QT数据库操作基本介绍

1. 关于MVC

MVC全名是Model View Controller，是模型(model)－视图(view)－控制器(controller)的缩写。对于上位机PC的编程，MVC是较好的编程方式。

文档重点描述QT“所见即所得”的数据库编程方式和思维。

1. 常规高级语言连接数据库方式

笔者用过C#，C++，VB。连接的数据的方式大致相同，如下：

* 首先需要自己用外部程序建立一个数据库，他可以是一个大型集合，也可以是一个文件。
* 使用自己的编程语言中提供的类，建立一个链接数据库的对象。该对象一般需要指定数据库的地址，访问密码。大型数据库，还需要用户名，权限设置等。
* 链接建立后，就可以通过连接对数据库中的表进行操作了。但是这样的操作比较麻烦。一般高级语言都提供了自己的容器和接口，与链接数据库的对象对接，像二维数组一样平面化的操作数据库。
* 在以上基础上，再提供与可视化操作接口的类，用于直接可视化编辑数据库。

一般来说，高级语言操作数据库都不需要使用SQL语言，避免来回的折腾数据。

1. QT使用到的类
   1. QSqlTableModel

构造函数QSqlTableModel ( QObject \* parent = 0, QSqlDatabase db = QSqlDatabase() )。

QT介绍翻译：该类为数据基础表格提供了一个可供编辑的数据模型。它提供了数据表格记录的高级读写操作（用户可以不再使用Select，where，insert等SQL语句），但这仅限于特定的一张表。该类继承了QSqlQuery的功能，并且更高级。该类还可以与显示控件QTableView对接，直接用于显示。

* 1. QSqlDatabase

QT介绍翻译：该类代理一个基础的数据链接。

个人介绍：该类需要输入数据库的地址，如果有用户名，密码等，就是用它进行输入。设置好后，用open()函数进行打开数据库操作。

* 1. 可视化QtableView

他就是QT的一个可视化控件。QT把他设计成可以与QSqlTableModel接口，用于数据库的可视化操作。

1. 例子程序的重点介绍

例子程序的工程名为“SqlMVC”，请不要修改文件夹和工程名。QT构建版本为5.2.1

程序的设计是：

* 主线程负责界面的刷新。用Qtimer myTick 的TimeOut 使用”select()”定时刷新数据库的数据。
* 然后新建两个线程，分别对数据库中1、2行的“Wight”列进行累加修改。
* 这样，程序就相当于对于数据库有3个异步操作，主线程中的读取显示，次线程中的读取和修改。但是要注意到，程序只使用了一个QSqlDatabase用于数据库的链接。