



Curso de Java Standard



Ing. Octavio Robleto



octavio.robleto@gmail.com



<https://octaviorobleto.com>



Introducción

Los gestores de base de datos relacionales como MySQL, PostgreSQL, SQL Server, etc. Nos proporcionan mecanismos para optimizar el acceso y ejecución de sentencias.

Uno de estos mecanismos es la **declaración preparada**; forma que le indicamos a la base de datos que vamos a utilizar una sentencia, la guarde y la “**precompile**” con las condiciones necesarias para poder utilizarla.

Muchas veces vamos a necesitar ejecutar mas de una vez la misma sentencia pero con valores diferentes; al hacerlo con una declaración preparada evitamos el que el gestor evalúe cada vez si lo que le enviamos tiene la sintaxis correcta, los campos y tablas existen y la conversión al tipo de dato correcto.

```
-- USAR BASE DE DATOS
USE autosEducacionIT;

-- INSERTAR ESPECIFICANDO LOS CAMPOS PREPARANDO LA
PREPARE INSERTAR FROM "INSERT INTO autosEducacionIT.autoFamiliar
(patenteNumero, patenteActiva, marca, categoria, color, puestos)
VALUES (?, ?, ?, ?, ?, ?);";

-- SETEAMOS LAS VARIABLES
SET @patenteNumero = 'ABC-001';
SET @patenteActiva = 1;
SET @marca = 'Audi';
SET @categoria = 'SEDAN';
SET @Color = 'ROJO';
SET @puestos = 6;

-- EJECUTAR LA CONSULTA PREPARADA
EXECUTE INSERTAR USING @patenteNumero, @patenteActiva, @marca, @categoria, @Color, @puestos;

-- SETEAMOS LAS VARIABLES
SET @patenteNumero = 'ZXM-901';
SET @patenteActiva = 1;
SET @marca = 'Ford';
SET @categoria = 'COMPACTO';
SET @Color = 'NEGRO';
SET @puestos = 4;

-- EJECUTAR LA CONSULTA PREPARADA
EXECUTE INSERTAR USING @patenteNumero, @patenteActiva, @marca, @categoria, @Color, @puestos;
```

Recordando Connection

Esta interfaz provee una sesión de trabajo con una base de datos. Sus métodos, aparte de permitir modificarla y consultar sus tablas, también permiten obtener información sobre su estructura.

Tipo	Método	Descripción
void	commit()	Confirma los cambios realizados provisionalmente por las transacciones.
void	close()	Libera la conexión a la base de datos.
Statement	createStatement()	Crea un objeto Statement para enviar sentencias SQL a la base de datos.
PreparedStatement	prepareStatement(String sql)	Crea un objeto PreparedStatement para enviar sentencias SQL parametrizadas a la base de datos.
void	rollback()	Deshace todos los cambios realizados en la transacción actual y libera los bloqueos de la base de datos.
void	setAutoCommit(boolean autoCommit)	Establece el modo de confirmación automática.
boolean	isClosed()	Recupera si la conexión se ha cerrado.

Ejecutar Declaraciones Preparadas SQL

La interfaz **PreparedStatement** que hereda **Statement** nos proporciona un objeto que representa una instrucción SQL precompilada, además nos provee una serie de métodos que al ser ejecutados nos devuelven los resultados que producen dichas sentencias.

Al crear la conexión el objeto nos provee como pudimos ver en el cuadro de métodos de **Connection** el método **prepareStatement(String sql)** que nos devuelve un **PreparedStatement**.

Tipo	Método	Descripción
Boolean	execute(String sql)	Ejecuta la instrucción SQL dada, que puede devolver varios resultados.
ResultSet	executeQuery()	Ejecuta la instrucción SQL dada, que devuelve un solo Objeto ResultSet.
Int	executeUpdate()	Ejecuta la instrucción SQL dada, que debe ser de tipo DML y devuelve uno (1) si se realizó correctamente o cero (0)

Parámetros

La interfaz nos provee además varios métodos setters para establecer los valores de los parámetro: **setTipoDato(int indiceParametro, TipoDato valorDato)**.

Estos parámetros están representados por el signo de interrogación “?” en el String que contiene el SQL, al usar los métodos setter debemos indicar el índice del parámetro a sustituir comenzando desde el numero 1.

¿Qué ganamos con esto además de la optimización del proceso? Que al sustituir un valor no se arme de manera dinámica el SQL y así evitamos la Inyección SQL.

```
String sql = "INSERT ?, UPDATE ?, DELETE ?";

PreparedStatement declaracionPreparadaSQL = conexion.prepareStatement(sql);

declaracionPreparadaSQL.setTipo(1, valor);

declaracionPreparadaSQL.execute();
```

