# 📘 Intermediate SQL – Lesson 7: Using PARTITION BY

## 🎯 What You'll Learn

In this lesson, you'll learn how to:

- Use the `PARTITION BY` clause in SQL
- Understand how it compares to `GROUP BY`
- Apply window functions that calculate values across subsets of rows without reducing your dataset

---

## 🧠 What is PARTITION BY?

The `PARTITION BY` clause is used with **window functions** to divide result sets into **logical partitions** (or groups) — similar to how `GROUP BY` works — but without reducing the number of rows in your result.

Each row retains its detail, while an aggregate function (like `COUNT` or `AVG`) is calculated **within its group**.

---

## 🔄 PARTITION BY vs GROUP BY

| Feature | GROUP BY | PARTITION BY |
|---|---|---|
| Output | Reduces number of rows | Keeps all original rows |
| Use Case | Summarize grouped data | Add group-level info to each row |
| Flexibility | Limited to grouped columns only | Can include non-aggregated columns |

---

## 📊 Example Use Case: Counting Gender Distribution

Imagine you want to see **how many males and females** exist in your company — **alongside each employee's name and salary**.

- `GROUP BY` can tell you how many males or females exist — but **not while showing individual names**.
- `PARTITION BY` allows you to display **first name, last name, salary**, and still add a column that says “there are X others like you.”

---

## ✅ Why PARTITION BY Is Valuable

- Helps enrich each row with **group-level insights**
- Keeps your dataset intact — every row is preserved
- Perfect for dashboards, analytics, and reports where row-level detail is needed along with grouped summaries

---

## 💡 Key Tip

With `PARTITION BY`, your aggregate function (e.g., `COUNT`, `AVG`, `SUM`) is applied **within each group** but shown **on every row**.
This is different from `GROUP BY`, which aggregates data into one row **per group**.

---

## 📌 Recap

✅ `PARTITION BY` divides your data for **window functions** like COUNT or AVG
✅ It **does not reduce** the number of rows
✅ Perfect for showing group-level metrics **next to row-level data**
✅ Use it when `GROUP BY` is too restrictive and you want to keep more detail