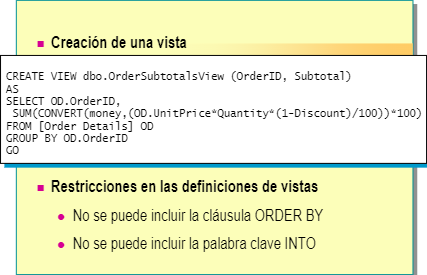
**Vistas**

Definición:

Es una tabla virtual basada en el resultado de una consulta SQL. Aunque una vista no almacena datos en sí misma, su contenido es el resultado de una consulta que se puede tratar como una tabla normal.



Usos y ventajas de las vistas:

* Centrar el interés en los datos de los usuarios
* Centrarse sólo en los datos importantes o adecuados
* Limitar el acceso a los datos confidenciales
* Enmascarar la complejidad de la base de datos
* Ocultar el diseño de la base de datos compleja
* Simplificar las consultas complejas, incluyendo las consultas distribuidas a datos heterogéneos
* Simplificar la administración de los permisos de usuario
* Mejorar el rendimiento
* Organizar los datos para exportarse a otras aplicaciones

Desventajas de las vistas:

Restricciones en las definiciones de vistas:

Al crear vistas, tenga presentes las restricciones siguientes:

* La instrucción CREATE VIEW no puede incluir las cláusulas COMPUTE o COMPUTE BY. La instrucción CREATE VIEW no puede incluir la palabra clave INTO.
* La instrucción CREATE VIEW puede incluir la cláusula ORDER BY, sólo si se utiliza la palabra clave TOP.
* Las vistas no pueden hacer referencia a tablas temporales.
* Las vistas no pueden hacer referencia a más de 1.024 columnas.
* La instrucción CREATE VIEW no puede combinarse con otras instrucciones de Transact-SQL en un mismo lote.

**Stored Procedure:**

Un procedimiento almacenado es una colección con nombre de instrucciones de Transact-SQL que se almacena en el servidor. Los procedimientos almacenados son un método para encapsular tareas repetitivas. Admiten variables declaradas por el usuario, ejecución condicional y otras características de programación muy eficaces.

Usos y ventajas de los Stored Procedure:

* Compartir la lógica de la aplicación
* Exposición de los detalles de las tablas de la base de datos
* Proporcionar mecanismos de seguridad
* Mejorar el rendimiento Reducir el tráfico de red

Desventajas de los Stored Procedure:

* Portabilidad
* Depuración
* Complejidad
* Pruebas
* Versionamiento

**Funciones:**

Las funciones en SQL son rutinas almacenadas que aceptan parámetros, realizan una acción, como una operación compleja, y devuelven el resultado de esa acción. Al igual que los procedimientos almacenados, las funciones son formas de encapsular código para reutilización y eficiencia, pero a diferencia de los procedimientos almacenados, las funciones siempre devuelven un valor.

Usos y ventajas de las funciones:

* Reutilización de código
* Simplicidad
* Modularidad
* Flexibilidad

Desventajas de las funciones:

* Rendimiento
* Portabilidad
* Depuración
* Limitaciones de las funciones

**Triggers:**

Un desencadenador es un procedimiento almacenado que se ejecuta cuando se modifican los datos de una tabla determinada. Los desencadenadores se suelen crear para exigir integridad referencial o coherencia entre datos relacionados de forma lógica en diferentes tablas. Como los usuarios no pueden evitar los desencadenadores, éstos se pueden utilizar para exigir reglas de empresa complejas que mantengan la integridad de los datos.

Usos y ventajas de los Triggers:

Los desencadenadores son adecuados para mantener la integridad de los datos en el nivel inferior, pero no para obtener resultados de consultas. La ventaja principal de los desencadenadores consiste en que pueden contener lógica compleja de proceso. Los desencadenadores pueden hacer cambios en cascada en tablas relacionadas de una base de datos, exigir integridad de datos más compleja que una restricción CHECK, definir mensajes de error personalizados, mantener datos no normalizados y comparar el estado de los datos antes y después de su modificación.

Desventajas de los Triggers:

* Rendimiento
* Complejidad
* Mantenimiento
* Depuración
* Comportamiento inesperado