



博客：<https://www.cnblogs.com/HOsystem/p/14116443.html>

## 2、具体内容

当获取了java.sql.Connection接口对象之后，那么其核心目的一定不是只为了连接，而是为了进行数据库的操作，而进行数据库的开发操作应该使用标准SQL语句来完成，所以需要有一个SQL的执行器，而执行器就可以利用Statement接口实现。

### ■Statement简介

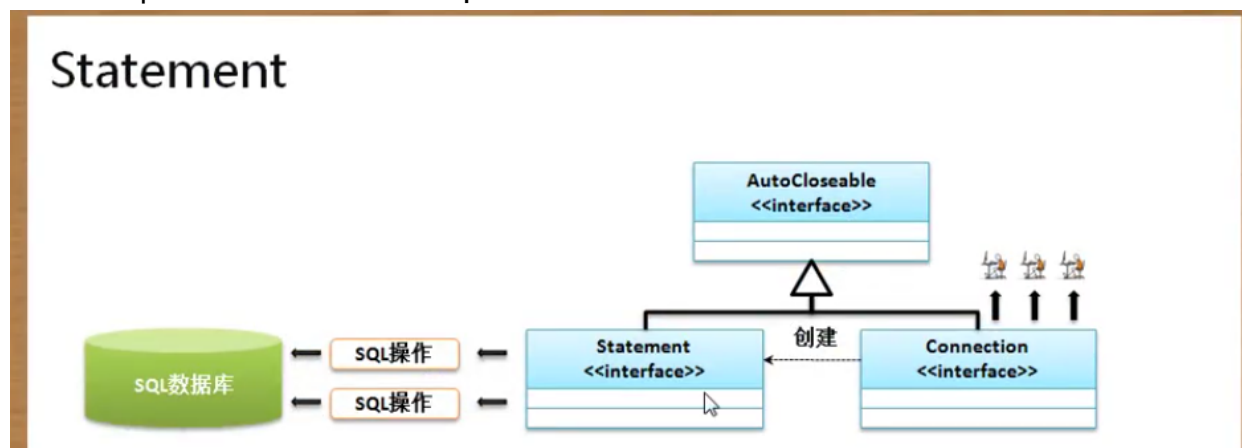
java.sql.Statement是JDBC之中提供的数据库的操作接口，利用其可以实现数据的更新与查询的处理操作，该接口定义如下：

```
public interface Statement extends Wrapper, AutoCloseable
```

该接口是AutoCloseable子接口，所以可以得出结论：每一次进行数据库操作完成之后都应该关闭Statement操作，即：一条SQL的执行一定是一个Statement接口对象，但是如果要想获得Statement接口对象，那么必须依靠Connection接口提供的方法：

·获取Statement接口对象：public Statement createStatement() throws  
SQLException;

|- 此时抛出的SQLException是JDBC数据库开发之中的最大异常；



当获取了Statement接口对象之后，就可以使用SQL进行处理了，而这里面需要两个方法的支持：

·**数据更新处理(INSERT、UPDATE、DELETE):**public int executeUpdate(String sql) throws SQLException;

·**数据查询处理(SELECT、统计查询、复杂查询):**public **ResultSet** executeQuery (String sql) throws SQLException;

这两个数据库的操作方法里面都需要接收SQL的字符串，也就是说Statement接口可以直接使用SQL语句实现开发。

### 范例：定义数据库脚本

```
DROP TABLE news PURGE ;
DROP SEQUENCE news_seq ;
CREATE SEQUENCE news_seq ;
CREATE TABLE news(
    nid      NUMBER ,
    title  VARCHAR2(30) ,
    read  NUMBER ,
    price  NUMBER ,
    content CLOB ,
    pubdate DATE ,
    CONSTRAINT pk_nid PRIMARY KEY(nid)
);
```

随后将利用此表实现数据的更新与查询处理操作。

## ■数据更新操作

在SQL语句之中数据的更新操作一共分为三种：增加(INSERT)、修改(UPDATE)、删除(DELETE)，Statement接口的最大的特点是可以直接执行一个标准的SQL语句。

### 范例：实现数据的增加处理

·增加SQL语法：INSERT INTO 表名称 (字段,字段,...) VALUES(值,值,...);

```
INSERT INTO news(nid,title,read,price,content,pubdate) VALUES (news_seq.nextval,'MLDN-News',10,9.9,'这个春天很美丽，有雾霾，有沙尘暴，有大雨，有大风，还下大雪',TO_DATE('1988-04-05','yyyy-mm-dd')) ;
```

```
package cn.mldn.demo;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;
public class JDBCdemo {
    private static final String DATABASE_DRVIER = "oracle.jdbc.driver.OracleDriver" ;
    private static final String DATABASE_URL = "jdbc:oracle:thin:@localhost:1521:mldn" ;
    private static final String DATABASE_USER = "scott" ;
    private static final String DATABASE_PASSWORD = "tiger" ;
    public static void main(String[] args) throws Exception {
        String sql = " INSERT INTO news(nid,title,read,price,content,pubdate) VALUES "
```

```

        + " (news_seq.nextval,'MLDN-News',10,9.9, "
        + " '这个春天很美丽, 有雾霾, 有沙尘暴, 有大雨, 有大风, 还下大雪', "
        + " TO_DATE('1988-04-05','yyyy-mm-dd'))" ;
    Connection conn = null ; // 每一个Connection接口对象描述的就是一个用户连接
    Class.forName(DATABASE_DRVIER) ; // 向容器之中加载数据库驱动程序
    conn = DriverManager.getConnection(DATABASE_URL, DATABASE_USER,
    DATABASE_PASSWORD) ;
    Statement stmt = conn.createStatement() ; // 创建数据库的操作对象
    int count = stmt.executeUpdate(sql) ; // 返回影响的行数
    System.out.println("更新操作影响的数据行数: " + count);
    conn.close(); // 数据库的连接资源有限一定要关闭
}
}

```

对于SQL语句而言, 由于一般都比较长, 所以一定要考虑换行编写, 一旦换行了, 强烈建议在每个字符串的前后多追加一个空格(避免错误带来的麻烦)。

### 范例: 更新操作,在进行更新操作的时候一般都是做条件性的更新;

·更新的SQL语法: UPDATE 表名称 SET 字段=值... WHERE 更新条件 ;

```

UPDATE news SET title='MLDN重大新闻',content='据说四月的最低温度为1度',read=99998
WHERE nid=5 ;

package cn.mldn.demo;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;
public class JDBCdemo {
    private static final String DATABASE_DRVIER = "oracle.jdbc.driver.OracleDriver" ;
    private static final String DATABASE_URL = "jdbc:oracle:thin:@localhost:1521:mldn" ;
    private static final String DATABASE_USER = "scott" ;
    private static final String DATABASE_PASSWORD = "tiger" ;
    public static void main(String[] args) throws Exception {
        String sql = "UPDATE news SET title='MLDN重大新闻', "
        + " content='据说四月的最低温度为1度',read=99998 WHERE nid=5 " ;
        Connection conn = null ; // 每一个Connection接口对象描述的就是一个用户连接
        Class.forName(DATABASE_DRVIER) ; // 向容器之中加载数据库驱动程序
        conn = DriverManager.getConnection(DATABASE_URL, DATABASE_USER,
        DATABASE_PASSWORD) ;
        Statement stmt = conn.createStatement() ; // 创建数据库的操作对象
        int count = stmt.executeUpdate(sql) ; // 返回影响的行数
        System.out.println("更新操作影响的数据行数: " + count);
        conn.close(); // 数据库的连接资源有限一定要关闭
    }
}

```

### 范例: 数据删除

·删除语法: DELETE FROM news WHERE nid IN (11,13,15,17,19) ;;

```

DELETE FROM news WHERE nid IN (11,13,15,17,19) ;

package cn.mldn.demo;
import java.sql.Connection;
import java.sql.DriverManager;

```

```

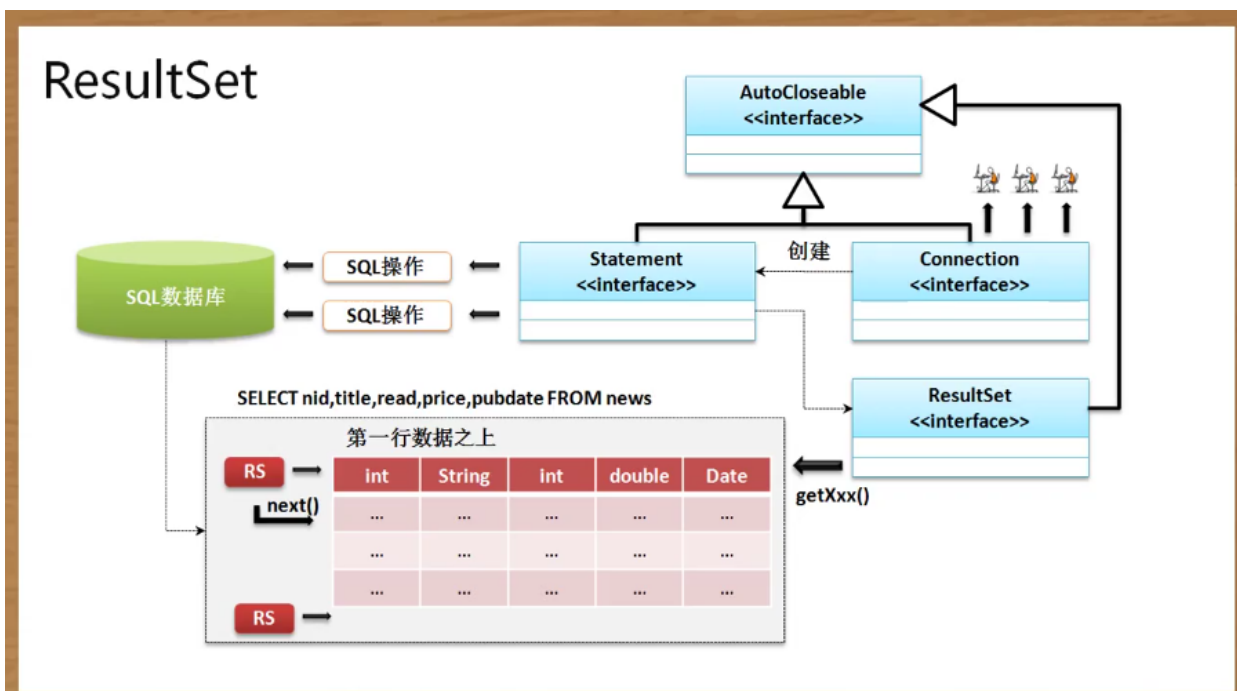
import java.sql.Statement;
public class JDBCdemo {
    private static final String DATABASE_DRIVER = "oracle.jdbc.driver.OracleDriver" ;
    private static final String DATABASE_URL = "jdbc:oracle:thin:@localhost:1521:mldn" ;
    private static final String DATABASE_USER = "scott" ;
    private static final String DATABASE_PASSWORD = "tiger" ;
    public static void main(String[] args) throws Exception {
        String sql = "DELETE FROM news WHERE nid IN (11,13,15,17,19) " ;
        Connection conn = null ; // 每一个Connection接口对象描述的就是一个用户连接
        Class.forName(DATABASE_DRIVER) ; // 向容器之中加载数据库驱动程序
        conn = DriverManager.getConnection(DATABASE_URL, DATABASE_USER,
DATABASE_PASSWORD)
        Statement stmt = conn.createStatement() ; // 创建数据库的操作对象
        int count = stmt.executeUpdate(sql) ; // 返回影响的行数
        System.out.println("更新操作影响的数据行数: " + count);
        conn.close(); // 数据库的连接资源有限一定要关闭
    }
}

```

数据修改里面只需要考虑到不同的SQL语句即可，这也是Statement接口的最大特点(直接执行SQL语句)。

## ■数据查询操作

数据更新主要是接收其影响的数据行数，但是数据查询就比较麻烦了，因为查询一定要将结果返回给程序，由程序来进行结果的处理，所以在java里面通过ResultSet接口来描述查询结果。



### 范例：实现数据的查询处理

```

package cn.mldn.demo;
import java.sql.Connection;

```

```

import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.Date;
public class JDBCdemo {
    private static final String DATABASE_DRIVER = "oracle.jdbc.driver.OracleDriver" ;
    private static final String DATABASE_URL = "jdbc:oracle:thin:@localhost:1521:mldn" ;
    private static final String DATABASE_USER = "scott" ;
    private static final String DATABASE_PASSWORD = "tiger" ;
    public static void main(String[] args) throws Exception {
        // 在程序开发之中SELECT子句后面必须跟上具体的字段名称，写 "*" 的都叫垃圾代码
        String sql = "SELECT nid,title,read,price,content,pubdate FROM news " ;
        Connection conn = null ; // 每一个Connection接口对象描述的就是一个用户连接
        Class.forName(DATABASE_DRIVER) ; // 向容器之中加载数据库驱动程序
        conn = DriverManager.getConnection(DATABASE_URL, DATABASE_USER,
DATABASE_PASSWORD) ;
        Statement stmt = conn.createStatement() ; // 创建数据库的操作对象
        ResultSet rs = stmt.executeQuery(sql) ; // 执行查询
        while (rs.next()) { // 现在如果发现还有数据行未输出
            int nid = rs.getInt(1) ;
            String title = rs.getString(2) ;
            int read = rs.getInt(3) ;
            double price = rs.getDouble(4) ;
            String content = rs.getString(5) ;
            Date pubdate = rs.getDate(6) ;
            System.out.println(nid + "、" + title + "、" + read + "、" + price + "、" +
content + "、" + pubdate);
        }
        conn.close(); // 数据库的连接资源有限一定要关闭
    }
}

```

需要注意的是，ResultSet对象时保存在内存之中的，如果说你查询数据的返回结果过大，程序也将出现问题。