数据库的学习

（odbc和jdbc的区别）

ODBC（Open Database Connectivity）是一组对数据库访问的标准API，这些API通过SQL来完成大部分任务，而且它本身也支持SQL语言，支持用户发来的SQL。ODBC定义了访问数据库API的一组规范，这些API独立于形色各异的DBMS和编程语言。

也就是说，一个基于ODBC的应用程序，对数据库的操作不依赖任何DBMS，不直接与DBMS打交道，所有的数据库操作由对应的DBMS的ODBC驱动程序完成。不论是SQL Server、Access还是Oracle数据库，均可用ODBC API进行访问。

由此可见，ODBC的最大优点是能以统一的方式处理所有的数据库。

JDBC（JavaDatabase Connectivity）是Java与数据库的接口规范，JDBC定义了一个支持标准SQL功能的通用低层API，它由Java 语言编写的类和接口组成，旨在让各数据库开发商为Java程序员提供标准的数据库API。

JDBC API定义了若干Java中的类，表示数据库连接、SQL指令、结果集、数据库元数据等。它允许Java程序员发送SQL指令并处理结果

**共同点** ：并不是直接来链接数据库 都是通过驱动进行来管理数据

一般都是四个组件 应用程序、驱动程序管理器、驱动程序和数据源,工作原 理亦大体相同；

**不同点**：Java 可以使用 ODBC，但最好是以JDBC-ODBC桥的形式使用（Java连接总体分为Java直连和JDBC-ODBC桥两种形式）。

那为什么还需要 JDBC？

因为ODBC 不适合直接在 Java 中使用，因为它使用 C 语言接口。从Java 调用本地 C代码在安全性、实现、坚固性和程序的自动移植性方面都有许多缺点。从 ODBC C API 到 Java API 的字面翻译是不可取的。例如，Java 没有指针，而 ODBC 却对指针用得很广泛（包括很容易出错的指针"void \*"）。

另外，ODBC 比较复杂，而JDBC 尽量保证简单功能的简便性，同时在必要时允许使用高级功能。如果使用ODBC，就必须手动地将 ODBC 驱动程序管理器和驱动程序安装在每台客户机上。如果完全用 Java 编写 JDBC 驱动程序则 JDBC代码在所有 Java 平台上（从网络计算机到大型机）都可以自 动安装、移植并保证安全性。

总之，JDBC 在很大程度上是借鉴了ODBC的，从他的基础上发展而来。JDBC 保留了 ODBC 的基本设计特征，因此，熟悉 ODBC 的程序员将发现 JDBC 很容易使用。它们之间最大的区别在于：JDBC 以 Java 风格与优点为基础并进行优化，因此更加易于使用。

具体进行相应的连接操作

、连接

public class DbUtil {

public static Connection getConnection(){

Connection conn=null;

try {

Class.forName("oracle.jdbc.driver.OracleDriver");//找到oracle驱动器所在的类

String url="jdbc:oracle:thin:@localhost:1521:bjpowernode"; //URL地址

String username="drp";

String password="drp";

conn=DriverManager.getConnection(url, username, password);

} catch (ClassNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return conn;

}

<pre name="code" class="java">

public static void close(PreparedStatement pstmt){

if(pstmt !=null){

try {

pstmt.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

}

public static void close(ResultSet rs){

if(rs !=null){

try {

rs.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

}

在运行Class.forName时，须要找到OracleDriver所在的地址，路径为：

1、首先找到

D:\oracle\product\10.2.0\db\_1\jdbc\lib 找到ojdbc14.jar

2、其次再找到 ojdbc14.jar\oracle\jdbc\driver 以下的oraceldriver这样就找到了要使用的驱动程序文件

2、操作数据库--加入

public void addUser(User user){

String sql="insert into t\_user(user\_id,user\_name,PASSWORD,CONTACT\_TEL,EMAIL,CREATE\_DATE)values(?

,?

,?,?,?

,?)"; //?为參数占位符

Connection conn=null;

PreparedStatement pstmt=null; //通常利用PreparedStatement进行操作，性能得到优化

try{

conn=DbUtil.getConnection();

pstmt=conn.prepareStatement(sql);

pstmt.setString(1, user.getUserId());

pstmt.setString(2,user.getUserName());

pstmt.setString(3, user.getPassword());

pstmt.setString(4, user.getContactTel());

pstmt.setString(5,user.getEmail());

//pstmt.setTimestamp(6,new Timestamp(System.currentTimeMillis()));

pstmt.setTimestamp(6, new Timestamp(new Date().getTime()));//获取当前系统时间

pstmt.executeUpdate();//运行增删改操作

}catch(SQLException e){

e.printStackTrace();

}finally{

DbUtil.close(conn);

DbUtil.close(pstmt);

}

}

3、操作数据库--查询

public User findUserById(String userId){

String sql = "select user\_id, user\_name, password, contact\_tel, email, create\_date from t\_user where user\_id=?";

Connection conn=null;

PreparedStatement pstmt=null;

ResultSet rs=null;//定义存放查询结果的结果集

User user=null;

try{

conn=DbUtil.getConnection();

pstmt=conn.prepareStatement(sql);

pstmt.setString(1,userId);

rs=pstmt.executeQuery();//运行查询操作

if(rs.next()){

user=new User();

user.setUserId(rs.getString("user\_Id"));

user.setUserName(rs.getString("user\_name"));

user.setPassword(rs.getString("password"));

user.setContactTel(rs.getString("contact\_Tel"));

user.setEmail(rs.getString("email"));

user.setCreateDate(rs.getTimestamp("create\_date"));

}

}catch(SQLException e){

e.printStackTrace();

}finally{

//按顺序进行关闭

DbUtil.close(rs);

DbUtil.close(pstmt);

DbUtil.close(conn);

}

return user;

}

四、总结

1、PreparedStatement与Statement的简单差别

Statement为一条SQL语句生成运行计划，假设參数值不同。会生成不同的sql语句，运行相应參数值个数的次数。假设仅仅有一条SQL语句运行时，最好採用Statement进行。

PreparedStatement使用绑定变量重用运行计划。不同的參数值相应的查询语句。仅仅会生成一个sql语句。大大提高了运行的效率。是预编译的。在大批量语句操作时，提高了效率，同一时候可採用'?'来代表參数，能够防止SQL注入。安全性更高。

2、与之前的知识进行联系

眼下接触到的额JDBC中的连接对象有Connection这个与之前ODBC的Connection作用同样，为数据库连接对象。而PraparedStatement与Statement与ODBC中的command对象类似。都是用来运行SQL语句的。

查询方法中用到的ResultSet则与之前用到的DataSet或者DataTable功能类似，

1. oracle的下载

2可视化界面

PLsql developer 下载链接<https://www.allroundautomations.com/plsqldev.html>

Oracle数据库主要能对五种不同类型的存储单元： 函数，过程，类型，触发器，包