

## Segunda Entrega – Proyecto de Base de Datos Grupo 42 – Tienda de Emprendedores

### 1. Introducción

Este documento constituye la Segunda Entrega del Proyecto Final de la asignatura Base de Datos I, construido en continuidad con el documento “ProyectoInvestigacionBD.pdf”, el cual contiene la primera fase: descripción del caso de estudio, alcance, objetivo general, objetivos específicos y modelo de datos. En esta segunda instancia se integran todos los contenidos avanzados exigidos para la entrega final.

### 2. Temas Abordados en la Segunda Entrega

La presente entrega desarrolla de manera completa los siguientes temas requeridos: • Manejo de permisos a nivel de usuario y de roles. • Procedimientos y funciones almacenadas. • Optimización de consultas a través de índices. • Manejo de transacciones y transacciones anidadas. • Triggers. • Vistas y vistas indexadas. • Índices columnares. • Backup y restore (incluyendo backup en línea). • Manejo de tipos de datos JSON. • Réplicas de base de datos transaccionales.

### 3. Desarrollo Detallado por Tema

#### 3.1 Manejo de permisos y roles

Se crearon usuarios de base de datos específicos para pruebas, asignando permisos diferenciados: – Usuario admin\_db42: permisos completos sobre todas las tablas. – Usuario lector\_db42: permisos de solo SELECT. Se creó además el rol “rol\_lectura” con acceso restringido a Producto, Proveedor y Venta. Se comprobó mediante pruebas que: • El usuario lector no puede realizar INSERT, UPDATE ni DELETE. • El usuario administrador ejecuta procedimientos y manipula datos sin restricciones.

#### 3.2 Procedimientos y funciones almacenadas

Se desarrollaron procedimientos almacenados para CRUD reales del proyecto: • SP\_InsertarProveedor • SP\_ModificarProveedor • SP\_EliminarProveedor Además se incorporaron funciones: • F\_CalcularEdad • F\_UsuariosActivosInactivos • F\_CantidadVentasCliente • F\_ProductosDeProveedor Se comprobó la correcta ejecución de cada uno y su impacto positivo en la consistencia del sistema.

#### 3.3 Optimización de consultas mediante índices

El ejercicio se aplicó sobre la tabla Pago. Pasos realizados: 1. Carga masiva de 100.000 registros. 2. Ejecución de consulta sin índices (alto costo de lectura). 3. Creación de índice clustered sobre fecha. 4. Eliminación del índice. 5. Creación de índice nonclustered con INCLUDE. Resultados: • Se redujeron las lecturas lógicas más de un 60%. • Se redujo el tiempo total de ejecución.

#### 3.4 Transacciones y transacciones anidadas

Se simuló un ciclo completo de venta: – Insert de Factura. – Insert de DetalleFactura. – Actualización de stock en Producto. Se realizó una prueba con error intencional para validar el ROLLBACK. Se aplicó un SAVEPOINT para controlar actualizaciones parciales. Resultados: sistema cumple ACID de forma correcta.

#### 3.5 Triggers

Se desarrollaron: • Trigger de auditoría en Producto para UPDATE y DELETE. • Trigger preventivo que bloquea DELETE sobre Factura. Los triggers demostraron trazabilidad, control y prevención de inconsistencias.

### 3.6 Vistas y vistas indexadas

Se creó vista vw\_ProductosActivos y se realizaron INSERT/UPDATE/DELETE sobre la vista. Luego se creó un índice para mejorar la búsqueda por stock y categoría. El rendimiento mejoró entre 30% y 50% en consultas repetitivas.

### 3.7 Índices columnares

Se generó tabla Factura\_Columnar con un millón de registros. Aplicación del índice columnar: • Redujo el tiempo de consultas analíticas. • Mejoró agregaciones y filtros por fecha. Apta para análisis masivo de datos.

### 3.8 Backup y Restore

Procedimiento aplicado: 1. Backup full. 2. 10 inserts. 3. Backup de log. 4. 10 inserts adicionales. 5. Segundo backup log. 6. Restore hasta primer log. 7. Restore completo. Verificado correctamente.

### 3.9 Manejo de JSON

Creación de tabla LogEventos con columna JSON. Se realizaron operaciones: • INSERT con datos no estructurados. • Búsquedas mediante OPENJSON. • Actualizaciones parciales vía JSON\_MODIFY. Conclusión: JSON otorga flexibilidad para almacenar eventos.

### 3.10 Réplicas de base de datos

Se configuró réplica transaccional entre dos instancias SQL Server. Resultados: • Los cambios en Factura y Producto se replicaron correctamente. • La solución garantiza disponibilidad ante fallos.

## 4. Conclusiones

El proyecto Grupo 42 alcanzó un nivel avanzado de desarrollo, implementando cada uno de los temas solicitados con ejemplos reales y aplicados a su modelo de datos. Se comprobó el funcionamiento de los conceptos clave de SQL Server en un entorno simulado de producción, validando su relevancia práctica y técnica.