

Java的JDBC编程

本节目标

- 数据库驱动
- JDBC的概念及作用
- 掌握JDBC的工作原理
- 掌握JDBC中几个常用接口和类
- 掌握基于数据库的应用程序开发流程

1. 数据库编程的必备条件

- 编程语言，如Java, C、C++、Python等
- 数据库，如Oracle, MySQL, SQL Server等
- 数据库驱动包：不同的数据库，对应不同的编程语言提供了不同的数据库驱动包，如：MySQL提供了Java的驱动包mysql-connector-java，需要基于Java操作MySQL即需要该驱动包。同样的，要基于Java操作Oracle数据库则需要Oracle的数据库驱动包ojdbc。

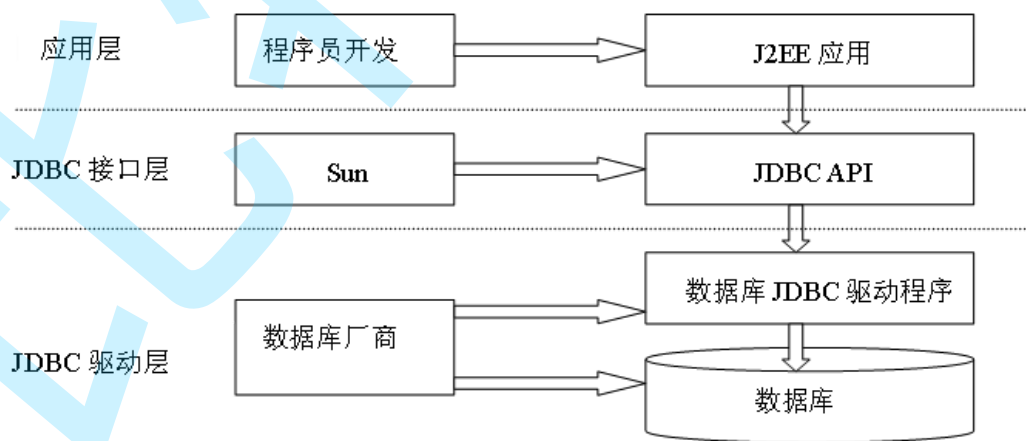
2. Java的数据库编程：JDBC

JDBC，即Java Database Connectivity，java数据库连接。是一种用于执行SQL语句的Java API，它是Java中的数据库连接规范。这个API由 `java.sql.*`、`javax.sql.*` 包中的一些类和接口组成，它为Java开发人员操作数据库提供了一个标准的API，可以为多种关系数据库提供统一访问。

3. JDBC工作原理

JDBC 为多种关系数据库提供了统一访问方式，作为特定厂商数据库访问API的一种高级抽象，它主要包含一些通用的接口类。

JDBC访问数据库层次结构：



JDBC优势：

- Java语言访问数据库操作完全面向抽象接口编程
- 开发数据库应用不用限定在特定数据库厂商的API
- 程序的可移植性大大增强

4. JDBC使用

4.1 JDBC开发案例

- 准备数据库驱动包，并添加到项目的依赖中：

在项目中创建文件夹lib，并将依赖包mysql-connector-java-5.1.47.jar复制到lib中。再配置该jar包到本项目的依赖中：右键点击项目Open Module Settings，在Modules中，点击项目，配置Dependencies，点击+，JARS or Directories，将该lib文件夹配置进依赖中，表示该文件夹下的jar包都引入作为依赖。

- 建立数据库连接

```
// 加载JDBC驱动程序：反射，这样调用初始化com.mysql.jdbc.Driver类，即将该类加载到JVM方法区，并执行该类的静态方法块、静态属性。
```

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

```
// 创建数据库连接
```

```
Connection connection =  
DriverManager.getConnection("jdbc:mysql://localhost:3306/test?  
user=root&password=root&useUnicode=true&characterEncoding=UTF-8");
```

```
//MySQL数据连接的URL参数格式如下：
```

```
jdbc:mysql://服务器地址:端口/数据库名?参数名=参数值
```

- 创建操作命令 (Statement)

```
Statement statement = connection.createStatement();
```

- 执行SQL语句

```
ResultSet resultSet= statement.executeQuery(  
    "select id, sn, name, qq_mail, classes_id from student");
```

- 处理结果集

```
while (resultSet.next()) {  
    int id = resultSet.getInt("id");  
    String sn = resultSet.getString("sn");  
    String name = resultSet.getString("name");  
    int classesId = resultSet.getInt("classes_id");  
    System.out.println(String.format("Student: id=%d, sn=%s, name=%s, classesId=%s", id, sn, name, classesId));  
}
```

- 释放资源 (关闭结果集，命令，连接)

```
//关闭结果集
```

```
if (resultSet != null) {  
    try {  
        resultSet.close();  
    } catch (SQLException e) {  
        e.printStackTrace();  
    }  
}
```

```

}
//关闭命令
if (statement != null) {
    try {
        statement.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
//关闭连接命令
if (connection != null) {
    try {
        connection.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
}

```

4.2 JDBC使用步骤总结

1. 创建数据库连接Connection
2. 创建操作命令Statement
3. 使用操作命令来执行SQL
4. 处理结果集ResultSet
5. 释放资源

5. JDBC常用接口和类

5.1 JDBC API

在Java JDBC编程中对数据库的操作均使用JDK自带的API统一处理，通常与特定数据库的驱动类是完全解耦的。所以掌握Java JDBC API（位于 `java.sql` 包下）即可掌握Java数据库编程。

5.2 数据库连接Connection

Connection接口实现类由数据库提供，获取Connection对象通常有两种方式：

- 一种是通过DriverManager（驱动管理类）的静态方法获取：

```

// 加载JDBC驱动程序
Class.forName("com.mysql.jdbc.Driver");

// 创建数据库连接
Connection connection = DriverManager.getConnection(url);

```

- 一种是通过DataSource（数据源）对象获取。**实际应用中会使用DataSource对象。**

```

DataSource ds = new MySQLDataSource();
((MySQLDataSource) ds).setUrl("jdbc:mysql://localhost:3306/test");
((MySQLDataSource) ds).setUser("root");
((MySQLDataSource) ds).setPassword("root");
Connection connection = ds.getConnection();

```

- 以上两种方式的区别是：

1. DriverManager类来获取的Connection连接，是无法重复利用的，每次使用完以后释放资源时，通过connection.close()都是关闭物理连接。
2. DataSource提供连接池的支持。连接池在初始化时将创建一定数量的数据库连接，这些连接是可以复用的，每次使用完数据库连接，释放资源调用connection.close()都是将Connction连接对象回收。

5.3 Statement对象

Statement对象主要是将SQL语句发送到数据库中。JDBC API中主要提供了三种Statement对象。

Statement

- 用于执行不带参数的简单SQL语句

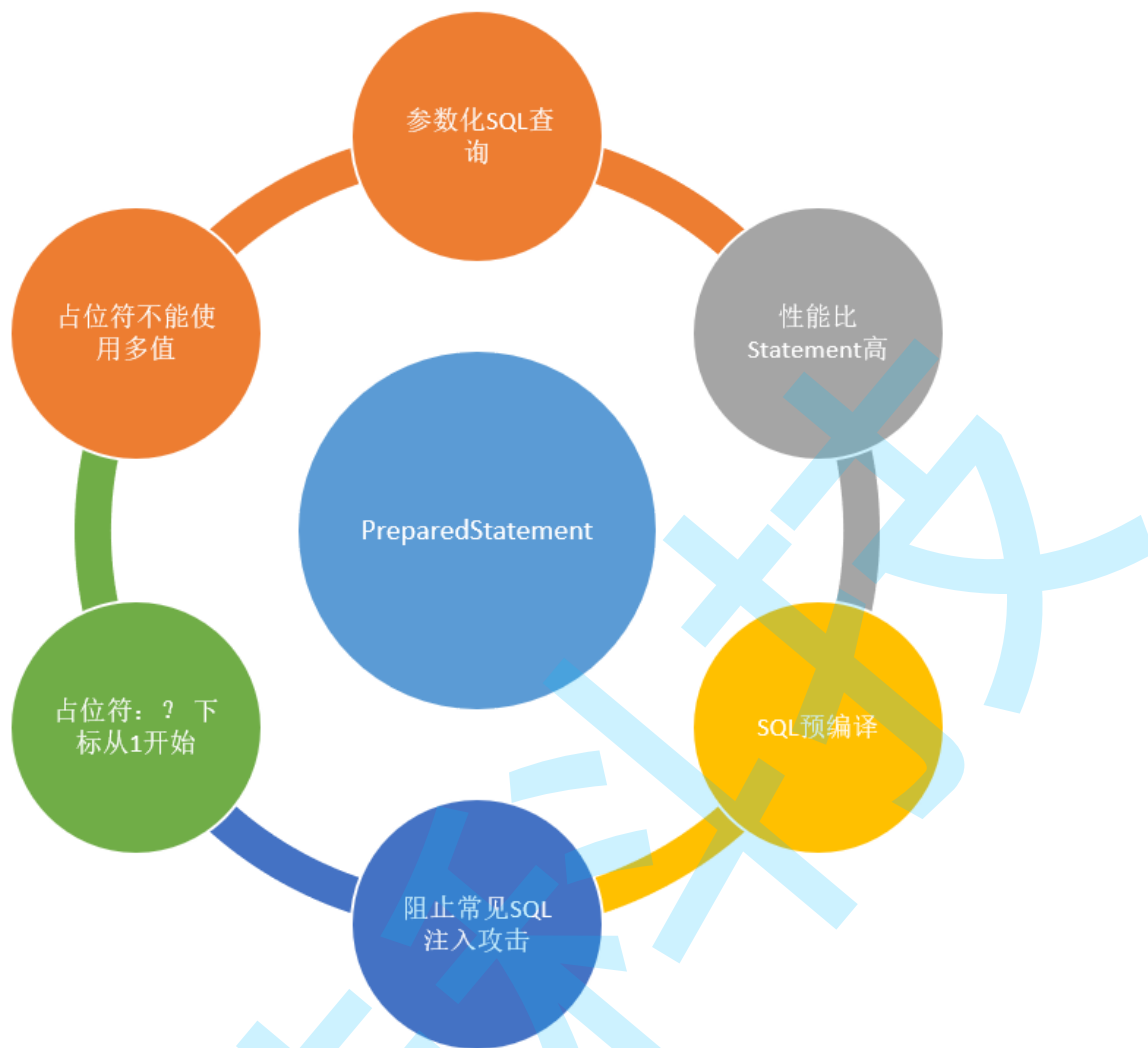
PreparedStatement

- 用于执行带或者不带参数的SQL语句
- SQL语句会预编译在数据库系统
- 执行速度快于Statement对象

CallableStatement

- 用于执行数据库存储过程的调用

实际开发中最常用的是PreparedStatement对象，以下对其的总结：



主要掌握两种执行SQL的方法：

- `executeQuery()` 方法执行后返回单个结果集的，通常用于select语句
- `executeUpdate()`方法返回值是一个整数，指示受影响的行数，通常用于update、insert、delete语句

5.4 ResultSet对象

ResultSet对象它被称为结果集，它代表符合SQL语句条件的所有行，并且它通过一套getXXX方法提供了对这些行中数据的访问。

ResultSet里的数据一行一行排列，每行有多个字段，并且有一个记录指针，指针所指的数据行叫做当前数据行，我们只能来操作当前的数据行。我们如果想要取得某一条记录，就要使用ResultSet的`next()`方法,如果我们想要得到ResultSet里的所有记录，就应该使用while循环。

6. 应用案例

技术知识点：

- JDBC API的CRUD
- JDBC API的事务控制

功能要求：

- 学生表
 - 自动化2019级5班新增一名同学程咬金
 - 修改该同学的班级为中文系2019级3班

- 查询所有中文系2019级3班的同学
- 删除名叫程咬金的同学
- 成绩表
 - 新增许仙同学的成绩：英文80分，Java65分，计算机原理76分，语文59分
 - 修改许仙同学的成绩：英文81分
 - 删除许仙同学的语文成绩
 - 查询中文系2019级3班同学的成绩

7. 内容重点总结

JDBC使用步骤：

1. 创建数据库连接Connection
 - DriverManager创建
 - DataSource获取
2. 创建操作命令Statement
 - PreparedStatement
3. 使用操作命令来执行SQL

```
// 查询操作
preparedStatement.executeQuery();

// 新增、修改、删除操作
preparedStatement.executeUpdate();
```

4. 处理结果集ResultSet

```
while (resultSet.next()) {
    int xxx = resultSet.getInt("xxx");
    String yyy= resultSet.getString("yyy");
    ...
}
```

5. 释放资源

```
try {
    if(resultSet != null){
        resultSet.close();
    }
    if(preparedStatement != null){
        preparedStatement.close();
    }
    if(connection != null){
        connection.close();
    }
} catch (SQLException e) {
    e.printStackTrace();
    throw new RuntimeException("数据库错误");
}
```

面试问答：

1. 数据库连接有哪些方式？分别有什么区别

2. 数据库Statement和PreparedStatement有什么区别？

8. 课后作业

- 图书管理系统
 - 新增貂蝉同学的借阅记录：诗经，从2019年9月25日17:50到2019年10月25日17:50
 - 查询计算机分类下的图书借阅信息
 - 修改图书《深入理解Java虚拟机》的价格为61.20
 - 删除id最大的一条借阅记录