

SQLite

2018.12

3160103113 曾显珺

编写一个命令行程序，它具有三个命令行参数：

1. 数据库名
2. xlsx文件名，带.xlsx后缀
3. Excel中的页名，当这个参数省略时，取第一个页数据库表名，当这个参数省略时，用页名或当第三个参数也省略时用xlsx文件名前缀作为表名
4. 当然第3个参数省略时不能单独带有第四个参数。

程序要打开xlsx文件，找到指定的页，在本地的SQLite数据库中建立指定名称的表。xlsx里的第一行作为表中的字段名，字段类型由xlsx数据推断得到。推断的规则为：

- 整数的为int类型；
- 浮点数为real类型；
- 其他为char类型，大小为所有行中最大的长度。

然后程序将所有的行数据导入该表，并为该表建立一个PK，以表达行号。程序输出表的结构（SQL）和行的数量。

命令行规则：

打开xlsx文件： **open \$filepath\$.xlsx**

读取页面： **page \$pagenumber\$** （没有此命令时默认page为0）

建立表格： **create table \$tablename\$**

数据库规则： primary key 为行号，其中行号为0的行（即第一行）为字段名。

代码实现：

Table 类中判断输入格式：

```
if (cell.getCellType() == Cell.CELL_TYPE_NUMERIC) { //数字格式
    if (DateUtil.isCellDateFormatted(cell)) {
        SimpleDateFormat sdf = null;
        if (cellCellStyle().getDataFormat() ==
HSSFDataFormat
            .get_builtinFormat("h:mm")) {
            sdf = new SimpleDateFormat("HH:mm");
        } else {// 日期
            sdf = new SimpleDateFormat("yyyy-MM-dd");
        }
        Date date = cell.getDateCellValue();
        vs = sdf.format(date);
        v = new Char(vs);
    } else {
        cell.setCellType(Cell.CELL_TYPE_STRING);
        String temp = cell.getStringCellValue();
        // 判断是否包含小数点，如果不含小数点，则转换为int类，如果含小数点，则转换为real类
        if (temp.indexOf(".") > -1) {
            singleRow[n] = String.valueOf(new Double(temp))
                .trim();
            vs = singleRow[n];
            v = new Real(vs);
        } else {
            singleRow[n] = temp.trim();
            vs = singleRow[n];
            v = new Int(vs);
        }
    }
} else { //字符串
    singleRow[n] = cell.getStringCellValue().trim();
    vs = singleRow[n];
    v = new Char(vs);
}
```

Value 类：

```
public abstract class Value { //抽象类Value
    public String val;
    value(String S){
        this.val = S;
    }
}
```

```

}

class Int extends value{ //int整型value

    Int(String S) {
        super(S);
    }
}

class Real extends value{ //real实数型value

    Real(String S) {
        super(S);
    }
}

class Char extends value{ // char 字符串型value

    Char(String S) {
        super(S);
    }
}

```

最后输出：

```

public void printout() {
    if(lines.size()>1) {
        ArrayList<value> line = Lines.get(0);
        System.out.println("-----");
        System.out.print("field name"+":\t"); //第一行：字段名(index=0的信息)
        for(value v: line) {
            System.out.print(v.val+"\t"+'\t');
        }
        System.out.println("");
        Lines.remove(0);
        for(int index: Lines.keySet()) { //第二行开始是列表内容
            System.out.print(index+":\t"+'\t');
            ArrayList<value> line1 = Lines.get(index);
            for(value v: line1) {
                System.out.print(v.val+"\t"+'\t');
            }
            System.out.println("");
        }
    }
}

```

```

        System.out.println("-----");
    );
    System.out.print("keys"\t"\t"); //最后一行，输出各列的类型
    line = lines.get(1);
    for(value v: line) {
        System.out.print(v.getClass()+"\t");
    }
    System.out.println("");
}
}

```

输出示例：

```

>>>open D:\\excel.xlsx
>>>create table a
>>>print table a
-----
field name:      name          code          state
4:              wed           233.6         1
3:              dog           4.72          0
2:              cat           3.45          1
1:              bill          2.24          0
-----
keys            class Char     class Real    class Int
>>>

```

本实验中仍然存在待改进的问题，比如excel文件中一列可以是不同的字符类型，比如都是数据的情况下，可能全是real也有可能有部分int，不能按照列为数据进行区分。同样，也可能在同一字段名下有空数据，此时应用“<null>”填充而不是直接忽略，空数据的存在也影响了字段类型的判断（本实验以第一行数据的字段类型为标准，即默认第一行没有空数据）。