## **ASSIGNMENT**

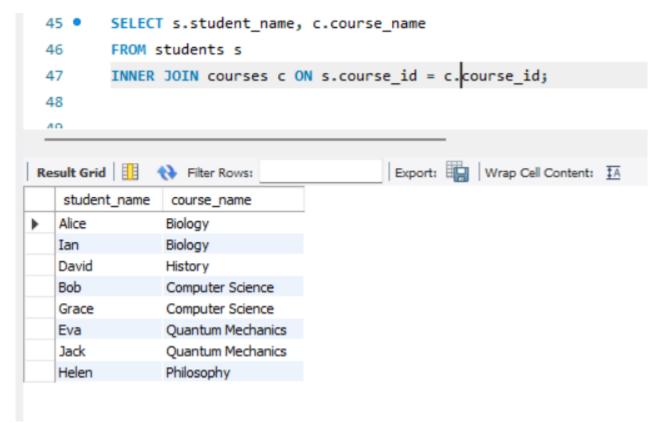
## SQL JOINS 2

```
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,
```

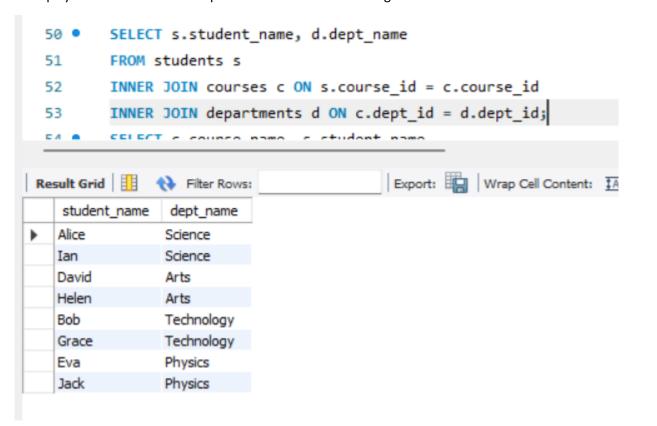
## FOREIGN KEY (course\_id) REFERENCES courses(course\_id)

);

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



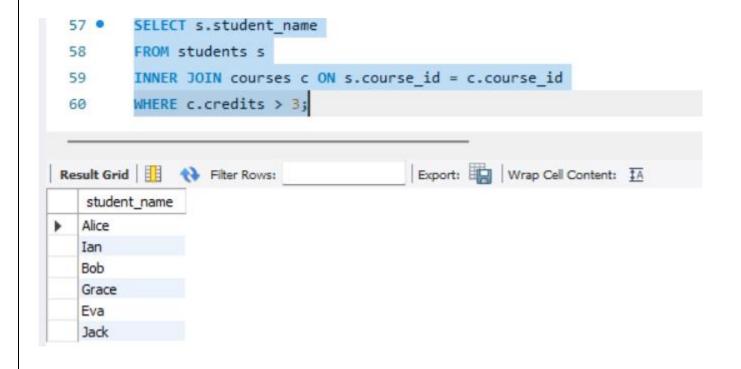
2. Display all students and the department of their course using an INNER JOIN.



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

```
50 •
         SELECT s.student name, d.dept name
         FROM students s
 51
         INNER JOIN courses c ON s.course id = c.course id
 52
         INNER JOIN departments d ON c.dept id = d.dept id;
 53
         CELECT & COURCE NAME & chudent name
                                           Export: Wrap Cell Content: IA
Result Grid
              Filter Rows:
   student_name
                dept_name
  Alice
                Science
  Ian
                Science
  David
                Arts
  Helen
                Arts
  Bob
                Technology
                Technology
  Grace
  Eva
                Physics
                Physics
  Jack
```

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

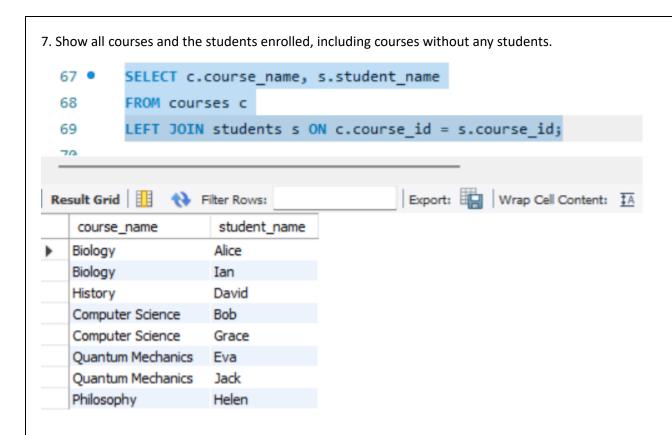


5. Find students who are in the 'Science' department.

```
SELECT s.student name
 61 •
         FROM students s
 62
        INNER JOIN courses c ON s.course_id = c.course_id
 63
        INNER JOIN departments d ON c.dept_id = d.dept_id
 64
        WHERE d.dept name = 'Science';
 65
 cc
                                          Export: Wrap Cell Content: TA
Result Grid
             Filter Rows:
   student_name
  Alice
  Ian
```

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

```
SELECT s.student_name, c.course_name
 64
         FROM students s
 65
         LEFT JOIN courses c ON s.course_id = c.course_id;
 66
 67
                                          Export: Wrap Cell Content: TA
student_name
                course_name
   Alice
               Biology
  Bob
               Computer Science
  Charlie
  David
               History
  Eva
               Quantum Mechanics
  Frank
  Grace
               Computer Science
  Helen
               Philosophy
  Ian
               Biology
               Quantum Mechanics
  Jack
```



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

```
FROM students

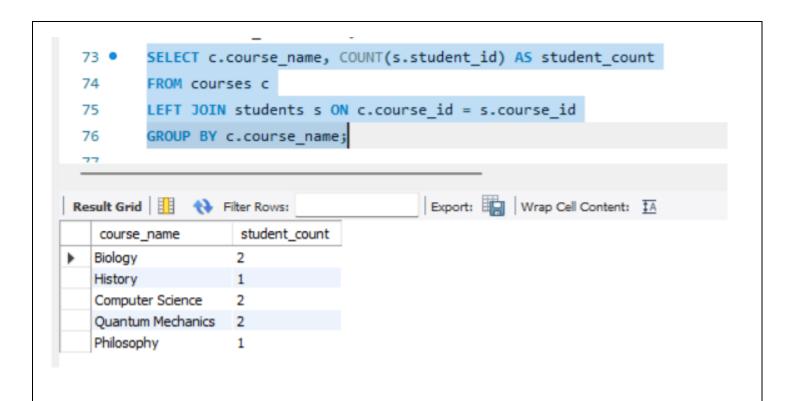
WHERE course_id IS NULL;

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

student_name

Charlie
Frank
```

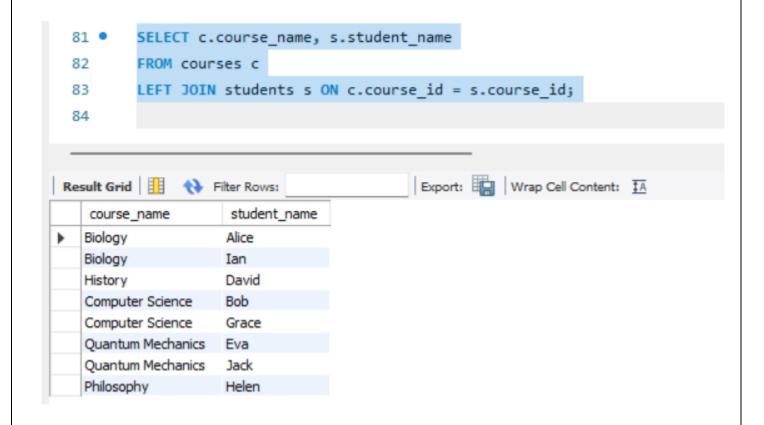
9. Show all courses along with the number of students enrolled (0 if none).



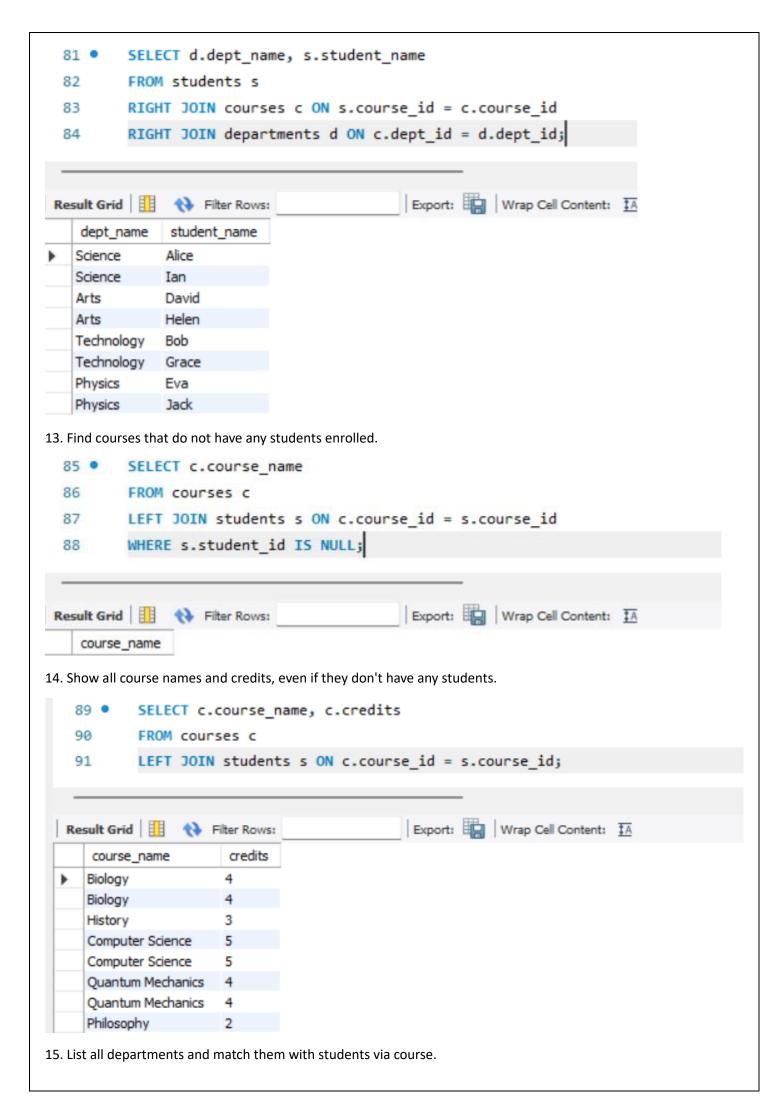
10. Display all students with their course department (NULL if not assigned).

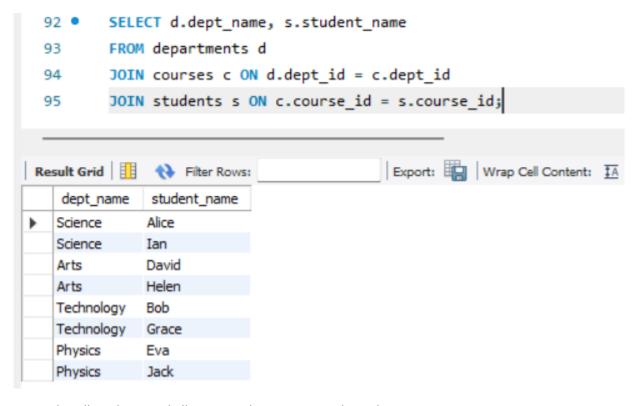
```
SELECT s.student name, d.dept name
 77 •
 78
         FROM students s
         LEFT JOIN courses c ON s.course id = c.course id
 79
         LEFT JOIN departments d ON c.dept id = d.dept id;
 80
                                              Export: Wrap Cell Content: TA
Result Grid
               Filter Rows:
   student_name
                 dept_name
  Alice
                Science
  Bob
                Technology
                NULL
  Charlie
  David
                Arts
  Eva
                Physics
                NULL
  Frank
  Grace
                Technology
  Helen
                Arts
  Ian
                 Science
  Jack
                Physics
```

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

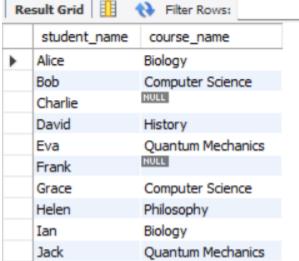


12. Show all departments and the students studying in them, using RIGHT JOIN.

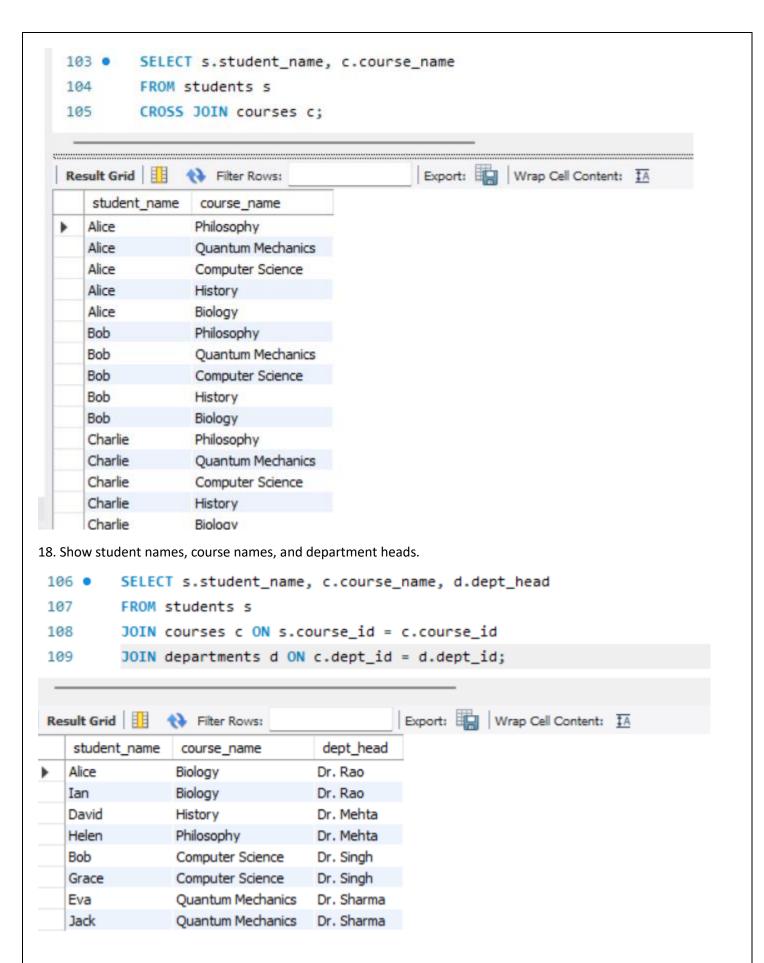




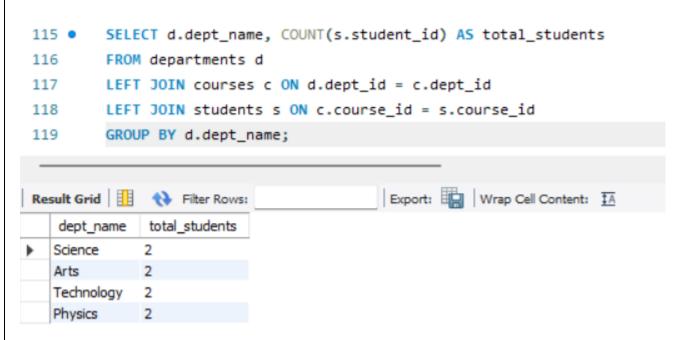
16. Display all students and all courses, showing NULL where data is missing.



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



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 JOIN departments d ON c.dept\_id = d.dept\_id 113 114 WHERE d.dept\_name = 'Technology'; Export: Wrap Cell Content: IA student\_name Bob Grace 20. Show department names and total students in each department.



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

```
SELECT s.student_name, d.dept_name, c.credits
120 •
121
        FROM students s
122
        JOIN courses c ON s.course_id = c.course_id
        JOIN departments d ON c.dept_id = d.dept_id;
123
                                        Export: Wrap Cell Content: TA
student_name | dept_name
                          credits
  Alice
               Science
                         4
                         4
  Ian
               Science
  David
               Arts
  Helen
               Arts
                         2
  Bob
               Technology
                         5
  Grace
               Technology
                         5
  Eva
               Physics
  Jack
               Physics
```

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

```
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;

ult Grid	N Filter Rows:		Export:	Wrap Cell Conten	t: <u>‡A</u>
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			

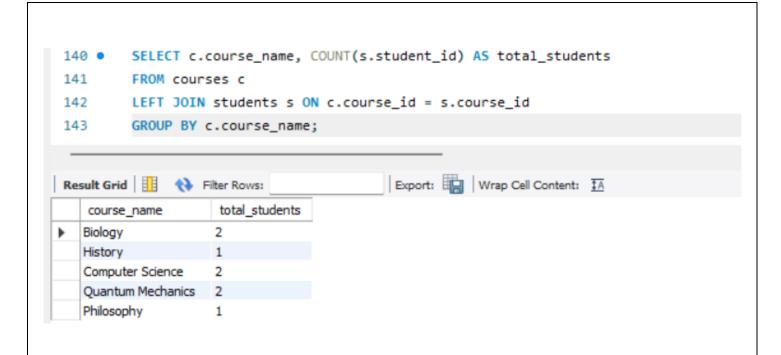
23. Find pairs of students studying the same course.

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
        JOIN students s2 ON s2.course_id = c1.course_id AND s1.student_id < s2.student_id;</pre>
135
                                      Export: Wrap Cell Content: IA
student1 student2
  Alice
          Ian
  Bob
          Grace
  Eva
          Jack
```

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

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



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

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

```
147
           SELECT d.dept_name, COUNT(s.student_id) AS total_students
           FROM departments d
 148
           JOIN courses c ON d.dept_id = c.dept_id
           JOIN students s ON c.course_id * s.course_id
 150
 151
           GROUP BY d.dept name
           ORDER BY total_students DESC
 152
           LIMIT 1;
 153
                                                      Export: Wrap Cell Content: TA Fetch rows:
dept_name total_students
   Science
                 2
```

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

