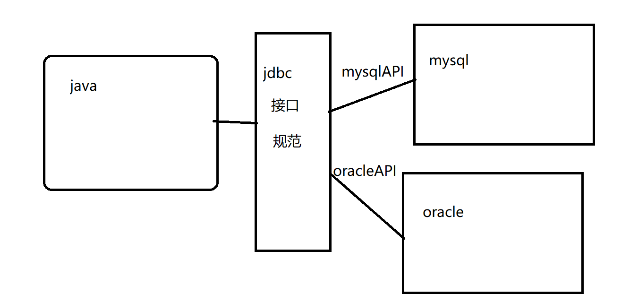
JDBC

# JDBC简介

由于各大数据库厂商提供的驱动(操作数据库的jar包)，各不相同，导致开发人员学习成本改哦，sun公司为简化操作，提供一套规范，本质是一大堆接口，要求各数据库厂商实现这套规范，这个规范就是JDBC，只要学会jdbc这套接口，即可操作各类数据库。



# JAVA JDBC

Jdbc由二个包组成，分别是java.sql和javax.sql，目前已经集成到javase中

但jdbc只是一套接口，真正实现在具体的数据库的jar包中，所以连接数据库需要导入具体的驱动jar包。

JDBC提供了对Java程序员，数据库厂商及第三方中间件厂商的API。

# JDBC相关类或接口

### DriverManager类

获取连接注册驱动

### Connection接口

代表数据库连接的对象，客户端与数据库的交互都是通过Connection对象完成的

CreateStatement():创建传输器，用与发送sql和返回结果

### Statement接口

用于向数据库发送SQL语句的方法

#### ExecuteQuery(String sql):

执行查询操作，返回ResultSet结果集

#### ExecuteUpdate(String sql)：

执行增删改操作，返回的时被影响的行数

#### 子接口

##### [CallableStatement](mk:@MSITStore:E:\JAVA\JDK_API_1_6_zh_CN.CHM::/java/sql/CallableStatement.html)

用于执行 SQL 存储过程的接口

##### [PreparedStatement](mk:@MSITStore:E:\JAVA\JDK_API_1_6_zh_CN.CHM::/java/sql/PreparedStatement.html)

注入攻击：用户提交一些特殊字符，后台在拼接用户名和密码形成SQL语句时，特殊字符有了特殊语义，使SQL语句语义改变，获得了不可预知的结果。

该接口为预编译，可解决注入攻击。

表示预编译的 SQL 语句的对象。

### ResultSet接口

ResultSet 对象具有指向其当前数据行的光标。最初，光标被置于第一行之前。next 方法将光标移动到下一行；因为该方法在 ResultSet 对象没有下一行时返回 false，所以可以在 while 循环中使用它来迭代结果集。

此对象用于封装查询结果，封装时类似于表格的方式，ResultSet中维护了一个游标，初始位置在第一行之前，调用next()方法，游标向下移动一行

#### 操作游标的方法：

next()：移动到下一行，有数据返回true，没有返回false

previous()：移动到上一行

absolute(int row)：移动到指定行

beforeFirst()：移动到第一行的前面

afterLast()：移动到最后一行的后面

#### 获取数据的方法：

getInt(String col\_name)

getString(String col\_name)

getDouble(String col\_name)

getObject(String col\_name)

# JAVA连接数据库步骤

### 加载驱动

#### 代码：

Class.*forName*("com.mysql.jdbc.Driver");

#### 说明：

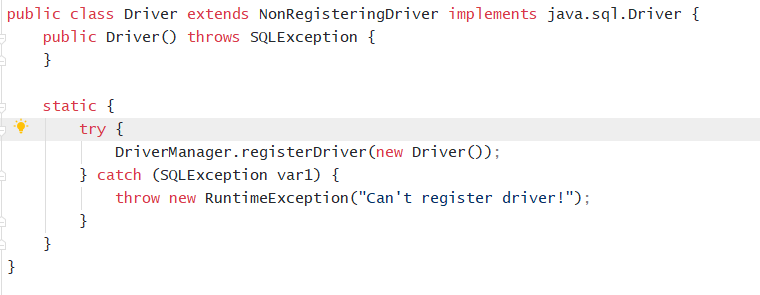
通过反射加载 com.mysql.jdbc.Driver类，这个类就是在我们导入的MySql JAR包中，该类中有静态代码块,一旦反射，static代码块执行，构造该类

所以理论上可以直接

DriverManager.*registerDriver*(new com.mysql.jdbc.Driver());

但是这里new了一次，static代码块new了一次，重复了

编译时异常ClassNotFoundException



### 创建连接

#### 代码

*Connection* conn=DriverManager.*getConnection*

("jdbc:mysql://localhost:3306/mydb1?useSSL=false&serverTimezone=UTC","root","admin");

#### 说明

##### URL(类似httpurl) "jdbc:mysql://localhost:3306/mydb1"

协议：jdbc:mysql

地址：localhost本机，如果是本机可忽略不写

端口：mysql 为3306如果是默认的可以省略不写

数据库名：mysb1

*参数：?参数名1=参数值1&参数名2……和URL类似*

##### 用户名：root

##### 密码：admin

##### 注意：

编译时异常：SQLException

URL省略后：jdbc:mysql:///mydb1

### 创建传输器

#### 代码1：

*Statement* statement=conn.createStatement();

编译时异常：SQLException

#### 代码2：

PrepareStatementstatement=conn.prepareStatement

(“select \* from exam WHERE USER=?”);

编译时异常：SQLException

##### “?”为占位符

##### 传值 :

//注意序号从1开始

statement.setString(1,"123");

//表示第一个问号的值替换为123

最终语句：“select \* from exam WHERE USER=’123’”

##### 特点(Statement相反)

预编译：效率高

提前将SQL骨架编译

防注入攻击：

预编译：就是定死了，后面不管填什么,都当作字符串，不能当作sql语句

SQL语句固定，不灵活

### 发送SQL

#### *Statement发送sql*

*ResultSet* resultSet= statement.executeQuery("select \* from exam;");

编译时异常：SQLException

#### *PrepareStatement发送sql*

*ResultSet* resultSet= = statement.executeQuery();

### 处理结果

#### 代码：

while (resultSet.next()){  
 System.*out*.println(resultSet.getString("name"));

或者getString(1) 第一列数据

//注意序号从1开始  
}

#### 说明

初始时指针指向表头，调用next()方法移动指针，第一次就是第一行记录，如果没有记录，next()返回false

### 释放资源

#### 代码



#### 从后往前依次关闭

先关闭ResultSet->Statement->Connection

Connectiong是非常稀有的资源，应尽量晚创建、早释放

当生成 ResultSet 对象的 Statement 对象关闭、重新执行或用来从多个结果的序列获取下一个 结果时，ResultSet 对象将自动关闭。

# 数据库连接池(C3P0)

### C3p0常见连接池，性能优秀开源

### 会读取默认配置文件，配置文件：c3p0.properties或c3p0-config.xml

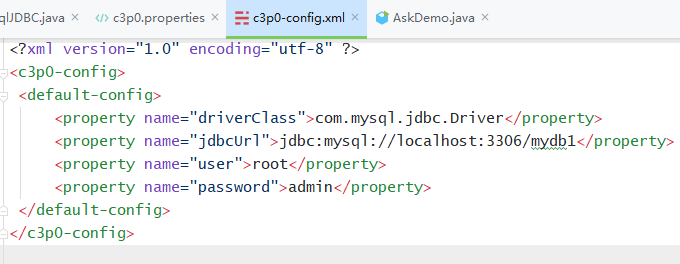
### 使用

#### 配置文件内容

Properties:

c3p0.driverClass=com.mysql.jdbc.Driver  
c3p0.jdbcUrl=jdbc:mysql://localhost:3306/mydb1  
c3p0.user=root  
c3p0.password=admin

XML:



#### 创建连接池对象

ComboPooledDataSource pool = new ComboPooledDataSource();

#### 获取Connection对象

*Connection* conn=pool.getConnection();

#### 关闭连接

提问：Connection使用完不是应该放回队列中吗，我看老师在finally中直接关闭了?

回答：在后面老师会讲设计模式，和代理模式有关？讲了后更好理解

4.22已讲，装饰模式