

ASSIGNMENT

SQL JOINS 2

```
CREATE TABLE departments (
```

```
    dept_id INT PRIMARY KEY,
```

```
    dept_name VARCHAR(50),
```

```
    dept_head VARCHAR(50)
```

```
);
```

```
CREATE TABLE courses (
```

```
    course_id INT PRIMARY KEY,
```

```
    course_name VARCHAR(50),
```

```
    credits INT,
```

```
    dept_id INT,
```

```
    FOREIGN KEY (dept_id) REFERENCES departments(dept_id)
```

```
);
```

```
CREATE TABLE students (
```

```
    student_id INT PRIMARY KEY,
```

```
    student_name VARCHAR(50),
```



```
    course_id INT,
```

FOREIGN KEY (course_id) REFERENCES courses(course_id)

);

1. Show each student's first name and their course name using an INNER JOIN.

```
45 • SELECT s.student_name, c.course_name
46 FROM students s
47 INNER JOIN courses c ON s.course_id = c.course_id;
48
49
```

Result Grid |  Filter Rows: | Export:  | Wrap Cell Content: 

	student_name	course_name
▶	Alice	Biology
	Ian	Biology
	David	History
	Bob	Computer Science
	Grace	Computer Science
	Eva	Quantum Mechanics
	Jack	Quantum Mechanics
	Helen	Philosophy

2. Display all students and the department of their course using an INNER JOIN.

```
50 • SELECT s.student_name, d.dept_name
51 FROM students s
52 INNER JOIN courses c ON s.course_id = c.course_id
53 INNER JOIN departments d ON c.dept_id = d.dept_id;
54 • SELECT c.course_name, s.student_name
```

Result Grid |  Filter Rows: | Export:  | Wrap Cell Content: 

	student_name	dept_name
▶	Alice	Science
	Ian	Science
	David	Arts
	Helen	Arts
	Bob	Technology
	Grace	Technology
	Eva	Physics
	Jack	Physics

3. Show course names and the students enrolled in them (only matching rows).

```
50 • SELECT s.student_name, d.dept_name
51 FROM students s
52 INNER JOIN courses c ON s.course_id = c.course_id
53 INNER JOIN departments d ON c.dept_id = d.dept_id;
54 • SELECT c.course_name, s.student_name
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	student_name	dept_name
▶	Alice	Science
	Ian	Science
	David	Arts
	Helen	Arts
	Bob	Technology
	Grace	Technology
	Eva	Physics
	Jack	Physics

4. List all students whose course credits are greater than 3.

```
57 • SELECT s.student_name
58 FROM students s
59 INNER JOIN courses c ON s.course_id = c.course_id
60 WHERE c.credits > 3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	student_name
▶	Alice
	Ian
	Bob
	Grace
	Eva
	Jack

5. Find students who are in the 'Science' department.

```

61 • SELECT s.student_name
62 FROM students s
63 INNER JOIN courses c ON s.course_id = c.course_id
64 INNER JOIN departments d ON c.dept_id = d.dept_id
65 WHERE d.dept_name = 'Science';

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	student_name			
▶	Alice			
	Ian			

6. Show all students and their course names, including students without a course.

```

64 • SELECT s.student_name, c.course_name
65 FROM students s
66 LEFT JOIN courses c ON s.course_id = c.course_id;
67

```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	student_name	course_name			
▶	Alice	Biology			
	Bob	Computer Science			
	Charlie	NULL			
	David	History			
	Eva	Quantum Mechanics			
	Frank	NULL			
	Grace	Computer Science			
	Helen	Philosophy			
	Ian	Biology			
	Jack	Quantum Mechanics			

7. Show all courses and the students enrolled, including courses without any students.

```
67 • SELECT c.course_name, s.student_name
68 FROM courses c
69 LEFT JOIN students s ON c.course_id = s.course_id;
70
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
course_name	student_name		
Biology	Alice		
Biology	Ian		
History	David		
Computer Science	Bob		
Computer Science	Grace		
Quantum Mechanics	Eva		
Quantum Mechanics	Jack		
Philosophy	Helen		

8. List students who have not been assigned a course yet.

```
70 • SELECT student_name
71 FROM students
72 WHERE course_id IS NULL;
73
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
student_name			
Charlie			
Frank			

9. Show all courses along with the number of students enrolled (0 if none).

```
73 • SELECT c.course_name, COUNT(s.student_id) AS student_count
74 FROM courses c
75 LEFT JOIN students s ON c.course_id = s.course_id
76 GROUP BY c.course_name;
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:

	course_name	student_count
►	Biology	2
	History	1
	Computer Science	2
	Quantum Mechanics	2
	Philosophy	1

10. Display all students with their course department (NULL if not assigned).

```

77 • SELECT s.student_name, d.dept_name
78 FROM students s
79 LEFT JOIN courses c ON s.course_id = c.course_id
80 LEFT JOIN departments d ON c.dept_id = d.dept_id;

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
student_name	dept_name			
▶ Alice	Science			
Bob	Technology			
Charlie	NULL			
David	Arts			
Eva	Physics			
Frank	NULL			
Grace	Technology			
Helen	Arts			
Ian	Science			
Jack	Physics			

11. Display all course names with their student names, even if no students are enrolled.

```

81 • SELECT c.course_name, s.student_name
82 FROM courses c
83 LEFT JOIN students s ON c.course_id = s.course_id;
84

```





Result Grid		Filter Rows:	Export:	Wrap Cell Content:
course_name	student_name			
▶ Biology	Alice			
Biology	Ian			
History	David			
Computer Science	Bob			
Computer Science	Grace			
Quantum Mechanics	Eva			
Quantum Mechanics	Jack			
Philosophy	Helen			

12. Show all departments and the students studying in them, using RIGHT JOIN.

```

81 • SELECT d.dept_name, s.student_name
82 FROM students s
83 RIGHT JOIN courses c ON s.course_id = c.course_id
84 RIGHT JOIN departments d ON c.dept_id = d.dept_id;

```

Result Grid |   Filter Rows: | Export:  | Wrap Cell Content: 

	dept_name	student_name
▶	Science	Alice
	Science	Ian
	Arts	David
	Arts	Helen
	Technology	Bob
	Technology	Grace
	Physics	Eva
	Physics	Jack

13. Find courses that do not have any students enrolled.

```

85 • SELECT c.course_name
86 FROM courses c
87 LEFT JOIN students s ON c.course_id = s.course_id
88 WHERE s.student_id IS NULL;

```

Result Grid |   Filter Rows: | Export:  | Wrap Cell Content: 

14. Show all course names and credits, even if they don't have any students.

```

89 • SELECT c.course_name, c.credits
90 FROM courses c
91 LEFT JOIN students s ON c.course_id = s.course_id;

```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	course_name	credits
▶	Biology	4
	Biology	4
	History	3
	Computer Science	5
	Computer Science	5
	Quantum Mechanics	4
	Quantum Mechanics	4
	Philosophy	2

15. List all departments and match them with students via course.


```

92 • SELECT d.dept_name, s.student_name
93 FROM departments d
94 JOIN courses c ON d.dept_id = c.dept_id
95 JOIN students s ON c.course_id = s.course_id;

```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	dept_name	student_name
▶	Science	Alice
	Science	Ian
	Arts	David
	Arts	Helen
	Technology	Bob
	Technology	Grace
	Physics	Eva
	Physics	Jack

16. Display all students and all courses, showing NULL where data is missing.

```

96 • SELECT s.student_name, c.course_name
97 FROM students s
98 LEFT JOIN courses c ON s.course_id = c.course_id
99 UNION
100 SELECT s.student_name, c.course_name
101 FROM courses c
102 LEFT JOIN students s ON c.course_id = s.course_id;

```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:


	student_name	course_name
▶	Alice	Biology
	Bob	Computer Science
	Charlie	NULL
	David	History
	Eva	Quantum Mechanics
	Frank	NULL
	Grace	Computer Science
	Helen	Philosophy
	Ian	Biology
	Jack	Quantum Mechanics

17. List all student names and all course names, even if they don't match.

```

103 • SELECT s.student_name, c.course_name
104 FROM students s
105 CROSS JOIN courses c;

```

Result Grid |  Filter Rows: | Export:  | Wrap Cell Content: 

	student_name	course_name
▶	Alice	Philosophy
	Alice	Quantum Mechanics
	Alice	Computer Science
	Alice	History
	Alice	Biology
	Bob	Philosophy
	Bob	Quantum Mechanics
	Bob	Computer Science
	Bob	History
	Bob	Biology
	Charlie	Philosophy
	Charlie	Quantum Mechanics
	Charlie	Computer Science
	Charlie	History
	Charlie	Biology

18. Show student names, course names, and department heads.

```

106 • SELECT s.student_name, c.course_name, d.dept_head
107 FROM students s
108 JOIN courses c ON s.course_id = c.course_id
109 JOIN departments d ON c.dept_id = d.dept_id;

```

Result Grid |  Filter Rows: | Export:  | Wrap Cell Content: 

	student_name	course_name	dept_head
▶	Alice	Biology	Dr. Rao
	Ian	Biology	Dr. Rao
	David	History	Dr. Mehta
	Helen	Philosophy	Dr. Mehta
	Bob	Computer Science	Dr. Singh
	Grace	Computer Science	Dr. Singh
	Eva	Quantum Mechanics	Dr. Sharma
	Jack	Quantum Mechanics	Dr. Sharma

19. Find students who are in courses from the 'Technology' department.

```
110 • SELECT s.student_name
111 FROM students s
112 JOIN courses c ON s.course_id = c.course_id
113 JOIN departments d ON c.dept_id = d.dept_id
114 WHERE d.dept_name = 'Technology';
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	student_name			
▶	Bob			
	Grace			

20. Show department names and total students in each department.

```
115 • SELECT d.dept_name, COUNT(s.student_id) AS total_students
116 FROM departments d
117 LEFT JOIN courses c ON d.dept_id = c.dept_id
118 LEFT JOIN students s ON c.course_id = s.course_id
119 GROUP BY d.dept_name;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	dept_name	total_students		
▶	Science	2		
	Arts	2		
	Technology	2		
	Physics	2		

21. Display student names, their department, and credits for their course.

```

120 • SELECT s.student_name, d.dept_name, c.credits
121 FROM students s
122 JOIN courses c ON s.course_id = c.course_id
123 JOIN departments d ON c.dept_id = d.dept_id;

```

Result Grid |  Filter Rows: | Export:  | Wrap Cell Content: 

	student_name	dept_name	credits
▶	Alice	Science	4
	Ian	Science	4
	David	Arts	3
	Helen	Arts	2
	Bob	Technology	5
	Grace	Technology	5
	Eva	Physics	4
	Jack	Physics	4

22. Show all departments with courses and students (include those with no students).

```

124 • SELECT d.dept_name, c.course_name, s.student_name
125 FROM departments d
126 LEFT JOIN courses c ON d.dept_id = c.dept_id
127 LEFT JOIN students s ON c.course_id = s.course_id;



```

Result Grid |  Filter Rows: | Export:  | Wrap Cell Content: 

	dept_name	course_name	student_name
	Science	Biology	Alice
	Science	Biology	Ian
	Arts	History	David
	Arts	Philosophy	Helen
	Technology	Computer Science	Bob
	Technology	Computer Science	Grace
	Physics	Quantum Mechanics	Eva
	Physics	Quantum Mechanics	Jack

23. Find pairs of students studying the same course.




```
128 • SELECT s1.student_name AS student1, s2.student_name AS student2
129 FROM students s1
130 JOIN students s2 ON s1.course_id = s2.course_id AND s1.student_id < s2.student_id;
```

Result Grid |  Filter Rows: | Export:  | Wrap Cell Content: 

	student1	student2
▶	Alice	Ian
	Bob	Grace
	Eva	Jack

24. Show all possible student pairs where both are in the same department.


```
131 • SELECT s1.student_name AS student1, s2.student_name AS student2
132 FROM students s1
133 JOIN courses c1 ON s1.course_id = c1.course_id
134 JOIN departments d1 ON c1.dept_id = d1.dept_id
135 JOIN students s2 ON s2.course_id = c1.course_id AND s1.student_id < s2.student_id;
```

Result Grid |  Filter Rows: | Export:  | Wrap Cell Content: 

	student1	student2
▶	Alice	Ian
	Bob	Grace
	Eva	Jack

25. List all students who share a course with 'Alice'.

```
136 • SELECT s2.student_name
137 FROM students s1
138 JOIN students s2 ON s1.course_id = s2.course_id
139 WHERE s1.student_name = 'Alice' AND s2.student_name != 'Alice';
```

Result Grid |  Filter Rows: | Export:  | Wrap Cell Content: 



	student_name
▶	Ian



26. Find the total number of students in each course.

```

140 • SELECT c.course_name, COUNT(s.student_id) AS total_students
141 FROM courses c
142 LEFT JOIN students s ON c.course_id = s.course_id
143 GROUP BY c.course_name;

```

Result Grid   Filter Rows:

Export:  Wrap Cell Content: 

	course_name	total_students
▶	Biology	2
	History	1
	Computer Science	2
	Quantum Mechanics	2
	Philosophy	1

27. Find the average credits for courses that have students enrolled.

```

144 • SELECT AVG(c.credits) AS avg_credits
145 FROM courses c
146 JOIN students s ON c.course_id = s.course_id;

```

Result Grid




Filter Rows:

Export:



Wrap Cell Content:



	avg_credits
▶	3.8750

28. Show the department with the highest number of enrolled students.

```

147 • SELECT d.dept_name, COUNT(s.student_id) AS total_students
148 FROM departments d
149 JOIN courses c ON d.dept_id = c.dept_id
150 JOIN students s ON c.course_id = s.course_id
151 GROUP BY d.dept_name
152 ORDER BY total_students DESC
153 LIMIT 1;

```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

Fetch rows:

	dept_name	total_students
▶	Science	2

29. Display students who are in courses with credits above the average credits.

```

154 • SELECT s.student_name
155 FROM students s
156 JOIN courses c ON s.course_id = c.course_id
157 WHERE c.credits > (
158     SELECT AVG(credits) FROM courses
159 );

```

Result Grid |  Filter Rows: | Export:  | Wrap Cell Content: 

	student_name
▶	Alice
	Ian
	Bob
	Grace
	Eva
	Jack

30. Show students whose course is in the same department as 'Physics'.

```

160 • SELECT s.student_name
161 FROM students s
162 JOIN courses c ON s.course_id = c.course_id
163 JOIN departments d ON c.dept_id = d.dept_id
164 WHERE d.dept_id = (
165     SELECT dept_id FROM departments WHERE dept_name = 'Physics'
166 );

```

Result Grid |  Filter Rows: | Export:  | Wrap Cell Content: 

	student_name
▶	Eva
	Jack