DESCRIPTION

Table: Students

```
+---------------+---------+
| Column Name   | Type    |
+---------------+---------+
| student_id    | int     |
| student_name  | varchar |
+---------------+---------+
```

student_id is the primary key (column with unique values) for this table.

Each row of this table contains the ID and the name of one student in the school.

Table: Subjects

```
+--------------+---------+
| Column Name  | Type    |
+--------------+---------+
| subject_name | varchar |
+--------------+---------+
```

subject_name is the primary key (column with unique values) for this table.

Each row of this table contains the name of one subject in the school.

Table: Examinations

```
+--------------+---------+
| Column Name  | Type    |
+--------------+---------+
| student_id   | int     |
| subject_name | varchar |
+--------------+---------+
```

There is no primary key (column with unique values) for this table. It may contain duplicates.

Each student from the Students table takes every course from the Subjects table.

Each row of this table indicates that a student with ID student_id attended the exam of subject_name.

Write a solution to find the number of times each student attended each exam.

Return the result table ordered by student_id and subject_name.

The result format is in the following example.

Example 1:

Input:

Students table:

```
+------------+--------------+
| student_id | student_name |
+------------+--------------+
| 1          | Alice        |
| 2          | Bob          |
| 13         | John         |
| 6          | Alex         |
+------------+--------------+
```

Subjects table:

```
+--------------+
| subject_name |
+--------------+
| Math         |
| Physics      |
| Programming  |
+--------------+
```

Examinations table:

```
+------------+--------------+
| student_id | subject_name |
+------------+--------------+
| 1          | Math         |
| 1          | Physics      |
| 1          | Programming  |
| 2          | Programming  |
| 1          | Physics      |
| 1          | Math         |
| 13         | Math         |
| 13         | Programming  |
| 13         | Physics      |
| 2          | Math         |
| 1          | Math         |
+------------+--------------+
```

Output:

```
+------------+--------------+--------------+----------------+
| student_id | student_name | subject_name | attended_exams |
+------------+--------------+--------------+----------------+
| 1          | Alice        | Math         | 3              |
| 1          | Alice        | Physics      | 2              |
| 1          | Alice        | Programming  | 1              |
| 2          | Bob          | Math         | 1              |
| 2          | Bob          | Physics      | 0              |
| 2          | Bob          | Programming  | 1              |
| 6          | Alex         | Math         | 0              |
| 6          | Alex         | Physics      | 0              |
| 6          | Alex         | Programming  | 0              |
| 13         | John         | Math         | 1              |
| 13         | John         | Physics      | 1              |
| 13         | John         | Programming  | 1              |
+------------+--------------+--------------+----------------+
```

Explanation:

The result table should contain all students and all subjects.

Alice attended the Math exam 3 times, the Physics exam 2 times, and the Programming exam 1 time.

Bob attended the Math exam 1 time, the Programming exam 1 time, and did not attend the Physics exam.

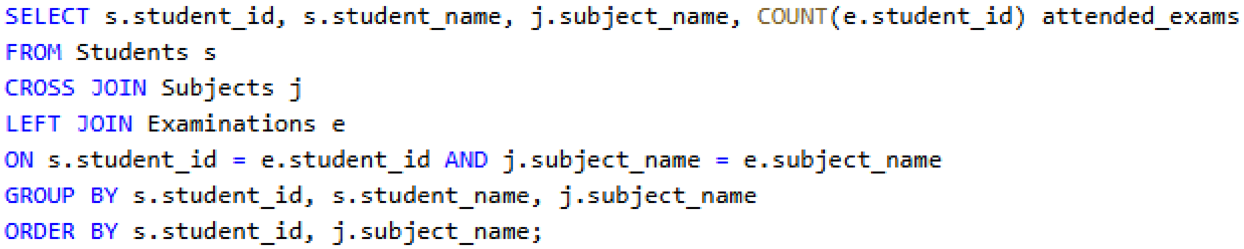
Alex did not attend any exams.

John attended the Math exam 1 time, the Physics exam 1 time, and the Programming exam 1 time.

SOLUTION

MySQL:

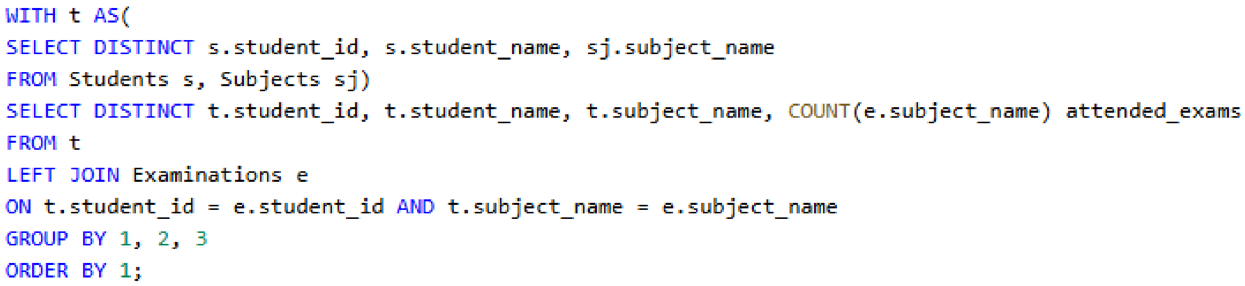
- Using is CROSS JOIN, LEFT JOIN and COUNT() function



```
SELECT s.student_id, s.student_name, j.subject_name, COUNT(e.student_id) attended_exams
FROM Students s
CROSS JOIN Subjects j
LEFT JOIN Examinations e
ON s.student_id = e.student_id AND j.subject_name = e.subject_name
GROUP BY s.student_id, s.student_name, j.subject_name
ORDER BY s.student_id, j.subject_name;
```

PostgreSQL:

- Using is DISTINCT, Subquery, WITH, LEFT JOIN and COUNT() function



```
WITH t AS(
SELECT DISTINCT s.student_id, s.student_name, sj.subject_name
FROM Students s, Subjects sj)
SELECT DISTINCT t.student_id, t.student_name, t.subject_name, COUNT(e.subject_name) attended_exams
FROM t
LEFT JOIN Examinations e
ON t.student_id = e.student_id AND t.subject_name = e.subject_name
GROUP BY 1, 2, 3
ORDER BY 1;
```