

实验 1 抽取样本

1. 相关知识

经常使用 JDBC 进行的操作是：与一个数据库建立连接，向已连接的数据库发送 SQL 语句，处理 SQL 语句返回的结果。

采用 JDBC-ODBC 方式访问数据库需要事先设置 ODBC 数据源，比如将 shop 数据库设置为名字是 dataSource 的数据源。那么操作 dataSource 数据源的步骤如下。

(1) 加载 JDBC-ODBC 桥接器。

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
```

(2) 建立和 dataSource 数据源的连接。

```
Connection con = DriverManager.getConnection("jdbc:odbc:dataSource",  
"loginName"," password ");
```

(3) 得到 Statement 语句对象。

```
Statement sql = con.createStatement();
```

(4) 向数据源发送 SQL 语句，必要时返回 ResultSet 对象（结果集）。

```
ResultSet rs = sql.executeQuery(SQL中的查询语句);  
sql.execute(SQL中的更新、插入和删除语句);
```

2. 实验目的

本实验的目的是让学生掌握操作数据库的基本步骤。

3. 实验要求

使用某种数据库管理系统，比如 Microsoft Access 或 SQL Server 2000 建立一个名字为 shop 的数据库。在 shop 数据库中创建名字为 goods 的表，表的字段（属性）为：

number(文本) name(文本) madeTime(日期) price(数字,双精度)

其中，number 字段为主键。

将数据库设置为名字是 dataSource 的 ODBC 数据源。

编写程序，在 goods 表中随机查询 10 条记录，并计算出这 10 条记录 price 字段值的平均值，即计算平均价格。

4. 运行效果示例

程序运行效果如图 11.1 所示。

5. 程序模板

请按模板要求，将【代码】替换为 Java 程序代码。

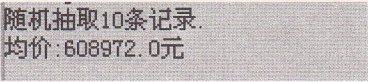


图 11.1 随机查询记录

RandomGetRecord.java

```
import java.sql.*;
import java.util.*;

public class RandomGetRecord {
    public static void main(String args[]) {
        int wantRecordAmount = 10;    //随机抽取的记录数目
        Random random =new Random();
        try{    【代码1】 //加载JDBC-ODBC桥接器
        }
        catch(ClassNotFoundException e) {
            System.out.print(e);
        }
        Connection con;
        Statement sql;
        ResultSet rs;
        try {
            String uri="jdbc:odbc:datasource";
            String id="";
            String password="";
            con=DriverManager.getConnection(uri,id,password);
            sql=con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
                                    ResultSet.CONCUR_READ_ONLY);
            rs = 【代码2】 //sql调用.executeQuery方法查询goods表中的全部记录
            rs.last();    //将rs的游标移到rs的最后一行
            int count=rs.getRow();
            Vector<Integer> vector=new Vector<Integer>();
            for(int i=1;i<=count;i++) {
                vector.add(new Integer(i));
            }
            int itemAmount=Math.min(wantRecordAmount,count);
            System.out.println("随机抽取"+itemAmount+"条记录.");
            double sum =0, n = itemAmount;
            while(itemAmount>0) {
                int randomIndex = random.nextInt(vector.size());
                int index=(vector.elementAt(randomIndex)).intValue();
                【代码3】 //将rs的游标移到index
                String number = rs.getString(1);
                String name = rs.getString(2);
                java.util.Date date=rs.getDate(3);
                double price=rs.getDouble(4);
```



```
        sum = sum+price;
        itemAmount--;
        vector.removeElementAt(randomIndex);
    }
    con.close();
    double aver = sum/n;
    System.out.println("均价:"+aver+"元");
}
catch(SQLException e) {
    System.out.println(""+e);
}
}
```

6. 实验指导

- ✧ 为了能进行随机查询，Statement 对必须返回一个可滚动的结果集。
- ✧ absolute(int row)方法可以将结果集中的游标移到参数 row 指定的行。
- ✧ java.util 包中的 Vector 类负责创建一个向量对象。创建一个向量时不用像数组那样必须要给出数组的大小。向量创建后，例如“Vector<Integer> a=new Vector<Integer>();”，a 可以使用 add(Integer n)把 Integer 对象 n 添加到向量的末尾，向量的大小会自动地增加。向量 a 可以使用 elementAt(int index)获取指定索引处的向量的元素（索引初始位置是 0）。

7. 实验后的练习

参照本实验编写一个数据库查询的程序，可以在若干学生中随机抽取 20 名学生，并计算这 20 名学生的平均成绩。

8. 填写实验报告

实验编号：1101 学生姓名： 实验时间： 教师签字：

实验效果评价	A	B	C	D	E
模板完成情况					
实验后的练习效果评价	A	B	C	D	E
练习完成情况					
总评					

实验 2 用户转账

1. 相关知识

事务由一组 SQL 语句组成，所谓事务处理是指：应用程序保证事务中的 SQL 语句要么全部都执行，要么一个都不执行。

处理事务的步骤如下：

- (1) 关闭自动提交模式，即关闭 SQL 语句的即可生效性。Connection 对象 con 使用

setAutoCommit()关闭自动提交模式“con.setAutoCommit(false);”。

(2) 执行事物中的 SQL 语句, 然后执行 Connection 对象 con 调用 commit()方法恢复 SQL 语句的有效性“con.commit();”。

(3) 撤销事务所做的操作, 即处理事物失败。如果事物中的 SQL 语句未能全部成功, 需在该步骤撤销 SQL 语句对数据库的操作, 即 con 对象调用 rollback()方法“con.rollback();”。

2. 实验目的

本实验的目的是让学生掌握事务处理的基本步骤。

3. 实验要求

使用某种数据库管理系统, 比如 Microsoft Access 或 SQL Server 2000 建立一个名字为 bank 的数据库。在 bank 数据库中创建 car1 表和 car2 表, card1 表和 card2 表的字段如下(二者相同):

number(文本) name(文本) amount(数字, 双精度)

其中, number 字段为主键。

将数据库设置为名字是 tom 的 ODBC 数据源。

程序进行两个操作, 一是将 card1 表中某记录的 amount 字段的值减去 100, 二是将 card2 表中某记录的 amount 字段的值增加 100, 必须保证这两个操作要么都成功, 要么都失败。

4. 运行效果示例

程序运行效果如图 11.2 所示。

5. 程序模板

请按模板要求, 将【代码】替换为 Java 程序代码。

TurnMoney.java

```
import java.sql.*;

public class TurnMoney {
    public static void main(String args[]){
        Connection con = null;
        Statement sql;
        ResultSet rs;
        try { Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        }
        catch(ClassNotFoundException e){
            System.out.println(""+e);
        }
        try{ double n = 100;
            con = DriverManager.getConnection("jdbc:odbc:tom","","");
            【代码1】//关闭自动提交模式
            sql = con.createStatement();
            rs = sql.executeQuery("SELECT * FROM card1 WHERE number=
            'zhangsan'");
```

转账操作之前zhangsan的钱款数额:200.0
转账操作之前xidanshop的钱款数额:160.0
转账操作之后zhangsan的钱款数额:100.0
转账操作之后xidanshop的钱款数额:260.0

图 11.2 转账操作


```

rs.next();
double amountOne = rs.getDouble("amount");
System.out.println("转账操作之前zhangsan的钱款数额:"+amountOne);
rs = sql.executeQuery("SELECT * FROM card2 WHERE number=
'xidanShop'");
rs.next();
double amountTwo = rs.getDouble("amount");
System.out.println("转账操作之前xidanShop的钱款数额:"+amountTwo);
amountOne = amountOne-n;
amountTwo = amountTwo+n;
sql.executeUpdate(
"UPDATE card1 SET amount ="+amountOne+" WHERE number = 'zhangsan'");
sql.executeUpdate(
"UPDATE card2 SET amount ="+amountTwo+" WHERE number = 'xidanShop'");
con.commit(); //开始事务处理,如果发生异常直接执行catch块
【代码2】//恢复自动提交模式
rs = sql.executeQuery("SELECT * FROM card1 WHERE number='zhangsan'");
rs.next();
amountOne = rs.getDouble("amount");
System.out.println("转账操作之后zhangsan的钱款数额:"+amountOne);
rs=sql.executeQuery("SELECT * FROM card2 WHERE number='xidanShop'");
rs.next();
amountTwo = rs.getDouble("amount");
System.out.println("转账操作之后xidanShop的钱款数额:"+amountTwo);
con.close();
}
catch(SQLException e){
try{ 【代码3】//撤销事务所做的操作
}
catch(SQLException exp){}
System.out.println(e.toString());
}
}
}

```

6. 实验指导

- ✧ 在设置数据源时, 请关闭 Microsoft Access 打开的数据库, 否则提示“非法路径”。
- ✧ “con.commit();”语句进行事务处理的过程中, 如果发现无法保证事务中的所有 SQL 语句要么都成功要么都不成功, 就抛出异常。

7. 实验后的练习

参照本实验编写事务处理程序。

8. 填写实验报告

实验编号：1102 学生姓名： 实验时间： 教师签字：

实验效果评价	A	B	C	D	E
模板完成情况					
实验后的练习效果评价	A	B	C	D	E
练习完成情况					
总评					

实验 3 查询 Excel 电子表格

1. 相关知识点

有时需要查询、更新和删除 Excel 电子表格的内容，可以通过 JDBC-ODBC 桥接器访问 Excel 电子表格。假设有电子表格：goods.xls，见图 11.3。

连接 Excel 电子表格的步骤如下。

1) 在电子表格中创建表

与访问其他数据库不同的是，必须在电子表格中选出一工作区作为连接时使用的表。在 Excel 电子表格中拖动鼠标选出范围，如图 11.3 所示。然后在 Excel 菜单中选择“插入”→“名称”→“定义”，给选中的工作区命名（这一工作区的名称将作为连接时使用的表名），如图 11.4 所示。这样就在电子表格中创建了一个名字是 message、有 5 个字段的表。现在就可以查询、更新、删除这个表中的记录了。

	A	B	C	D
1	品名	单价(元)	库存量	生产者
2	电视	2900	4	长城公司
3	冰箱	3400	5	里泊公司
4	长虹电视	3600	3	佳美集团
5	PC电脑	9088	12	IBM
6				
7				

图 11.3 在 Excel 电子表格中设置表

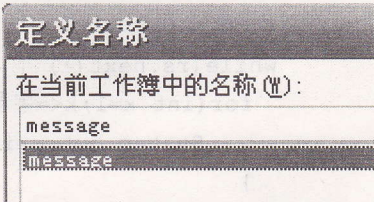


图 11.4 创建 message 表

2) 设置数据源

设置数据源名称为 myExcel，为数据源选择的是 goods.xls 电子表格（在该表格中创建了名字是 message 的表），选择的驱动程序必须是：Microsoft Excel Driver。在设置数据源时，可以单击其中的“选项”，将“只读”属性设置为未选中状态。

2. 实验目的

本实验的目的是让学生掌握操作 Excel 的基本步骤。

3. 实验要求

编写程序，查询 Excel 电子表格中设置的表，比如图 11.3 中设置的 message 表。

4. 运行效果示例

程序运行效果如图 11.5 所示。

5. 程序模板

上机调试模板代码。

QueyExcel.java

```
import java.sql.*;

public class QueryExcel {
    public static void main(String args[]) {
        try{
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        }
        catch(ClassNotFoundException e) {
            System.out.print(e);
        }
        Connection con;
        Statement sql;
        ResultSet rs;
        try {
            con=DriverManager.getConnection("jdbc:odbc:myExcel","","");
            DatabaseMetaData metadata=con.getMetaData();
            ResultSet rs1=metadata.getColumns(null,null,"message",null);
            int 字段个数=0;
            while(rs1.next()) {
                字段个数++;
            }
            sql=con.createStatement();
            rs=sql.executeQuery("select * from message");
            while(rs.next()) {
                for(int k=1;k<=字段个数;k++) {
                    System.out.print(" "+rs.getString(k)+" ");
                }
                System.out.println("");
            }
        }
        catch(SQLException e) {
            System.out.println(""+e);
        }
    }
}
```

图 11.5 查询电子表格

电视	2900.0
冰箱	3400.0
长虹电视	3600.0

6. 实验指导

可以选中电子表格中的任意区域来创建表,但需要注意的是表的第一行一定是所创建的表的字段名称。

7. 实验后的练习

参照本实验在电子表格中创建更多的表,然后查询或修改这些表。

8. 填写实验报告

实验编号：1103 学生姓名： 实验时间： 教师签字：

实验效果评价	A	B	C	D	E
模板完成情况					
实验后的练习效果评价	A	B	C	D	E
练习完成情况					
总评					

实验答案

实验 1:

【代码 1】`Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");`

【代码 2】`sql.executeQuery("SELECT * FROM goods");`

【代码 3】`rs.absolute(index);`

实验 2:

【代码 1】`con.setAutoCommit(false);`

【代码 2】`con.setAutoCommit(true);;`

【代码 3】`con.rollback();`

自 测 题

- 1. 在没有输出结果集中的数据之前能否关闭数据库连接?
- 2. 可滚动结果集的好处是什么?
- 3. 事务处理的第一步骤是什么?

答案:

- 1. 不能
- 2. 可以方便的随机查询记录。
- 3. 关闭默认的提交方式，即关闭 SQL 语句的即可生效性。