

✓ **Hecho:** Ver

Por hacer: Hacer un envío

Evaluación del módulo

Objetivo

El objetivo de esta actividad es que puedas diseñar y desarrollar un sistema de gestión de inventario para una empresa que utiliza una base de datos relacional para almacenar y consultar información sobre productos, proveedores y transacciones. Además, aprenderás cómo manejar transacciones, restricciones de integridad referencial, y consultas complejas utilizando SQL.

Contexto

En una empresa de ventas, es necesario gestionar un inventario de productos que se van almacenando en diferentes proveedores, y a su vez, realizar transacciones de compra y venta. Para esto, se utiliza una base de datos relacional (RDBMS) para organizar toda la información de manera estructurada. El sistema debe permitir agregar productos, actualizar el inventario, registrar compras y ventas, y consultar los datos de manera eficiente.

Requisitos del Proyecto

1. Diseño del Modelo Relacional

- Crea un modelo de datos para representar la siguiente información:
 - **Productos:** nombre, descripción, precio, cantidad en inventario.
 - **Proveedores:** nombre, dirección, teléfono, email.
 - **Transacciones:** tipo (compra o venta), fecha, cantidad, id de producto, id de proveedor.
- Utiliza el modelo **Entidad-Relación** (ER) para abstraer las entidades y sus relaciones, y luego tradúcelas al modelo relacional.
- Asegúrate de identificar claves primarias y foráneas para establecer la integridad referencial entre las tablas.

2. Creación de la Base de Datos y Tablas

- Utiliza SQL para crear las tablas **productos**, **proveedores** y **transacciones** en la base de datos.
- Define las restricciones de **nulidad**, **llaves primarias** y **llaves foráneas** para garantizar la integridad de los datos.
- Establece el tipo de dato adecuado para cada atributo (por ejemplo, **VARCHAR**, **INT**, **DECIMAL**).

3. Consultas Básicas

- Realiza consultas básicas utilizando el lenguaje SQL:
 - Recupera todos los productos disponibles en el inventario.
 - Recupera todos los proveedores que suministran productos específicos.
 - Consulta las transacciones realizadas en una fecha específica.
 - Realiza consultas de selección con funciones de agrupación, como **COUNT()** y **SUM()**, para calcular el número total de productos vendidos o el valor total de las compras.

4. Manipulación de Datos (DML)

- Inserta datos en las tablas **productos**, **proveedores** y **transacciones**.
- Actualiza la cantidad de inventario de un producto después de una venta o compra.
- Elimina un producto de la base de datos si ya no está disponible.
- Asegúrate de aplicar **integridad referencial** al actualizar o eliminar registros relacionados.

5. Transacciones SQL

- Realiza una transacción para registrar una compra de productos. Utiliza el comando **BEGIN TRANSACTION**, **COMMIT** y **ROLLBACK** para asegurar que los cambios se apliquen correctamente.

- Asegúrate de que los cambios en la cantidad de inventario y las transacciones se realicen de forma atómica.
- Utiliza el modo **AUTOCOMMIT** para manejar operaciones individuales si es necesario.

6. Consultas Complejas

- Realiza una consulta que recupere el total de ventas de un producto durante el mes anterior.
- Utiliza **JOINS** (INNER, LEFT) para obtener información relacionada entre las tablas **productos**, **proveedores** y **transacciones**.
- Implementa una consulta con subconsultas (subqueries) para obtener productos que no se han vendido durante un período determinado.

7. Normalización y Desnormalización

- Asegúrate de que las tablas estén **normalizadas** hasta la tercera forma normal (3NF) para evitar redundancias y asegurar la integridad de los datos.
- Si es necesario, discute en tu informe los casos en los que la desnormalización podría ser útil para mejorar el rendimiento de las consultas.

8. Manejo de Excepciones y Restricciones

- Implementa **restricciones** en los campos de las tablas para garantizar que los datos ingresados sean válidos (por ejemplo, asegurando que la cantidad de inventario no sea negativa o que los precios sean mayores que cero).
- Utiliza el comando **TRY/CATCH** para manejar excepciones en caso de errores durante las transacciones.

9. Documentación

- Documenta el proceso de creación del modelo de datos y las decisiones tomadas al diseñar las tablas, restricciones y relaciones entre entidades.
- Incluye una breve explicación de la normalización aplicada en el modelo de datos y su impacto en la estructura de la base de datos.
- Presenta ejemplos de las consultas SQL utilizadas y explica cómo funcionan.

Agregar entrega

Estado de la entrega

Estado de la entrega	Todavía no se han realizado envíos
Estado de la calificación	Sin calificar
Última modificación	-

Contacta

✉ Correo electrónico : contacto@skillnest.com

Copyright © 2017 -Desarrollado por LMSACE.com. Desarrollado por [Moodle](https://moodle.org/)