1. What’s the difference between Full, Differential, and Transactional Backup?

A **Full Backup** is a complete backup of the entire database, including all the data, schema, and objects. It captures the current state of the database and can be used alone to restore the database to the point when the backup was taken.

A **Differential Backup** stores only the data that has changed since the last full backup. It is faster and smaller than a full backup but cannot be used alone; it requires the last full backup to perform a complete restore.

A **Transaction Log Backup** (also known as Log Backup) records all the transactions that have occurred since the last log backup. It allows for point-in-time recovery of the database and is mainly used in combination with full and differential backups for high-availability and disaster recovery strategies.

---

2. What is Permission, and What’s the Difference Between GRANT and DENY? On What Level Are They Used?

Permissions in SQL Server define what actions a user or role can perform on various database objects such as tables, views, stored procedures, etc. They are essential for controlling access and maintaining security within the database environment.

- **GRANT** is used to give a user or role permission to perform specific actions like SELECT, INSERT, UPDATE, DELETE, EXECUTE, etc.
- **DENY** explicitly prevents a user or role from performing a specific action, even if the permission was granted indirectly through a role or group.

Permissions can be applied at different levels:

- **Server level** (e.g., logging in, creating databases)
- **Database level** (e.g., access to specific databases)
- **Schema and Object level** (e.g., tables, views, stored procedures)

Note: DENY always overrides GRANT unless the user is a member of the sysadmin role.

---

3. What is SQL Profiler, and When to Use It?

SQL Profiler is a performance monitoring and debugging tool provided by Microsoft SQL Server. It allows database administrators and developers to capture and analyze SQL Server events in real-time.

It is used to:

- Monitor and trace SQL queries and stored procedures.
- Identify slow-performing queries.
- Track user activity on the server.
- Debug application behavior related to SQL operations.
- Audit security-related events (like failed login attempts).

SQL Profiler is especially useful during performance tuning, debugging complex issues, or auditing system usage. However, it should be used carefully in production environments because it can impact performance if overused.

---

4. What is a Trigger, Why Use It, On What Level, and How Is It Different from a Stored Procedure?

A **Trigger** is a special type of stored procedure that automatically executes in response to specific events on a table or view, such as INSERT, UPDATE, or DELETE operations.

Why Use It?

- To enforce business rules automatically.
- To maintain audit trails (e.g., track changes to critical data).
- To automate system-level responses without manual intervention.

Level of Use:
Triggers are defined at the **table level** or **view level**, meaning they are associated directly with the specific table or view that the event occurs on.

Difference from a Stored Procedure:

- A stored procedure is executed manually or via application logic.
- A trigger runs **automatically** when the specified event occurs.
- Triggers are not called directly by users.
- Stored procedures offer more flexibility and are typically used for business logic, while triggers are best used for automated enforcement of rules.
