**Lab 1: ROW\_NUMBER()**

**Task:** Assign a unique row number to each employee based on their hire date, ordered by the most recent hire first.

SELECT EmployeeID, Name, HireDate,

ROW\_NUMBER() OVER (ORDER BY HireDate DESC) AS RowNum

FROM Employees;

**Lab 2: RANK()**

**Task:** Rank employees by salary in descending order. Employees with the same salary should receive the same rank, and the next rank should be skipped accordingly.

SELECT EmployeeID, Name, Salary,

RANK() OVER (ORDER BY Salary DESC) AS SalaryRank

FROM Employees;

**Lab 3: DENSE\_RANK()**

**Task:** Assign a dense rank to employees based on their salary. Unlike RANK(), there should be no gaps in ranking.

SELECT EmployeeID, Name, Salary,

DENSE\_RANK() OVER (ORDER BY Salary DESC) AS SalaryDenseRank

FROM Employees;

**Lab 4: NTILE()**

**Task:** Divide employees into 4 salary groups (quartiles).

SELECT EmployeeID, Name, Salary,

NTILE(4) OVER (ORDER BY Salary DESC) AS SalaryQuartile

FROM Employees;

**Lab 5: LAG()**

**Task:** Retrieve the previous employee's salary for each employee based on hire date order.

SELECT EmployeeID, Name, Salary,

LAG(Salary, 1, NULL) OVER (ORDER BY HireDate) AS PrevSalary

FROM Employees;

**Lab 6: LEAD()**

**Task:** Retrieve the next employee's salary for each employee based on hire date order.

SELECT EmployeeID, Name, Salary,

LEAD(Salary, 1, NULL) OVER (ORDER BY HireDate) AS NextSalary

FROM Employees;

**Lab 7: FIRST\_VALUE()**

**Task:** Retrieve the highest salary in the company for each employee's row.

SELECT EmployeeID, Name, Salary,

FIRST\_VALUE(Salary) OVER (ORDER BY Salary DESC) AS HighestSalary

FROM Employees;

**Lab 8: LAST\_VALUE()**

**Task:** Retrieve the lowest salary in the company for each employee's row.

SELECT EmployeeID, Name, Salary,

LAST\_VALUE(Salary) OVER (ORDER BY Salary DESC

ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS LowestSalary

FROM Employees;

**Lab 9: SUM() OVER PARTITION**

**Task:** Calculate the total salary for each department.

SELECT EmployeeID, Name, Department, Salary,

SUM(Salary) OVER (PARTITION BY Department) AS TotalDeptSalary

FROM Employees;

**Lab 10: AVG() OVER PARTITION**

**Task:** Calculate the average salary for each department.

SELECT EmployeeID, Name, Department, Salary,

AVG(Salary) OVER (PARTITION BY Department) AS AvgDeptSalary

FROM Employees;