**What Is ORDER BY?**

The ORDER BY clause is used to sort the result set of a query based on one or more columns or expressions. It’s evaluated after filtering (WHERE) and before pagination (LIMIT, OFFSET).

Syntax:

**SELECT column1, column2 FROM table\_name ORDER BY column1 [ASC | DESC], column2 [ASC | DESC];**

1. ASC (ascending) is the default.
2. DESC (descending) must be explicitly stated.

**You can sort by:**

* Column names
* Column positions (e.g., ORDER BY 2)
* Expressions (e.g., ORDER BY price \* qty)
* Aliases (e.g., ORDER BY total\_price if defined in SELECT)

**SELECT product\_id, qty FROM sales ORDER BY product\_id, qty DESC;**

* **ORDER BY product\_id** defaults to **ascending** order.
* **qty DESC** explicitly sorts qty in **descending** order.
* However, because ORDER BY evaluates **left to right**, this query:
* First sorts by product\_id in ascending order.
* Then, **within each product\_id group**, it sorts by qty in descending order.

**What Is Multi-Column ORDER BY?**

When you use multiple columns in the ORDER BY clause, SQL sorts your result set **hierarchically** — meaning it sorts by the **first column**, then **breaks ties** using the second column, then the third, and so on.

**SELECT product\_id, qty, price FROM sales ORDER BY product\_id ASC, qty DESC, price ASC;**

This means:

1. Sort by product\_id in ascending order.
2. If multiple rows have the same product\_id, sort those by qty in descending order.
3. If there are still ties, sort those by price in ascending order.

**What Is Pagination?**

Pagination is the process of dividing a large result set into smaller, manageable chunks — or "pages" — so users can view data incrementally (like page 1, page 2, etc.).

**Why Use Pagination?**

* Improves performance by loading only a subset of data.
* Enhances user experience (especially in web apps).
* Reduces memory and bandwidth usage.

**Syntax:**

SELECT \* FROM table\_name ORDER BY column\_name OFFSET x LIMIT y;

* OFFSET x: Skips the first x rows.
* LIMIT y: Returns the next y rows.

**Example:**

**SELECT \* FROM college2 ORDER BY id OFFSET 20 LIMIT 5;**

**What Does OFFSET Mean?**

* OFFSET tells SQL to **skip** a certain number of rows **before** starting to return results.
* In your query, OFFSET 20 means: **skip the first 20 rows** after sorting by id.

1. **ORDER BY id**: Sorts all rows in the college2 table by the id column in ascending order.
2. **OFFSET 20**: Skips the first 20 rows of that sorted result.
3. **LIMIT 5**: Then returns the **next 5 rows** after those 20.

or

SELECT \* FROM college2 ORDER BY id LIMIT 5 OFFSET 20;

This is **functionally identical**, and many databases (like PostgreSQL) prefer this order: LIMIT first, then OFFSET.