使用元数据分析数据库

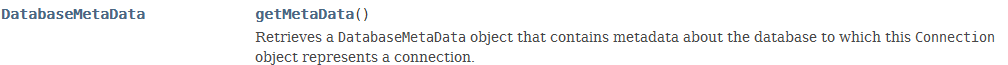
1. **元数据的意义：元数据**（**Metadata**），又称**中介数据**、**中继数据**，为描述[数据](http://baike.baidu.com/view/38752.htm)的数据（data about data），主要是描述数据[属性](http://baike.baidu.com/view/77730.htm)（property）的[信息](http://baike.baidu.com/view/1527.htm)，用来支持如指示存储位置、[历史](http://baike.baidu.com/view/8081.htm)数据、[资源](http://baike.baidu.com/view/8439.htm)查找、文件记录等功能。元数据算是一种[电子](http://baike.baidu.com/view/3476.htm)式目录，为了达到编制目录的目的，必须在描述并收藏数据的内容或特色，进而达成协助数据检索的目的。
2. **元数据信息：**

**（1）数据库的元数据信息：DatabaseMetaData。**

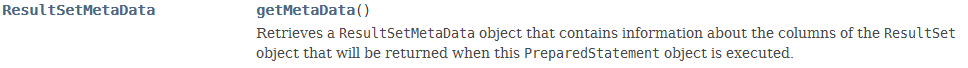
**（2）参数的元数据信息：ParameterMetaData。**

**（3）结果集的元数据信息：ResultSetMetaData。**

1. 三者的获取方法：
2. **DatabaseMetaData对象的获取方法**：利用**Connection的getMetaData()方法**可以获取DatabaseMetaData对象。



1. **ResultSetMetaData对象**的获取方法：利用ResultSet对象的getMetaData()方法或者直接利用**PreparedStatement对象**的getMetaData()方法即可，主要用于查询结果时使用，获取有多少列以及列名等功能。

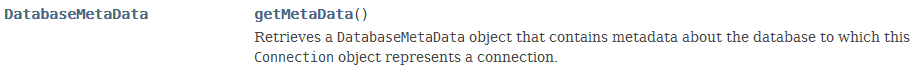


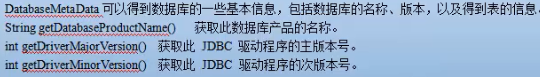


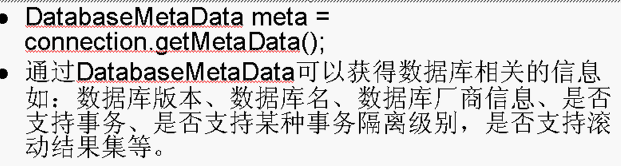
1. **ParameterMetaData对象**的获取方法：利用PreparedStatement对象的getParameterMetaData()方法。



1. **DatabaseMetaData接口**：用来**获取数据库的基本信息**。需要驱动程序实现此接口，利用**Connection接口的getMetaData（）方法**获取其**实现对象**：







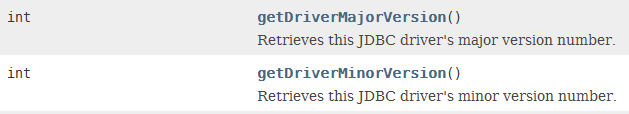
此接口具有**很多字段Field和许多方法Method**：

重要方法如：

1. **getDatabaseProductName（）**



1. **getDriverMajorVersion（）和getDriverMinorVersion（）方法**：返回int值。



1. 方法：**supportsXxx()方法**：返回Boolean值，判断此数据库是否支持某个功能。如**supportsTransaction（）**方法：

判断是否**支持事务**。

1. 示例：

Connection con = (new DbUtil()).getConnection();

DatabaseMetaData dmd = con.getMetaData();

System.out.println(dmd.getDriverName()) ;

System.out.println( dmd.getDriverMajorVersion());

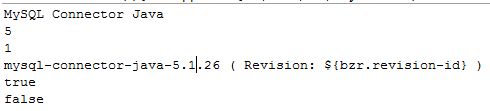
System.out.println(dmd.getDriverMinorVersion());

System.out.println(dmd.getDriverVersion());

System.out.println(dmd.supportsAlterTableWithAddColumn());

System.out.println(dmd.supportsExtendedSQLGrammar());

结果：

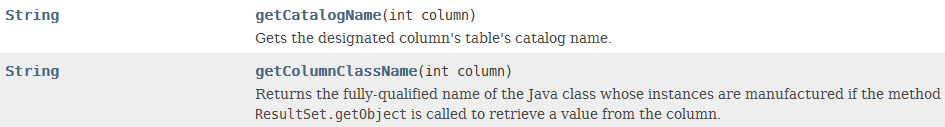


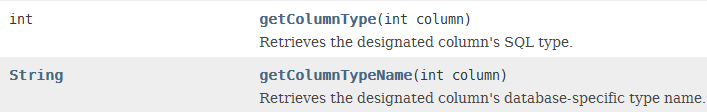
1. **结果集元数据信息**：

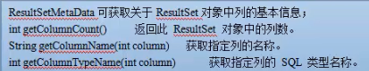
使用**ResultSetMetaData**获取ResultSet对象中的信息。

主要方法是**getXxx方法和判断方法isXxx**。





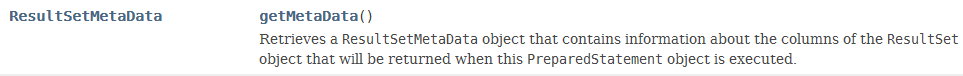




1. **参数元数据信息**：

利用**PreparedStatement接口**的方法获取对象：

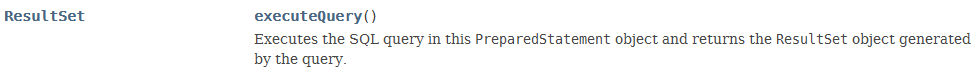
1. **getMetaData（）方法**：获取ResultSetMetaData实现对象。



1. **getParameterMetaData（）方法**：获取ParameterMetaData实现对象。



1. **executeQuery（）方法**：获取ResultSet的实现对象。



1. ResultSetMetaData的示例：

DbUtil du = new DbUtil();

Connection con = du.getConnection();

String sql = "select \* from t\_book";

PreparedStatement psm = con.prepareStatement(sql);

ResultSetMetaData rsmd = psm.getMetaData();

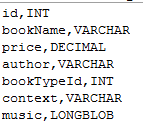
int num = rsmd.getColumnCount();

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

System.out.println(rsmd.getColumnName(i)+","+rsmd.getColumnTypeName(i));

}

结果：



示例2：

public static void main(String[] args) throws Exception {

Connection conn = DriverUtil.getConnection();

String sql= "select id,name,birthday from users where id < 35";

PreparedStatement psm = conn.prepareStatement(sql);

ResultSet rs = psm.executeQuery();

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

String[] cols = new String[3];

**int count = rsmd.getColumnCount();//找出列数**

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

**cols[i-1] = rsmd.getColumnName(i);//找出所有的列名**

System.out.println(cols[i-1]);

}

ArrayList<Map<String,Object>> list = new ArrayList<Map<String, Object>>();

Map<String, Object> map = new HashMap<String, Object>();

while(rs.next()) {

for(int i = 0;i<count;i++) {

map.put(cols[i], rs.getObject(i+1));

}

list.add(map);

}

DriverUtil.close(rs,psm,conn);

System.out.println(list);

}

结果：

