Let’s go **in detail about the SQL ORDER BY clause** with examples. ORDER BY is used to **sort the result set** of a query.

**1. Basic Syntax**

SELECT column1, column2, ...

FROM table\_name

ORDER BY column1 [ASC|DESC], column2 [ASC|DESC];

* **ASC** → Ascending order (default)
* **DESC** → Descending order

**2. Simple Example – Single Column**

SELECT Name, Salary

FROM Employees

ORDER BY Salary ASC;

* Sorts employees by Salary **low to high**.

SELECT Name, Salary

FROM Employees

ORDER BY Salary DESC;

* Sorts employees by Salary **high to low**.

**3. Multiple Column Sorting**

* You can sort by more than one column.

SELECT Name, DepartmentID, Salary

FROM Employees

ORDER BY DepartmentID ASC, Salary DESC;

* **First:** Sorted by DepartmentID ascending
* **Second:** Within each department, sorted by Salary descending

**4. Sort by Column Position**

* You can use **column index** instead of column name.

SELECT Name, Salary

FROM Employees

ORDER BY 2 DESC;

* 2 refers to the **second column**, i.e., Salary.

Not recommended for readability, but valid.

**5. Sorting Strings (Alphabetical Order)**

SELECT Name

FROM Employees

ORDER BY Name ASC;

* Sorts names **A → Z**

SELECT Name

FROM Employees

ORDER BY Name DESC;

* Sorts names **Z → A**

**6. Sorting by Expressions**

* You can sort by calculations or expressions.

SELECT Name, Salary, Salary \* 1.10 AS NewSalary

FROM Employees

ORDER BY Salary \* 1.10 DESC;

* Sorts by **10% incremented salary** in descending order.

**7. Sorting with NULL Values**

* **MySQL / PostgreSQL:**
  + NULLs **come last** by default in ASC order.
  + Use NULLS FIRST or NULLS LAST in PostgreSQL/Oracle.

SELECT Name, Bonus

FROM Employees

ORDER BY Bonus DESC NULLS LAST;

* Sorts by Bonus descending, **NULL values at the end**.

**8. Example – Combined Query**

SELECT Name, DepartmentID, Salary

FROM Employees

WHERE Salary > 30000

ORDER BY DepartmentID ASC, Salary DESC;

* Filters employees with salary > 30,000
* **Sorts by department** ascending
* **Within department, sorts by salary** descending

**✅ Key Notes**

1. **Default is ASC** if not specified.
2. Can sort **by one or more columns**.
3. Can sort by **expressions** or **aliases** in some DBMS.
4. Works with **all SQL queries**: simple SELECT, joins, unions, etc.

If you want, I can make a **complete SQL cheat sheet PDF** that includes:

* **String functions**
* **Aggregate functions**
* **Mathematical functions**
* **Date functions**
* **Set operators**
* **ORDER BY with examples**

All in **one page for quick revision**.

Do you want me to make it?

**You said:**

limit offset

**ChatGPT said:**

Absolutely! Let’s go **in detail about LIMIT and OFFSET in SQL** with examples. These are used to **control how many rows are returned** and from which point in the result set.

**1. LIMIT – Restrict Number of Rows**

* **Purpose:** Returns only a **specified number of rows**.
* **Syntax (MySQL / PostgreSQL / SQLite):**

SELECT column1, column2

FROM table\_name

LIMIT number\_of\_rows;

**Example: Get first 5 employees**

SELECT Name, Salary

FROM Employees

ORDER BY Salary DESC

LIMIT 5;

* Returns the **top 5 highest-paid employees**.

**2. OFFSET – Skip Rows**

* **Purpose:** Skip a number of rows before returning results.
* **Syntax (MySQL / PostgreSQL):**

SELECT column1, column2

FROM table\_name

ORDER BY column1

LIMIT number\_of\_rows OFFSET skip\_rows;

**Example: Get next 5 employees after first 5**

SELECT Name, Salary

FROM Employees

ORDER BY Salary DESC

LIMIT 5 OFFSET 5;

* Skips the **first 5 rows** and returns the **next 5**.

**3. Alternative Syntax in MySQL**

* MySQL also allows **LIMIT offset, count**:

SELECT Name, Salary

FROM Employees

ORDER BY Salary DESC

LIMIT 5, 5;

* 5,5 → skip 5 rows, then take 5 rows (same as LIMIT 5 OFFSET 5).

**4. PostgreSQL Syntax**

* PostgreSQL supports both:

SELECT Name, Salary

FROM Employees

ORDER BY Salary DESC

LIMIT 5 OFFSET 5;

* Or using FETCH:

SELECT Name, Salary

FROM Employees

ORDER BY Salary DESC

OFFSET 5 ROWS

FETCH NEXT 5 ROWS ONLY;

**5. SQL Server Equivalent**

* SQL Server uses TOP or OFFSET-FETCH:

**a) Using TOP**

SELECT TOP 5 Name, Salary

FROM Employees

ORDER BY Salary DESC;

**b) Using OFFSET-FETCH**

SELECT Name, Salary

FROM Employees

ORDER BY Salary DESC

OFFSET 5 ROWS

FETCH NEXT 5 ROWS ONLY;

**6. Pagination Example**

* Useful for **pages in a web app**.

-- Page 1 (rows 1-5)

SELECT Name, Salary

FROM Employees

ORDER BY Salary DESC

LIMIT 5 OFFSET 0;

-- Page 2 (rows 6-10)

SELECT Name, Salary

FROM Employees

ORDER BY Salary DESC

LIMIT 5 OFFSET 5;

**✅ Key Points**

1. LIMIT controls **number of rows returned**.
2. OFFSET controls **starting point**.
3. MySQL allows **LIMIT offset, count** syntax.
4. PostgreSQL uses **LIMIT/OFFSET** or **FETCH**.
5. SQL Server uses **TOP** or **OFFSET-FETCH**.
6. Essential for **pagination** in web apps.