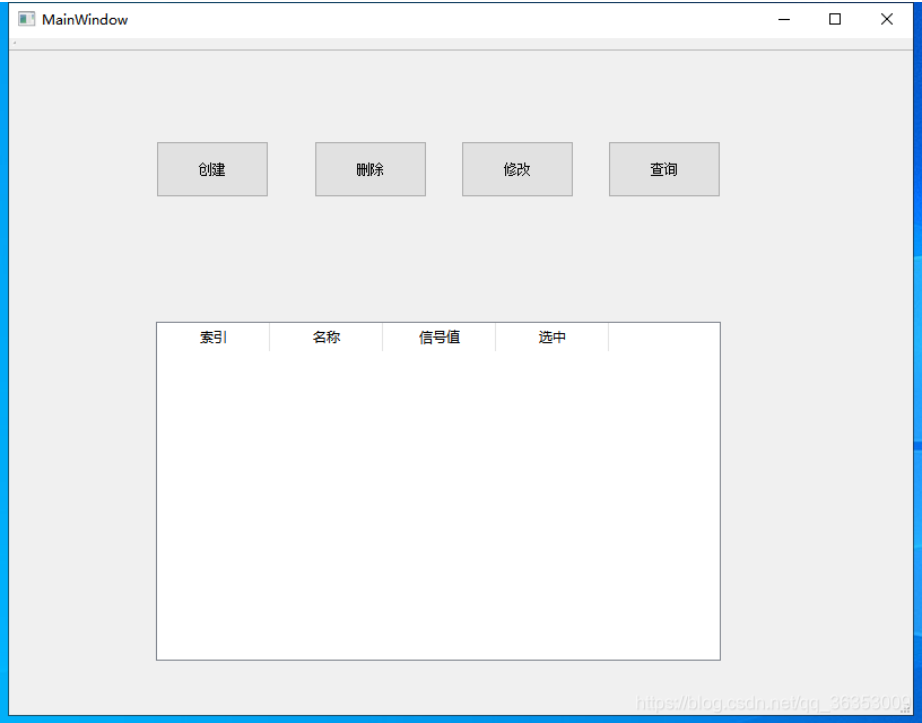


QT应用SQL数据库，简单全面的应用，增删改查。

对于从来没接触过数据库的人来说，网上很多教程都太学术了。很多基本概念不了解的萌新，理解起来比较困难。这里一步一步操作，让小白也能掌握QT环境下SQL数据库的基本操作。

先看界面

基本功能就是“增删改查”四个按钮。



1.头文件和 .pro 文件

.pro 文件中

```
1 | QT += sql
```

.h文件中，添加头文件

```
1 | #include <QSqlQuery>
2 | #include <QtSql>
3 | #include <qsqldatabase.h>
```

2.打开数据库

```
1 | QSqlDatabase db;//建立套接字, 建议设置从全局变量.
2 | db = QSqlDatabase::addDatabase("QSQLITE","QtSql");//数据库名称 QtSql是数据库的名称
3 | db.setDatabaseName("QtSql.db");//数据库文件
4 | if(db.open())
5 | {
6 |     qDebug()<<"打开数据库成功";
7 | }
```

3.查

```
1 | if(db.open())//打开数据库
2 | {
3 |     QSqlQuery qry(db);//获取数据库套接字
4 |     if(qry.exec("select * from Table1"))//Table1是数据库表格的名称
5 |     {
6 |         Sql_Size = 0;//全局变量 记录数据库有多少条数据
7 |         while(qry.next())
8 |         {
9 |             Sql_TablePart[Sql_Size].Sql_Index = qry.value(0).toInt();
10 |             Sql_TablePart[Sql_Size].Sql_Name = qry.value(1).toString();
11 |             Sql_TablePart[Sql_Size].Sql_Values = qry.value(2).toDouble();
12 |             Sql_TablePart[Sql_Size].Sql_Select = qry.value(3).toBool();
13 |             Sql_Size++;//全局变量
14 |         }
15 |     }
16 | }
```

查询数据库的内容，这里利用了一个结构体，储存并显示。
这里我提前建立好啦一个数据库，名称是QtSql，表格名称是Table1。

Index	Name	Declared Type	Type	Size	Precision	Not Null	
1	AA	INT	INT	0	0		
2	BB	TEXT	TEXT	0	0		
3	CC	DOUBLE	DOUBLE	0	0		
4	DD	BOOL	BOOL	0	0		

4个数据的Fields：

- int 类型的 索引 AA
- text 类型的 名称 BB
- double 类型的 CC
- bool 类型的 DD

RecNo	AA	BB	CC	DD
Click here to define a filter				
1	0	烟雾报警器	3.14	

4.增

```
1 | QSqlQuery qry(db);//获取套接字
2 | qry.prepare("insert into Table1(AA,BB,CC,DD) " "values(:AA,:BB,:CC,:DD)");//4个类型
3 | qry.bindValue(":AA",Sql_Size);
4 | qry.bindValue(":BB","烟雾报警器");
5 | qry.bindValue(":CC",3.14);
6 | qry.bindValue(":DD",0);
7 | qry.exec();//写入数据库
```

5.删

```
1 | QSqlQuery qry(db);//获取套接字
2 | qry.prepare("delete from Table1 where AA=?");//要删除AA行数据为Sql_Size-1的那一行
3 | qry.addBindValue(Sql_Size-1);
4 | qry.exec();//写入数据库
```

6.改

```
1 | QSqlQuery qry(db);
2 | qry.prepare("update Table1 set BB=?,CC=? where AA=?");//改变AA行为Sql_Size-1的那一行
3 | qry.addBindValue(3.14); //BB那一个元素 改成 3.14
4 | qry.addBindValue(5.72); //CC那一个元素 改成 5.72
5 | qry.addBindValue(Sql_Size-1);
6 | qry.exec();
```

7.补充数据库软件

我用的数据软件是SQLiteExpertSetup


网上都可以下到。版本比较低，可能有BUG，基本功能都能用。
具体操作如下：

1. 创建数据库 File > New DataBase 然后选中路径，文件名称。
2. 创建表格 邮件创建好的数据库 > New Table > 起名
3. 创建Fields 点击Add添加表头，名称尽量用英文。完成后点击Apply保存。
4. 最后就可以自定义数据了，用程序添加也可以。

总结

- 数据库简单的可以理解成一个**结构体数组**。
- 用QT使用数据库，其实用的还是**SQL的命令语句**。基本功能都可以这样实现：

```
1 | qry.exec("insert into Table1 values(2,'烟雾传感器',2.56,0)");//增
2 | qry.exec("delete from Table1 where AA=1");//删
3 | qry.exec("update Table1 set BB=3.14,CC=5.17 where AA=3");//改
4 |
5 | qry.next();
6 | Sql_Index = qry.value(0).toUInt();//查询
```

- 对于没有SQL的基础的我来说，记住几个操作的方法，简单的功能都可以实现。
- 数据库的数据，关闭软件，关闭电脑数据也能保留。简单的记录一下软件的登陆账号密码，非常有用。

8.“增”的进阶写法

数据库有好多元素，需要

```
1  if(db.open())//打开数据库
2  {
3      QSqlQuery qry(db);//获取数据库
4      if(qry.exec("select * from LSSJ"))//获取数据表
5      {
6          //*****整理字符串成 qry.prepare("insert into LSSJ(SJ,SJ1,SJ2,SJ3)" "VALUES(:SJ,:SJ1,:SJ2,:SJ3)");
7          QString Qry_Str = "insert into LSSJ(SJ";
8          QString Qry_Str1 = "VALUES(:SJ";
9          for(int i=1;i<45;i++)
10         {
11             Qry_Str += ",SJ";
12             Qry_Str += QString::number(i);
13
14             Qry_Str1 += ",:SJ";
15             Qry_Str1 += QString::number(i);
16         }
17         Qry_Str += ")";
18         Qry_Str1 += ")";
19
20         Qry_Str += ' ';
21         Qry_Str += Qry_Str1;
22
23         qDebug()<<Qry_Str;//*****整理字符串成
24
25         qry.prepare(Qry_Str);//写入字符串
26         qry.bindValue(":SJ",Qt_Time_String);//写入第一个数据
27         for(int i=0;i<44;i++)//写入44个数据
28         {
29             Qry_Str1 = ":SJ";
30             Qry_Str1 += QString::number(i+1);
31             qry.bindValue(Qry_Str1,PKX_Value[i]);
32         }
33         if(qry.exec())//看看写入是否成功
34         {
35             qDebug()<<"写入并判断是否成功OK";
36         }
37         else
38         {
39             qDebug()<<"写入并判断是否成功err";
40         }
41
42         //      qry.prepare("insert into LSSJ(SJ,SJ1,SJ2,SJ3,SJ4,SJ5,SJ6,SJ7,SJ8)" "VALUES(:SJ,:SJ1,:SJ2,:SJ3,:SJ4,:SJ5,:SJ6,:SJ7,:SJ8)");
43         //      qry.bindValue(":SJ",Qt_Time_String);
44         //      qry.bindValue(":SJ1",PKX_Value[0]);
45         //      qry.bindValue(":SJ2",PKX_Value[1]);
46         //      qry.bindValue(":SJ3",PKX_Value[2]);
47         //      qry.bindValue(":SJ4",PKX_Value[3]);
48         //      qry.bindValue(":SJ5",PKX_Value[4]);
49         //      qry.bindValue(":SJ6",PKX_Value[5]);
50         //      qry.bindValue(":SJ7",PKX_Value[6]);
51         //      qry.bindValue(":SJ8",PKX_Value[7]);
52     }
53 }
```