

1. SQL JOINS

1. Retrieve all employees and their assigned projects using an **INNER JOIN**.
 - Return: EmployeeID, FirstName, LastName, Department, Salary, ProjectID, ProjectName, Budget, Status.

```
15 -- SQL JOINS
16
17
18 -- 1. Retrieve all employees and their assigned projects using an INNER JOIN.Return: EmployeeID, FirstName, LastName, Department, Salary,ProjectID, ProjectName,
19 | Budget, Status.
20 | SELECT e.ID, e.first_name, e.last_name, e.Department, e.Salary, p.ProjectID, p.ProjectName, p.Budget, p.Status
21 | FROM "EMPLOYEES"."EMPLOYEES_SCHEMA"."EMPLOYEES" AS e
22 | INNER JOIN
23 | "EMPLOYEES"."EMPLOYEES_SCHEMA"."PROJECTS" AS P ON e.id = P.EMPLOYEEID;
24 |
25
```

Results

# ID	A FIRST_NAME	A LAST_NAME	A DEPARTMENT	# SALARY	# PROJECTID	A PROJECTNAME	# BUDGET	A STATUS
1 1	John	Doe	IT	55000	101	AI Development	100000	Completed
2 1	John	Doe	IT	55000	103	Cybersecurity Audit	75000	Pending
3 2	Jane	Smith	HR	48000	102	Employee Training	50000	Ongoing
4 3	Mike	Johnson	Finance	60000	104	Financial Analysis	90000	Ongoing
5 5	David	White	Marketing	52000	105	Market Expansion	65000	Completed
6 6	Emily	Davis	IT	62000	106	Risk Management	80000	Pending

Query Details

Query duration 138ms

Rows 6

Query ID 01bc9a59-0001-0cd5-0...

Show more

2. Retrieve all employees and their assigned projects, including employees who have no projects using a **LEFT JOIN**.
 - Return: EmployeeID, FirstName, LastName, Department, Salary, ProjectID, ProjectName, Budget, Status.

```
25
26
27
28 --2. Retrieve all employees and their assigned projects, including employees who have no projects using a LEFT JOIN.
29
30 | SELECT e.ID, e.first_name, e.last_name, e.Department, e.Salary, p.ProjectID, p.ProjectName, p.Budget, p.Status
31 | FROM "EMPLOYEES"."EMPLOYEES_SCHEMA"."EMPLOYEES" AS e
32 | LEFT JOIN
33 | "EMPLOYEES"."EMPLOYEES_SCHEMA"."PROJECTS" AS P ON e.id = P.EMPLOYEEID;
34
35
```

Results

# ID	A FIRST_NAME	A LAST_NAME	A DEPARTMENT	# SALARY	# PROJECTID	A PROJECTNAME	# BUDGET	A STATUS
1 1	John	Doe	IT	55000	101	AI Development	100000	Completed
2 1	John	Doe	IT	55000	103	Cybersecurity Audit	75000	Pending
3 2	Jane	Smith	HR	48000	102	Employee Training	50000	Ongoing
4 3	Mike	Johnson	Finance	60000	104	Financial Analysis	90000	Ongoing
5 4	Sarah	Brown	IT	53000	null	null	null	null
6 5	David	White	Marketing	52000	105	Market Expansion	65000	Completed
7 6	Emily	Davis	IT	62000	106	Risk Management	80000	Pending
8 7	Robert	Wilson	Finance	59000	null	null	null	null

Query Details

Query duration 42ms

Rows 11

Query ID 01bc9a70-0001-0b38-0...

Show more

ID #

3. Retrieve all projects and their assigned employees, including projects that have no employees using a **RIGHT JOIN**.

- Return: ProjectID, ProjectName, Budget, Status, EmployeeID, FirstName, LastName, Department, Salary.

```

33  -----
34  -- Retrieve all projects and their assigned employees, including projects that have no employees using a RIGHT JOIN. Return: ProjectID, ProjectName, Budget,
35  Status, EmployeeID, FirstName, LastName, Department, Salary.
36
37  SELECT p.PROJECTID, p.projectname, p.budget, p.status, e.id, e.first_name, e.last_name, e.department
38  FROM "EMPLOYEES"."EMPLOYEES_SCHEMA"."EMPLOYEES" AS e
39  RIGHT JOIN
40  "EMPLOYEES"."EMPLOYEES_SCHEMA"."PROJECTS" AS P ON e.id = P.EMPLOYEEID;

```

Results **Chart**

#	PROJECTID	PROJECTNAME	BUDGET	STATUS	ID	FIRST_NAME	LAST_NAME	DEPARTMENT
1	101	AI Development	100000	Completed	1	John	Doe	IT
2	103	Cybersecurity Audit	75000	Pending	1	John	Doe	IT
3	102	Employee Training	50000	Ongoing	2	Jane	Smith	HR
4	104	Financial Analysis	90000	Ongoing	3	Mike	Johnson	Finance
5	105	Market Expansion	65000	Completed	5	David	White	Marketing
6	106	Risk Management	80000	Pending	6	Emily	Davis	IT

Query Details

- Query duration: 2.3s
- Rows: 6
- Query ID: 01bc9a85-0001-0cb3-0...

PROJECTID

4. Retrieve all employees and projects, including those without a match in either table using a **FULL OUTER JOIN**.

- Return: EmployeeID, FirstName, LastName, Department, Salary, ProjectID, ProjectName, Budget, Status.

EMPLOYEES.EMPLOYEES_SCHEMA ▾ Settings ▾

```

37 EMPLOYEE . EMPLOYEES_SCHEMA . PROJECTS AS P ON E.ID = P.EMPLOYEEID;
38 -----
39
40 --4. Retrieve all employees and projects, including those without a match in either table using a FULL OUTER JOIN. Return: EmployeeID, FirstName, LastName, Department, Salary, ProjectID, ProjectName, Budget, Status.
41
42
43 > SELECT e.ID, e.first_name, e.last_name, e.Department, e.Salary, p.ProjectID, p.ProjectName, p.Budget, p.Status
44 FROM "EMPLOYEES"."EMPLOYEES_SCHEMA"."EMPLOYEES" AS e
45 FULL OUTER JOIN
46 "EMPLOYEES"."EMPLOYEES_SCHEMA"."PROJECTS" AS P ON e.id = P.EMPLOYEEID;
47
48
49
50

```

↳ Results ▾ Chart

# ID	▲ FIRST_NAME	▲ LAST_NAME	▲ DEPARTMENT	# SALARY	# PROJECTID	▲ PROJECTNAME	# BUDGET	▲ STATUS	
1	1	John	Doe	IT	55000	101	AI Development	100000	Completed
2	1	John	Doe	IT	55000	103	Cybersecurity Audit	75000	Pending
3	2	Jane	Smith	HR	48000	102	Employee Training	50000	Ongoing
4	3	Mike	Johnson	Finance	60000	104	Financial Analysis	90000	Ongoing
5	4	Sarah	Brown	IT	53000	null	null	null	null
6	5	David	White	Marketing	52000	105	Market Expansion	65000	Completed
7	6	Emily	Davis	IT	62000	106	Risk Management	80000	Pending
8	7	Robert	Wilson	Finance	59000	null	null	null	null

Query Details

- Query duration 621ms
- Rows 11
- Query ID 01bc9a91-0001-0be2-0...

Show more ▾

ID #

2. UNION & UNION ALL

5. Retrieve a list of all unique cities where employees are located and project statuses. o Return: Location (Rename the column to Location using an alias).

```

48 -----
49
50
51 --2. UNION & UNION ALL
52 --5. Retrieve a list of all unique cities where employees are located and project statuses. o Return: Location (Rename the column to Location using an alias)
53
54 > SELECT city AS Location
55 FROM "EMPLOYEES"."EMPLOYEES_SCHEMA"."EMPLOYEES"
56 union
57 SELECT status FROM "EMPLOYEES"."EMPLOYEES_SCHEMA"."PROJECTS";
58

```

↳ Results ▾ Chart

▲ LOCATION
1 New York
2 Chicago
3 Los Angeles
4 San Francisco
5 Houston
6 Completed
7 Ongoing
8 Pending

Query Details

- Query duration 741ms
- Rows 8
- Query ID 01bc9a9e-0001-0cd5-0...

Show more ▾

LOCATION A

100% filled

6. Retrieve a list of all cities where employees are located and project statuses, allowing duplicates. o Return: Location (Rename the column to Location using an alias).

The screenshot shows a SQL query editor interface with three tabs at the top: '15 May, SQL Fundamentals', '2025-05-17 Exercise 1 - E...', and '2025-05-26 10:18am Exer...'. The main area displays a query:

```

53 | SELECT city AS Location
54 | FROM "EMPLOYEES"."EMPLOYEES_SCHEMA"."EMPLOYEES"
55 | union ALL
56 | SELECT status FROM "EMPLOYEES"."EMPLOYEES_SCHEMA"."PROJECTS";
57 |

```

The results tab shows the output:

LOCATION
San Francisco
Chicago
Houston
Los Angeles
Chicago
San Francisco
Completed
Ongoing
Pending
Ongoing
Completed
Pending

On the right, the 'Query Details' panel shows:

- Query duration: 239ms
- Rows: 16
- Query ID: 01bc9aa1-0001-0be2-0...

Below the results table, a summary table shows the count of rows for each location:

LOCATION	Count
Chicago	3
New York	2
Los Angeles	2
+ 5 more	

3. Filtering Statements

7. Retrieve employees who earn more than 70,000. o Return: EmployeeID, FirstName, LastName, Department, Salary.

NO RESULTS

The screenshot shows a SQL query editor interface with three tabs at the top: '15 May, SQL Fundamentals', '2025-05-17 Exercise 1 - E...', and '2025-05-26 10:18am Exer...'. The main area displays a query:

```

60 |
61 | --3. Filtering Statements
62 | --7. Retrieve employees who earn more than 70,000. o Return: EmployeeID, FirstName, LastName, Department, Salary.
63 |
64 | SELECT id, first_name, last_name, department, salary
65 | FROM "EMPLOYEES"."EMPLOYEES_SCHEMA"."EMPLOYEES"
66 | WHERE SALARY > 70000;
67 |
68 |

```

The results tab shows the output:

ID	FIRST_NAME	LAST_NAME	DEPARTMENT	SALARY
Query produced no results				

On the right, the 'Query Details' panel shows:

- Query duration: 28ms
- Rows: 0
- Query ID: 01bc9bdd-0001-0cb3-0...

8. Retrieve employees working in either IT or Finance departments. o Return: EmployeeID, FirstName, LastName, Department, Salary.

```

67
68
69
70 -- 8. Retrieve employees working in either IT or Finance departments. Return: EmployeeID, FirstName, LastName, Department, Salary.
71
72 | SELECT id, first_name, last_name, department, salary
73 | FROM "EMPLOYEES"."EMPLOYEES_SCHEMA"."EMPLOYEES"
74 | WHERE department IN ('IT','Finance');
75
76

```

Results

# ID	A FIRST_NAME	A LAST_NAME	A DEPARTMENT	# SALARY
1	John	Doe	IT	55000
2	Mike	Johnson	Finance	60000
3	Sarah	Brown	IT	53000
4	Emily	Davis	IT	62000
5	Robert	Wilson	Finance	59000
6	Laura	Hall	IT	50000

Query Details

- Query duration 747ms
- Rows 6
- Query ID 01bc9be1-0001-0b38-0...

9. Retrieve projects that are not yet completed. o Return: ProjectID, ProjectName, Budget, Status.

```

78
79 | SELECT PROJECTID, PROJECTNAME, BUDGET, STATUS
80 | FROM "EMPLOYEES"."EMPLOYEES_SCHEMA"."PROJECTS"
81 | WHERE status != 'Completed';
82
83

```

Results

# PROJECTID	A PROJECTNAME	# BUDGET	A STATUS
1	Employee Training	50000	Ongoing
2	Cybersecurity Audit	75000	Pending
3	Financial Analysis	90000	Ongoing
4	Risk Management	80000	Pending

Query Details

- Query duration 226ms
- Rows 4
- Query ID 01bc9be4-0001-0cb3-0...

10. Retrieve projects that have a budget greater than 70,000 and are not completed. •
Return: ProjectID, ProjectName, Budget, Status.

```

82
83
84
85 --10. Retrieve projects that have a budget greater than 70,000 and are not completed. • Return: ProjectID, ProjectName, Budget, Status.
86
87 | SELECT PROJECTID, PROJECTNAME, BUDGET, STATUS
88 | FROM "EMPLOYEES"."EMPLOYEES_SCHEMA"."PROJECTS"
89 | WHERE status != 'Completed' AND BUDGET > 70000;
90
91
92
93

```

Results

# PROJECTID	A PROJECTNAME	# BUDGET	A STATUS
1	Cybersecurity Audit	75000	Pending
2	Financial Analysis	90000	Ongoing
3	Risk Management	80000	Pending

Query Details

- Query duration 311ms
- Rows 3
- Query ID 01bc9be8-0001-0ce9-0...

11. Retrieve employees from New York OR Toronto, ordered by salary in descending order.

- Return: EmployeeID, FirstName, LastName, Department, Salary, City.

```
92  --11. Retrieve employees from New York OR Toronto, ordered by salary in descending order. • Return: EmployeeID, FirstName, LastName, Department, Salary, City.
93
94  | SELECT id, first_name, last_name, department, salary, CITY
95  | FROM "EMPLOYEES"."EMPLOYEES_SCHEMA"."EMPLOYEES"
96  | WHERE city IN ('New York','Toronto')
97  | ORDER BY salary DESC;
98
```

Results Chart Q ↻ ⌂ ⌄ ⌁

#	ID	FIRST_NAME	LAST_NAME	DEPARTMENT	SALARY	CITY	Query Details	...
1	1	John	Doe	IT	55000	New York	Query duration	24ms
2	4	Sarah	Brown	IT	53000	New York	Rows	2

Show more ▾

Query ID 01bc9beb-0001-0cb3-0...

12. Retrieve the top 3 highest-paid employees. • Return: EmployeeID, FirstName, LastName, Department, Salary.

```
101
102  | SELECT id, first_name, last_name, department, salary
103  | FROM "EMPLOYEES"."EMPLOYEES_SCHEMA"."EMPLOYEES"
104  | ORDER BY salary DESC
105  | LIMIT 3;
106
107
```

Results Chart Q ↻ ⌂ ⌄ ⌁

#	ID	FIRST_NAME	LAST_NAME	DEPARTMENT	SALARY	Query Details	...
1	6	Emily	Davis	IT	62000	Query duration	18ms
2	3	Mike	Johnson	Finance	60000	Rows	3
3	7	Robert	Wilson	Finance	59000	Query ID	01bc9bed-0001-0ce9-0...

Show more ▾

4. Aggregate Functions with GROUP BY & HAVING

13. Find the total salary per department, sorted in descending order. • Return: Department, TotalSalary (Rename SUM(Salary) as TotalSalary).

```

108 -- 14. Aggregate Functions with GROUP BY & HAVING
109 -- 13. Find the total salary per department, sorted in descending order. • Return: Department, TotalSalary (Rename SUM(Salary) as TotalSalary).
110
111
112 | SELECT DEPARTMENT, SUM(SALARY) AS TotalSalary
113 | FROM "EMPLOYEES"."EMPLOYEES_SCHEMA"."EMPLOYEES"
114 | GROUP BY DEPARTMENT;
115

```

↳ Results ↵ Chart

The screenshot shows a table with two columns: 'DEPARTMENT' and 'TOTALSALARY'. The data rows are:

DEPARTMENT	TOTALSALARY
IT	220000
HR	99000
Finance	119000
Marketing	105000

Query Details: Query duration 169ms, Rows 4, Query ID 01bc9bfe-0001-0ce9-0...

14. Find the average salary per city, but only include cities where the average salary is greater than 65,000. • Return: City, AverageSalary (Rename AVG(Salary) as AverageSalary).

```

116 -----
117
118 -- 14. Find the average salary per city, but only include cities where the average salary is greater than 65,000. • Return: City, AverageSalary (Rename
119 | AVG(Salary) as AverageSalary).
120 | SELECT CITY, AVG(SALARY) AS AverageSalary
121 | FROM "EMPLOYEES"."EMPLOYEES_SCHEMA"."EMPLOYEES"
122 | GROUP BY CITY
123 | HAVING AverageSalary > 65000;
124

```

↳ Results ↵ Chart

The screenshot shows a table with two columns: 'CITY' and 'AVERAGESALARY'. The message in the body says 'Query produced no results'.

CITY	AVERAGESALARY
Query produced no results	

Query Details: Query duration 51ms, Rows 0, Query ID 01bc9bfe-0001-0ce9-0...

15. Count the number of employees per department, including only departments with more than 1 employee. • Return: Department, EmployeeCount (Rename COUNT(EmployeeID) as EmployeeCount).

```

125 -----
126 -- 15. Count the number of employees per department, including only departments with more than 1 employee. • Return: Department, EmployeeCount (Rename COUNT(EmployeeID) as EmployeeCount).
127
128 | SELECT Department, COUNT(ID) AS EmployeeCount
129 | FROM "EMPLOYEES"."EMPLOYEES_SCHEMA"."EMPLOYEES"
130 | GROUP BY DEPARTMENT
131 | HAVING EmployeeCount > 1;
132
133
134

```

↳ Results ↵ Chart

DEPARTMENT	EMPLOYEECOUNT
IT	4
HR	2
Finance	2
Marketing	2

Query Details ...
 Query duration 266ms
 Rows 4
 Query ID 01bc9c01-0001-0b38-0...
 Show more ▾

16. Retrieve the number of projects per status, but only include statuses with at least 2 projects. • Return: Status, ProjectCount (Rename COUNT(ProjectID) as ProjectCount).

```

133 -----
134 --16. Retrieve the number of projects per status, but only include statuses with at least 2 projects. • Return: Status, ProjectCount (Rename COUNT(ProjectID) as ProjectCount).
135
136 | SELECT STATUS, COUNT(PROJECTID) AS ProjectCount
137 | FROM "EMPLOYEES"."EMPLOYEES_SCHEMA"."PROJECTS"
138 | GROUP BY STATUS
139 | HAVING ProjectCount >= 2;
140
141

```

↳ Results ↵ Chart

STATUS	PROJECTCOUNT
Completed	2
Ongoing	2
Pending	2

Query Details ...
 Query duration 384ms
 Rows 3
 Query ID 01bc9c06-0001-0d09-0...
 Show more ▾

17. Retrieve the total project budget per employee, but only for employees who are managing projects worth more than 150,000. • Return: EmployeeID, FirstName, LastName, TotalProjectBudget (Rename SUM(Budget) as TotalProjectBudget).

```
140  
141 -----  
142 -- 17. Retrieve the total project budget per employee, but only for employees who are managing projects worth more than 150,000. · Return: EmployeeID,  
143 FirstName, LastName, TotalProjectBudget (Rename SUM(Budget) as TotalProjectBudget).  
144  
145 SELECT e.ID, e.first_name, e.last_name, SUM(p.Budget) AS TotalProjectBudget  
146 FROM "EMPLOYEES"."EMPLOYEES_SCHEMA"."EMPLOYEES" AS e  
147 inner join "EMPLOYEES"."EMPLOYEES_SCHEMA"."PROJECTS" as p  
148 ON e.ID = p.EMPLOYEEID  
149 group by e.ID, e.first_name, e.last_name  
150 HAVING TotalProjectBudget > 150000;  
151  
152
```

↳ Results ⚡ Chart

#	ID	FIRST_NAME	LAST_NAME	TOTALPROJECTBUDGET
1	1	John	Doe	175000

Query Details ...
Query duration 91ms
Rows 1
Query ID: 2021-03-06-2001-21-0-0