Perfect question 👍 Let's go through this SQL query step by step, **line by line**, and I'll explain *why* we write each part.

Query:

sql Copy Edit

```sql
SELECT stu.student_id, stu.student_name, sub.subject_name,
       COUNT(exam.subject_name) AS attended_exams
FROM Students stu
CROSS JOIN Subjects sub
LEFT JOIN Examinations exam
    ON stu.student_id = exam.student_id
   AND sub.subject_name = exam.subject_name
GROUP BY stu.student_id, stu.student_name, sub.subject_name
ORDER BY stu.student_id, sub.subject_name;
```

🔎 Explanation:

1. SELECT stu.student_id, stu.student_name, sub.subject_name, COUNT(exam.subject_name) AS attended_exams

- `stu.student_id, stu.student_name` → We want the **student details**.
- `sub.subject_name` → We want the **subject**.
- `COUNT(exam.subject_name) AS attended_exams`
  - We check **how many times this student attended this subject exam**.
  - `COUNT()` is used because we are grouping results (so one row per student + subject).

---

2. FROM Students stu

- `Students` table → Contains **list of students**.
- `stu` → is just an **alias** (shortcut name) for `Students`, so we don't write full table name again and again.
  Example: Instead of writing `Students.student_id`, we write `stu.student_id`.

---

3. CROSS JOIN Subjects sub

- `CROSS JOIN` → Creates all **possible combinations** of students and subjects.
  - Example: If there are 3 students and 4 subjects, CROSS JOIN makes `3 × 4 = 12` rows.
  - Why? Because every student can appear with every subject.
- `Subjects sub` → Table of all subjects. `sub` is an alias.

👉 This ensures we check **each student with every subject**, even if they didn't take the exam.

---

4. LEFT JOIN Examinations exam

- `Examinations` table → Stores actual exam records (`student_id`, `subject_name`).
- `LEFT JOIN` → Means:
  - Take **all rows from the left side** (Students × Subjects combination).

- Match rows from `Examinations` if available.
- If no match, put `NULL` (meaning the student didn't attend that subject exam).

👉 This ensures students who didn't attend still appear in results.

---

5. ON stu.student_id = exam.student_id AND sub.subject_name = exam.subject_name

- Join condition:
  - `stu.student_id = exam.student_id` → Match exam records of the same student.
  - `sub.subject_name = exam.subject_name` → Match subject names correctly.
- Together: Finds whether a student attended that specific subject.

---

6. GROUP BY stu.student_id, stu.student_name, sub.subject_name

- We group results by **student + subject** so we can count how many times they attended.
- Example: If student `1` attended `Math` exam twice, COUNT will give `2`.

---

7. ORDER BY stu.student_id, sub.subject_name

- Finally, we order the results:
  - First by `student_id` → Students appear in order.
  - Then by `subject_name` → Subjects for each student appear alphabetically (or as stored).

📊 Example Walkthrough

Let's say:

Students table

student_id	student_name
1	Alice
2	Bob

Subjects table

subject_name
Math
English

Examinations table

student_id	subject_name
1	Math
2	English
2	English

👉 After query:

student_id	student_name	subject_name	attended_exams
1	Alice	English	0
1	Alice	Math	1
2	Bob	English	2
2	Bob	Math	0

✅ This query is used to create a **report of all students and all subjects**, with the number of times they attended each exam.

Do you want me to also **draw the step-by-step table transformations** (like CROSS JOIN result → after LEFT JOIN → after GROUP BY), so you can **visualize** how the query builds the final output?