

Unidad 2

Conceptos básicos de JDBC y MySQL

Temas Selectos de Ingeniería en Computación I

Universidad Autónoma Metropolitana

Trimestre 10-I

Contenido de la unidad

- 1 Introducción
- 2 Funcionamiento de JDBC
- 3 Implementación

JDBC

- La conectividad de base de datos de Java (JDBC, Java Database Connectivity) es un marco de programación para los desarrolladores de Java que escriben los programas que tienen acceso a la información guardada en bases de datos
- Permite el acceso a una base de datos sin importar qué software de administración o manejo de base de datos se utilice para controlarlo

MySQL

- El Lenguaje de Consulta Estructurado (SQL, Structured Query Language) es considerado como un estándar para el manejo de bases de datos relacionales
- MySQL es una idea de la empresa de código abierto MySQL AB cuyo objetivo es que MySQL cumpla con el estándar SQL pero de una manera más eficiente
- MySQL es un sistema de administración de bases de datos. Una base de datos es una colección estructurada de tablas que contienen datos

Características de MySQL

Propiedades

Entre las características disponibles en las últimas versiones se pueden destacar

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Diferentes opciones de almacenamiento según si se desea velocidad en las operaciones o el mayor número de operaciones disponibles.
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

Contenido de la unidad

- 1 Introducción
- 2 Funcionamiento de JDBC
- 3 Implementación

JDBC

Introducción

El API (Interfaz de Programación de Aplicaciones) JDBC está compuesto por una gran cantidad de Clases e Interfaces que contienen métodos que permiten la gestión de operaciones con bases de datos

Interfaz Connection

Descripción

Es una conexión (sesión) con una base de datos específica

Métodos a utilizar

- `createStatement()`
- `close()`

Método createStatement()

Descripción

Crea un objeto Statement para enviar sentencias de SQL a la base de datos

Sintaxis

Statement createStatement() throws SQLException

Método close()

Descripción

Libera el objeto tipo Connection y los recursos de JDBC

Sintaxis

void close() throws SQLException

clase DriverManager

Descripción

Es el servicio básico para administrar un conjunto de manejadores de JDBC

Métodos a utilizar

- registerDriver()
- getConnection()

Método registerDriver()

Descripción

Realiza el registro del manejador de la base de datos a utilizar

Sintaxis

```
public static void registerDriver(Driver driver) throws  
SQLException
```

driver

Es el manejador que se desea, en el caso de MySQL, la sintaxis es la siguiente:

- `DriverManager.registerDriver(new com.mysql.jdbc.Driver());`

Método getConnection()

Descripción

Intenta realizar una conexión con la base de datos

Sintaxis

```
public static Connection getConnection(String url) throws  
SQLException
```

url

- contiene especificaciones de la base de datos de la forma jdbc:subprotocol:subname
- Adicionalmente se puede incluir el usuario y el password como campos extra

Interfaz Statement

Descripción

Un objeto Statement es utilizado para ejecutar sentencias de SQL y regresar los resultados que produce

Métodos a utilizar

- `executeQuery()`
- `executeUpdate()`

Método executeQuery()

Descripción

Ejecuta una sentencia SQL normalmente SELECT que regresa un objeto tipo ResultSet

Sintaxis

ResultSet executeQuery(String sql) throws SQLException

Método executeUpdate()

Descripción

Ejecuta una sentencia SQL la cuál debe ser un INSERT, UPDATE, o DELETE

Sintaxis

```
int executeUpdate(String sql) throws SQLException
```


Interfaz ResultSet

Descripción

Es una tabla de datos que representa un conjunto de resultados de una base de datos generado normalmente después de una consulta

Contenido de la unidad

- 1 Introducción
- 2 Funcionamiento de JDBC
- 3 Implementación

Implementación

Ejemplo

- Crear una base de datos en MySQL
- Configurar el entorno de desarrollo
- Implementar una aplicación con las operaciones básicas de manejo de base de datos utilizando JDBC