In SQL, **UNION** and **JOIN** are both used to combine data from two or more tables, but they serve very different purposes and work differently.

### ****UNION****:

* **Purpose**: Combines the results of two or more **queries** into a single result set.
* **How it works**: Stacks rows from multiple result sets **vertically**, as long as they have the same number of columns and compatible data types.
* **Key difference**:
  + UNION removes duplicate rows by default. To include duplicates, use UNION ALL.
* **Use case**: When you need to combine rows from separate queries that produce similar results but come from different tables.

**Example**:

sql

Copy code

SELECT name, age FROM students

UNION

SELECT name, age FROM teachers;

This combines all names and ages from both students and teachers tables into a single list.

### ****JOIN****:

* **Purpose**: Combines rows from two or more tables **horizontally** based on a related column.
* **How it works**: Matches rows from one table to rows in another using a condition (e.g., foreign key or common field).
* **Key difference**:
  + JOIN does not merge queries but rather combines columns from related rows.
  + JOIN requires a logical relationship between tables (e.g., ON condition).
* **Use case**: When you need to pull related data from multiple tables based on relationships between them.

**Example**:

sql

Copy code

SELECT students.name, classes.class\_name

FROM students

JOIN classes

ON students.class\_id = classes.id;

This retrieves each student’s name and their corresponding class name by matching the class\_id in students with the id in classes.

### ****Key Differences****:

| Aspect | UNION | JOIN |
| --- | --- | --- |
| Combines | Result sets (rows) vertically | Tables (columns) horizontally |
| Requirement | Same number of columns, compatible types | Logical relationship between tables |
| Removes duplicates | Yes, unless UNION ALL is used | No |
| Data Structure | Stacks rows | Matches rows |
| Typical Use Case | Merge results of separate queries | Relate and retrieve data from multiple tables |

If you’re looking for overlapping concepts:

* Use **UNION** when dealing with **multiple independent datasets** that share a similar structure.
* Use **JOIN** when working with **related datasets** that share a logical relationship.

4o

##### You said:

what do you mean by horizontal and vertical.

###### ChatGPT said:

In the context of SQL, **horizontal** and **vertical** refer to the way data is combined when using operations like JOIN and UNION.

### ****Horizontal Combination****:

* Data is combined **side-by-side**, meaning additional columns are added to each row.
* This happens when you use **JOIN**, as rows from one table are matched with rows from another based on a condition, and their columns are combined.

**Example of Horizontal Combination**:

plaintext

Copy code

Table 1 (students): Table 2 (classes):

+----+--------+ +----+-------------+

| id | name | | id | class\_name |

+----+--------+ +----+-------------+

| 1 | Alice | | 1 | Math |

| 2 | Bob | | 2 | Science |

Result of a JOIN (Horizontal Combination):

+--------+-------------+

| name | class\_name |

+--------+-------------+

| Alice | Math |

| Bob | Science |

Here, columns from students and classes are combined **side-by-side** for matched rows.

### ****Vertical Combination****:

* Data is combined **top-to-bottom**, meaning additional rows are appended to the result.
* This happens when you use **UNION**, as rows from multiple queries or tables are stacked one below the other.

**Example of Vertical Combination**:

plaintext

Copy code

Table 1 (students): Table 2 (teachers):

+--------+-----+ +--------+-----+

| name | age | | name | age |

+--------+-----+ +--------+-----+

| Alice | 15 | | Mr. Lee| 40 |

| Bob | 16 | | Ms. May| 35 |

Result of a UNION (Vertical Combination):

+--------+-----+

| name | age |

+--------+-----+

| Alice | 15 |

| Bob | 16 |

| Mr. Lee| 40 |

| Ms. May| 35 |

Here, rows from students and teachers are combined **top-to-bottom**.

### ****Key Differences****:

| Aspect | Horizontal Combination | Vertical Combination |
| --- | --- | --- |
| Operation | JOIN | UNION |
| Data Alignment | Combines rows by adding columns | Combines rows by adding more rows |
| Structure Impact | Wider result set (more columns) | Taller result set (more rows) |

### Visual Analogy:

* Horizontal: Imagine aligning two books **side-by-side** to create a wider collection.
* Vertical: Imagine stacking two books **on top of each other** to create a taller pile.

4o