

# 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 el objeto **DatabaseMetaData** y su método **getMetaData()**

```
Class.forName("com.mysql.jdbc.Driver");  
Connection conexion = DriverManager.getConnection  
    ("jdbc:mysql://localhost/ejemplo", "root", "");  
  
DatabaseMetaData dbmd = conexion.getMetaData();
```

# 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 (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 para obtener todas las tablas del catálogo "DAM2":*

```
ResultSet resul = null;
```

```
String[] tipos = {"TABLE"};
```

```
resul = dbmd.getTables(null, "DAM2" ,null, tipos);
```



# OBTENCIÓN DE INFORMACIÓN

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

Las columnas principales que devuelve el ResultSet son:

- TABLE\_CAT
- TABLE\_SCHEM
- TABLE\_NAME
- TABLE\_TYPE
- REMARKS,

(aunque se pueden utilizar los números 1, 2, 3, 4, 5)

```
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 “ejemplo”.*

```
ResultSet columnas = null;  
columnas = dbmd.getColumns(null, "ejemplo", "departamentos", null);
```

# OBTENCIÓN DE INFORMACIÓN

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

Las columnas principales que devuelve el ResultSet 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)
COLUMN_NAME, DATA_TYPE, COLUMN_SIZE, BUFFER_LENGTH, DECIMAL_DIGITS NUM_PREC_RADIX NULLABLE	

```
String nombCol = columnas.getString("COLUMN_NAME");
```

# OBTENCIÓN DE INFORMACIÓN

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

Obtiene información de las claves primarias

*Ejemplo: Obtener las claves primarias de la tabla “departamentos” del esquema “ejemplo”*

```
ResultSet pk = dbmd.getPrimaryKeys(null, "ejemplo", "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”.*

```
ResultSet fk = dbmd.getExportedKeys(null, "ejemplo", "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.*

```
ResultSet fk = dbmd.getImportedKeys(null, "ejemplo", "empleados");
```



# OBTENCIÓN DE INFORMACIÓN

- Columnas para **getImportedKeys** y **getExportedKeys**:

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

# OBTENCIÓN DE INFORMACIÓN

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

Obtiene información de procedimientos almacenados

*Ejemplo: Obtener los procedimientos.*

```
ResultSet proc = dbmd.getProcedures(null, null, null);
```

# EJECUCIÓN DE SENTENCIAS DE DESCRIPCIÓN DE DATOS

OBTENCIÓN DE INFORMACIÓN SOBRE RESULTSET



# OBTENCIÓN DE INFORMACIÓN SOBRE RESULTSET

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 RESULTSET

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

Método	Descripción
getColumnCount()	Devuelve el número de columnas devueltas por la tabla
columnName(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 RESULTSET

Para saber el número de filas que devuelve el ResultSet:

```
Statement sentencia=conexion.createStatement();
```

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

```
// Nos movemos al ultimo registro.
```

```
// El método last() devolverá false si el contenido de ResultSet esta vacío.
```

```
if (rs.last()) {
```

```
    int rows = rs.getRow();
```

```
    rs.beforeFirst();
```

```
// Devuelve cuál es el número del ultimo registro
```

```
// Volvemos al primer registro
```

```
...  
}
```