# 阶段4 - JDBC

### 上节课回顾

• 数据库是什么

• 数据库中的数据是怎么进行保存的

• 创建使用Mysql数据库、表

• Mysql的基本操作

• Mysql中常见的数据类型

• 常用sql语句

• 查询语句：as，条件查询，模糊查询，连接查询，排序(升序降序)，分组，去重，范围

• 部分函数使用

### 二、作业点评

### 三、本次课知识点，重点难点

【知识点目标】

• 使用JDBC操作数据库

• JDBC API

• Statement和PreparedStatement

• ConnectionFactory

【重点】

• JDBC API

【难点】

• Statement和PreparedStatement

### 四、本次课程任务知识点讲解

JDBC (Java DB Connection)---Java数据库连接

JDBC是一种可用于运行SQL语句的JAVA API（ApplicationProgramming Interface应用程序设计接口）。它由一些Java语言编写的类和界面组成。

JDBC为数据库应用开发者和数据库前台工具开发者提供了一种标准的应用程序设计接口，使开发者能够用纯JAVA语言编写完整的数据库应用程序。

JDBC代表JAVA数据库连接。

自从Java语言于1995年5月正式发布以来，Java风靡全球。出现大量的用java语言编写的程序。当中也包含数据库应用程序。因为没有一个Java语言的API，编程人员不得不在Java程序中添加C语言的ODBC函数调用。这就使非常多Java的优秀特性无法充分发挥，比方平台无关性、面向对象特性等。随着越来越多的编程人员对Java语言的日益喜爱。越来越多的公司在Java程序开发上投入的精力日益添加，对java语言接口的訪问数据库的API的要求越来越强烈。也因为ODBC的有其不足之处。比方它并不easy使用。没有面向对象的特性等等，SUN公司决定开发一Java语言为接口的数据库应用程序开发接口。

在JDK1．x版本号中，JDBC仅仅是一个可选部件。到了JDK1．1发布时，SQL类包（也就是JDBCAPI）就成为Java语言的标准部件。

二、JDBC技术概述

JDBC是一种可用于运行SQL语句的JavaAPI（ApplicationProgrammingInterface。应用程序设计接口）。

通过使用JDBC，开发者能够非常方便地将SQL语句传送给差点儿不论什么一种数据库。

也就是说，开发者能够不必写一个程序訪问Sybase，写还有一个程序訪问Oracle。再写一个程序訪问Microsoft的SQLServer。

用JDBC写的程序能够自己主动地将SQL语句传送给对应的数据库管理系统（DBMS）。不但如此，使用Java编写的应用程序能够在不论什么支持Java的平台上执行，不必在不同的平台上编写不同的应用。Java和JDBC的结合能够让开发者在开发数据库应用时真正实现“WriteOnce。RunEverywhere。”

**JDBC 编程步骤**

加载驱动程序：

**通过JDBC操作数据库——步骤：**

第1步：注冊驱动 (仅仅做一次)

第2步：建立连接(Connection)

第3步：创建运行SQL的语句(Statement)

第4步：运行语句

第5步：处理运行结果(ResultSet)

第6步：释放资源

使用JDBC第一步：载入驱动

注冊驱动有三种方式：

1. Class.forName(“com.mysql.jdbc.Driver”);

推荐这样的方式，不会对详细的驱动类产生依赖

2. DriverManager.registerDriver(com.mysql.jdbc.Driver);

会对详细的驱动类产生依赖

3. System.setProperty(“jdbc.drivers”, “driver1:driver2”);

尽管不会对详细的驱动类产生依赖；但注冊不太方便。所以非常少使用

使用JDBC第二步：建立连接

通过Connection建立连接，Connection是一个接口类。其功能是与数据库进行连接（会话）。

建立Connection接口类对象：

Connection conn =DriverManager.getConnection(url, user, password);

当中URL的格式要求为：

JDBC:子协议:子名称//主机名:port/数据库名？属性名=属性值&…

如："jdbc:mysql://localhost:3306/test“

user即为登录数据库的username，如root

password即为登录数据库的密码，为空就填””

使用JDBC第三步：创建运行对象

运行对象Statement负责运行SQL语句。由Connection对象产生。

Statement接口类还派生出两个接口类PreparedStatement和CallableStatement，这两个接口类对象为我们提供了更加强大的数据訪问功能。

创建Statement的语法为：

Statement st = conn.createStatement();

使用JDBC第四步：运行SQL语句

运行对象Statement提供两个经常使用的方法来运行SQL语句。

executeQuery(Stringsql),该方法用于运行实现查询功能的sql语句。返回类型为ResultSet（结果集）。

如：ResultSet  rs =st.executeQuery(sql);

executeUpdate(Stringsql),该方法用于运行实现增、删、改功能的sql语句，返回类型为int，即受影响的行数。

如：int flag = st.executeUpdate(sql);

使用JDBC第五步：处理运行结果

ResultSet对象

ResultSet对象负责保存Statement运行后所产生的查询结果。

结果集ResultSet是通过游标来操作的。

游标就是一个可控制的、能够指向随意一条记录的指针。

有了这个指针我们就能轻易地指出我们要对结果集中的哪一条记录进行改动、删除，或者要在哪一条记录之前插入数据。一个结果集对象中仅仅包括一个游标。

使用JDBC 第六步——释放资源

Connection对象的close方法用于关闭连接，并释放和连接相关的资源。

接口

PreperedStatement接口

PreperedStatement从Statement扩展而来。

须要多次运行SQL语句，能够使用PreparedStatement。

PreparedStatement能够对SQL语句进行预编译

而且能够存储在PreparedStatement对象中。当多次运行SQL语句时能够提高效率。

作为Statement的子类，PreparedStatement继承了Statement的全部函数。

创建PreperedStatement

PreparedStatementstr=con.prepareStatement("update user set id=? where username=?”);//此处?为通配符

其它的CRUD方法和Statement基本一致。

CallableStatement接口

CallableStatement类继承了PreparedStatement类，他主要用于运行SQL存储过程。

在JDBC中运行SQL存储过程须要转义。

JDBC API提供了一个SQL存储过程的转义语法：

{call<procedure-name>[<arg1>,<arg2>, ...]}

procedure-name：是所要运行的SQL存储过程的名字

[<arg1>,<arg2>, ...]：是相相应的SQL存储过程所须要的參数

ResultSetMeta接口

前面使用ResultSet接口类的对象来获取了查询结果集中的数据。

但ResultSet功能有限，假设我们想要得到诸如查询结果集中有多少列、列名称各自是什么就必须使用ResultSetMeta接口了。

ResultSetMeta是ResultSet的元数据。

元数据就是数据的数据， “有关数据的数据”或者“描写叙述数据的数据”。

ResultSetMeta的使用

获得元数据对象

ResultSetMetaData rsmd=rst.getMetaData();

返回全部字段的数目

public int getColumCount() throwsSQLException

依据字段的索引值取得字段的名称

public String getColumName (int colum)throws SQLException

依据字段的索引值取得字段的类型

public String getColumType (int colum)throws SQLException

PreperedStatement接口和Statement的差别

(1) 使用Statement发送和运行sql语句

Statement stmt = con.creatStatement();//载入时，无參数。不编译sql语句

String sql = "selete \* from emp";

ResultSet rs = stmt.executeQuery(sql);//运行时编译sql语句。返回查询结果集

(2) 使用PreparStatement发送和运行sql语句

String sql = "selete \* from emp";

PreparStatement ps=con.prepareStatement(sql);//载入时就编译sql语句

ResultSet rs = ps.executeQuery();//此处运行时，不须要传參

还有就是

Statement stmt；支持能够通过字符串拼接(来传递參数)，如StringdeleteStu = "delete from student where id="+id;但easyＳＱＬ注入

PreparedStatement   ps; 不支持字符串拼接，提供？通配符传递參数。更安全；如String deleteStu = "delete from student where id=?

";

ps.setInt(1, id);  //设置第一个？通配值为id

JDBC事务

何谓事务？

所谓事务是指一组逻辑操作单元，使数据从一种状态变换到还有一种状态。

事务的ACID属性：

1.原子性（Atomicity）：指事务是一个不可切割的工作单位，事务中的操作要么都发生。要么都不发生。

2.一致性（Consistency）事务必须使数据库从一个一致性状态变换到另外一个一致性状态。

3.隔离性（Isolation）事务的隔离性是指一个事务的运行不能被其它事务干扰。

4.持久性（Durability）持久性是指一个事务一旦被提交。它对数据库中数据的改变就是永久性的，接下来的其它操作和数据库故障不应该对其有不论什么影响。

JDBC中使用COMMIT 和 ROLLBACK语句来实现事务。

COMMIT用于提交事务，ROLLBACK用于数据库回滚。

通过这两个语句能够确保数据完整性。数据改变被提交之前预览；将逻辑上相关的操作分组。

RollBack语句数据改变被取消使之再一次事务中的改动前的数据状态能够被恢复。

为了让多个 SQL 语句作为一个事务运行，基本步骤为：

先调用 Connection 对象的 setAutoCommit(false); 以取消自己主动提交事务。

然后在全部的 SQL 语句都成功运行后，调用 commit();

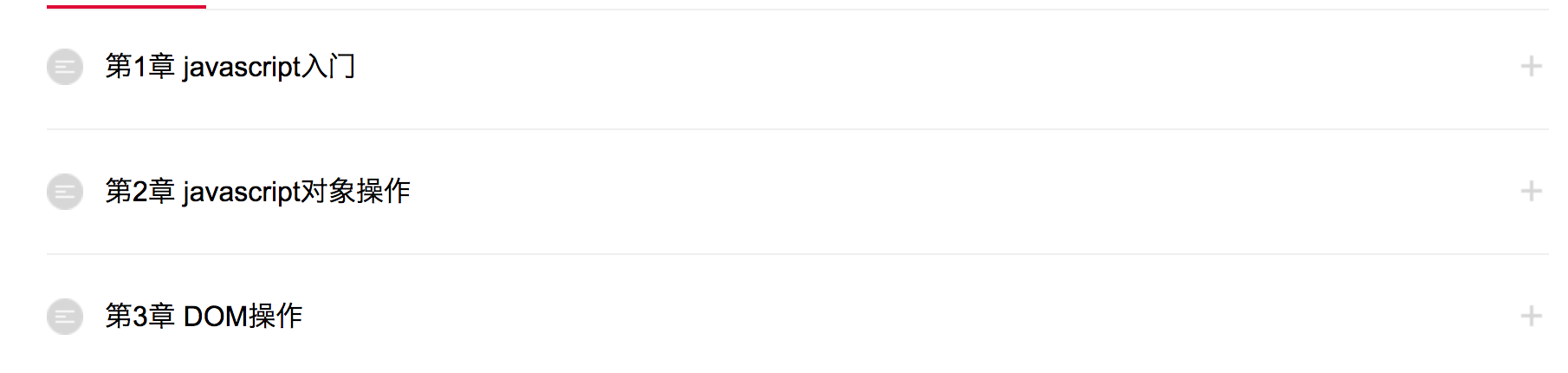
假设在出现异常时，能够调用 rollback();使方法回滚事务。

### 五、作业讲解

### 六、当日课程内容总结

### 本次直播课(后)复习巩固补充题

### 八、下次自主学习任务布置



需要通过平台提交作业。完成平台上所布置的作业。