实验四 基本用户界面实验

一、 实验目的:

进一步熟悉使用QtCreator进行程序开发和调试的基本方法；

学习使用命令行编译的方法，并编译mysql驱动（以插件形式）；

掌握使用界面编辑器进行用户界面的创建、布局方法；

进一步学习和理解qt中的信号槽概念；

学习数据库相关知识及基本数据库操作方法。

二、 实验软件

- Qt 5.7.0
- MinGW 5.3.0 32bit
- Windows 7 系统及以上

三、 实验内容及说明

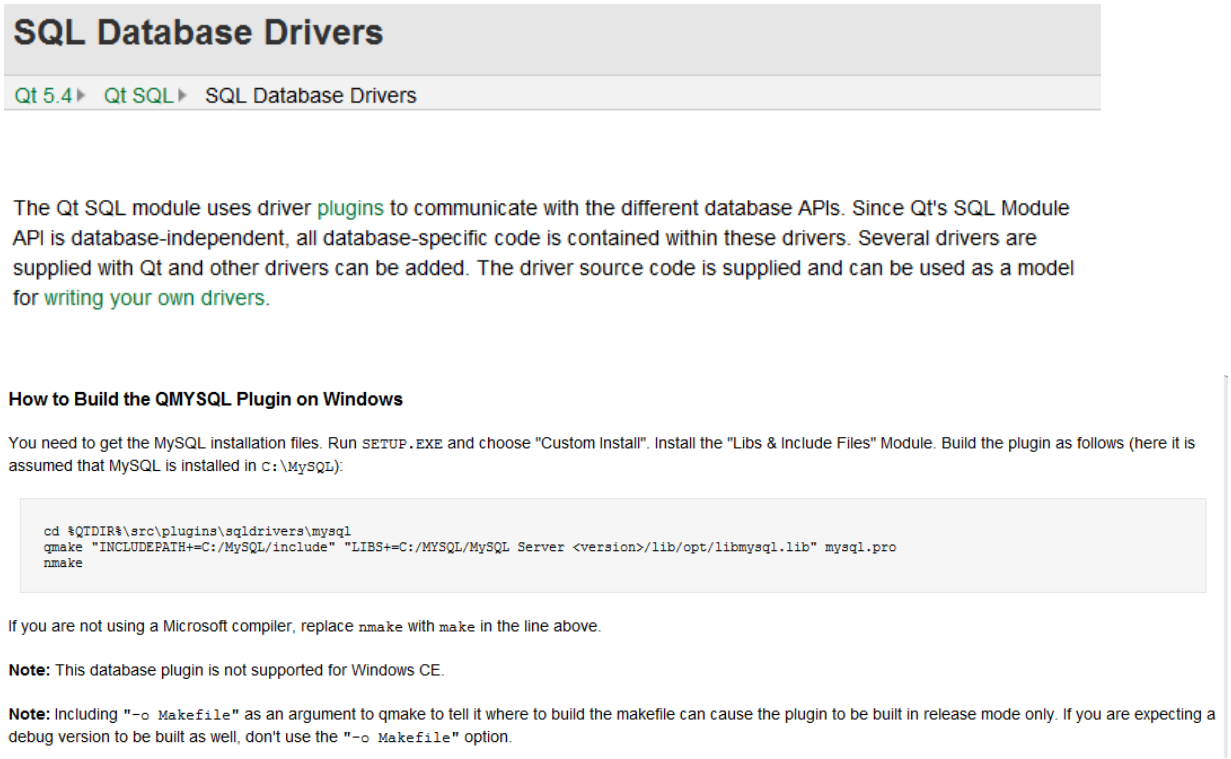
编写一个能够用来计算绩点的绩点计算器。程序能按照指定要求检索保存在 mysql 数据库中的学生成绩，并计算该同学的成绩绩点后进行显示。用户可以选择使用姓名查询和学号查询。

图 4.1 基本界面

本实验没有详细代码，要求各位同学参考 Qt 文档，自己决定加入哪些头文件和撰写具体代码。

四、 编译mysql驱动

本实验将使用 mysql 数据库，从数据库中查询相关数据并进行处理。因此第一步为 mysql 数据库驱动的编译。编译的方法有 2 中，一种是在编译 Qt 库本身的时候进行编译，另一种是 Qt 库已经编译完成后，单独编译相应驱动。详细内容可参考 Qt 文档。



SQL Database Drivers

Qt 5.4 ▸ Qt SQL ▸ SQL Database Drivers

The Qt SQL module uses driver plugins to communicate with the different database APIs. Since Qt's SQL Module API is database-independent, all database-specific code is contained within these drivers. Several drivers are supplied with Qt and other drivers can be added. The driver source code is supplied and can be used as a model for writing your own drivers.

How to Build the QMYSQL Plugin on Windows

You need to get the MySQL installation files. Run `SETUP.EXE` and choose "Custom Install". Install the "Libs & Include Files" Module. Build the plugin as follows (here it is assumed that MySQL is installed in `C:\MySQL`):

```
cd %QTDIR%\src\plugins\sqldrivers\mysql
qmake "INCLUDEPATH+=C:/MySQL/include" "LIBS+=C:/MYSQL/MySQL Server <version>/lib/opt/libmysql.lib" mysql.pro
nmake
```

If you are not using a Microsoft compiler, replace `nmake` with `make` in the line above.

Note: This database plugin is not supported for Windows CE.

Note: Including `"-o Makefile"` as an argument to qmake to tell it where to build the makefile can cause the plugin to be built in release mode only. If you are expecting a debug version to be built as well, don't use the `"-o Makefile"` option.

图 4.2 Qt 文档中对 mysql 数据库驱动编译的说明

编译数据库驱动的核心是需要有 mysql 的客户端及服务端的 dev 库文件。编译完成后产生的 libmysql.dll 及 libqsqlmysql.a 和 libqsqlmysqld.a 文件分别复制至 bin 目录(.dll 文件)和 lib(.a 文件)目录下，本实验提供的 mysql 数据库已经包含上述文件。

五、 数据库的创建

本实验使用的 mysql 版本为 5.6.20，解压 mysql 至 D:\Dev 目录下（不要更改该路径），进入 D:\Dev\mysql 目录，右键单击“mysql_installservice.bat”文件，单击“以管理员身份运行”(Win7/10)，或直接双击运行（WinXP）。打开命令行窗口，输

入命令“sc start mysql”，然后输入“sc query mysql”，如果出现“STATE ：RUNNING”字样，则 mysql 已正确安装。如图 4-3 所示。

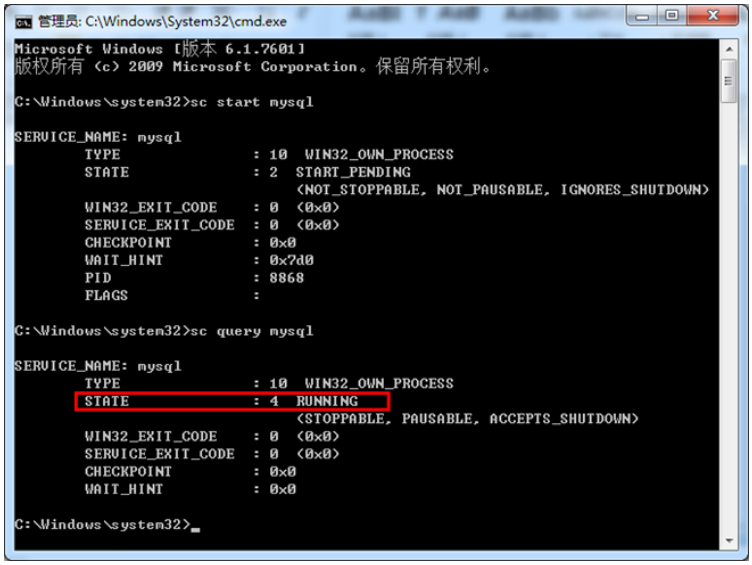


图 4.3 安装并运行 mysql 服务

有多种 GUI 工具很方便的进行 mysql 数据库的管理。本课程使用的是 Mysql WorkBench 软件，版本为 6.2。

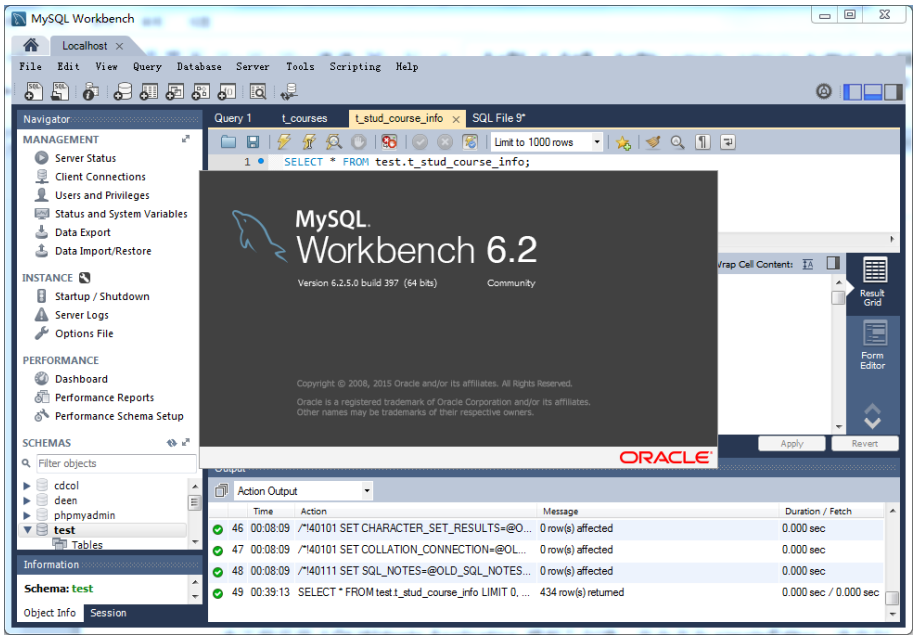


图 4.4 mysql Workbech 界面

打开 mysq Workbench，点击“Database->manage connection”，弹出连接管理对话框，如下图所示，“connection name” 填入 localhost，Default Schema 填入“test”，然后关闭对话框。此时，程序主界面处应当出现刚才配置的数据库连接，双击这个连接，进入 mysql 数据库。在左下角 SCHEMAS 下方空白处右键，创建 test 数据库，并双击。

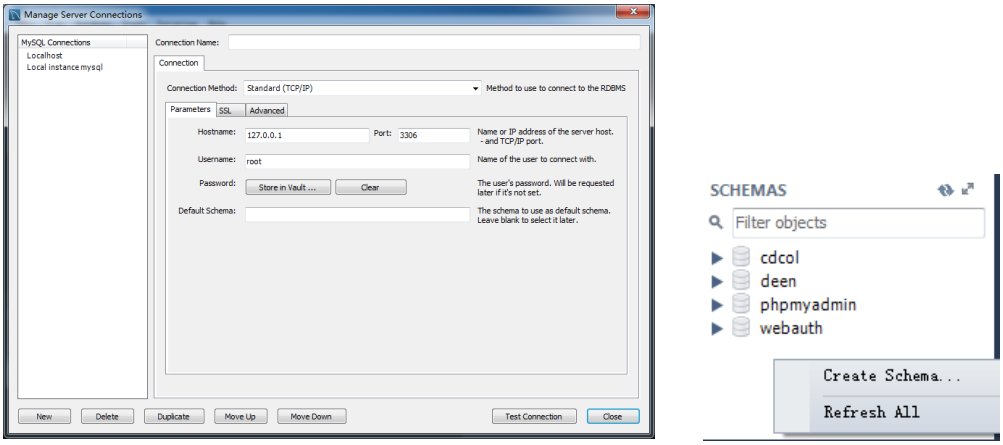


图 4.5 配置数据库连接及创建新数据库

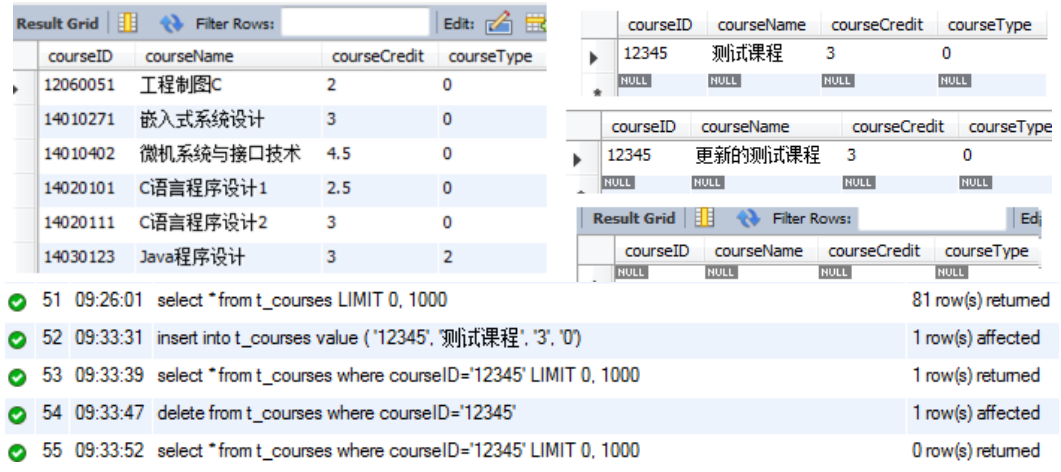
然后点击“File->Open Sql script”，打开 Dump20150514.sql 文件，然后点击“Query->Execute”或点击脚本对话框左上角的闪电图标，执行该 sql 脚本。至此，已经创建完成 test 数据库（含表结构和数据）。

六、 Mysql数据库的基本使用

确认左下角 test 数据库为粗体显示（表示选定当前数据库），新建一个“new query tab”（File 菜单下），在这个 query 窗口下**分别**执行以下语句（每句执行一次）：

```
select * from t_courses;
insert into t_courses value ( '12345', '测试课程', '3', '0');
select * from t_courses where courseID='12345';
update t_courses set courseName='更新的测试课程' where courseID='12345';
select * from t_courses where courseID='12345';
delete from t_courses where courseID='12345';
select * from t_courses where courseID='12345';
```

窗口下方会返回上述语句执行结果：



这是一般关系型数据库基本具备的几个语句，select、insert、update 和 delete。具体这几个命令的用法，请参考相关资料。以下，给一个多表级联的使用例程。

```
select t_stud_info.sName, t_courses.courseName,t_stud_course_info.scScores
from
( t_stud_course_info inner join t_courses
    on t_stud_course_info.scCourseID=t_courses.courseID)
inner join t_stud_info
    on t_stud_info.sNumber=t_stud_course_info.scNumber
where t_stud_info.sName='韩子川';
```

返回如下：

sName	courseName	scScores
韩子川	复变函数	96
韩子川	毛泽东思想和中国特色社会主义理论体系概论	83
韩子川	大学物理B-2	77
韩子川	大学体育-3	90
韩子川	模拟电子线路	80
韩子川	信号与系统	93
韩子川	模拟电子线路课程设计	优秀

Message

7 row(s) returned

完成以上内容，基本上可以正常操作数据库，一般来说，对数据库的操作就是上述的过程，只不过有时需要兼顾效率和功能，需要特殊设计，这是一个专门的内容。

注：关于数据库库表结构设计，这是非常专业的问题，本课程不做详细说明。

七、 Qt的数据库支持

Qt Sql 模块是提供平台独立的 SQL 数据库支持的基本模块。Qt 的 SQL API 可以分为三个不同的层次：

(1) 数据库驱动层；

(2) SQL API 接口层；

(3) 用户接口层。

Qt 数据库模块的使用有两大部分组成：

(1) 在工程文件中打开 Qt 的 SQL 模块，并包含相应的头文件；

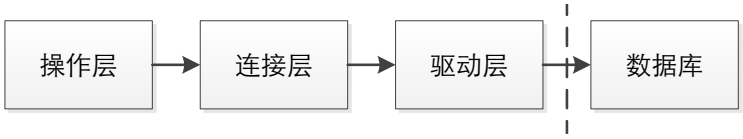
```
QT  += sql

#include <QtSql>
```

(2) 在程序中添加数据库的连接，并开始进行相应的数据库操作。

Qt Sql 模块还可以很方便的与模型/视图框架整合，特别是对数据库以表格形式进行操作时，模型/视图框架非常适合。限于时间关系，本课程不涉及模型/视图框架部分内容，请自行查阅相关文档。

与 Qt 绘制系统结构非常类似，Qt 的 SQL 模块可以划分为 3 个层次：操作层、数据库层和驱动层。操作层提供了一个统一的数据库操作接口(最为常见的是 QSqlQuery 类)，执行相应的 SQL 命令，如数据的检索、更新、删除等。数据库层负责数据库的统一管

理，包括数据库的打开、连接、断开、关闭等（使用 QSqlDatabase 类）。数据库层将查询的指令与目标数据库通过驱动层相关联，不同的数据库其数据库驱动各不相同。



Qt 提供了如下几种驱动：

Driver name	DBMS
QDB2	IBM DB2 (version 7.1 and above)
QIBASE	Borland InterBase
QMYSQL	MySQL
QOCI	Oracle Call Interface Driver
QODBC	Open Database Connectivity (ODBC) - Microsoft SQL Server and other ODBC-compliant databases
QPSQL	PostgreSQL (versions 7.3 and above)
QSQLITE2	SQLite version 2
QSQLITE	SQLite version 3
QTDS	Sybase Adaptive Server (**Note:** obsolete from Qt 4.7)

由于版权问题 Qt 仅自带了 SQLite 的驱动。SQLite 是一个开源免费的文件型 SQL 数据库，非常适合在小规模数据处理中使用（有兴趣的同学可以将实验三的数据，存储至一个 SQLite 数据库中），MySQL 的驱动需要下载 Oracle 提供的开发库并自行编译。

八、 工程创建

本工程采用“Qt Widgets Application 模板”创建，命名为 qtMysql，设定类名为“ScoreWindow”、基类为 QMainWindow，并勾选“创建界面”。

按照实验三的内容，使用界面编辑器设计界面，并添加相应资源，总体要求操作简单、界面美观，功能完备。

1. 实现功能

本程序的功能很简单，从数据库查询相应的数据，计算后显示在用户区。因此，在界面设计上，需要有查询输入框、详细信息显示框以及结果显示等，依次添加 2 个 QRadioButton，2 个 QGroupbox，5 个 QLineEdit、1 个 QTextEdit 以及 1 个 QPushButton（及相关的 QLabel），并使用布局编辑器布好局。

用户在使用时，会在编辑框中输入查询条件，并点击 QPushButton 进行查询。因此，创建 QPushButton(命名为 btnQuery)的槽函数: on_btnQuery_clicked()。程序的主要功能都在该槽函数中体现。

2. 数据库的连接

本程序需要使用前述编译的 mysql 插件。在 pro 文件中添加如下配置：

```
QT          += sql
```

在头文件 scorewindow.h 文件中添加#include <QtSql>，并增加一个私有成员变量：

```
#include <QtSql>
 …
QSqlDatabase db;
```

在 scorewindow.cpp 文件中构造函数中添加如下代码：

```
db = QSqlDatabase::addDatabase("QMYSQL");
db.setHostName("localhost");
db.setPort(3306);
db.setUserName("root");
db.setPassword("");
db.setDatabaseName("test");
```

至此，我们已在程序内创建了一个 db 对象，该对象主要负责数据的连接。

3. QPushButton 槽函数

on_btnQuery_clicked()槽函数是本程序的主要处理函数。主要代码如下所示，功能的核心内容是根据用户的要求，构造出相应的 MySQL 查询语句。基本的方法是先使用 mysql workbench 编写 SQL 查询脚本，然后在程序中构造这些脚本，最后将查询返回结果进行处理。

```
void ScoreWindow::on_btnQuery_clicked()
{
    if(db.isOpen())
        db.close();

    if(!db.open()){
        QMessageBox::critical(this,tr("错误"),tr("无法连接数据库！\n请检查数据库连接配置。"));
        return;
    }else{
        QString str;
        if(ui->radioStNumber->isChecked())
            str=QString("where sNumber='%1'").arg(ui->edtInput->text());
        else if(ui->radioStName->isChecked())
            str=QString("where sName='%1'").arg(ui->edtInput->text());
        else if(ui->radioStAll->isChecked())
            str="";
        QString queryStr = QString("SELECT * FROM t_stud_info %1;").arg(str);
        queryStudentInfo(queryStr);

        queryStr =QString("select t_stud_info.sName as '姓名', t_courses.courseName as '课程名称',"
                " t_stud_course_info.scScores as '成绩',  t_courses.courseCredit as '学分' "
                " from "
                " ( t_stud_course_info inner join t_courses on t_stud_course_info.scCourseID=t_cou
                " inner join t_stud_info on t_stud_info.sNumber=t_stud_course_info.scNumber "
                " %1 "
                " order by t_stud_course_info.scNumber"
                ";").arg(str);
        QStringList queryResult = queryStudentScore(queryStr);

        calculatePoints(queryResult);
        updateInfoWindow(queryResult);

        db.close();
    }
```

4. 几个需要注意的问题

(1) Sql 语句的编写；

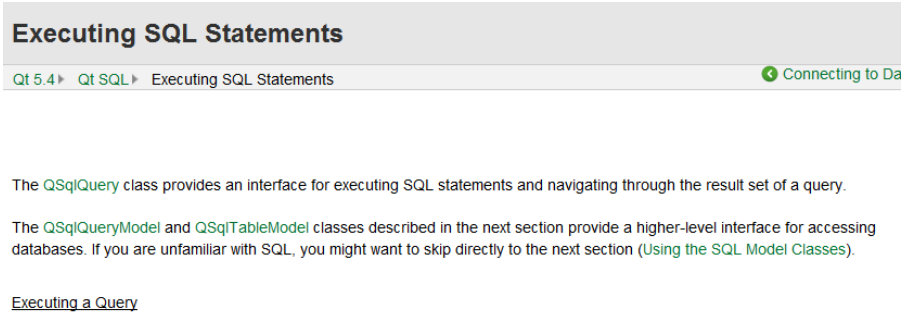
(2) 查询结果的处理；

(3) 绩点规则及计算；

(4) 结果显示，使用 QTextEditor + Html 方式，实现一个简单的表格。

九、 预习要求：

1、 阅读 Qt 文档：Executing SQL Statements，摘要写出其主要内容。



Executing SQL Statements

Qt 5.4 ▸ Qt SQL ▸ Executing SQL Statements ◂ Connecting to Da

The QSqlQuery class provides an interface for executing SQL statements and navigating through the result set of a query.

The QSqlQueryModel and QSqlTableModel classes described in the next section provide a higher-level interface for accessing databases. If you are unfamiliar with SQL, you might want to skip directly to the next section (Using the SQL Model Classes).

Executing a Query

2、 增加一个学生信息和成绩录入功能，可以每次录入一条信息（包括个人信息：学号、姓名、专业、班级、入学年份、联系电话，以及课程成绩信息：嵌入式系统设计(A~E)、微波技术与天线(0~100)、通信原理(0~100)、科技英语(A~E)），输入完成后自动计算绩点。

3、 增加批量导入功能，数据为 csv 逗号分隔符文本文件。每行为一位同学的成绩信息。格式如下：

2016-2017（1）通信14成绩

序号	学号	姓名	专业	班级	嵌入式系统设计	微波技术与天线	通信原理	科技英语
1	1403140101	布拉比·玉努斯	通信工程	通信1401	良好	79	93	良好
2	1403140102	陈妍荷	通信工程	通信1401	优秀	84	91	优秀
3	1403140103	李莲实	通信工程	通信1401	优秀	72	93	优秀
4	1403140104	李诗嘉	通信工程	通信1401	优秀	84	77	优秀
5	1403140105	尚静	通信工程	通信1401	优秀	94	74	优秀
6	1403140106	张涵	通信工程	通信1401	良好	78	94	优秀
7	1403140107	周钰	通信工程	通信1401	良好	88	80	优秀

4、 增加数据库导出功能，将 mysql 数据库导出为 sqlite3 数据库

5、 使用 QTableWidget 无成绩。