📘 Intermediate SQL – Lesson 2: Combining Data with UNION

🎯 What You'll Learn

In this lesson, you'll learn how to:

- Combine results from multiple tables using `UNION`
- Understand the difference between `UNION` and `UNION ALL`
- Identify when to use unions instead of joins
- Avoid common mistakes when merging datasets

🔍 What is a UNION?

A **UNION** allows you to **stack results** from two or more tables (or queries) that share the same structure.
Unlike a join — which merges tables side-by-side — a union **merges data vertically**, appending rows from one dataset to another.

🆚 UNION vs JOIN

JOIN	UNION
Combines data **side by side**	Combines data **top to bottom**
Requires a common column	Requires the **same number of columns** and **compatible data types**
Useful for combining related attributes	Useful for combining similar records from different sources

---

🔄 UNION vs UNION ALL

✅ `UNION`

- Combines and removes **duplicate rows**
- Similar to applying a **DISTINCT** automatically
- Ensures the final result has only unique rows

🔁 `UNION ALL`

- Combines and **keeps all** rows, including duplicates
- Useful when you want to retain full details or analyze repetition

---

📂 Real-World Example

You have two employee tables:

- `EmployeeDemographics`: Office staff
- `WarehouseEmployeeDemographics`: Warehouse staff

Both have the same structure (e.g., `first name`, `last name`, `age`, `gender`), so using a union combines them into a single unified list of all employees.

If a person exists in both tables (like Darryl), a `UNION` would **remove the duplicate**, while `UNION ALL` would **keep both rows**.

⚠️ Caution: Mismatched Columns

If you try to use a union on two queries with:

- **Different numbers of columns**, or
- **Incompatible data types** (like mixing numbers and names),

…it will either fail or return misleading results.

Example mistake:
Combining a `first name` with a `job title`, and an `age` with a `salary`, just because they align by position — will produce nonsense like:

- First Name: "Salesman"
- Age: "$45,000"

So always double-check that your unioned columns **match logically and by type**.

📌 Recap

✅ Use `UNION` to vertically stack results from two compatible tables
✅ Use `UNION ALL` if you want to **include duplicates**
✅ Columns must be in the same **order**, have the same **data types**, and **column count**
✅ Unions are great for merging identical structures (like multiple employee tables)
✅ Watch out for confusing mismatched columns

🚀 Coming Up Next: CASE Statements

In the next video, we'll explore how to write `CASE` statements — a powerful way to add conditional logic directly into your queries.