SQL Practice Exercise: JOINS, UNION, Filtering & Aggregates

SQL JOINS

① INNER JOIN

```
1. SELECT    EmployeeID,
             FirstName,
             LastName,
             Department,
             Salary,
             ProjectID,
             ProjectName,
             Budget,
             Status
   FROM Employees AS A
   INNER JOIN Projects AS B
        ON A.EmployeeID = B.EmployeeID;
```

EmployeeID	FirstName	LastName	Department	Salary	ProjectID	ProjectName	Budgets	Status
1	John	Doe	IT	70000	101	AI Development	100 000	Complete
2	Alice	Smith	HR	60000	102	Employee Training	50 000	ongoing
3	Bob	Johnson	Finance	75 000	104	Financial Analysis	90 000	ongoing
5	Emma	Wilson	Sales	65 000	105	Market Expansion	65 000	Complete
6	Michael	Clark	Finance	80 000	106	Risk Management	80 000	Pending.

② LEFT JOIN

```
2. SELECT    EmployeeID,
             FirstName,
             LastName,
             Department,
             Salary,
             ProjectID,
             ProjectName,
             Budget,
             Status
      FROM
   ~~LEFT JOIN~~    Employees AS A
      LEFT JOIN Project AS B
      ON A.EmployeeID = B.Employee.ID;
```

	Status
1	completed
2	ongoing
3	ongoing
4	NULL
5	Completed
6	Pending.

EmployeeID	FirstName	LastName	Department	Salary	ProjectID	ProjectName	Budget
1	John	Doe	IT	70000	~~AI~~ 101 ~~Development~~	AI Development	100000
2	Alice	Smith	HR	60000	102	Employee Training	50000
3	Bob	Johnson	Finance	75000	104	Financial Analyst	90000
4	David	Brown	IT	72000	NULL	NULL	NULL
5	Emma	Wilson	Sales	65000	105	Market Expansion	65000
6	Michael	Clark	Finance	80000	106	Risk Management	8000

③ RIGHT JOIN

```
3. SELECT
            ProjectID,
            ProjectName,
            Budget,
            Status,
            EmployeeID,
            FirstName,
            LastName,
            Department,
            Salary
   FROM     Projects AS A
   RIGHT JOIN Employees AS B
        ON A.~~Projects~~ EmployeeID = B.EmployeeID;
```

ProjectID	ProjectName	Budget	Status	EmployeeID	FirstName	LastName	Department	Salary
101	AI Development	100000	Completed	1	John	Doe	IT	70000
102	Employee Train	50000	ongoing	2	Alice	Smith	HR	60000
103	Cyber security	75000	Pending	1	NULL	NULL	NULL	NULL
104	Financial Anal	90000	Ongoing	3	Bob	Johnson	Finance	75000
105	Market Exp	65000	Completed	5	Emma	Wilson	Sales	68000
106	Risk Management	80000	Pending	6	Micheael	Clark	Finance	80000

④ RULL OUTER JOIN

```
4. SELECT
          EmployeeID,
          FirstName,
          LastName,
          Department,
          Salary
          ProjectID,
          ProjectName,
          Budget,
          Status
    FROM  Employees AS A
FULL OUTER JOIN Project AS B
       ON A. employeeID = B.employeeID;
```

Employee ID	First Name	Last Name	Department	Salary	Project ID	ProjectName	Budget	Status
1	John	Doe	IT	70000	101	AI Development	100000	completed
2	Alice	Smith	HR	60000	102	Employee Training	50000	ongoing
3	Bob	Johnson	Finance	75000	104	Financial Analyst	90000	ongoing
4	David	Brown	IT	72000	NULL	NULL	Null	Null
5	Emma	Wilson	Sales	65000	105	Market Expanson	65000	completed
6	Michael	Clark	Finance	80000	106	Risk Management	80000	Pending

Filtering Statements:

```
7. SELECT
          EmployeeID,
          FirstName,
          LastName,
          Department,
          Salary
    FROM  Employees
    WHERE salary > 70000;
```

EmployeeID	FirstName	LastName	Department	Salary
1	John	Doe	IT	70000
3	Bob	Johnson	Finance	75000
4	David	Brown	IT	72000
6	Michael	Clark	Finance	80000

```
8. SELECT  EmployeeID,
           FirstName,
           LastName,
           Department,
           Salary
    FROM   employees   ('IT', 'Finance'));
    WHERE  department ~~= 'IT' OR department = 'Finance'~~);
```

Employee ID	FirstName	LastName	Department	Salary
1	John	Doe	IT	70000
3	Bob	Johnson	Finance	75000
4	David	Brown	IT	72000
6	Michael	Clark	Finance	80000

```
9.  SELECT   ProjectID,
             ProjectName,
             Budget,
             Status
    FROM     Projects
    WHERE    status = 'ongoing' OR status = 'pending'
```

ProjectID	ProjectName	Budget	Status
102	Employee Training	50000	Ongoing
103	Cybersecurity Audit	75 000	Pending
104	Financial Analyst	90 000	Ongoing
106	Risk Management	80 000	Pending.

```
10. SELECT
        ProjectID,
        ProjectName,
        Budget,
        Status
    FROM Projects
    WHERE budget > 70000
    AND status = 'ongoing' OR Status 'pending'
```

ProjectID	Project~~Name~~	Budget	Status
103	Cybersecurity Audit	75000	Pending
104	Financial Analysis	90000	ongoing
106	Ris~~k~~ Management	80000	Pending

```
11. SELECT
        EmployeeID,
        FirstName,
        LastName,
        Department,
        Salary,
        City
    FROM Employees
    WHERE city IN ('New York', 'Toronto')
    ORDER BY Salary DESC;
```

Employee ID	FirstName	LastName	Department	Salary	City
1	John	Doe	IT	70000	New York
3	Bob	Johnson	Finance	75000	Toronto
6	Michael	Clark	Finance	80000	New York

```
12. SELECT
        EmployeeID,
        FirstName,
        LastName
        Department,
        Salary
    FROM employees
    ORDER BY Salary DESC
    LIMIT 3;
```

EmployeeID	FirstName	LastName	Department	Salary
6	Michael	Clark	Finance	80000
3	Bob	Johnson	Finance	78000
4	David	Brown	IT	75000

Aggregate functions with GROUPBY & HAVING

```
13. SELECT department,
    SUM (salary) AS total_salary
    FROM employees
    GROUP BY department DESC;

14. SELECT city,
      AVG(salary) AS AverageSalary
     FROM Employees
    WHERE salary >65;

15. SELECT department
     COUNT (EmployeeID) AS EmployeeCount
    FROM employees
    GROUP BY department
    HAVING COUNT (*) >1;
    ORDER BY Employee_Count;

16. SELECT Status,
     COUNT (ProjectID) AS ProjectCount
     FROM Project
     GROUPBY Status
     HAVING COUNT (*) >2
     ORDER BY projects;

17. SELECT EmployeeID,
           FirstName,
           lastName,
    SUM (Budget) AS TotalProjectBudget
    FROM Employees
    GROUP BY Budget
     HAVING SUM (Budget) >150000;
```