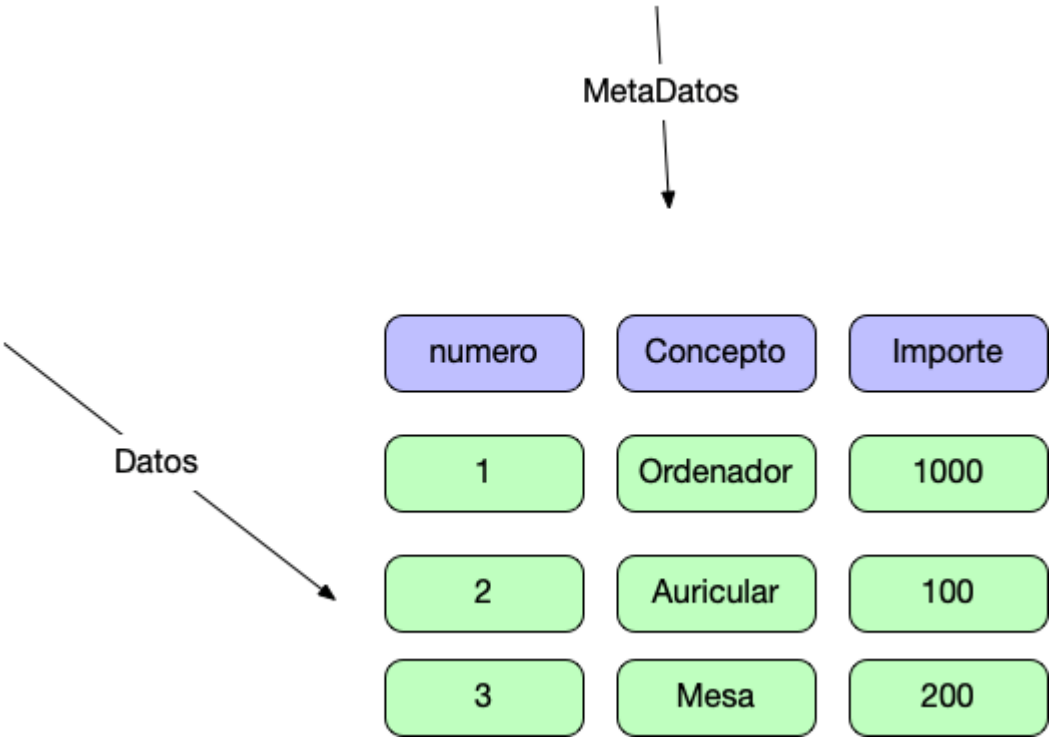
El concepto de ResultSetMetaData es uno de los conceptos que aporta flexibilidad al API de JDBC hay situaciones en la que por ejemplo deseamos leer cualquier tipo de tabla de la base de datos con cualquier consulta y presentar un resultado flexible . Para realizar esta operación necesitamos acceder a los que se denomina los metadatos de la base de datos y no solo ser capaz de leer la información de la tabla con las filas contenidas sino también el nombre de las columnas que son usadas ya que cada tabla tendrá las suyas.



Vamos a ver un ejemplo sencillo de ResultSetMetaData con JDBC.

```
package com.arquitecturajava;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.Statement;
```

```
public class PruebaMetaData {

	public static void main(String[] args) {

		ConfiguradorDB c = ConfiguradorDB.getInstance();

		try (Connection con =
DriverManager.getConnection(c.getUrl(), c.getUser(), c.getPassword());
				Statement s = con.createStatement();
				ResultSet rs = s.executeQuery("select
* from Facturas")) {

			ResultSetMetaData rsmd = rs.getMetaData();
			int columnas = rsmd.getColumnCount();

			for (int i = 1; i <= columnas; i++) {

				System.out.print(rsmd.getColumnName(i)
+ "     ");

			}
			System.out.println("");

			while (rs.next()) {

				for (int i = 1; i <= columnas; i++) {

System.out.print(rs.getString(i) + "     ");

				}
				System.out.println("");
```

```
                }
            } catch (Exception e) {
                e.printStackTrace();
            }
        }

}
```

JDBC ResultSetMetaData

Acabamos de usar un ResultSetMetaData para leer la información de las columnas de la tabla de la base de datos.

```
ResultSetMetaData rsmd = rs.getMetaData();
int columnas = rsmd.getColumnCount();
```

Una vez que usamos el metodo getColumnCount() nos dice el numero de columnas que tenemos . El siguiente paso es usar un bucle for para recorrer las columnas y obtener sus nombres . Hecho esto imprimimos por la consola los nombres de las columnas y estamos ya accediendo a los metadatos.

```
for (int i = 1; i <= columnas; i++) {

System.out.print(rsmd.getColumnName(i) + "     ");

}
```

Realizado este paso el nombre de las columnas queda impreso por la consola :

```
numero     concepto     importe
```

Es momento de recorrer la tabla y presentar los datos apoyándonos en la información obtenida del ResultSetMetaData.

```
while (rs.next()) {

        or (int i = 1; i <= columnas; i++) {

                System.out.print(rs.getString(i) + "     ");
        }
        System.out.println("");
}
```

Vemos aparecer los datos en la consola junto con los ya escritos de nombre de columna:

```
numero     concepto     importe
1     ordenador     300.0
2     tablet     200.0
```

Podemos cambiar la consulta del código por otra tabla y nos mostrará la información de la nueva tabla con sus datos correspondientes.

```
try (Connection con = DriverManager.getConnection(c.getUrl(),
c.getUser(), c.getPassword());
Statement s = con.createStatement();
ResultSet rs = s.executeQuery("select * from Personas"))
```

En este caso la tabla Personas.

```
nombre     apellidos     edad
juan     gomez     20
miguel     sanchez     30
```

Aprendamos usar ResultSetMetaData en JDBC para obtener soluciones flexibles a nuestros problemas de una forma muy directa y compacta.

Otros artículos relacionados

- Spring Boot JDBC y su configuración
- Spring BeanPropertyRowMapper y jdbcTemplates
- Spring JdbcTemplate y el principio DRY
- JDBC Batch y rendimiento
- ¿Qué es un Java DataSource?
- Que es JDBC