03

En este paso vamos a descargar el driver apropiado para java y mariadb, y una vez hecho esto vamos a crear un proyecto que haga alguna operación sobre el sistema gestor y las bases de datos que gestiona.

Descarga del driver jdbc para mariadb

Otra vez desde la página de mariadb.org

Botón de descarga (download (2023: solapa lateral derecho))

En una de las solapas de la página de descargas están los conectores (connectors, drivers), como se dijo, aplicaciones intermediarias entre los distintos lenguajes de programación y el sistema gestor mariadb.

Hay varias, cada uno para un lenguaje o entorno distinto: para C, para java (J8+ y J7) , para python, también un conector ODBC y otro para Node.js.

Descargar el correspondiente a java, la última (20230205) versión estable 3.1.2 GA en el sistema operativo "universal".

El driver es un archivo jar, y por tanto se puede utilizar en cualquier sustema operativo donde esté también disponible el entorno java.

mariadb—java-client-#.#.#.jar

También se puede descargar la documentación asociada a este archivo, lo cual luego permitirá obtener la ayuda en el entorno de desarrollo.

Atención mucha atención, WARNING/ACHTUNG

Cuando un programa java necesite utilizar este conector para realizar operaciones sobre una base de datos utilizando el gestor mariadb, hay que indicar en las propiedades del proyecto cuál es la ruta donde se encuentra este archivo.

Y si queremos, que queremos siempre, que nuestro proyecto sea autónomo, y funcione esté donde esté debemos configurarlo para que las librerías estén incluidas en el proyecto.

Esto lo digo porque he visto cómo algunos de ustedes han puesto muy poco cuidado en este aspecto que es vital. Sobre todo, si queremos que nos paguen por nuestro trabajo, que es algo que, poco a poco, cada vez es más habitual.

A continuación, veamos qué va a cambiar respecto al caso de sqlite.

Ejemplo sencillo

Recuerden ustedes los inicios del acceso a datos en un sistema gestor de bases de datos.

En el primer ejercicio lo hicimos con el sistema gestor sqlite, que no es un sistema gestor que funcione en modo cliente—servidor pero el esquema del programa java para acceder a los datos es igual en todos los casos.

Importante: si se quieren realizar peticiones (queries) a un sistema gestor de base de datos con servidor, el servidor debe estar iniciado.

// carga del driver

Si bien este paso suele ser innecesario lo dejo aquí porque todavía en algunos sistemas gestores es necesario.

// crear la conexión con la base de datos

Connection conexion = DriverManager.getConnection("cadena de conexión, usuario, contraseña, etc.")

La cadena de conexión depende del modo de funcionamiento del sistema gestor (cliente—servidor, no servidor), de forma que tendrá más o menos parámetros.

En los casos de sistemas gestores de base de datos que funcionan en el modo cliente servidor, la cadena de conexión suele tener el siguiente esquema genérico:

"jdbc:sgbd://ip\_servidor:puerto/BaseDatos","usuario", "contraseña")

// crear el objeto java de petición o consulta

// crear la cadena sql de petición o consulta

// enviar la sentencia a través del driver jdbc para su procesamiento en el sistema gestor, y recoger el resultado de la ejecución de la sentencia

ResultSet resultado = query.executeQuery(sentencia sql);

El resultado, un objeto de la clase ResultSet, es análogo a una tabla y los métodos para acceder a los datos que contiene son similares a los que se utilizan en las colecciones como las del tipo ArrayList.

Hay tres métodos de los objetos de la clase Statement: execute, executeQuery y executeUpdate. Considero que tienen los suficientes "galones" y horas de programación para que no les indique cuándo se debe utilizar cada uno de ellos y cuál es el resultado que devuelven, pero estos detalles son importantes.

// recorrido del resultado para mostrar los datos de cada fila

// se hace con un bucle que continúa mientras haya registros y en cada pasada se muestran los elementos de la fila, para lo cual hay que conocer cuántos elementos tiene cada fila, es decir, cuántas columnas tiene el resultado (en este ejemplo suponemos que hay cinco columnas, y que cada columna es del tipo de datos que indica el método) while (resultado.next()) 

System.out.println("|" + resultado.getInt(1) + "|" + resultado.getString(2) + "|" + resultado.getFloat(3) + "|" + resultado.getString(4) + "|" + resultado.getString(5) + "|";

}

También se puede utilizar resultado.getInt("nombreCamp01").

// Por último hay que cerrar todos los objetos que se han abierto resultado.close();

query.close();

conexion.close();

Lo anterior es un recordatorio de Io que hicimos con sqlite cambiando Io que hay que cambiar. Pueden echarle un vistazo.

Ejemplo jdbc con mariadb

Llega el momento de aplicar lo anterior al caso de mariadb.

Realiza un proyecto con el nombre **apellidos\_mariadb\_jdbc3**, que contenga varios programas:

* un programa java con el nombre generaBDmariadb.java, que realiza la tarea de generar una base de datos con el nombre apellidosMariaDB, de la misma forma que se hizo con el sistema gestor sqlite,(nota: esta vez las dos tablas tienen un campo apellidos que no aparecía como tal en las tablas del primer ejercicio, si bien luego se hacía referencia a ellas)
* un programa java, consulta.java, que solicita por teclado una de las dos tablas, y luego un nombre, a continuación, devuelve al terminal, en formato tabulado, todos los datos de la tabla elegida cuyos registros contienen el nombre elegido,
* un programa java, muestraBBDD.java, que al ejecutarse muestra por pantalla mensajes indicando si el servidor de mariadb está iniciado o no, y si lo está, muestra en formato de columnas la lista de bases de datos a las que se puede acceder.

Notas:

* Deben enviar el proyecto comprimido en formato zip.
* No debe haber errores de sintaxis ni de ejecución, y debe estar configurado correctamente, y contener todo lo necesario para su ejecución de forma autónoma.
* El puerto de funcionamiento del servidor debe ser 3303.
* No debería hacer falta que lo repita, pero si algún ejercicio se resiste más allá de media hora, sin visos de solución, es el momento de dejarlo, enviarme un correo con suficiente información descriptiva sobre los intentos realizados, los errores obtenidos, etcétera, y esperar la respuesta.