📘 Advanced SQL – Lesson 4: Stored Procedures

🎯 What You'll Learn

In this lesson, you'll learn how to:

- Understand what stored procedures are and why they're useful
- Create and execute stored procedures
- Pass parameters into stored procedures for dynamic results
- Modify stored procedures to improve reusability and performance

---

🧠 What is a Stored Procedure?

A **stored procedure** is a reusable block of SQL code that's stored in the database. It can contain one or more SQL statements — such as `SELECT`, `INSERT`, or `UPDATE` — and can accept **parameters** to make it dynamic.

💡 Why Use Stored Procedures?

Stored procedures:

- **Save time** by letting you reuse logic across multiple queries
- **Improve performance** by reducing network traffic (executed on the server)
- **Centralize logic**, so when you update the procedure, everyone gets the latest version
- **Allow parameters**, making the procedure dynamic and adaptable

---

🔧 Key Features

- **Reusable**: Once created, stored procedures can be executed again and again
- **Parameterizable**: Accept user input to customize output
- **Maintainable**: Centralized logic makes updating and maintaining easier
- **Efficient**: Executes directly on the server, speeding up heavy operations

---

🗂️ Use Cases

- Creating temporary tables and pre-processing data
- Generating reports or views based on input values (e.g., job title, department)
- Wrapping complex logic in one call for dashboards or applications
- Encapsulating query logic for use in apps, systems, or by multiple users

---

📝 Quick Notes

- To **create** a stored procedure, you define it once and save it to the database
- To **execute** it, you simply call it by name — optionally passing in parameters
- You can **alter** it later to add logic, conditions, or parameters
- Parameters can filter results dynamically — such as retrieving only "Salesman" data

---

🔁 Real-World Example

You might create a stored procedure to:

- Join employee and salary tables
- Filter by job title using a parameter
- Return only relevant rows for that role — instantly accessible by other users

For example, users could pass "Accountant" or "Salesman" as a filter without writing the SQL themselves.

📌 Recap

✅ Stored procedures are **saved queries** that live in your database
✅ They can take input **parameters** to return customized results
✅ They reduce redundancy, improve efficiency, and make SQL logic easier to manage
✅ You can **modify and extend** them with ease