Los componentes de JDBC

El gestor de drivers (java.sql.DriverManager).
La conexión con la base de datos (java.sql.Connection).
La sentencia a ejecutar (java.sql.Statement).
El resultado (java.sql.ResultSet).

Otros:
otro tipo de sentencia (java.sql.PreparedStatement).

Java.sql.DriverManager
Lleva el control de los Drivers cargados en la JDBC Memoria. Encargado de realizar conexión con la Base de Datos.
Se carga mediante el método estático:

forName() de la clase java.lang.Class.

Ejemplo: Class.forName(“com.mysql.jdbc.Driver”);

java.sql.Connection
Representa la conexión con la base de datos. El encargado de abrir una conexión es el Driver Manager mediante el método estático:

public static Connection getConnection(url, usr, pwr) trows java.sql.SQLException

Donde:
url: Identificador de la Base de Datos
usr: Usuario con el que se abre la conexión (opcional)
pwr: Contraseña del Usuario (opcional)

A través de la conexión nos comunicamos con la Base de Datos, enviándole sentencias SQL. Las sentencias SQL se envían a través de “Statements”.
Existen tres tipos de “Statements” y un método para generar cada tipo.

Una vez terminada una Conexión, se debe “Liberar”, que es cerrarla, de modo análogo a como se trabaja con flujos (Streams).
Las conexiones se cierran con el método:

public void close() trows java.sql.SQLException;

Statements

java.sql.Statement:

createStatement();

java.sql.PreparedStatement:

prepareStatement();

java.sql.Statement
Se usa para ejecutar sentencias SQL. Lleva asociada una conexión que sirvió como origen para su creación.
Se crea con el método de la clase:

java.sql.Connection:

public Statement createStament() trows java.sql.SQLException;

Las sentencias se cierran con el método:

public void close() trows java.sql.SQLException;

El método para ejecutarla depende del tipo de sentencia SQL que contenga.
Sentencias SELEC: se usa el método:

executeQuery(String sql)

devuelve una instancia de java.sql.ResultSet.

Sentencias INSERT, UPDATE, DELETE: se usa el método:

executeUpdate(String sql).

devuelve un int con el número de filas afectadas.

Java.sql.ResultSet
Representa el resultado de la ejecución de una sentencia SQL. Lleva asociadas las filas y columnas que cumplían con la sentencia SQL.
Implementa métodos para:
-Acceder a las filas que componen el resultado.
-Acceder al valor de cada columna de la fila seleccionada.

Los ResultSet se cierran mediante el método: public boolean close() trows java.sql.SQLException; El ResultSet se cierra automáticamente al cerrar el Statement que la creó. No obstante no está demás cerrarlo.

Java Database Connectivity (**JDBC**) es una interfase de acceso a bases de datos estándar SQL que proporciona un acceso uniforme a una gran variedad de bases de datos relacionales. **JDBC** también proporciona una base común para la construcción de herramientas y utilidades de alto nivel.

La interfaz **PreparedStatement** hereda de Statement y difiere de esta en dos maneras. Las instancias de **PreparedStatement** contienen una sentencia SQL que ya ha sido compilada. Esto es lo que hace que se le llame 'preparada'. La sentencia SQL contenida en un objeto **PreparedStatement** pueden tener uno o más parámetros.

El objeto **ResultSet** proporciona varios métodos para obtener los datos de columna correspondientes a un fila.

Statement devuelve una instancia de **java**.sql.ResultSet. Sentencias INSERT, UPDATE, DELETE: se usa el método: executeUpdate(String sql). devuelve un int con el número de filas afectadas.

Page 18 **java**.sql.ResultSet

Representa el resultado de la ejecución de una sentencia SQL.