

1. Uso de funciones frente a procedimientos almacenados para cálculos

En **SQL Server**, tanto las **funciones** como los **procedimientos almacenados** son bloques de código reutilizables, pero se diferencian en su propósito y forma de uso.

- Una **función** (Function) está diseñada para **realizar cálculos y devolver un valor** (escalar o una tabla).
Por ejemplo, puede servir para calcular el **promedio de ventas** o el **impuesto de un producto**.
- Un **procedimiento almacenado** (Stored Procedure) se utiliza para ejecutar **un conjunto de instrucciones complejas**, como insertar, actualizar o eliminar datos, e incluso llamar a otras funciones o procedimientos.

Ventaja de usar funciones para cálculos:

- Se pueden **utilizar dentro de consultas SELECT**, cláusulas WHERE, HAVING, etc.
- Permiten **encapsular lógica matemática o de negocio** para evitar repetir código.
- Facilitan la **legibilidad y mantenimiento** del código, ya que el cálculo está centralizado en una sola función.

Ejemplo conceptual:

Si necesitas calcular el descuento de un producto en varias consultas, una función te permitirá hacerlo automáticamente sin repetir la fórmula cada vez.

2. Manejo de variables en T-SQL y datos dinámicos

Las **variables en T-SQL** permiten **almacenar valores temporales** durante la ejecución de una consulta o procedimiento. Son útiles cuando los valores cambian según las condiciones del proceso.

Ventajas del uso de variables:

- Permiten **manejar datos dinámicos**, por ejemplo, valores de entrada del usuario, fechas actuales o resultados intermedios.
- Ayudan a **controlar el flujo de ejecución** mediante condicionales (IF, CASE, etc.).
- Mejoran la **flexibilidad** de las consultas, ya que el mismo bloque de código puede adaptarse a distintos escenarios.

Ejemplo teórico:

Si se desea consultar las ventas de un año determinado, se puede guardar ese año en una variable (@Año) y utilizarla dentro de la consulta. Así, solo cambiando el valor de la variable, la misma consulta se ajusta dinámicamente.

3. Uso de cursores en T-SQL y su ventaja frente a JOIN o SELECT

Los **cursores** en T-SQL permiten **recorrer los resultados de una consulta fila por fila**, lo cual es útil cuando se requiere realizar operaciones individualizadas sobre cada registro.

Aunque las instrucciones **JOIN o SELECT** son más eficientes para manejar conjuntos de datos grandes, los cursores son ventajosos cuando:

- Es necesario **procesar registros uno a uno** (por ejemplo, cálculos secuenciales o validaciones personalizadas).
- Se requiere **aplicar lógica condicional compleja** que no puede resolverse fácilmente con una sola instrucción SQL.
- Se deben **ejecutar acciones distintas por registro**, como enviar notificaciones o actualizar tablas relacionadas según condiciones específicas.

Ejemplo teórico:

Si se necesita recorrer una lista de clientes y enviar un correo personalizado a cada uno según su historial de compras, un cursor permite hacerlo paso a paso, procesando cada cliente individualmente.

Conclusión general

El uso de **funciones, procedimientos almacenados, variables, condicionales y cursores en T-SQL** permite:

- **Automatizar procesos repetitivos**, reduciendo errores humanos.
- **Optimizar el rendimiento**, centralizando la lógica en el servidor y evitando múltiples consultas desde la aplicación.
- **Personalizar la manipulación y consulta de datos**, adaptando el comportamiento del código a las necesidades del negocio.

En conjunto, estas herramientas proporcionan una **base sólida para el desarrollo eficiente, modular y seguro** dentro de **SQL Server**, facilitando la administración y explotación de la información.