**SQL ASSIGNMENT-5**

**QUESTIONS:**

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).

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

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

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

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

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

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).

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.

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

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

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

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.

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

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

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

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

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

23. Find pairs of students studying the same course.

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

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.

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

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

**ANSWER:**

create database students;

use students;

create table courses (course\_id int primary key,course\_name varchar(50),department varchar(50),credits int, dept\_head varchar(50));

create table students (student\_id int primary key,first\_name varchar(50),course\_id int,foreign key (course\_id) references courses(course\_id));

insert into courses (course\_id, course\_name, department, credits, dept\_head) values(1, 'Math', 'Science', 4, 'Dr. Smith'),(2, 'Physics', 'Science', 3, 'Dr. Johnson'),(3, 'Computer Science', 'Technology', 5, 'Dr. Lee'),(4, 'History', 'Arts', 2, 'Dr. Brown'),(5, 'Electronics', 'Technology', 4, 'Dr. Green');

insert into students (student\_id, first\_name, course\_id) values(1, 'Alice', 1),(2, 'Bob', 3),(3, 'Charlie', NULL),(4, 'David', 2),(5, 'Eve', 5),(6, 'Frank', 3),(7, 'Grace', NULL),(8, 'Hank', 4),(9, 'Ivy', 1);

select \*from students;

select\* from courses;

alter table students add (last\_name varchar(100));

update students set last\_name = case student\_id when 1 then 'Johnson' when 2 then 'Smith' when 3 then 'Brown' when 4 then 'Wilson' when 5 then 'Davis' when 6 then 'Miller' when 7 then 'Taylor' when 8 then 'Anderson' when 9 then 'Clark' end;

1. select s.first\_name,c.course\_name from students s inner join courses c on s.course\_id = c.course\_id;
2. select s.first\_name, c.department from students s inner join courses c on s.course\_id = c.course\_id;
3. select c.course\_name, s.first\_name from courses c inner join students s on c.course\_id = s.course\_id;
4. select s.first\_name from students s inner join courses c on s.course\_id = c.course\_id where c.credits > 3;
5. select s.first\_name from students s inner join courses c on s.course\_id = c.course\_id where c.department = 'Science';
6. select s.first\_name, c.course\_name from students s left join courses c on s.course\_id = c.course\_id;
7. select c.course\_name, s.first\_name from courses c left join students s on c.course\_id = s.course\_id;
8. select first\_name from students where course\_id is null;
9. select c.course\_name, count(s.student\_id) as total\_students from courses c left join students s on c.course\_id = s.course\_id group by c.course\_name;
10. select s.first\_name, c.department from students s left join courses c on s.course\_id = c.course\_id;
11. select c.course\_name, s.first\_name from courses c left join students s on c.course\_id = s.course\_id;
12. select c.department, s.first\_name from students s right join courses c on s.course\_id = c.course\_id;
13. select c.course\_name from courses c left join students s on c.course\_id = s.course\_id where s.student\_id is null;
14. select c.course\_name, c.credits from courses c;
15. select distinct c.department, s.first\_name from students s left join courses c on s.course\_id = c.course\_id;
16. select s.first\_name, c.course\_name from students s left join courses c on s.course\_id = c.course\_id union

select s.first\_name, c.course\_name from students s right join courses c on s.course\_id = c.course\_id;

1. select s.first\_name, c.course\_name from students s left join courses c on s.course\_id = c.course\_id union

select s.first\_name, c.course\_name from students s right join courses c on s.course\_id = c.course\_id;

1. select s.first\_name, c.course\_name, c.dept\_head from students s left join courses c on s.course\_id = c.course\_id;
2. select s.first\_name from students s inner join courses c on s.course\_id = c.course\_id where c.department = 'technology';
3. select c.department, count(s.student\_id) as total\_students from courses c left join students s on c.course\_id = s.course\_id group by c.department;
4. select s.first\_name, c.department, c.credits from students s left join courses c on s.course\_id = c.course\_id;
5. select c.department, c.course\_name, s.first\_name from courses c left join students s on c.course\_id = s.course\_id;
6. select s1.first\_name as student1, s2.first\_name as student2, c.course\_name from students s1 join students s2 on s1.course\_id = s2.course\_id and s1.student\_id < s2.student\_id join courses c on s1.course\_id = c.course\_id;
7. select s1.first\_name, s2.first\_name, c.department from students s1 join students s2 on s1.student\_id < s2.student\_id join courses c on s1.course\_id = c.course\_id and s2.course\_id = c.course\_id;
8. select s2.first\_name from students s1 join students s2 on s1.course\_id = s2.course\_id where s1.first\_name = 'alice' and s2.first\_name <> 'alice';
9. select c.course\_name, count(s.student\_id) as total\_students from courses c left join students s on c.course\_id = s.course\_id group by c.course\_name;
10. select avg(c.credits) as avg\_credits from courses c join students s on c.course\_id = s.course\_id;
11. select c.department, count(s.student\_id) as total\_students from courses c left join students s on c.course\_id = s.course\_id group by c.department order by total\_students desc limit 1;
12. select s.first\_name from students s join courses c on s.course\_id = c.course\_id where c.credits > (select avg(credits) from courses);
13. select s.first\_name from students s join courses c on s.course\_id = c.course\_id where c.department = (select department from courses where course\_name = 'physics');



