# Day12 JDBC

## 一.JDBC简介

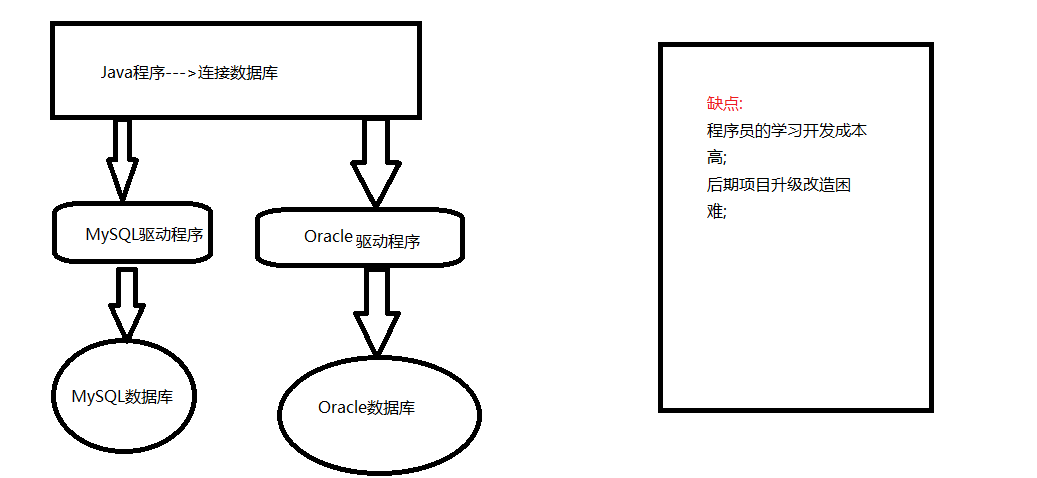
### 1.概念:

JDBC,Java DataBase Connectivity,Java数据库连接.

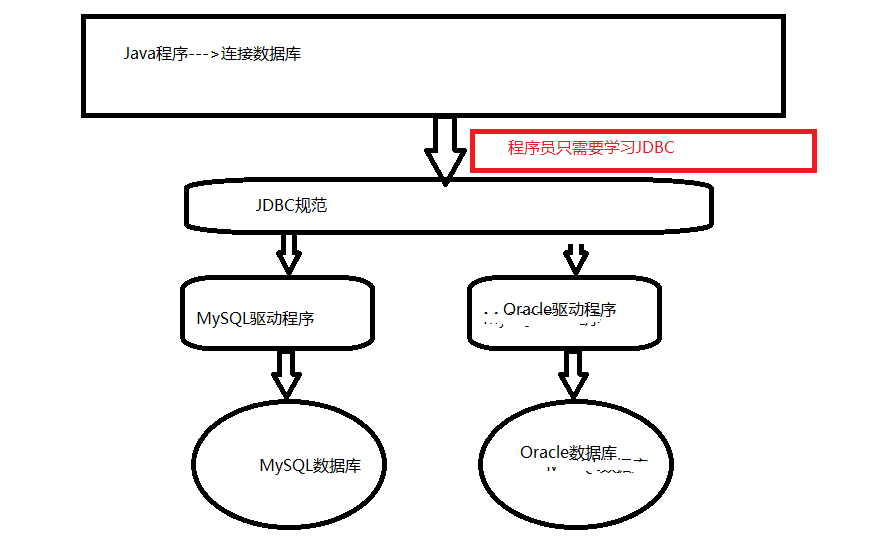
### 2.作用:

为了简化,统一对数据库的操作,Sun公司就指定了一套针对数据库操作的标准规范,这个标准规范就是所谓的JDBC,这个JDBC主要是由一些列的接口来组成,这些接口由不同的数据库厂商来实现,程序员只需要负责去调用这些接口中的方法就可以了.

**如果没有JDBC:**



**JDBC的作用:**



### 3.JDBC的API组成

JDBC的API存在于两个包中:

java.sql;

javax.sql;

在JavaSE中,本身就自带这两个包,所以不需要导入第三方的包.

### 4.驱动程序

MySQL的驱动程序:

Oracle的驱动程序:

SQL Server的驱动程序:

要想能够真正的操作数据库,进行CRUD操作,还必须导入各个数据库的驱动程序包.

## 二.JDBC中的4个核心对象

### 1.DriverManager:驱动管理器.

主要用来加载驱动程序,并且建立与数据库的连接.

**主要的方法:**

DriverManager.registerDriver(new Driver());//加载驱动程序

DriverManager.getConnection(url,user,pass);//建立连接

### 2.Connection:数据库连接对象.

该对象是JDBC中最核心的一个对象.主要用来表示与数据库的连接.

一般情况下,会利用这个Connection对象创建Statement对象.

常用方法:

createStatement();

prepareStatement(sql);

prepareCall(sql);调用存储过程.

### 3.Statement:用来向数据库发送SQL语句的对象.

常用方法:

executeUpdate(sql);

executeQuery(sql);

execute(sql);

...

### 4.ResultSet:结果集,”虚拟表”.

常用方法:

ResultSet.next();

ResultSet.getXXX(key|index);

## 三.JDBC的URL地址

MySQL的URL:

jdbc:mysql://localhost:3306/db01

jdbc:总的协议的名称;

mysql:子协议的名称;

localhost:数据库所在服务器的地址;

3306:MySQL数据库的端口号;

db01:要加载的数据库的名称.

Oracle的URL:

jdbc:oracle:thin:@localhost:1521:db01

## 四.JDBC开发的具体实现过程

### 1.建立数据库,建表,表中添加数据:

### 2.建立Java项目,搭建环境:

JavaSE本身就包含如下两个包,所以不需要导入第三方jar包.

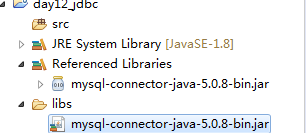
java.sql;

javax.sql;

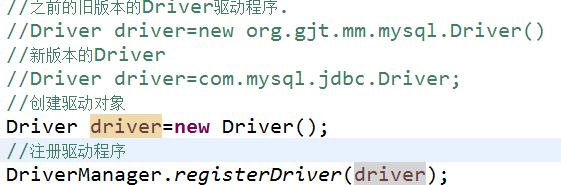
如果要连接具体的数据库,则需要导入相应的数据库jar包.

mysql-connector-java-xxx.jar

结构如下:



### 3.建立数据库驱动



### 4.建立数据库连接

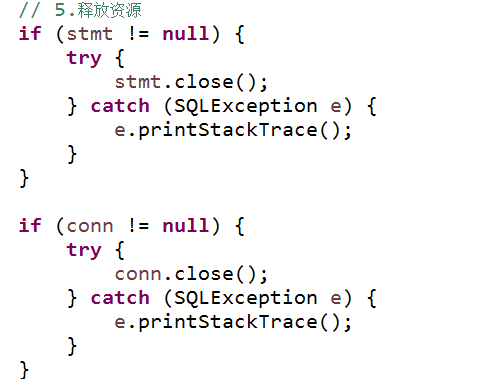


### 5.利用Statement向数据库发送具体的SQL语句



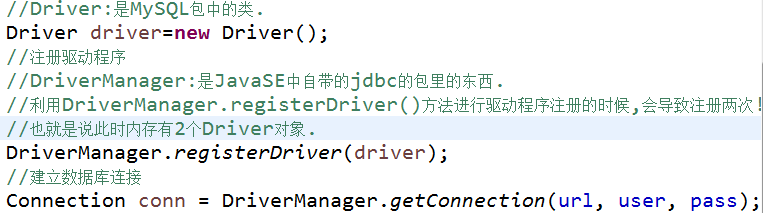
### 6.如果是查询,则从ResultSet中获取具体的结果.

### 7.断开与数据库的连接,释放占用的资源.



## 五.进行数据库驱动,获取数据库连接方式

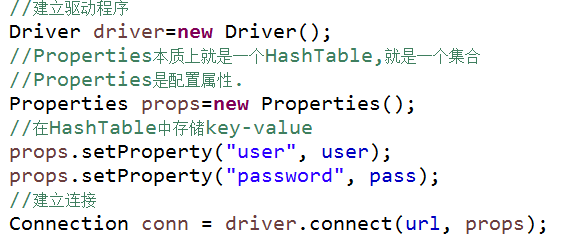
### 1.方式一:



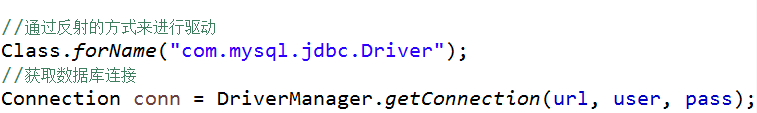
**该方式的弊端:**

1. .会导致Driver驱动程序重复注册,导致内存有2个Driver对象,无谓的占用了内存,造成浪费;
2. .DriverManager属于java.sql包下的类,Driver类属于com.mysql包下的类,在java中引用了MySQL的API方法,一旦有一天把MySQL的jar给去掉了,Driver就会失效.

### 2.方式二:



### 3.方式三:



## 六.Java的单元测试

单元测试用在方法上面.

@Test注解

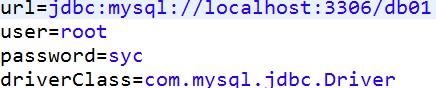
针对测试的方法,有些要求:

1. .必须是public,
2. .没有返回值,
3. .方法中不要带参数.
4. .通常还可以给该方法throws Exception;

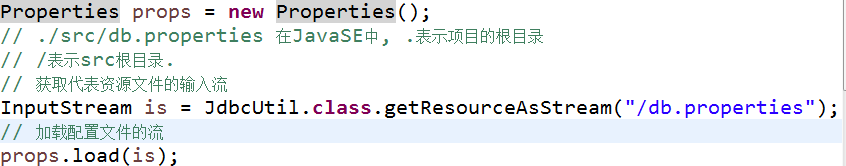
## 七.抽取出一个DbUtil

### 1.建立一个配置文件,db.properties

该配置文件应该在src目录下



### 2.在DbUtil工具类中,加载配置文件:



**注意:**

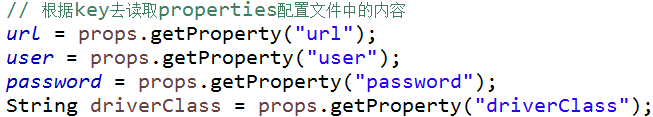
配置文件的路径问题.

/ 表示src目录;

. 表示JavaSE的根目录.

./src/db.properties 配置文件的完整路径

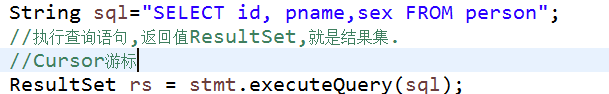
### 3.读取配置文件的内容:



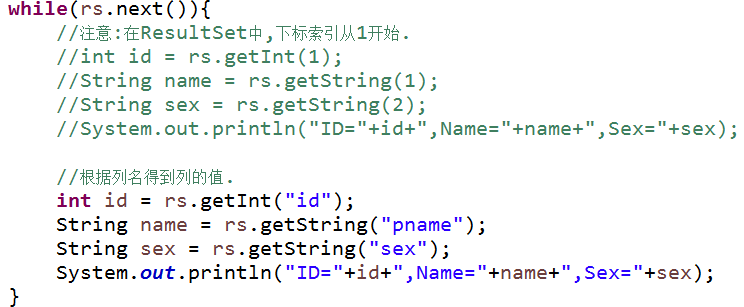
## 八.ResultSet总结

游标...

### 1.获取结果集:

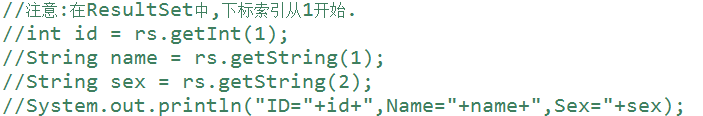


### 2.遍历结果集:

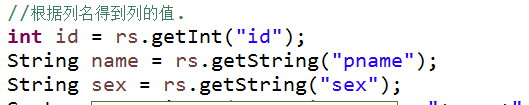


### 3.获取ResultSet中数据内容的方式:

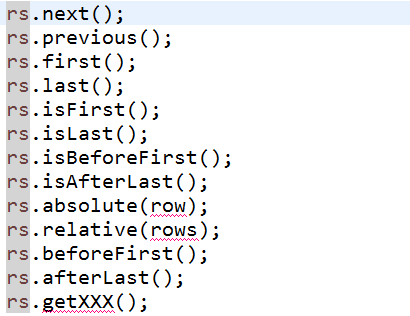
#### 3.1 根据列的索引来获取



#### 3.2 根据列名来获取



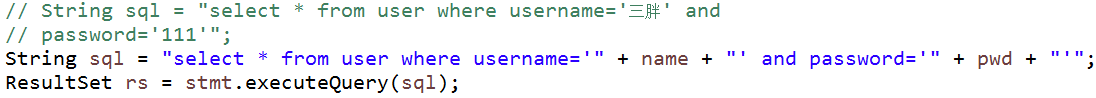
#### 4.常用方法



## 九.SQL注入攻击

实现:

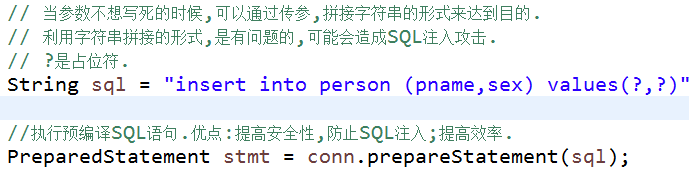




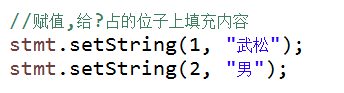
## 十.预编译的PreparedStatement

为了解决SQL注入攻击,可以使用PreparedStatement.

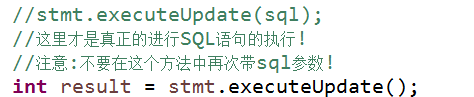
### 1.PreparedStatement具体使用



### 2.具体赋值



### 3.真正执行SQL语句



作业:

利用PreparedStatement来实现update,delete,query操作

update:

修改id=5的用户信息;

delete:

删除id=10的用户信息;

query:

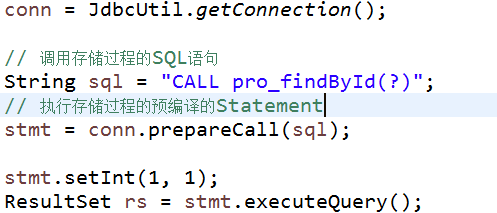
查询id=10的用户信息.

## 十一.调用存储过程

### 首先数据库中应该有一个存储过程|函数;

### Java代码实现:

案例:调用带有输入参数的存储过程.

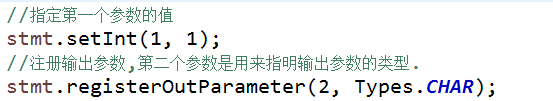


案例二:调用带有输入输出参数的存储过程

进行预编译:



给参数赋值:



执行SQL语句:



获取结果:

