MARKS)

## Scenario:

A company manages Employees, Departments, and Projects. You are required to perform various SQL operations on these tables.

## Tables Structure:

### Employees

* EmployeeID (INT, Primary Key)
* Name (VARCHAR(50))
* DepartmentID (INT, Foreign Key references Departments(DepartmentID))
* Salary (DECIMAL(10,2))
* HireDate (DATE)

### Departments

* DepartmentID (INT, Primary Key)
* DepartmentName (VARCHAR(50))

### Projects

* ProjectID (INT, Primary Key)
* ProjectName (VARCHAR(50))
* EmployeeID (INT, Foreign Key references Employees(EmployeeID))
* StartDate (DATE)

## Tasks:

1. Create an Employees record with relevant values.

2. Update the Salary of an employee with EmployeeID = 101 to 80000.

3. Delete an employee whose EmployeeID = 105.

4. Retrieve all employees who were hired after 2020-01-01.

5. Display the Employee Name, Department Name, and Project Name for all employees working on projects.

6. Create a Scalar Function named GetEmployeeCount that returns the total number of employees in a given department.

7. Create a View named EmployeeProjects that shows Employee Name, Project Name, and StartDate of the project.

8. Create a Stored Procedure named GetHighSalaryEmployees that takes a Salary value as input and returns employees earning more than the given value.

9. Create an AFTER INSERT trigger on the Employees table that logs the new employee details into a new table named EmployeeLog with columns:  
 LogID (INT, Primary Key)  
 EmployeeID (INT)  
 ActionDate (DATETIME)

10. Retrieve the Employee Name of the employee(s) earning the highest salary using a subquery.

11. Create a Non-clustered Index on the Salary column of the Employees table.

12. Grant SELECT permission on the Projects table to a user named User123.

13. Declare a variable @AvgSalary and store the average salary of all employees in it. Display the value.

## Submission:

Submit a `.sql` file containing all your solutions.