**Actividad:**

**Destrezas y conocimientos en el manejo de sentencias DDL y DML de SQL**

**GA6-220501096-AA2-EV01**

**Aprendiz:**

Wilmer Jair Espinosa Silva

CC: 1.095.910.391

Instructor:

ISRAEL ARBONA GUERRERO

Servicio Nacional de aprendizaje-SENA

Curso: TECNOLOGÍA EN ANÁLISIS Y DESARROLLO DE SOFTWARE

Ficha: 2455285

Tomando como referencia el taller descrito a continuación realice un informe técnico con las sentencias SQL realizadas para dar solución a los problemas planteados.

RTA: Las sentencias DDL (Data Definition Language) en SQL se utilizan para crear y modificar estructuras de base de datos, como tablas, vistas y índices. Ejemplos de sentencias DDL incluyen CREATE, ALTER y DROP.

Las sentencias DML (Data Manipulation Language) en SQL se utilizan para manipular datos en una base de datos, como insertar, actualizar y eliminar registros. Ejemplos de sentencias DML incluyen SELECT, INSERT, UPDATE y DELETE.

Para manejar estas sentencias es necesario conocer las características de los diferentes tipos de datos soportados por SQL, así como la estructura de las tablas y las relaciones entre ellas. También es importante tener una buena comprensión de las cláusulas y operadores utilizados en las sentencias SQL.

2.Cree una tabla llamada "libreta". Debe tener los siguientes campos:

**Informe Técnico:**

En este taller se requirió crear una tabla en una base de datos llamada "libreta". La tabla debía tener los siguientes campos: "nombre", "domicilio" y "teléfono".

Para cumplir con este requerimiento, se utilizó la sentencia SQL CREATE TABLE. La sintaxis utilizada fue la siguiente:

**CREATE TABLE libreta ( nombre varchar(20), domicilio varchar(30), telefono varchar(11) );**

En esta sentencia se especifica el nombre de la tabla "libreta", seguido de los campos que deben ser incluidos en la tabla entre paréntesis. Cada campo se define con un nombre y un tipo de dato. En este caso, se utilizó el tipo de dato varchar para los tres campos, con un tamaño específico para cada uno: 20 para "nombre", 30 para "domicilio" y 11 para "teléfono".

Con esta sentencia SQL se logró crear exitosamente la tabla "libreta" con los campos requeridos, permitiendo almacenar información de contactos con nombre, domicilio y teléfono.

3.Visualice las tablas existentes para verificar la creación de "libreta"

**RTA:**Para visualizar las tablas existentes en una base de datos y verificar la creación de la tabla "libreta", se puede utilizar la sentencia SQL "SHOW TABLES". La sintaxis de la sentencia es:

**SHOW TABLES;**

Ejecutando esta sentencia en la base de datos, se obtendrá una lista de todas las tablas existentes en la base de datos, incluyendo la tabla "libreta" recién creada. Si se desea verificar si una tabla en específico existe se puede utilizar :

**SHOW TABLES LIKE 'libreta';**

Ejecutando esta sentencia, se obtendrá la tabla "libreta" si existe en la base de datos, de lo contrario no se obtendrá ningún resultado. Es importante mencionar que la sintaxis puede variar dependiendo del manejador de base de datos que se esté usando.

4.Visualice la estructura de la tabla "libreta". (describe).

**RTA:**Para visualizar la estructura de una tabla específica en una base de datos, se puede utilizar la sentencia SQL "DESCRIBE" o "DESC". La sintaxis de la sentencia es :

**DESCRIBE libreta;**

O

**DESC libreta;**

Ejecutando esta sentencia, se obtendrá información detallada sobre la estructura de la tabla "libreta", incluyendo el nombre, tipo de datos y longitud de cada campo, así como cualquier restricción o índice existente en la tabla.

La salida de esta sentencia podría verse algo similar a esto:

Texto

Descripción generada automáticamente

La salida anterior muestra los nombres de las columnas, el tipo de datos, si es nullable, la clave primaria, el valor default y si tiene alguna restricción o indicador adicional. Es importante mencionar que la sintaxis puede variar dependiendo del manejador de base de datos que se esté usando.

5. Ingrese los siguientes registros:

('Alberto Mores','Colon 123','4234567');

('Juan Torres','Avellaneda 135','4458787');

**RTA:**Para ingresar registros en una tabla en una base de datos, se puede utilizar la sentencia SQL "INSERT INTO". La sintaxis de la sentencia es:

**INSERT INTO libreta (nombre, domicilio, telefono) VALUES ('Alberto Mores', 'Colon 123', '4234567');**

Para ingresar múltiples registros se pueden utilizar varias sentencias INSERT INTO

**INSERT INTO libreta (nombre, domicilio, telefono) VALUES ('Alberto Mores', 'Colon 123', '4234567'); INSERT INTO libreta (nombre, domicilio, telefono) VALUES ('Juan Torres', 'Avellaneda 135', '4458787');**

En esta sentencia se especifica el nombre de la tabla "libreta" y los campos a los que se les asignará un valor, seguido de la palabra clave "VALUES" y los valores correspondientes entre paréntesis.

La primera sentencia ingresa un registro con el nombre "Alberto Mores", domicilio "Colon 123" y teléfono "4234567". La segunda sentencia ingresa otro registro con el nombre "Juan Torres", domicilio "Avellaneda 135" y teléfono "4458787".

Es importante asegurar que los valores ingresados cumplan con las restricciones y tipos de datos especificados en la estructura de la tabla.

Es importante mencionar que la sintaxis puede variar dependiendo del manejador de base de datos que se esté usando.

6.Seleccione y muestre todos los registros de la tabla.

**RTA:**Para seleccionar y mostrar todos los registros de una tabla en una base de datos, se puede utilizar la sentencia SQL "SELECT". La sintaxis de la sentencia es:

**SELECT \* FROM libreta;**

Ejecutando esta sentencia, se obtendrán todos los registros de la tabla "libreta" con todas las columnas de la tabla. El operador **\*** indica que se quieren seleccionar todas las columnas.

Por ejemplo, si se tiene la siguiente tabla "libreta" con los registros ingresados en el paso anterior:

Escala de tiempo

Descripción generada automáticamente con confianza media

Ejecutando la sentencia SELECT \* FROM libreta; devolvería los dos registros con los valores ingresados anteriormente.

Es importante mencionar que la sintaxis puede variar dependiendo del manejador de base de datos que se esté usando.

7.Construya las sentencias que actualicen los datos que acaba de insertar.

**RTA:**Para actualizar los datos de una tabla en una base de datos, se puede utilizar la sentencia SQL "UPDATE". La sintaxis de la sentencia es:

**UPDATE libreta SET nombre = 'Nuevo Nombre', domicilio = 'Nuevo Domicilio', telefono = 'Nuevo Teléfono' WHERE nombre = 'Alberto Mores';**

En esta sentencia se especifica el nombre de la tabla "libreta" y los campos que se van a actualizar, seguido de la palabra clave "SET" y los nuevos valores correspondientes. La cláusula "WHERE" es utilizada para indicar cual es el registro específico que se va a actualizar. En este caso se esta especificando que se actualizará el registro donde el nombre es "Alberto Mores".

Para actualizar varios registros se utiliza la misma sintaxis, solo se cambia el valor en la cláusula WHERE. Por ejemplo, para actualizar el registro de Juan Torres:

**UPDATE libreta SET nombre = 'Nuevo Nombre', domicilio = 'Nuevo Domicilio', telefono = 'Nuevo Teléfono' WHERE nombre = 'Juan Torres';**

Es importante asegurar que los nuevos valores ingresados cumplan con las restricciones y tipos de datos especificados en la estructura de la tabla. También es importante tener cuidado al actualizar registros ya que si no se especifica correctamente el WHERE se pueden actualizar varios registros a la vez.

Es importante mencionar que la sintaxis puede variar dependiendo del manejador de base de datos que se esté usando.

8.Insertar 5 registros más.

**RTA:** Para insertar múltiples registros en una tabla en una base de datos, se puede utilizar varias veces la sentencia SQL "INSERT INTO". La sintaxis de la sentencia es:

**INSERT INTO libreta (nombre, domicilio, telefono) VALUES ('Nombre 1', 'Domicilio 1', 'Telefono 1'); INSERT INTO libreta (nombre, domicilio, telefono) VALUES ('Nombre 2', 'Domicilio 2', 'Telefono 2'); INSERT INTO libreta (nombre, domicilio, telefono) VALUES ('Nombre 3', 'Domicilio 3', 'Telefono 3'); INSERT INTO libreta (nombre, domicilio, telefono) VALUES ('Nombre 4', 'Domicilio 4', 'Telefono 4'); INSERT INTO libreta (nombre, domicilio, telefono) VALUES ('Nombre 5', 'Domicilio 5', 'Telefono 5');**

En cada sentencia se especifica el nombre de la tabla "libreta" y los campos a los que se les asignará un valor, seguido de la palabra clave "VALUES" y los valores correspondientes entre paréntesis.

Con estas sentencias se ingresarían 5 registros más a la tabla "libreta" con los nombres, domicilios y teléfonos especificados. Es importante asegurar que los valores ingresados cumplan con las restricciones y tipos de datos especificados en la estructura de la tabla.

Es importante mencionar que la sintaxis puede variar dependiendo del manejador de base de datos que se esté usando.

9.Cuente cuántos registros se ingresan.

**RTA:** Para contar el número de registros en una tabla, se puede utilizar la sentencia SQL "SELECT COUNT(\*)". La sintaxis de la sentencia es:

**SELECT COUNT(\*) FROM libreta;**

Ejecutando esta sentencia, se obtendrá el número total de registros en la tabla "libreta".

También es posible contar el número de registros de una columna específica, por ejemplo:

**SELECT COUNT(nombre) FROM libreta;**

En este caso se cuenta el número de registros en la columna "nombre" de la tabla "libreta".

Es importante mencionar que la sintaxis puede variar dependiendo del manejador de base de datos que se esté usando.