

Informe de Evaluación del Módulo

Sistema de Gestión de Inventario en SQL

1. Objetivo

El objetivo de esta actividad fue diseñar y desarrollar un sistema de gestión de inventario utilizando una base de datos relacional. El sistema debía contemplar la administración de productos, proveedores y transacciones de compra/venta, asegurando la integridad de los datos, el control de inventario y la posibilidad de realizar consultas simples y complejas para la toma de decisiones.

2. Diseño del Modelo de Datos

Se inició con un **Modelo Entidad-Relación (ERD)**, que permitió identificar las tres entidades principales:

- **Productos:** información básica (nombre, descripción, precio y stock).
- **Proveedores:** datos de identificación y contacto.
- **Transacciones:** registro de compras y ventas con fecha, cantidad y referencia a producto y proveedor.

El paso siguiente fue la traducción al **modelo relacional**, definiendo claves primarias (PK), claves foráneas (FK) y restricciones de integridad referencial. La relación principal es:

- Cada transacción referencia exactamente un producto y un proveedor.
- Un producto o proveedor puede estar asociado a múltiples transacciones.

3. Implementación de la Base de Datos

La base de datos fue creada en MySQL con el siguiente esquema:

- **Codificación:** utf8mb4 con COLLATE utf8mb4_0900_ai_ci, para garantizar soporte completo de caracteres y comparaciones insensibles a mayúsculas y acentos.
- **Modo estricto:** STRICT_TRANS_TABLES, ERROR_FOR_DIVISION_BY_ZERO y NO_ENGINE_SUBSTITUTION, con el fin de evitar inserciones inválidas o conversiones silenciosas.

Principales características:

- **Tipos de datos adecuados:** DECIMAL(10,2) para precios, DATETIME para fechas, VARCHAR en campos de texto.
- **Restricciones:**
 - CHECK para evitar precios negativos y stock inferior a cero.
 - UNIQUE en el nombre de producto y en el correo electrónico de proveedores.
- **Índices:** creados en las columnas más consultadas (producto_id, proveedor_id, fecha, tipo) para optimizar el rendimiento de las búsquedas.

4. Manipulación y Consultas

Se cargaron **datos de prueba** con proveedores, productos iniciales y compras que actualizan el inventario.

Posteriormente, se implementaron las siguientes operaciones:

4.1 Consultas básicas

- **Listado de productos disponibles en inventario.**
- **Proveedores que han abastecido un producto específico.**
- **Transacciones de una fecha determinada**, considerando que la columna incluye hora (resuelto con rangos BETWEEN).
- **Funciones de agregación** para calcular unidades vendidas y valor total de compras.

4.2 Manipulación de datos (DML)

- Inserción de nuevos productos y proveedores.
- Registro de compras con transacciones atómicas que incrementan el stock.
- Registro de ventas con validación de stock antes de descontar unidades, asegurando consistencia.

4.3 Consultas complejas

- **Ventas de un producto en el mes anterior**, utilizando variables de fecha y GROUP BY.
- **JOINS** (INNER y LEFT) para obtener información integrada de productos, proveedores y transacciones.

- **Subconsultas** para identificar productos que no fueron vendidos en un período determinado.

4.4 Eliminación de registros

Se probó la eliminación de productos sin transacciones asociadas. Cuando un producto tenía registros en transacciones, la restricción ON DELETE RESTRICT impidió su borrado, confirmando la correcta aplicación de integridad referencial.

5. Normalización

El modelo diseñado cumple con la **Tercera Forma Normal (3FN)**:

- 1FN: todos los atributos son atómicos.
- 2FN: no existen dependencias parciales, cada campo depende directamente de la PK.
- 3FN: no hay dependencias transitivas entre atributos no clave.

Se discutió la utilidad de una posible **desnormalización** (ejemplo: almacenar el último precio de compra en la tabla productos) para acelerar reportes frecuentes, asumiendo mayor complejidad en la actualización de datos.

6. Resultados

El sistema de gestión de inventario desarrollado permite:

- Mantener la trazabilidad de compras y ventas.
- Consultar información de manera flexible y eficiente.
- Asegurar la integridad de los datos mediante FKs, restricciones y transacciones.
- Responder preguntas de gestión, como volumen de ventas en un período, proveedores asociados a un producto o productos sin rotación.

7. Conclusiones

El desarrollo de este proyecto permitió aplicar en la práctica conceptos clave de bases de datos relacionales:

- **Modelado conceptual y lógico** (ERD → modelo relacional).

- **Creación de tablas con restricciones y FKs.**
- **Uso de SQL para DDL, DML y consultas avanzadas.**
- **Transacciones para garantizar atomicidad y consistencia.**
- **Normalización como herramienta para eliminar redundancias.**

En síntesis, el sistema cumple con los objetivos planteados, demostrando cómo un diseño estructurado de base de datos mejora la eficiencia en la gestión de inventario y reduce errores en la administración de productos y proveedores.