

Ejercicio grupal

Contexto

Un grupo de 3 amigos ha creado un emprendimiento llamado “**RECITRONIC**”, dedicado al reciclaje de electrónicos. Reciben pedidos para retirar artículos electrónicos en desuso, pero no tienen un sistema de gestión para coordinar las citas de manera eficiente. El proceso se lleva a cabo manualmente en un cuaderno, lo que ha ocasionado sobrecarga de trabajo, citas duplicadas y falta de control sobre los artículos reciclados.

Requerimientos de la Actividad

1. Base de Datos y Tablas:

- Crear las siguientes tablas en una base de datos:
 - **Clientes:** Para almacenar la información de los clientes.
 - **Artículos:** Para registrar los artículos electrónicos que los clientes desean reciclar.
 - **Citas:** Para registrar las citas agendadas para el retiro de los artículos.
 - **Pagos:** Para registrar los pagos realizados por los servicios de reciclaje.

2. Manipulación de Datos (DML):

- **Ingresar Información:**
 - Insertar registros de **clientes** que solicitan el servicio.
 - Insertar registros de **artículos reciclados** por cada cliente.
 - Insertar citas para agendar los retiros de los artículos reciclados.
 - Insertar pagos realizados por los clientes.
- **Actualizar Información:**
 - Actualizar una cita para un cliente, cambiando la fecha de la cita si se presenta un conflicto de horarios.
 - Actualizar el estado de un artículo reciclado (por ejemplo, de “pendiente” a “reciclado”).
- **Eliminar Información:**
 - Eliminar registros de **artículos reciclados** que hayan sido removidos por error.
 - Eliminar **citas** que hayan sido canceladas.

3. Restricciones y Transacciones:

- Utilizar restricciones para garantizar la **integridad referencial** entre las tablas (por ejemplo, no se debe poder crear una cita si el cliente no existe en la base de datos).
- Utilizar **secuencias** para asignar automáticamente un identificador único a cada cita, cliente y pago.
- Utilizar **transacciones** para realizar operaciones de actualización de manera segura (si una operación falla, todas las operaciones deben revertirse).
- Asegurar la **atomicidad, consistencia, aislamiento y durabilidad** (ACID) de las transacciones.
- Realizar un ejemplo de **confirmación de transacción** y **rollback** (si alguna parte del proceso falla, revertir todas las operaciones).

Pasos a Seguir

1. Crear la Base de Datos y Tablas:

- Diseñar la base de datos y crear las tablas correspondientes a **Clientes, Artículos, Citas, y Pagos**.
- Establecer claves primarias, claves foráneas y restricciones necesarias para garantizar la integridad de los datos.

2. Realizar Manipulaciones de Datos con SQL:

- Insertar registros de clientes, artículos reciclados, citas y pagos.



- Actualizar información sobre las citas y el estado de los artículos reciclados.
- Eliminar registros de artículos y citas cuando sea necesario.

3. Demostrar el Uso de Transacciones:

- Crear una transacción que incluya insertar datos, actualizar registros y eliminar registros. Asegurarse de que las operaciones sean atómicas, es decir, que todas las operaciones se completen correctamente o se reviertan en caso de error.

4. Implementar Secuencias y Restricciones:

- Usar secuencias para generar identificadores únicos automáticamente.
- Definir restricciones de integridad referencial entre las tablas para garantizar que no se puedan insertar o actualizar datos inválidos.

5. Ejemplo de Rollback y Commit:

- Realizar un ejemplo práctico donde se demuestre cómo confirmar (commit) o revertir (rollback) una transacción en caso de error.

Estructura de la Base de Datos

Clientes:

- **id_cliente** (INT, PK, AUTO_INCREMENT)
- **nombre** (VARCHAR)
- **telefono** (VARCHAR)
- **direccion** (VARCHAR)

Artículos:

- **id_articulo** (INT, PK, AUTO_INCREMENT)
- **id_cliente** (INT, FK, REFERENCIA a Clientes)
- **tipo_articulo** (VARCHAR)
- **estado** (VARCHAR)

Citas:

- **id_cita** (INT, PK, AUTO_INCREMENT)
- **id_cliente** (INT, FK, REFERENCIA a Clientes)
- **fecha_hora** (DATETIME)

Pagos:

- **id_pago** (INT, PK, AUTO_INCREMENT)
- **id_cliente** (INT, FK, REFERENCIA a Clientes)
- **monto** (DECIMAL)
- **fecha_pago** (DATETIME)

Componentes para Evaluar

- **Uso correcto de DML:** Insertar, actualizar y borrar registros correctamente.
- **Implementación de secuencias:** Asignación de identificadores automáticos para registros.
- **Integridad referencial:** Uso adecuado de claves primarias y foráneas para mantener la consistencia de los datos.
- **Transacciones SQL:** Uso de transacciones para asegurar la atomicidad de las operaciones.
- **Restricciones en las tablas:** Uso de restricciones como **NOT NULL**, **FOREIGN KEY**, etc., para mantener la integridad de los datos.
- **Confirmación de Transacciones:** Uso de **COMMIT** y **ROLLBACK** correctamente.

Duración: 1 jornada de clase

Ejecución: Grupal



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

