**Lab Exercise 1: UNION**

**Problem:** Retrieve all unique employee names from the HR\_Employees and Finance\_Employees tables.

SELECT EmployeeName FROM HR\_Employees

UNION

SELECT EmployeeName FROM Finance\_Employees;

**Explanation:** The UNION operator combines the results from both tables and removes duplicates.

**Lab Exercise 2: UNION ALL**

**Problem:** Retrieve all employee names from both HR\_Employees and Finance\_Employees, including duplicates.

SELECT EmployeeName FROM HR\_Employees

UNION ALL

SELECT EmployeeName FROM Finance\_Employees;

**Explanation:** The UNION ALL operator includes duplicates in the final result.

**Lab Exercise 3: INTERSECT**

**Problem:** Find employees who are present in both the HR\_Employees and Finance\_Employees tables.

SELECT EmployeeName FROM HR\_Employees

INTERSECT

SELECT EmployeeName FROM Finance\_Employees;

**Explanation:** The INTERSECT operator returns only common values between both tables.

**Lab Exercise 4: EXCEPT**

**Problem:** Retrieve employee names from HR\_Employees that are **not** in Finance\_Employees.

SELECT EmployeeName FROM HR\_Employees

EXCEPT

SELECT EmployeeName FROM Finance\_Employees;

**Explanation:** The EXCEPT operator returns values from the first query that do not appear in the second.

**Lab Exercise 5: UNION with Additional Columns**

**Problem:** Retrieve employee names along with their department (HR or Finance) from both tables.

SELECT EmployeeName, 'HR' AS Department FROM HR\_Employees

UNION

SELECT EmployeeName, 'Finance' AS Department FROM Finance\_Employees;

**Explanation:** This query merges data from both tables while adding a department label.

**Lab Exercise 6: Using ORDER BY with UNION**

**Problem:** Retrieve unique employee names from both tables and sort them in ascending order.

SELECT EmployeeName FROM HR\_Employees

UNION

SELECT EmployeeName FROM Finance\_Employees

ORDER BY EmployeeName ASC;

**Explanation:** The ORDER BY clause applies after the set operation.

**Lab Exercise 7: Using UNION with Different Column Names**

**Problem:** Retrieve unique employee emails from HR\_Employees (Email) and Finance\_Employees (EmpEmail).

SELECT Email AS EmployeeEmail FROM HR\_Employees

UNION

SELECT EmpEmail FROM Finance\_Employees;

**Explanation:** The column names can differ but should have the same data type.

**Lab Exercise 8: EXCEPT with Additional Condition**

**Problem:** Retrieve employee names from HR\_Employees who **do not exist** in Finance\_Employees and have a salary above 50000.

SELECT EmployeeName FROM HR\_Employees

WHERE Salary > 50000

EXCEPT

SELECT EmployeeName FROM Finance\_Employees;

**Explanation:** The WHERE condition filters before applying EXCEPT.

**Lab Exercise 9: INTERSECT with Additional Condition**

**Problem:** Find employees working in **both departments** whose salary is above 40000.

SELECT EmployeeName FROM HR\_Employees WHERE Salary > 40000

INTERSECT

SELECT EmployeeName FROM Finance\_Employees WHERE Salary > 40000;

**Explanation:** The query filters data **before** applying INTERSECT.

**Lab Exercise 10: UNION ALL with Grouping**

**Problem:** Retrieve the total count of employees in both tables.

SELECT 'HR' AS Department, COUNT(\*) AS TotalEmployees FROM HR\_Employees

UNION ALL

SELECT 'Finance', COUNT(\*) FROM Finance\_Employees;

**Explanation:** This query keeps separate counts from each department.