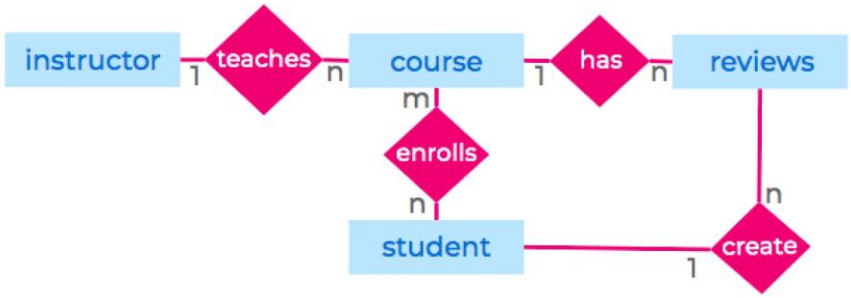
Coding Practice - 01

In this practice set, let's get the hold of SQL Joins operations using the following database.

Database:

The database stores the sample data of an e-learning platform. The database consists of instructor, course, review, and student tables.



- An instructor can teach many courses. A course is taught by only one instructor.
- A student can enroll for multiple courses. A course can have multiple students.
- A student can give multiple reviews.
- A course can have multiple reviews

Refer the tables in the code playground for a better understanding of the database.

QUESTIONS

1. Perform natural join between

   course and instructor table.

   Note:

   - Do not apply ORDER BY, LIMIT, OFFSET clauses as it is not required for this problem.

   Expected Output Format:

id	name	duration	instructor_id	full_name	gen
...	...	...	...	...	..

SHOW ANSWER

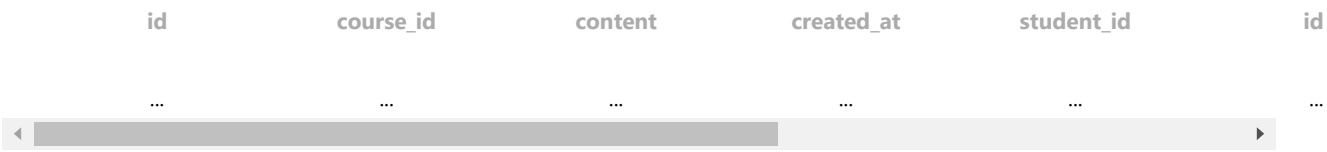
SELECT course.id, course.name, course.duration, course.instructor_id, instructor.full_name, instructor.gender FROM course JOIN instructor ON course.instructor_id = instructor.instructor_id;

2. Perform inner join between review and student table.

Note:

- Do not apply ORDER BY, LIMIT, OFFSET clauses as it is not required for this problem.

Expected Output Format:



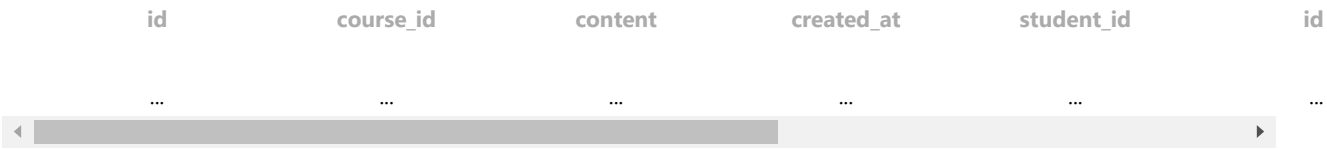
SELECT review.id, review.course_id, review.content, review.created_at, review.student_id, student.id, student.full_name, student.age, student.gender FROM review INNER JOIN student ON review.student_id = student.id;

3. Get all the reviews along with the course names.
Every review is associated with a course. So, we can perform an inner join on review and course tables.

Note:

- Do not apply ORDER BY, LIMIT, OFFSET clauses as it is not required for this problem.

Expected Output Format:



SELECT review.id, review.course_id, review.content, review.created_at, review.student_id, course.id, course.name, course.duration, course.instructor_id FROM review INNER JOIN course ON review.course_id = course.id;

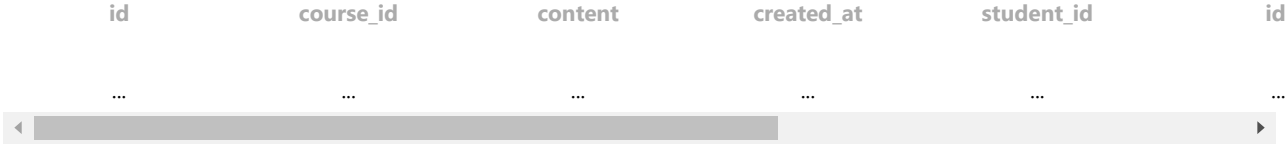
4. Continuation of question 3.

Get all the reviews on the "Cyber Security" course.

Note:

- We can perform inner join on review and course table.
- Do not apply ORDER BY, LIMIT, OFFSET clauses as it is not required for this problem.

Expected Output Format:



SELECT review.id, review.course_id, review.content, review.created_at, review.student_id, course.id, course.name, course.duration, course.instructor_id FROM review INNER JOIN course ON review.course_id = course.id WHERE course.name = 'Cyber Security';

5. Get all the courses and corresponding reviews.
For every course, there need not be a review. So, we need to perform a left join between the course and review tables.

Note:

Don't apply ORDER BY, LIMIT, OFFSET clauses as it is not required for this problem **Expected**

Output Format:

id	name	duration	instructor_id	id	course
...	...	...	...	...	...

SELECT course.id, course.name, course.duration, course.instructor_id, review.id, review.course_id, review.content, review.created_at, review.student_id FROM course LEFT JOIN review ON review.course_id = course.id;

6. Continuation of question 5.
For the “Cyber Security” course, get all the reviews using the left join between the course and review tables.

Note:

- Don't apply ORDER BY, LIMIT, OFFSET clauses as it is not required for this problem

Expected Output Format:

id	name	duration	instructor_id	id	course
...	...	...	...	...	...

SELECT course.id, course.name, course.duration, course.instructor_id, review.id, review.course_id, review.content, review.created_at, review.student_id FROM course LEFT JOIN review ON review.course_id = course.id WHERE course.name = 'Cyber Security';

7. Continuation of question 5.
For the “Linux” course, get all the reviews using the left join between the course and review tables.

Note:

- Don't apply ORDER BY, LIMIT, OFFSET clauses as it is not required for this problem

Expected Output Format:

id	name	duration	instructor_id	id	course
...	...	...	...	...	...

SELECT course.id, course.name, course.duration, course.instructor_id, review.id, review.course_id, review.content, review.created_at, review.student_id FROM course LEFT JOIN review ON review.course_id = course.id WHERE course.name = 'Linux';

8. In question 7, as the course “Linux” has no reviews, the columns corresponding to review table i.e id, content, etc., in the output has NULL value.

   Now, to get all the courses that does not have any reviews,
   We shall perform left join between

   course and review tables, and then filter the rows for which review.id is NULL

   Note:

   Don't apply ORDER BY, LIMIT, OFFSET clauses as it is not required for this problem **Expected**

   Output Format:

name
...

SELECT course.name FROM course LEFT JOIN review ON review.course_id = course.id WHERE review.id IS NULL;

9. Get the full_name of students who have not enrolled in any course.

   Note:

   - We can perform left join between student and student_course tables, and then filter the rows for which course_id is NULL
   - Don't apply ORDER BY, LIMIT, OFFSET clauses as it is not required for this problem.

   Expected Output Format:

full_name
...

SELECT student.full_name FROM student LEFT JOIN student_course ON student.id = student_course.student_id WHERE course_id IS NULL;

10. Get the course names in which no student has registered.

Note:

- We can perform left join between course and student_course tables, and then filter the rows for which student_id is NULL
- Don't apply ORDER BY, LIMIT, OFFSET clauses as it is not required for this problem.

Expected Output Format:

name

...

SELECT course.name FROM course LEFT JOIN student_course ON course.id = student_course.course_id WHERE student_id IS NULL;

1.Student Table

id	full_name	age	gender
1	Varun	16	M
2	Ajay	16	M
3	Sandhya	19	F
4	Afrin	16	F
5	Khyathi	18	F
6	vihu	16	M
7	Olivia	18	F
8	Isabella	19	F
9	Jack	19	M
10	Jacob	17	M
11	Ram	18	M

2. Course Table

id	name	duration	instructor_id
11	Machine Learning	90	102
12	Artificial Intelligence	90	102

id	name	duration	instructor_id
13	Data Science	60	103
14	Augmented Reality	80	104
15	Cyber Security	60	101
16	Virtual Reality	80	105
17	Data Mining	20	
18	Big Data	20	108
19	Data Structures	30	
20	Cloud Computing	15	101
21	Ethical Hacking	20	
22	Linux	20	102

3. Instructor Table

instructor_id	full_name	gender
101	Alex	M
102	Arun	M
103	Robert A. Iyer	M
104	Bhavani	F
105	Bentlee	M
106	Umesh Gupta	M
107	S. Radha Krishna	M
108	Nihonbashi	M
109	Miriyala Ravinder Reddy	M
110	D. Shivani	F

4. Review Table

id	course_id	content	created_at	student_id
1	11	Great course	2021-01-19	1
2	15	Good explanation	2021-01-19	2
3	15	Cyber Security is awesome	2021-01-20	2
4	12	Made understood well	2021-01-19	16
5	12	AI is next big thing	2021-02-20	6
6	14	Learning AR is made fun	2017-02-20	5
7	13	improved analytical skills	2018-02-20	11
8	12	Gained in-depth knowledge in AI field	2021-02-20	4

5. Student_course Table

id	student_id	course_id	score	enrollment_date
1	1	11	80	2021-01-16
2	2	15	60	2021-01-17
3	3	11	90	2021-01-19
4	4	12	45	2017-01-16
5	4	13	72	2019-02-12
6	5	14	88	2017-01-16
7	5	15	50	2019-02-12
8	6	15	75	2020-01-16
9	1	15	90	2020-01-16
10	6	12	88	2021-01-16
11	1	13	98	2021-08-16

id	student_id	course_id	score	enrollment_date
12	1	14	40	2021-08-16
13	11	11	85	2019-01-16
14	11	12	43	2020-03-16
15	11	13	43	2021-02-16
16	2	22	80	2020-02-16