**1. Types of SQL Server Backups: Full, Differential, and Transaction Log**

When managing data, you always need a safety net. Backups are your insurance policy. There are three main types:

* **Full Backup**: This is the complete snapshot of your database. It includes all data and objects, giving you a complete recovery point. It's usually taken daily or weekly, depending on the size of your database.
* **Differential Backup**: Think of it as a "difference checker." It captures only the changes made since the last **full backup**. It's quicker and smaller than full backups and is often taken more frequently (e.g., every few hours).
* **Transaction Log Backup**: This backup type logs every change made to the database after the last backup (full or differential). It allows **point-in-time recovery**, which is crucial for high-availability systems. It requires the database to be in **FULL recovery mode**.

**2. Permissions in SQL Server: GRANT vs. DENY**

Permissions control **who can do what** in your database.

* **GRANT** gives a user access to perform specific actions (like SELECT, INSERT, etc.).
* **DENY** explicitly blocks a user from doing something—even if they have that permission through a role or group.

**Levels** where permissions apply:

* **Server Level**: For things like creating databases or managing logins.
* **Database Level**: For accessing and modifying objects like tables, views, or stored procedures.

**Important Note**: DENY always overrides GRANT.

**3. SQL Server Profiler: What and When**

**SQL Server Profiler** is a performance-monitoring tool. It lets DBAs and developers **trace and analyze** what's happening in the SQL Server in real-time.

**Use cases**:

* Troubleshooting slow queries or performance issues
* Tracking specific user activity
* Debugging stored procedures
* Auditing security (e.g., who accessed which data)

It’s like a black box for your database — recording everything for later analysis.

**4. Triggers: What They Are and Why They Matter**

A **trigger** is a special kind of stored procedure that **automatically executes** in response to certain events — like **INSERT**, **UPDATE**, or **DELETE**.

**Why use triggers**:

* Enforce business rules (e.g., prevent deleting specific records)
* Track changes (e.g., logging modifications in an audit table)
* Automatically update related tables

**Key differences from stored procedures**:

* **Triggers fire automatically** based on events; stored procedures are called manually.
* Triggers can’t be executed on demand.
* Triggers are often used for **data integrity and automation**, while stored procedures are for **structured operations**.

