



# Java

## Tema 8

---

### Conexión a Base de Datos mediante JDBC

#### sentencia SELECT

1



### Contenido

---

- 1 - Interfaz PreparedStatement:**
  - sentencia Select
- 2 - Objetos ResultSet**
- 3 - Excepciones SQL**



## Objetos PreparedStatement

3



### Interface PreparedStatement

- ❑ Permiten **ejecutar sentencias SQL**
- ❑ El método `prepareStatement()` de un objeto `Connection`, tiene como parámetro una sentencia SQL, la compila (la revisa) y devuelve un objeto `PreparedStatement` :

```
conn = DriverManager.getConnection(URL, user, password);
```

```
String sentencia = "SELECT * FROM PARQUE";
```

```
PreparedStatement ps = conn.prepareStatement(sentencia);
```

4



## Método executeQuery()

- Para **ejecutar la sentencia SELECT** pasada al objeto PreparedStatement, se utiliza el método **executeQuery()**, y devuelve un objeto **ResultSet** con los datos encontrados.

Método	Uso
<b>ResultSet executeQuery()</b>	Ejecuta sentencia <b>SELECT</b> .  Devuelve un objeto <b>ResultSet</b>

5



## Parámetros de entrada en un SELECT

```
// Uso de parámetros para una consulta preparada

conn = DriverManager.getConnection(URL,user,password);

String consulta = "SELECT * " +
                  "FROM comunidad " +
                  "WHERE nombre = ?";

PreparedStatement ps = conn.prepareStatement(consulta);

ps.setString(1,"Comunidad Valenciana"); //1er parámetro

ResultSet rs = ps.executeQuery(); //ejecuta la select
```

6



## Objeto ResultSet

---

7



### Obteniendo datos de un ResultSet

---

- ❑ Los objetos `ResultSet` contienen las filas de datos devueltas por una consulta.
- ❑ Las columnas de una fila se numeran de izquierda a derecha empezando con la columna 1.
- ❑ Para **recuperar el valor de un columna**, de la fila actual del `ResultSet`, tenemos los métodos:
  - ❑ **`getXXX(int indiceColumna)`** o
  - ❑ **`getXXX(String NombreColumna)`**

donde **XXX** indica el tipo de dato al que queremos convertirlo.

```
Ejemplo:      rs.getInt("id")           //columna id, devuelve int
               rs.getString("nombre") //columna nombre, devuelve String
               rs.getString(2)        //segunda columna, devuelve String
```

8



## Obteniendo datos de un ResultSet

- ❑ Para saber si una columna, previamente recuperada, contiene un valor nulo, se utiliza el método `wasNull()` de un objeto `ResultSet`
- ❑ El método `next()` permite desplazarse a la siguiente fila del `ResultSet`.

```
ResultSet rs = ps.executeQuery();

// Recorremos las filas devueltas por la consulta
while (rs.next()) { //avanza a la fila siguiente
    Integer id      = rs.getInt("id");
    String nombre = rs.getString("nombre");
}
```

9



## Tipos de datos de un ResultSet

Tipo de dato SQL	Método de acceso
CHAR	String getString()
VARCHAR	String getString()
LONGVARCHAR	InputStream, getAsciiString() o getUnicodeString()
NUMERIC, DECIMAL	Java.math.BigDecimal getBigDecimal()
BIT	boolean getBoolean()
TINYINT	byte getByte()
SMALLINT	short getShort()
INTEGER	int getInt()
BIGINT	long getLong()
REAL	float getFloat()
FLOAT	double getDouble()
DOUBLE	double getDouble()
BINARY	byte[] getBytes()
VARBINARY	InputStream getBinaryStream()
DATE	java.sql.Date getDate()
TIME	java.sql.Time getTime()
TIMESTAMP	java.sql.Timestamp getTimeStamp()

10



## Uso del ResultSet

```
//Abrir la conexión con la Base de Datos
Connection conexion =
    DriverManager.getConnection(url,"root","root");

// Creamos un objeto PreparedStatement
PreparedStatement ps =
    conexion.prepareStatement("Select * from comunidad");

// Creamos un objeto ResultSet
ResultSet rs = ps.executeQuery();

// Recorremos los resultados de la consulta
while (rs.next()){ //avanza a la fila siguiente
    Integer id      = rs.getInt("id");
    String nombre = rs.getString("nombre");
}
```

11



## Tipos de ResultSet.

- ❑ El método **prepareStatement()** ofrece dos versiones para desplazarnos por un **ResultSet** :
  - ✓ **Sin parámetros** para desplazamientos solo hacia delante y de solo lectura
  - ✓ **Con dos parámetros** para definir el tipo de objeto **ResultSet**:
    - ❑ El primer parámetro, **tipo de movimiento**, puede tener los valores:
      - ✓ **TYPE\_FORWARD\_ONLY**: ResultSet con movimiento solo hacia delante.
      - ✓ **TYPE\_SCROLL\_INSENSITIVE**: ResultSet para todo tipo de movimientos que NO permite ver los cambios realizados en los datos.
      - ✓ **TYPE\_SCROLL\_SENSITIVE**: ResultSet para todo tipo de movimientos que permite ver los cambios realizados en los datos.
    - ❑ El segundo parámetro, **tipo de modificación**, puede tomar los valores:
      - ✓ **CONCUR\_UPDATABLE**: ResultSet modificable.
      - ✓ **CONCUR\_READONLY**: ResultSet de solo lectura.
- ❑ Si no se especifican parámetros, los valores por defecto del **ResultSet** serán **TYPE\_FORWARD\_ONLY** (hacia delante) y **CONCUR\_READONLY** (solo lectura)

12



## Uso de un ResultSet TYPE\_SCROLL\_INSENSITIVE y CONCUR\_READ\_ONLY

Ejemplo uso de un método **prepareStatement()** con parámetros

```
// Creamos un objeto PreparedStatement
ps = conexion.prepareStatement("select * from comunidad",
                               ResultSet.TYPE_SCROLL_INSENSITIVE,
                               ResultSet.CONCUR_READ_ONLY);

// Creamos un objeto ResultSet con método executeQuery()
ResultSet rs = ps.executeQuery();
```

13



## métodos de desplazamiento por un ResultSet.

- ❑ **boolean first()** : desplaza el cursor al primer registro. Devuelve true si el cursor esta en un registro válido, y false en otro caso, o si el objeto **ResultSet** no contiene registros. Valido solo en objetos **ResultSet** de tipo **scrollable** , en caso contrario lanzará una excepción **SQLException**.
- ❑ **boolean last()** : desplaza el cursor al último registro del objeto **ResultSet**. Devuelve true si el cursor esta en un registro válido, y false en otro caso o si el objeto **ResultSet** no tiene registros. Valido para objetos **ResultSet** de tipo **scrollable**, en caso contrario lanzará una excepción **SQLException**.
- ❑ **boolean previous()** : desplaza el cursor al anterior registro del objeto **ResultSet**. Devuelve true si el cursor esta en un registro válido, y false si está antes del primero. Válido para objetos **ResultSet** de tipo **scrollable**, en caso contrario lanzará una excepción **SQLException**.
- ❑ **void moveToCurrentRow()** : mueve el cursor a la posición recordada, normalmente el registro actual. Sólo tiene sentido cuando estamos situados en un registro que se ha insertado. Válido solo para objetos **ResultSet** que permiten la modificación (definidos con la constante **CONCUR\_UPDATABLE**)



## Otros métodos de desplazamiento por un ResultSet.

---

- ❑ **boolean absolute(int registro)** : se mueve al número de registro indicado. Si el valor es negativo, se posiciona en el número registro indicado pero empezando por el final. Este método devolverá false si nos hemos desplazado después del último registro o antes del primer registro del objeto **ResultSet**. Para poder utilizar este método el objeto **ResultSet** debe ser de tipo **TYPE\_SCROLL\_SENSITIVE** o **TYPE\_SCROLL\_INSENSITIVE**. Un **ResultSet** de cualquiera de estos dos tipos se dice que es de tipo **scrollable**.
- ❑ **void afterLast()** : se desplaza al final del objeto **ResultSet**, después del último registro. Si el **ResultSet** no posee registros no tiene ningún efecto. Este método sólo se puede utilizar en objetos **ResultSet** de tipo **scrollable**.
- ❑ **void beforeFirst()** : mueve el cursor al comienzo del objeto **ResultSet**, antes del primer registro. Solo sobre objetos **ResultSet** de tipo **scrollable**.

15



## Otros métodos de desplazamiento por un ResultSet.

---

- ❑ **boolean isAfterLast()** : indica si nos encontramos después del último registro del objeto **ResultSet**. Sólo se puede utilizar en objetos **ResultSet** de tipo **scrollable**.
- ❑ **boolean isBeforeFirst()** : indica si nos encontramos antes del primer registro del objeto **ResultSet**. Sólo se puede utilizar en objetos **ResultSet** de tipo **scrollable**.
- ❑ **boolean isFirst()** : indica si el cursor se encuentra en el primer registro. Sólo se puede utilizar en objetos **ResultSet** de tipo **scrollable**.
- ❑ **boolean isLast()** : indica si nos encontramos en el último registro del **ResultSet**. Sólo se puede utilizar en objetos **ResultSet** de tipo **scrollable**.
- ❑ **int getRow()**: devuelve el número de registro actual. El primer registro será el número 1, el segundo el 2, etc. Devolverá cero si no hay registro actual.

16





## Excepciones SQL

---

17



## Excepciones SQL

---

❑ **SQLException:** proporciona la siguiente información sobre el error:

- ✓ El método **getMessage()** devuelve un String describiendo el error generado por la BD
- ✓ El método **getErrorCode()** devuelve un entero con el código de error de la BD

18