El futuro digital es de todos

MinTIC

UNIVERSIDAD DEL NORTE

Vigilada Mineducación

CICLO II:

Programación Básica en Java

MisionTIC 2022

El futuro digital es de todos

MinTIC

UNIVERSIDAD DEL NORTE

Vigilada Mineducación

Sesión 15:

Introducción a Java

Conexión a Base de Datos

MisiónTIC2022

Objetivos de la sesión

Al finalizar esta sesión estarás en capacidad de:

1. Definir y diseñar una base de datos relacional
2. Construir una base de datos en SQLite de una tabla
3. Manipular la gestión de información en la base de datos construida
4. Explicar y aplicar el concepto de conexión a una base de datos relacional (JDBC)
5. Construir una aplicación con entorno gráfico que conecte a una base de datos relacional y lleve a cabo operaciones sobre esta

¿Qué es una base de datos relacional?

- Una ***base de datos relacional*** es un tipo de base de datos que almacena y proporciona acceso a puntos de datos relacionados entre sí. Las bases de datos relacionales se basan en el modelo relacional, una forma intuitiva y directa de representar datos en tablas.

- En una base de datos relacional, cada fila de la tabla es un registro con un ID único llamado clave. Las columnas de la tabla contienen atributos de los datos, y cada registro generalmente tiene un valor para cada atributo, lo que facilita el establecimiento de las relaciones entre los puntos de datos.

¿Cómo se estructuran las base de datos relacionales?

- El modelo relacional significa que las estructuras lógicas de datos (las tablas de datos, vistas e índices) están separadas de las estructuras físicas de almacenamiento. Esta separación significa que los administradores de bases de datos pueden administrar el almacenamiento físico de datos sin afectar el acceso a esos datos como una estructura lógica. Por ejemplo, cambiar el nombre de un archivo de base de datos no cambia el nombre de las tablas almacenadas en él.

¿Cómo se estructuran las base de datos relacionales?

- La distinción entre lógica y física también se aplica a las operaciones de la base de datos, que son acciones claramente definidas que permiten a las aplicaciones manipular los datos y las estructuras de la base de datos. Las operaciones lógicas permiten que una aplicación especifique el contenido que necesita, mientras que las operaciones físicas determinan cómo se debe acceder a esos datos y luego realizan la tarea.
- Para garantizar que los datos sean siempre precisos y accesibles, las bases de datos relacionales siguen ciertas reglas de integridad. Por ejemplo, una regla de integridad puede especificar que no se permiten filas duplicadas en una tabla, para eliminar la posibilidad de que ingrese información errónea en la base de datos.

SQL

- El lenguaje de consulta estructurado o SQL (Structured Query Language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas.

- Una de sus características es el manejo del álgebra y el cálculo relacional que permiten efectuar consultas con el fin de recuperar de forma sencilla información de interés de bases de datos, así como hacer cambios en ella.

SQL

¿Qué es?

Es un lenguaje declarativo de alto nivel

¿Por qué?

Nos permite el acceso a bases de datos relacionales y operar sobre ellas.

¿Para qué?

Manipular bases de datos desde cualquier lenguaje de programación

SQL = DML + DDL

Lenguaje de Definición de Datos (DDL)

Es un lenguaje de programación para definir estructuras de datos, proporcionado por los sistemas gestores de bases de datos:

- CREATE: se usa para crear una base de datos, tabla, vistas, etc.
- ALTER: se utiliza para modificar la estructura, por ejemplo añadir o borrar columnas de una tabla.
- DROP: con esta sentencia, podemos eliminar los objetos de la estructura, por ejemplo un índice o una secuencia.

SQL =DML+DDL.

Lenguaje de Manipulación de Datos (DML)

Permite a los usuarios introducir datos para posteriormente realizar tareas de consultas o modificación de los datos que contienen las Bases de Datos.

- SELECT: esta sentencia se utiliza para realizar consultas sobre los datos.
- INSERT: con esta instrucción podemos insertar los valores en una base de datos.
- UPDATE: sirve para modificar los valores de uno o varios registros.
- DELETE: se utiliza para eliminar las filas de una tabla.

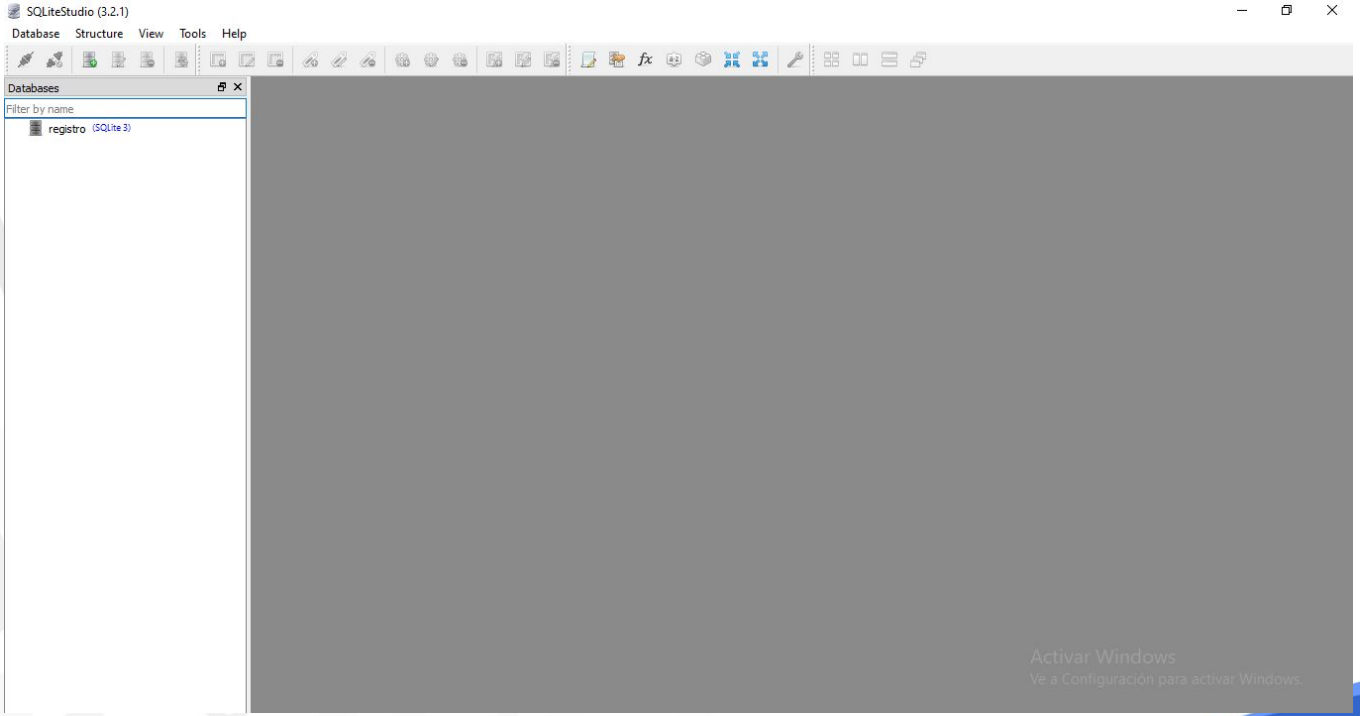
SQLite Studio

SQLiteStudio es una interfaz gráfica para bases de datos SQLite, potente, ligero, rápido e intuitivo. Se trata de una aplicación **multiplataforma**, disponible para los principales sistemas operativos *Windows*, *MacOSX* y **Linux**.

Entre las diferentes características que posee **SQLiteStudio** se pueden destacar las siguientes:

- No necesita instalación, solo hay que desempaquetar y ejecutar.
- Se puede exportar a diversos formatos: CSV, HTML, XML, PDF o JSON.
- Se pueden importar datos desde CSV
- Soporta **unicode**
- Permite configurar el aspecto, colores, fuentes y atajos.
- Código abierto y gratuito.
- Permite el uso de complementos.

Entorno SQLite Studio



Existen diferentes complementos, algunos que vienen por defecto, y otros que se pueden instalar.

De la misma forma, existen tanto complementos oficiales, como desarrollados por terceros.

Diseño y Creación (Create)

Empresa

Personas
P_id: int auto_increment Nombre: text(20) not null Apellidos: text(20) not null Dirección: text(40) not null Ciudad text(10) not null

```
CREATE DATABASE Empresa;

CREATE TABLE Personas(
 P_id integer autoincrement,
 Nombre text(20) not null,
 Apellidos text(20) not null,
 Direccion text(40) not null,
 Ciudad text(10) not null
);
```

Inserción de registros (Insert)

Sintaxis:

```
INSERT INTO nombre_de_la_tabla ( nombre_de_la_columna [, ...] )
    VALUES (valor_de_la_columna [, ...] )
```

Ejemplo:

```
INSERT  INTO Personas(P_id, Nombre, Apellidos, Direccion, Ciudad )
    VALUES ('2345', 'María', 'Pérez)', ‘clle 82 #42e-45', 'Barranquilla');

INSERT  INTO Personas(P_id, Nombre, Apellidos, Direccion, Ciudad )
    VALUES ('5347', 'Paola', 'López)', 'clle 62 #46-85', 'Barranquilla');
```

Mostrar registros (Select)

Sintaxis:

```
SELECT [ ALL | DISTINCT [ ON ( expression [, ...] ) ] ]
    * | expression [ [ AS ] output_name ] [, ...]
    [ FROM from_item [, ...] ]
    [ WHERE condition ]
    [ ORDER BY expression [ ASC | DESC  ]
```

SELECT FROM WHERE ORDER BY: las cláusulas SQL SELECT y SQL FROM son el ABC de las consultas de bases de datos, con estas dos cláusulas se muestran los datos de una tabla.
SELECT: indica que campos mostrar separándolos por comas, para mostrar todos los campos de una tabla se usa el símbolo *.
FROM: indica la tabla o tablas a consultar y obtener los datos.

Mostrar registros (Select)

Ejemplo:
Para la consulta de comprobación se usarán las clausulas SELECT y FROM:

```
SELECT *
   FROM Personas;
```

Los resultados mostrados son:

2345|María|Pérez|clle 82 #42e-45|Barranquilla

5347|Paola|López|clle 62 #46-85|Barranquilla

Eliminación de registros (Delete)

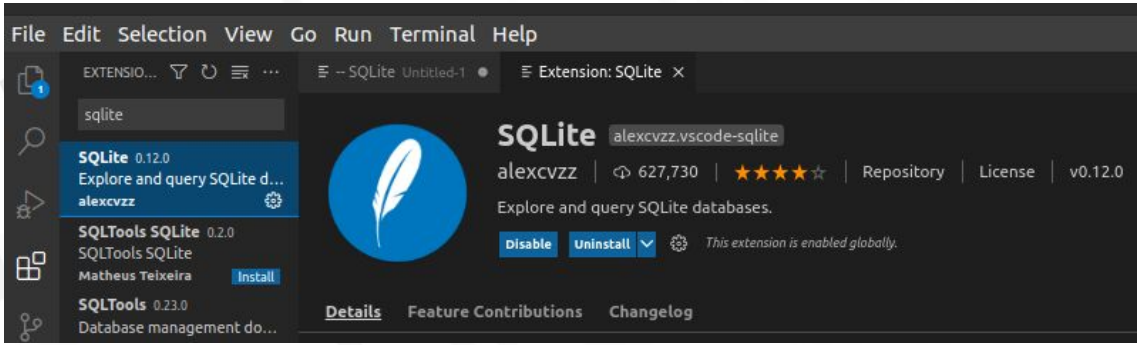
Sintaxis:
DELETE nombre_de_la_tabla
WHERE Expresion

Ejemplo:
DELETE Personas
WHERE P_id = '2345'

El futuro digital es de todos | MinTIC

UNIVERSIDAD DEL NORTE
Vigilada Mineducación

SQLite en Visual Studio Code



Para poder hacer uso del manejo de las bases de dato dentro del editor de código de visual studio, necesitamos instalar el plugin SQLite de alexcvzz.vscode-sqlite.

MisionTIC2022

Conexión con DriverManager y el objeto Connection

La clase **DriverManager** permite obtener objetos **Connection** con la base de datos. Para conectarse es necesario conocer:

- URL de conexión, que incluye:
  - Nombre del host donde está la base de datos.
  - Nombre de la base de datos a usar.
- Nombre del usuario en la base de datos.
- Contraseña del usuario en la base de datos.

El objeto **Connection** representa el contexto de una conexión con la base de datos:

- Permite obtener objetos **Statement** para realizar consultas SQL.
- Permite obtener metadatos acerca de la base de datos (nombres de tablas, etc.)
- Permite gestionar transacciones.

Conexión con DriverManager y el objeto Connection

```
Connection connection;
(...)
try {
    String url = "jdbc:mysql :// hostname/database -name";
    connection =
        DriverManager.getConnection(url , "user", "passwd");
} catch (SQLException ex) {
    connection = null;
    ex.printStackTrace ();
    System.out.println("SQLException: " + ex.getMessage ());
    System.out.println("SQLState: " + ex.getSQLState ());
    System.out.println("VendorError: " + ex.getErrorCode ());
}
```

Objeto Statement

Los objetos Statement permiten realizar consultas SQL en la base de datos:

- Se obtienen a partir de un objeto Connection.
- Disponen de distintos métodos para hacer consultas:
  - Método executeQuery: se utiliza para mostrar datos, normalmente mediante consultas SELECT.
  - Método executeUpdate: se utiliza para insertar, modificar o borrar datos, mediante sentencias INSERT, UPDATE y DELETE.

Objeto Statement - SELECT

El método executeQuery devuelve un objeto de tipo ResultSet, que dará acceso a los resultados de la consulta que se haya ejecutado.

```
String query = "SELECT nombre_columna [, . . .] "
            + "FROM nombre_tabla";
try (Statement stmt = connection.createStatement ()) {
  ResultSet rs = stmt.executeQuery(query);
  (...)
}
```

Objeto Statement – UPDATE

El método executeUpdate de Connection se utiliza para insertar, eliminar o modificar datos. Devuelve el número de filas afectadas (insertadas, eliminadas o modificadas) por la sentencia ejecutada.

```
String query = “UPDATE nombre_tabla  SET nombre_columna = valor_nuevo "
            + “WHERE Expresion ";
try (Statement stmt = connection.createStatement ()) {
    int rowCount = stmt.executeUpdate(query);
}
```

Objeto Statement – INSERT

```
String variable = "INSERT INTO nombre_tabla "
                + “(nombre_columna [, . . .]) "
                + "VALUES (valor_columna [, ...]) ";
 }
try (Statement stmt = connection.createStatement ()) {
    stmt.executeUpdate(q, Statement.RETURN_GENERATED_KEYS);
    ResultSet rs = stmt.getGeneratedKeys ();
    int rowId;
    if (rs.next()) {
       rowId = rs.getInt (1);
} else {
       // Esto no deber´ıa ocurrir ...
       rowId = -1;
}
```

El futuro digital es de todos | MinTIC

UNIVERSIDAD DEL NORTE
Vigilada Mineducación

Seguimiento Habilidades Digitales en Programación

* De modo general, ¿Cuál es grado de satisfacción con los siguientes aspectos?

	Nada Satisfecho	Un poco satisfecho	Neutra	Muy satisfecho	Totalmente satisfecho
Sesiones técnicas sincrónicas	○	○	○	○	○
Sesiones técnicas asincrónicas	○	○	○	○	○
Sesiones de inglés	○	○	○	○	○
Apoyo recibido	○	○	○	○	○
Material de apoyo: diapositivas	○	○	○	○	○

Completa la siguiente encuesta para darnos retroalimentación sobre esta semana ▼▼▼

https://www.questionpro.com/t/ALw8TZlxOJ

MisiónTIC2022

El futuro digital es de todos

MinTIC

UNIVERSIDAD DEL NORTE

Vigilada Mineducación

¡GRACIAS

POR SER PARTE DE ESTA EXPERIENCIA DE APRENDIZAJE!

Mision TIC 2022