EJECUCIÓN DE SENTENCIAS DE DESCRIPCIÓN DE DATOS

Javier García-Retamero Redondo

OBJETIVO

- El objetivo principal es obtener información sobre los elementos de la base de datos.

- La información la recuperaremos del diccionario de datos.

EJECUCIÓN DE SENTENCIAS DE DESCRIPCIÓN DE DATOS

FUNCIONAMIENTO

FUNCIONAMIENTO

- Podemos obtener información de la base de datos utilizando la interfaz **DatabaseMetaData** y su método **getMetaData()**.

```
DatabaseMetaData dbmd = conexion.getMetaData();
```

Descripción y todos los métodos de la interfaz DatabaseMetada

DatabaseMetaData (Java Platform SE 8)

EJECUCIÓN DE SENTENCIAS DE DESCRIPCIÓN DE DATOS

OBTENCIÓN DE INFORMACIÓN

OBTENCIÓN DE INFORMACIÓN

Obtener información de la conexión:

- **Nombre del gestor:**

String nombre = **dbmd**.getDatabaseProductName();

- **Driver utilizado:**

String driver = **dbmd**.getDriverName();

- **Dirección para acceder a la bbdd:**

String url = **dbmd**.getURL();

- **Nombre del usuario:**

String usuario = **dbmd**.getUserName();

OBTENCIÓN DE INFORMACIÓN

- **getTables(C,E,P,T):**

  Obtiene información de los objetos contenidos en la bbdd

  Catálogo: Catálogo (Nombre de la BBDD) que contiene las tablas.
  NULL = todos.

  Esquema: Esquema (usuario) que contiene las tablas.
  NULL = todos/actual (según BD)

  Patrón Tabla: Patrón del nombre de las tablas.
  Puedes utilizar _ y %. NULL = todas

  Tipo: Es un array de String con: TABLE/VIEW/INDEX...
  NULL = todo.

OBTENCIÓN DE INFORMACIÓN

Ejemplo: Obtener todas las tablas del usuario "DAM2" de la bbdd "FREE":

```
ResultSet resul = null;
String[] tipos = {"TABLE"};
resul = dbmd.getTables("FREE", "DAM2" ,null, tipos);
```

OBTENCIÓN DE INFORMACIÓN

Cada fila del **ResulSet** que devuelve **getTables()** tiene información de una tabla.

Las columnas que devuelve en cada una de las filas son:

TABLE_CAT String => table catalog (may be null)
TABLE_SCHEM String => table schema (may be null)
TABLE_NAME String => table name
TABLE_TYPE String => table type. Typical types are "TABLE", "VIEW", "SYSTEM TABLE", "GLOBAL TEMPORARY", "LOCAL TEMPORARY", "ALIAS", "SYNONYM".
REMARKS String => explanatory comment on the table
TYPE_CAT String => the types catalog (may be null)
TYPE_SCHEM String => the types schema (may be null)
TYPE_NAME String => type name (may be null)
SELF_REFERENCING_COL_NAME String => name of the designated "identifier" column of a typed table (may be null)
REF_GENERATION String => specifies how values in SELF_REFERENCING_COL_NAME are created. Values are "SYSTEM", "USER", "DERIVED". (may be null)

String catalogo = resul.getString("**TABLE_CAT**");

OBTENCIÓN DE INFORMACIÓN

- **getColumns (catálogo, esquema, nombre_tabla, nombre_columna):**
  Null = todos.

  Obtiene información de las columnas de las tablas

Ejemplo: Obtener los campos de la tabla "DEPARTAMENTOS" que se encuentra en el esquema "DAM2" en la bbdd "FREE".

```
ResultSet columnas = null;
columnas = dbmd.getColumns("FREE", "DAM2", "DEPARTAMENTOS", null);
```

OBTENCIÓN DE INFORMACIÓN

Cada fila del **ResulSet** que devuelve **getColumns()** tiene información de una columna.

Puedes ver la descripción de las columnas en:

DatabaseMetaData (Java Platform SE 8)

Las columnas que devuelve en cada una de las filas son:

TABLE_CAT, TABLE_SCHEM, TABLE_NAME, COLUMN_NAME, DATA_TYPE, TYPE_NAME, COLUMN_SIZE, BUFFER_LENGTH, DECIMAL_DIGITS, NUM_PREC_RADIX, NULLABLE, REMARKS, COLUMN_DEF, SQL_DATA_TYPE, CHAR_OCTET_LENGTH, ORDINAL_POSITION, IS_NULLABLE, SCOPE_CATALOG, SCOPE_SCHEMA, SCOPE_TABLE, SOURCE_DATA_TYPE, IS_AUTOINCREMENT, IS_GENERATEDCOLUMN

String nombCol = columnas.getString(“**COLUMN_NAME**”);

OBTENCIÓN DE INFORMACIÓN

- **getPrimayKeys(catalogo, esquema, tabla)**:
  Null = todos.

Obtiene información de las claves primarias

Ejemplo: Obtener la clave primaria de la tabla "DEPARTAMENTOS" del usuario "EJEMPLO" de la bbdd "FREE".

ResultSet pk = dbmd.getPrimaryKeys(**"FREE", "DAM2", "DEPARTAMENTOS");**

OBTENCIÓN DE INFORMACIÓN

- **getExportedKeys (catálogo, esquema, tabla):**
  Null = todos.

  Obtiene información de las claves externas que apuntan a una tabla

  Habría tantas filas como claves ajenas referenciaran a la tabla.

  Si la tabla no es referenciada por una clave ajena, no devolvería nada

Ejemplo: Obtener las claves ajenas que apuntan a la tabla "DEPARTAMENTOS" del usuario "EJEMPLO" de la bbdd "FREE".

ResultSet fk = dbmd.getExportedKeys(**"FREE, "DAM2", "DEPARTAMENTOS");**

OBTENCIÓN DE INFORMACIÓN

- **getImportedKeys (catálogo, esquema, tabla):** Devuelve la lista de claves ajenas existentes en la tabla.

Obtiene información de las claves externas que salen de la tabla

Si de la tabla no sale ninguna clave ajena, no devolvería nada.

Ejemplo: Obtener las claves ajenas que salen de la tabla "EMPLEADOS" del usuario "EJEMPLO" de la bbdd "FREE".

```
ResultSet fk = dbmd.getImportedKeys(“FREE”, “DAM2", “EMPLEADOS");
```

OBTENCIÓN DE INFORMACIÓN

Cada fila del **ResulSet** que devuelven **getExportedKeys()** o **getImportedKeys()** las siguientes columnas:

PKTABLE_CAT, PKTABLE_SCHEM, PKTABLE_NAME, PKCOLUMN_NAME, FKTABLE_CAT, FKTABLE_SCHEM, FKTABLE_NAME, FKCOLUMN_NAME, KEY_SEQ UPDATE_RULE, DELETE_RULE FK_NAME, PK_NAME, DEFERRABILITY

Puedes ver la descripción de las columnas que devuelve getImportedKeys() en:

DatabaseMetaData (Java Platform SE 8)

Puedes ver la descripción de las columnas que devuelve getExportedKeys() en:

DatabaseMetaData (Java Platform SE 8)

OBTENCIÓN DE INFORMACIÓN

- **getProcedures(catálogo, esquema, patrón nombreprocedimiento):**

  Obtiene información de procedimientos almacenados

Ejemplo: Obtener los procedimientos del usuario "EJEMPLO" de la bbdd "FREE".

ResultSet proc = dbmd.getProcedures(**"FREE", "EJEMPLO", null);**

OBTENCIÓN DE INFORMACIÓN

Cada fila del **ResulSet** que devuelven **getProcedures()** devuelve las siguientes columnas:

PROCEDURE_CAT String => procedure catalog (may be null)
PROCEDURE_SCHEM String => procedure schema (may be null)
PROCEDURE_NAME String => procedure name
reserved for future use
reserved for future use
reserved for future use
REMARKS String => explanatory comment on the procedure
PROCEDURE_TYPE short => kind of procedure:

- procedureResultUnknown - Cannot determine if a return value will be returned
- procedureNoResult - Does not return a return value
- procedureReturnsResult - Returns a return value

SPECIFIC_NAME String => The name which uniquely identifies this procedure within its schema.

EJECUCIÓN DE SENTENCIAS DE DESCRIPCIÓN DE DATOS

OBTENCIÓN DE INFORMACIÓN SOBRE RESULSET

OBTENCIÓN DE INFORMACIÓN SOBRE RESULSET

Si ejecutamos una sentencia SQL sobre una tabla de la siguiente manera:

```
Statement sentencia=conexion.createStatement();
ResultSet rs = sentencia.executeQuery ("Select * from departamentos");
```

- Sobre ese ResultSet podemos obtener metadatos (datos sobre los datos) aplicándole el método **getMetaData().**
- El resultado debe recogerlo una variable de tipo **ResultSetMetaData:**

```
ResultSetMetaData rsmd = rs.getMetaData();
```

OBTENCIÓN DE INFORMACIÓN SOBRE RESULSET

A una variable ResultSetMetaData (**rsmd**) se le pueden aplicar los siguientes métodos:

Método	Descripción
getColumnCount()	Devuelve el número de columnas devueltas por la tabla
getColumnName(i)	Devuelve el nombre de la columna de la posición “i”
getColumnTypeName(i)	Devuelve el tipo de datos de la columna de la posición “i”
isNullable(i)	Devuelve “1” si la columna de la posición “i” puede contener valores nulos
getColumnDisplaySize(i)	Devuelve el máximo número de caracteres de la columna de la posición “i”

OBTENCIÓN DE INFORMACIÓN SOBRE RESULSET

Número de filas que devuelve un ResultSet:

```
Statement sentencia=conexion.createStatement();
ResultSet rs = sentencia.executeQuery ("Select * from departamentos");
```

- Nos desplazamos al último registro (last()) y preguntamos en cual estamos (getRow()). Después volvemos a movernos antes del primero (beforeFirst()). El método last() devuelve False si no hay registros:

```
int rows = 0;
if (rs.last()) {
   rows = rs.getRow();
   rs.beforeFirst();
}
```