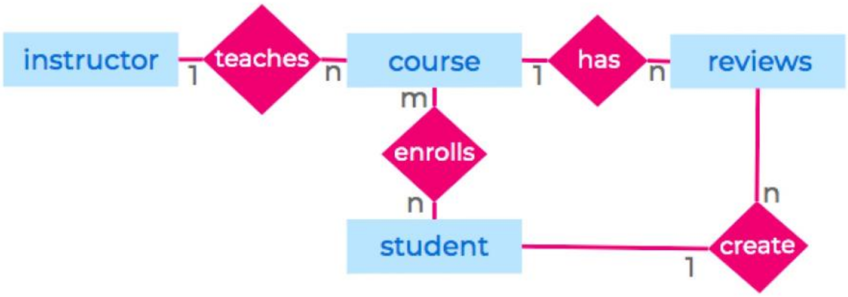
Coding Practice - 02

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. Fetch all the courses that are being taught by “Alex”.

Note:

- Solving this problem involves joining of `course` table and `instructor` table. Note that both the tables have `instructor_id` column in common.
- As we only want the courses taught by "Alex", we have to apply filter condition.
- Don't apply ORDER BY, LIMIT, OFFSET clauses as it is not required for this problem.

Expected Output Format:

course_id	course_name	instructor_name
...	...	...

SHOW ANSWER

```
SELECT course.id AS course_id, course.name AS course_name, instructor.full_name AS
instructor_name FROM course INNER JOIN instructor ON instructor.instructor_id =
course.instructor_id WHERE instructor.full_name = 'Alex';
```

2. Get all the reviews of “Cyber Security” course . ✓

Note:

- Solving this problem involves performing inner join on review and course tables.
- Don't apply ORDER BY, LIMIT, OFFSET clauses as it is not required for this problem.

Expected Output Format:

course_name	student_id	content
...	...	...

```
SELECT course.name AS course_name, review.student_id, review.content FROM review INNER JOIN
course ON review.course_id = course.id WHERE course.name = 'Cyber Security';
```

3. For a student with student (id = 1), get all the courses and the scores she/he secured in the year 2021.✓

Note:

- Solving this question involves performing inner join on student_course and course tables.
- You can get the year from the enrollment date.
- Don't apply ORDER BY, LIMIT, OFFSET clauses as it is not required for this problem.

Expected Output Format:

student_id	name	score
...	...	...

```
SELECT student_course.student_id, course.name, student_course.score FROM student_course
INNER JOIN course ON student_course.course_id = course.id WHERE student_course.student_id = 1
AND strftime("%Y", enrollment_date) = '2021';
```

4. Get all the student details who scored more than 70 in Cyber Security course (course_id = 15) in the year 2020.✓

Note:

- Solving this question involves performing inner join on student_course and student tables.
- You can get the year from the enrollment date.
- Don't apply ORDER BY, LIMIT, OFFSET clauses as it is not required for this problem.

Expected Output Format:

student_id	student_name	score	course_id	enrollment_date
...	...	...	...	...

```
SELECT student_course.student_id, student.full_name AS student_name, student_course.score,
student_course.course_id, student_course.enrollment_date FROM student_course INNER JOIN
student ON student.id = student_course.student_id WHERE student_course.course_id = 15 AND
strftime("%Y", enrollment_date) = '2020' AND student_course.score > 70;
```

5. Get all the student_ids who enrolled for the "Machine Learning" course in 2021. ✓

Note:

- Solving this question involves performing inner join on student_course and course tables.
- You can get the year from the enrollment date.

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

Expected Output Format:

student_id	course_name	enrollment_date
...	...	...

```
SELECT student_course.student_id, course.name AS course_name, student_course.enrollment_date
FROM student_course INNER JOIN course ON course.id = student_course.course_id WHERE
course.name = 'Machine Learning' AND strftime("%Y", enrollment_date) = '2021';
```

6. Continuation of question 5. ✓

Get the number of students who enrolled for the "Machine Learning" course in 2021.

Note:

- Solving this question involves performing inner join on course and student_course tables.
- You can get the year from the enrollment date.
- We have to perform the count() aggregation.

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

Expected Output Format:

course_name	no_of_students
...	...

```
SELECT course.name AS course_name, count(student_course.student_id) AS no_of_students FROM
student_course INNER JOIN course ON course.id = student_course.course_id WHERE course.name =
'Machine Learning' AND strftime("%Y", enrollment_date) = '2021' GROUP BY course_name;
```

7. Get the number of courses taken by “Ram”. ✓

Note:

- You can get the year from the enrollment date
-

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

Expected Output Format:

no_of_courses

...

SELECT count(course.id) AS no_of_courses FROM (course INNER JOIN student_course ON course.id = student_course.course_id) AS T INNER JOIN student ON student.id = T.student_id WHERE student.full_name = 'Ram';

8. For all the students, get the total number of courses taken by each student. ✓

A student need not register for any course as well. So, we need to perform a left join between the student and student_course tables.

Note:

- You can get the year from the enrollment date
-

As we have to calculate the number of courses for each student, we have to GROUP BY the students first and then perform the count() aggregation.

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

Expected Output Format:

full_name	no_of_courses
...	...

SELECT student.full_name, count(student_course.course_id) AS no_of_courses FROM student LEFT JOIN student_course ON student.id = student_course.student_id GROUP BY student.id;

9. Get the students who have taken at least 2 courses. ✓

Note:

- Solving this problem involves performing join operations on student and student_course tables.
-
- You can get the year from the enrollment date
-
- Use HAVING clause to filter the students who have taken at least two courses.

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

Expected Output Format:

full_name	no_of_courses
...	...

SELECT student.full_name, count(student_course.course_id) AS no_of_courses FROM student LEFT JOIN student_course ON student.id = student_course.student_id GROUP BY student.id HAVING no_of_courses >= 2;

10. Get all the students details and all the courses for which they have enrolled. ✓

Note:

- Here, we have to join student, student_course and course tables. Performing left join between
- the student and student_course tables perform left join on the combined table and course
- table.
- Don't apply ORDER BY, LIMIT, OFFSET clauses as it is not required for this problem.

Expected Output Format:

student_id	student_name	course_id	course_name	score	enrollmen
...	...	...	...	...	...

SELECT t.id AS student_id, t.full_name AS student_name, course.id AS course_id, course.name AS course_name, t.score, t.enrollment_date FROM (student LEFT JOIN student_course ON student.id = student_course.student_id) AS t LEFT JOIN course ON course.id = t.course_id;

11. Get all the student details who enrolled for the “Machine Learning” course in the year 2021.

Note:

- Here, we have to join student, student_course and course tables.
- Apply filters on the combined table.
- You can get the year from the enrollment date.
- Don't apply ORDER BY, LIMIT, OFFSET clauses as it is not required for this problem.

Expected Output Format:

student_id	full_name	course_id	course_name	enrollment_date
...	...	...	...	...

SELECT t.id AS student_id, t.full_name, course.id AS course_id, course.name AS course_name, t.enrollment_date FROM (student INNER JOIN student_course ON student.id = student_course.student_id) AS t INNER JOIN course ON course.id = t.course_id WHERE strftime("%Y", enrollment_date) = '2021' AND course.name = 'Machine Learning';

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
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	

id	name	duration	instructor_id
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

id	course_id	content	created_at	student_id
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
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