Universidad del Valle de Guatemala

Facultad de Ingeniería

Departamento de Ciencias de la Computación

Cristian Túnchez - 231359

Javier Benítez - 23405

Base de Datos 1

Ejercicio 5: Triggers

Instrucciones: Diseñen la base de datos que solucione el problema planteado y respondan las siguientes preguntas.

Diseño de la Base de Datos

Figura 1. Diseño de una Base de Datos que modela una Tienda de Decoración Temática

Entidad	Atributo	Tipo de Dato
Usuario	id_usuario (pk)	serial
	nombre	varchar(100)
	telefono	varchar(20)
	correo	varchar(100)
Categoria	id_categoria (pk)	serial
	nombre	varchar(100)
Tematica	id_tematica (pk)	serial
	nombre	varchar(100)
Articulo	id_articulo (pk)	serial
	nombre	varchar(100)
	precio	decimal(10,2)
	stock	integer
	id_categoria (fk)	integer
	id_tematica (fk)	integer
Transaccion	id_transaccion (pk)	serial
	id_usuario (fk)	integer
	fecha	date
	total	decimal(10,2)
	estado	varchar(50)
DetalleTransaccion	id_detalle (pk)	serial
	id_transaccion (fk)	integer
	id_articulo (fk)	integer
	cantidad	integer
	subtotal	decimal(10,2)

PREGUNTAS

1. ¿Cómo garantizaron la integridad de los datos?

   Garantizamos la integridad principalmente de estas maneras:

   - Usamos claves foráneas para asegurar que no haya referencias a datos que no existen.
   - Implementamos varios triggers BEFORE para validar datos antes de insertarlos (como evitar stock negativo).
   - Creamos triggers AFTER para mantener registros de auditoría cuando ocurren cambios importantes.
   - Añadimos validaciones que impiden borrar datos importantes (como el trigger que evita eliminar detalles de transacciones finalizadas).
   - Definimos restricciones como UNIQUE para el correo de usuario para evitar duplicados.

   Los triggers nos ayudaron mucho porque permiten validar condiciones complejas que no se pueden hacer solo con restricciones básicas.

2. Si el número de productos en la tienda aumentara significativamente, ¿qué modificaciones harían para garantizar el rendimiento?

   Si tuviéramos muchos más productos, haríamos estos cambios:

   - Agregar índices a las columnas que más se buscan (nombre, precio, categoría).
   - Crear índices compuestos para cuando filtramos por categoría y temática juntas.

- Revisar los triggers y hacerlos más eficientes, eliminando los que no sean tan necesarios.
- Particionar la tabla de Artículos por categorías para que las búsquedas sean más rápidas.
- Usar vistas materializadas para las consultas más frecuentes.
- Implementar algún tipo de caché para datos que no cambian mucho.

Las búsquedas por nombre y filtrado por categoría serían las más afectadas con mayor volumen, así que pondríamos más atención a esas partes.

3. ¿Qué pruebas pueden realizar para para verificar el rendimiento óptimo de la base de datos?

   Para comprobar si la base de datos funciona bien, nosotros haríamos:

   - Pruebas simulando varios usuarios comprando a la vez para ver si aguanta.
   - Usar EXPLAIN ANALYZE para ver si las consultas están usando los índices correctamente.
   - Medir cuánto tardan las operaciones más comunes (agregar productos, actualizar stock, etc.).
   - Probar específicamente los triggers para ver si hacen más lenta la base cuando hay muchas operaciones.
   - Revisar si hay bloqueos cuando varios usuarios intentan modificar los mismos datos.
   - Comprobar si las tablas de auditoría crecen demasiado rápido.

   Lo más importante sería simular un día con muchas ventas para ver dónde empiezan los problemas.

4. ¿Es su diseño escalable? ¿Por qué? Si la respuesta es no también respondan ¿qué están haciendo con su vida?

   El diseño tiene sus puntos fuertes y débiles para escalar. Entre “lo bueno” tenemos:

   - La estructura de tablas está bien normalizada y separada por funciones.
   - Las relaciones entre tablas están bien definidas con claves foráneas.
   - La separación de categorías y temáticas facilita agregar nuevos tipos de productos.

   Y entre “lo malo” está:

   - Tantos triggers pueden hacer más lentas las operaciones cuando hay muchos datos.
   - La tabla de Transacción podría saturarse en temporadas altas de ventas.
   - Las tablas de auditoría e historial van a crecer sin control si no las limpiamos.

   En general, funcionaría bien para una tienda mediana, pero necesitaríamos ajustes importantes para una tienda muy grande con miles de ventas diarias.

5. ¿Qué mejora pudieran hacer al sistema para mejorar su rendimiento?

   Para que funcione mejor, nosotros haríamos:

   - Agregar buenos índices en las columnas más usadas para búsquedas.
   - Simplificar algunos triggers o convertirlos en constraints cuando sea posible.
   - Mover transacciones viejas a tablas de historial (tipo archivado).
   - Implementar índices de texto para búsquedas por nombre de producto.
   - Configurar mantenimiento automático (VACUUM) para limpiar espacio no usado.
   - Mejorar las consultas para que sean más directas y usen bien los índices.
   - Añadir un sistema de caché para las consultas más frecuentes.