

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
16 -- SQL JOINS
17
18 -- 1. Retrieve all employees and their assigned projects using an INNER JOIN.Return: EmployeeID, FirstName, LastName, Department, Salary,ProjectID, ProjectName,
    Budget, Status.
19
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 Chart

	# ID	▲ FIRST_NAME	▲ LAST_NAME	▲ DEPARTMENT	# SALARY	# PROJECTID	▲ PROJECTNAME	# BUDGET	▲ STATUS	Query Details
1	1	John	Doe	IT	55000	101	AI Development	100000	Completed	Query duration 138ms
2	1	John	Doe	IT	55000	103	Cybersecurity Audit	75000	Pending	Rows 6
3	2	Jane	Smith	HR	48000	102	Employee Training	50000	Ongoing	Query ID 01bc9a59-0001-0cd5-0...
4	3	Mike	Johnson	Finance	60000	104	Financial Analysis	90000	Ongoing	Show more
5	5	David	White	Marketing	52000	105	Market Expansion	65000	Completed	
6	6	Emily	Davis	IT	62000	106	Risk Management	80000	Pending	

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

Results Chart

	# ID	▲ FIRST_NAME	▲ LAST_NAME	▲ DEPARTMENT	# SALARY	# PROJECTID	▲ PROJECTNAME	# BUDGET	▲ STATUS	Query Details
1	1	John	Doe	IT	55000	101	AI Development	100000	Completed	Query duration 42ms
2	1	John	Doe	IT	55000	103	Cybersecurity Audit	75000	Pending	Rows 11
3	2	Jane	Smith	HR	48000	102	Employee Training	50000	Ongoing	Query ID 01bc9a70-0001-0b38-0...
4	3	Mike	Johnson	Finance	60000	104	Financial Analysis	90000	Ongoing	Show more
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	

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...

Show more

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 Open in Workspaces

```

37  EMPLOYEES : EMPLOYEES_SCHEMA : PROJECTS AS P ON e.id = P.EMPLOYEEID,
40  -----
41
42  --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.
43
44  SELECT e.ID, e.first_name, e.Last_name, e.Department, e.Salary, p.ProjectID, p.ProjectName, p.Budget, p.Status
45  FROM "EMPLOYEES"."EMPLOYEES_SCHEMA"."EMPLOYEES" AS e
46  FULL OUTER JOIN
47  "EMPLOYEES"."EMPLOYEES_SCHEMA"."PROJECTS" AS P ON e.id = P.EMPLOYEEID;
48
49
50

```

Results Chart

	# 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 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  --2. UNION & UNION ALL
51
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

	A 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 IDE interface with a query editor and a results panel. The query editor contains the following SQL code:

```

53
54 SELECT city AS Location
55 FROM "EMPLOYEES"."EMPLOYEES_SCHEMA"."EMPLOYEES"
56 union ALL
57 SELECT status FROM "EMPLOYEES"."EMPLOYEES_SCHEMA"."PROJECTS";
58

```

The results panel displays a table with two columns: LOCATION and STATUS. The data is as follows:

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

The right sidebar shows query details: Query duration 239ms, Rows 16, Query ID 01bc9aa1-0001-0be2-0...

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 IDE interface with a query editor and a results panel. The query editor contains the following SQL code:

```

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 panel displays a table with columns: ID, FIRST_NAME, LAST_NAME, DEPARTMENT, SALARY. The message "Query produced no results" is shown in the center of the table.

The right sidebar shows query details: 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 Chart

#	ID	FIRST_NAME	LAST_NAME	DEPARTMENT	SALARY
1	1	John	Doe	IT	55000
2	3	Mike	Johnson	Finance	60000
3	4	Sarah	Brown	IT	53000
4	6	Emily	Davis	IT	62000
5	7	Robert	Wilson	Finance	59000
6	10	James	Hall	IT	50000

Query Details

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

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 Chart

#	PROJECTID	PROJECTNAME	BUDGET	STATUS
1	102	Employee Training	50000	Ongoing
2	103	Cybersecurity Audit	75000	Pending
3	104	Financial Analysis	90000	Ongoing
4	106	Risk Management	80000	Pending

Query Details

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

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 Chart

#	PROJECTID	PROJECTNAME	BUDGET	STATUS
1	103	Cybersecurity Audit	75000	Pending
2	104	Financial Analysis	90000	Ongoing
3	106	Risk Management	80000	Pending

Query Details

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

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
```

	# 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

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

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
```

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

```

	DEPARTMENT	TOTALSALARY
1	IT	220000
2	HR	99000
3	Finance	119000
4	Marketing	105000

Query Details

- Query duration: 169ms
- Rows: 4
- Query ID: 01bc9bf9-0001-0ce9-0...
- Show more

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
121 SELECT CITY, AVG(SALARY) AS AverageSalary
122 FROM "EMPLOYEES"."EMPLOYEES_SCHEMA"."EMPLOYEES"
123 GROUP BY CITY
124 HAVING AverageSalary > 65000;

```

CITY	AVERAGESALARY
Query produced no results	

Query Details

- Query duration: 51ms
- Rows: 0
- Query ID: 01bc9bfe-0001-0ce9-0...
- Show more

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		Query Details
1	IT	4	4	Query duration 266ms
2	HR	2	2	Rows 4
3	Finance	2	2	Query ID 01bc9c01-0001-0b38-0...
4	Marketing	2	2	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		Query Details
1	Completed	2	2	Query duration 384ms
2	Ongoing	2	2	Rows 3
3	Pending	2	2	Query ID 01bc9c06-0001-0d09-0...

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

Results Chart

Search Filter Download Copy Refresh

#	ID	FIRST_NAME	LAST_NAME	TOTALPROJECTBUDGET
1	1	John	Doe	175000

Query Details

Query duration 91ms

Rows 1