****

学生课程报告

报告名称：**运动会信息管理系统设计与实现**

第六组

姓名： 阮正鑫、李宇凡

学号： 516021910587、516021910554

邮箱： ruanzhengxin@sjtu.edu.cn

lyf1049443838@sjtu.edu.cn

项目地址：<https://github.com/liyufan/Sport>

2019 年 11 月7 日

**1、数据库需求分析与功能描述**

在校田径运动会中设置了各类比赛，许多同学积极参加各类比赛，都取得了不错的成绩。在这个过程中，由于比赛项目多样、参与学生众多，就产生了大量的成绩数据需要管理与统计。针对这个需要，我们设计了这个运动会信息管理系统数据库，以方便管理在运动会举办的过程中产生的大量的数据。

数据管理系统中，我们能够实现运动员信息、比赛项目信息的储存、更新与删除。同时能够实现根据存储的项目的编号信息或项目名称信息进行获奖结果的排名，也能够根据运动员的编号、运动员的姓名查询运动员参加的项目以及最好成绩。在数据库中，包含运动员的信息、项目的信息、系团队的信息以及比赛类型的信息。这几个主体之间的信息流动，我们通过以下的数据流图来体现。

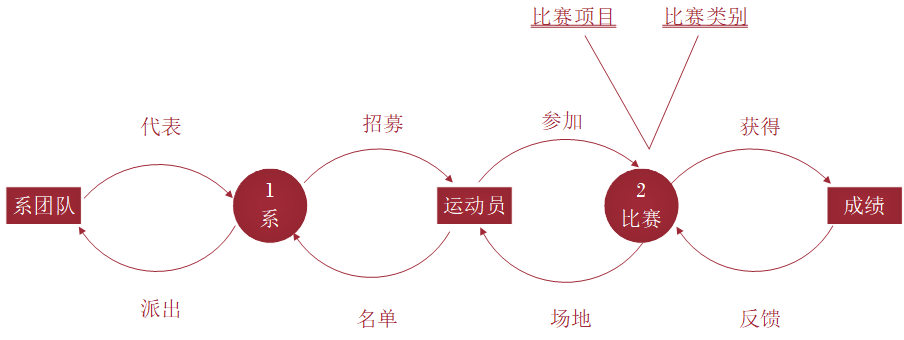


图1 数据流图

通过以上的数据流图我们可以看出，整个数据库的信息流动以运动员为中心。运动员在系团队的管辖之下，参加各类属于不同类别的比赛项目，得到了相应的成绩。根据以上的基本关系，我们开始了数据库的系统设计。

**2、E－R概念模型设计与分析**

根据以上的信息，我们开始设计数据库系统的ER图以及相应的关系模型。在校田径运动会中设置了各类比赛，每一比赛类别有类别编号、类别名称和主管等属性，每一比赛类别包含很多比赛项目；每一比赛项目有项目编号、项目名称、比赛时间等属性；各个系团队有团编号、团名称、领队等属性，每一代表团有多名运动员组成，运动员有编号，姓名，年龄，性别等属性；每一名运动员可以参加多个比赛项目，每一比赛项目也有多名运动员参加，运动员参加比赛有成绩属性，成绩限定在0-7分

根据以上信息，我们可以看到，这个数据库系统中共有4个实体：分别是运动员、系团队、比赛项目、比赛类别，各自包含其属性，之间有一定的关联关系，由此设计完成数据库系统的ER模型图

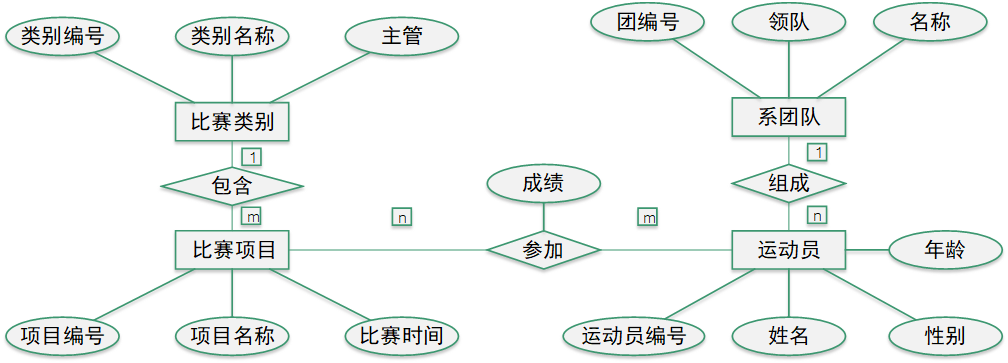


图2 数据库ER模型图

根据ER图，可将其转换为相应的关系模式：

1）比赛类别（类别名称，类别编号，主管）

2）比赛项目（项目编号，项目名称，比赛时间，类别编号）

3）系团队（系团队编号，团名称，领队）

4）运动员（运动员编号，姓名，年龄，性别，系团队编号）

5）参赛（项目编号，运动员编号，成绩）

关系模式中下划线属性为主码，红色字体属性为外码

以上构成了数据库系统的关系模式。

主要的依赖关系如下

F1={类别编号->类别名称, 类别编号->主管, 类别名称->类别编号}

F2={项目编号->项目名称, 项目名称->项目编号,项目编号->比赛时间, 项目编号->类别编号}

F3={系团队编号->团名称, 系团队编号->领队}

F4={运动员编号->姓名, 运动员编号->年龄, 运动员编号->性别, 运动员编号->系团队编号}

F5={(项目编号，运动员编号) ->成绩}

数据库系统的完整性包括，运动员、系团队、比赛项目、比赛类别四个实体的主码的唯一性，四个关系中的主码分别为运动员编号、系团队编号、项目编号、类别编号，可由英文字母加上数字构成，确保主码的唯一性。参照完整性包括比赛项目、运动员、参赛这三个关系的外码与相应的关系的主码的一致性。数据库检查约束中包括成绩限定在0~7名，运动员年龄限定在18~24岁，系团队名称所属仅在（电院、机动、媒设、生物等），项目名称所属仅在运动会开展的项目当中。关系模式中，五张表都属于BCNF模式，不会产生更新异常。

根据设计完成的关系模式，我们设计了相应的数据字典。

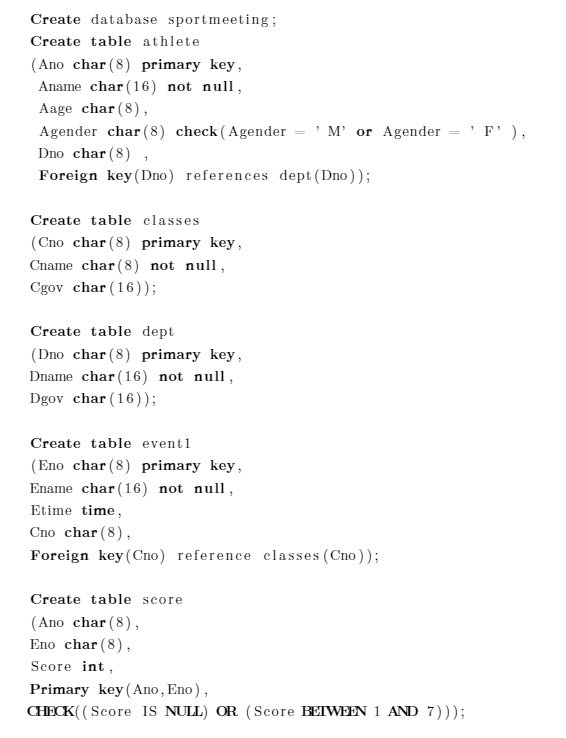
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据项 |  |  |  |  |  |
| 数据项名 | 含义说明 | 别名 | 类型 | 长度 | 取值范围 |
| 类别编号 | 唯一标识比赛类别 | Cno | char | 8 | C1~Cx |
| 类别名称 | 与编号对应的比赛类别的名称 | Cname | char | 16 |  |
| 主管 | 主管老师的姓名 | Cgov | char | 16 |  |
| 项目编号 | 唯一标识比赛项目 | Gno | char | 8 | G1~Gx |
| 项目名称 | 与编号对应的比赛项目的名称 | Gname | char | 16 |  |
| 比赛时间 | 对应的比赛项目开始的时间 | Gtime | time |  |  |
| 运动员编号 | 唯一标识运动员 | Ano | char | 8 | A1~Ax |
| 姓名 | 与编号对应的运动员的姓名 | Aname | char | 16 |  |
| 年龄 | 与编号对应的运动员的年龄 | Aage | char | 8 | 18~25 |
| 性别 | 与编号对应的运动员的性别 | Agender | char | 8 | 男、女 |
| 系团队编号 | 唯一标识系团队 | Dno | char | 8 | D1~Dx |
| 系团队名称 | 与系团队编号对应的系团队名称 | Dname | char | 16 |  |
| 领队 | 团队领队名字 | Dgov | char | 16 |  |

|  |  |  |
| --- | --- | --- |
| 数据结构 |  |  |
| 结构名 | 含义说明 | 组成 |
| 运动员 | 定义一个参加比赛的运动员有关信息，是运动会信息管理系统的主体 | 运动员编号、姓名、年龄、性别 |
| 系团队 | 定义一个系参加运动会的比赛团队 | 系团队编号、名称、领队 |
| 比赛项目 | 定义一个运动会中开展的比赛项目 | 项目编号、项目名称、比赛时间、类别编号 |
| 比赛类别 | 定义一个运动会中开展的比赛的大类 | 类别编号、类别名称、主管 |

**3、构建数据库**

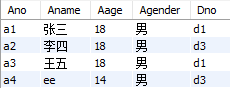
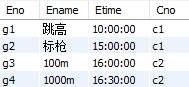
根据以上的理论设计以及数据字典，我们小组准备开始完成数据库的实现。选择了MySQL平台，在平台上创建数据库，建立相关的索引、完整性约束等，并输入实验数据。

构建数据库表的SQL脚本如下：



（由于代码在两个同学的电脑上测试，而报告由两人共同撰写，变量和表的命名可能稍有不同，但不影响功能）

由以上的sql脚本构建的数据库中插入一些基本数据，得到以下的基本表

Athlete 表 Event表

Dept表 classes表

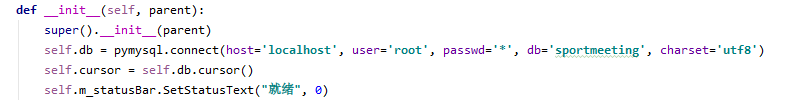
图3 数据库内容实例

**4、编写数据库访问程序实现相应功能**

完成数据库设计后，我们开始编写含数据库访问的应用程序，使用数据互连方法实现数据库的本地调用，完成一定的结构化数据检索、数据检索、数据增删改操纵和数据分类统计等操作。

我们选择使用python来完成数据库的访问以及GUI的设计。在python3.7的环境下，导入pymysql可以实现对MySQL的访问。通过对相关编程语法的学习，我们开始了应用的编写。

**SQL访问的功能：**



使用初始化函数，connect函数连接mysql本地数据库实现对sportmeet的数据库访问，以能够实现后续的增删、更新、查询功能。

**更新、增加数据功能的实现**：由于参加运动会的运动员以及项目是可以在使用数据库之前录入好的。我们设计的应用程序主要是针对运动会进行的过程中，各个比赛项目结束后成绩录入的功能，针对的是score表的更新与插入。

考虑到，有可能当前录入的运动员以及项目的成绩是之前在表中就存在的，要实现的是更新功能而原本不存在的信息要实现的是插入功能。我们将这两个功能合并在了一个按键中，实现自动检查更新还是插入。

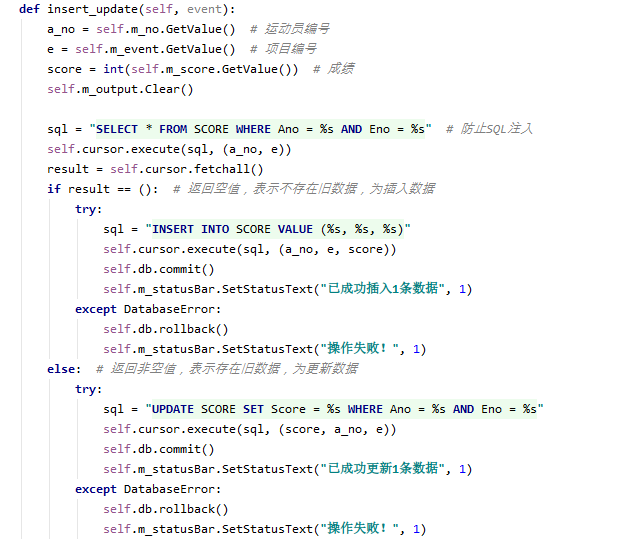


图4 插入、更新程序段

首先按照输入的信息进行表的查询，判断有无相同的元组，即按输入的运动员编号和项目编号查询。如果有相同的元组，即返回非空值，调用update语句更新数据，否则调用insert语句插入数据。

为了防止SQL注入，特意将输入参数加在了execute()函数中，而不是



或



的形式，下同。

**查询功能的实现**：



图5 查询程序段

我们实现了根据运动员姓名、运动员编号、比赛项目编号、比赛项目名称四个关键词的查询功能，并且输出的成绩由低到高进行排序。

**删除功能的实现：**

我们实现了根据运动员姓名、运动员编号、比赛项目编号、比赛项目名称四个关键词的删除功能。系统能够准确的删除不需要的数据，并且返回删除信息条数等信息，供使用者参考。

cursor.execute()函数可以返回受影响的行数，即删除的行数。获取此值后，可以显示在StatusBar上。

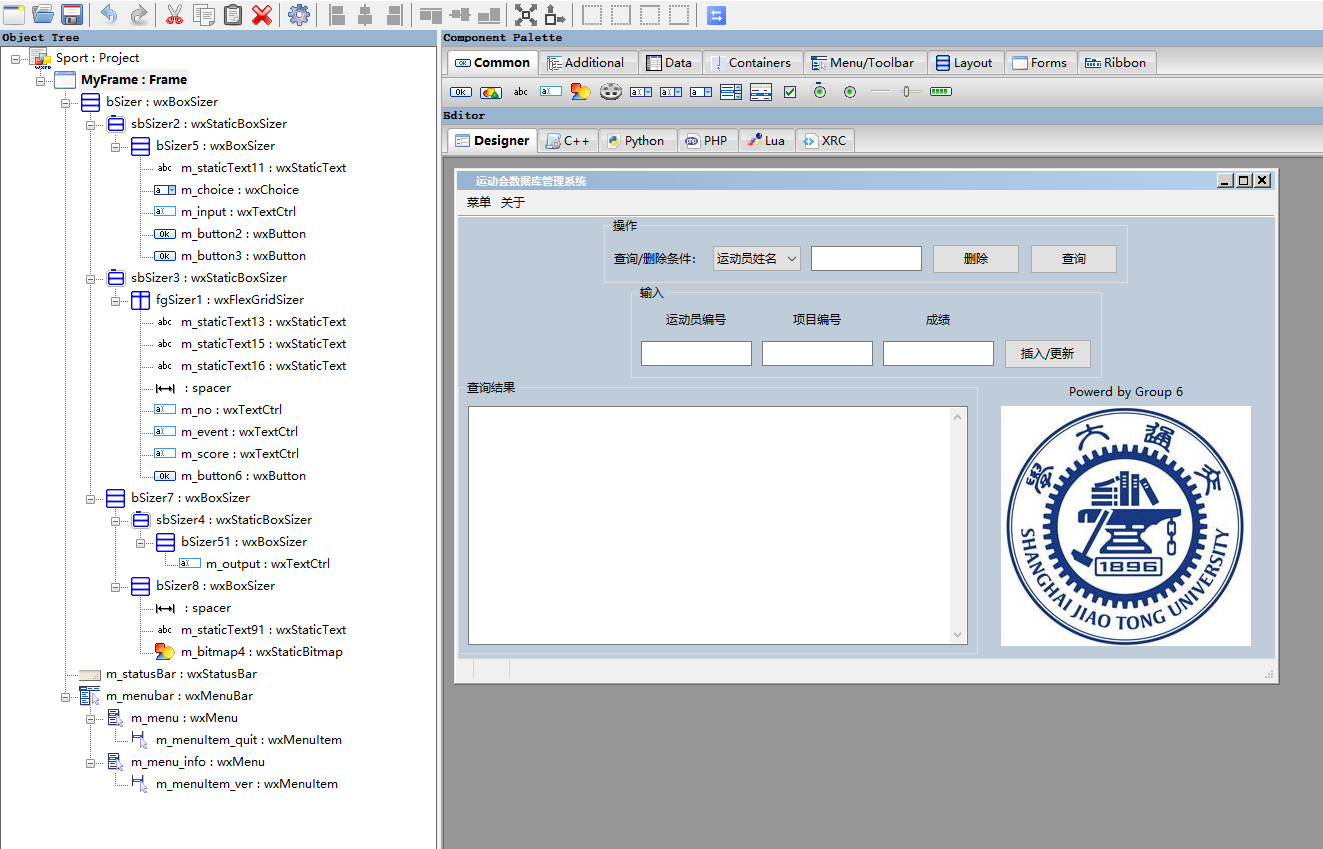


图6 删除功能程序段

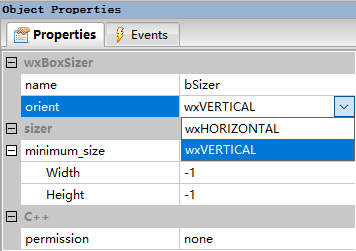
至此我们基本完成了应用程序所需要的功能的实现，接下来我们开始了应用程序GUI的设计。

**GUI的设计：**

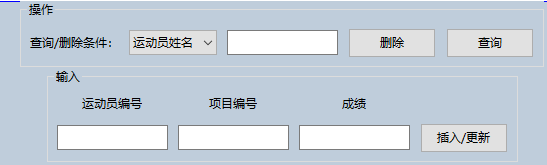
经过比较python GUI三个库QT、WX、tkinter的优缺点之后，我们认为使用wxpython的界面设计工具wxFormBuilder实现GUI较为方便与简单。此工具的优点是可以通过添加控件的方式直接生成代码，免去手写的麻烦。



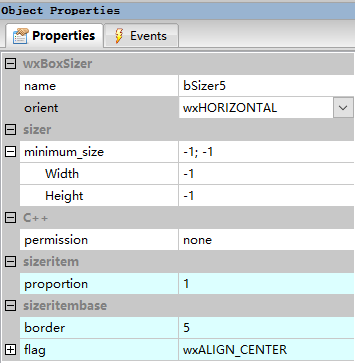
此为在wxFormBuilder中设计的布局。首先添加一个frame窗口，在其中建立一个vertical的layout（wxBoxSizer）：



建立两个wxStaticBoxSizer，proportion设为0，关掉EXPAND属性，分别将label设置为操作和输入：

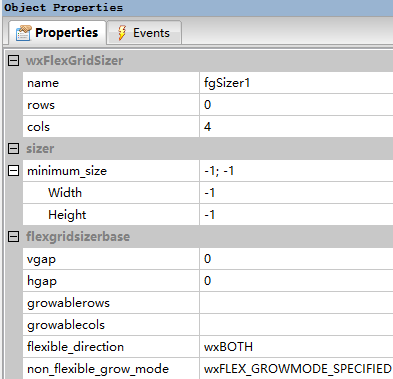
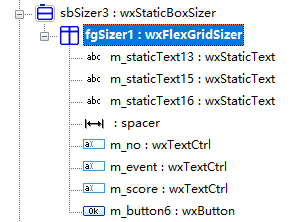


在第一个wxStaticBoxSizer中，建立一个horizontal的wxBoxSizer：



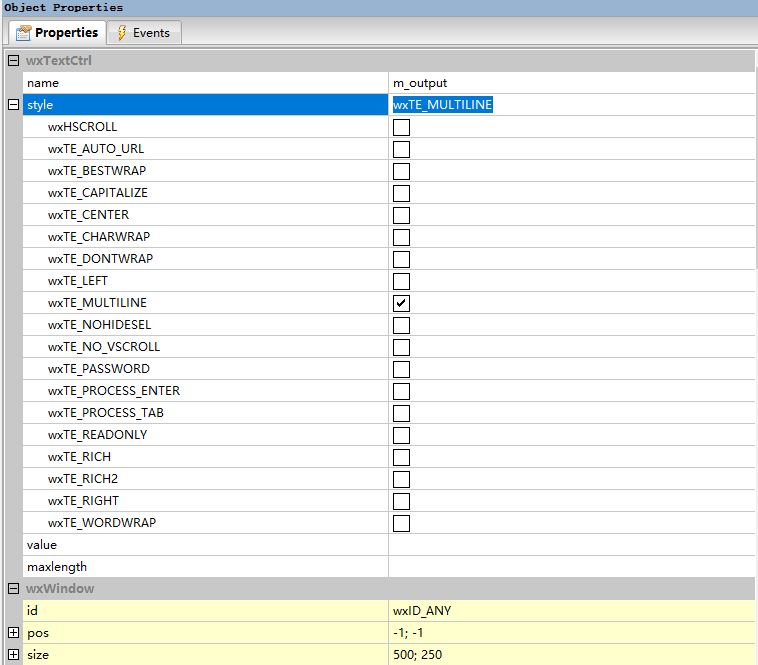
在其中添加按钮、静态文本、输入框，全部添加ALIGN\_CENTER属性，以保证居中。

第二个wxStaticBoxSizer中，添加更为灵活的wxFlexGridSizer，cols设为4，以便添加两行控件。其中插入/更新按钮上是一个spacer。



更改各控件的name和onClick属性，name用于在python中引用变量，onClick用于绑定当点击事件发生时调用的函数。其中wxChoice可以调用m\_choice.GetSelection()函数来确定选了哪一项。

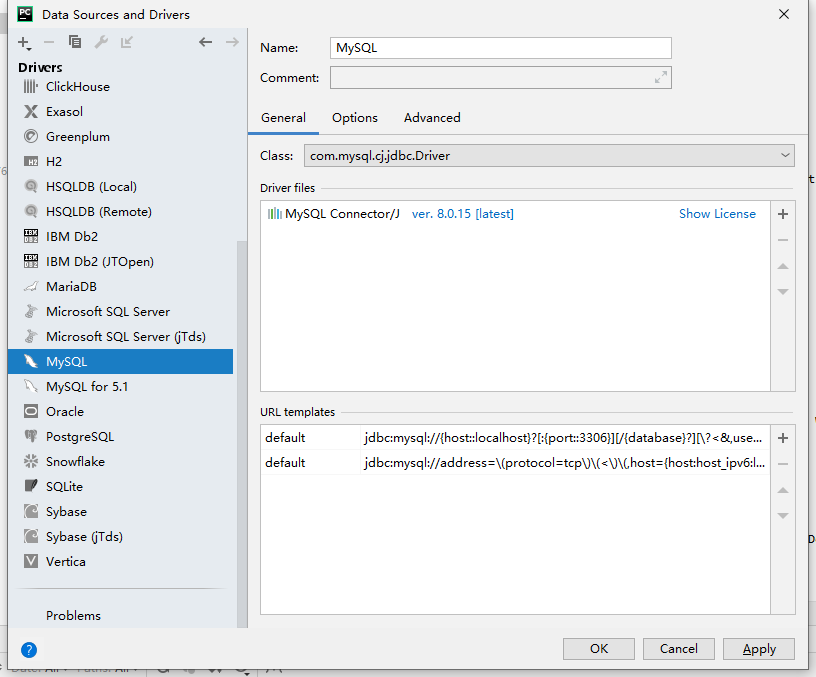
接着添加一个大的文本框用于展示结果，设置wxTE\_MULTILINE属性以展示多行数据。最后加上图片，美化。生成python代码，可以直接继承类使用。



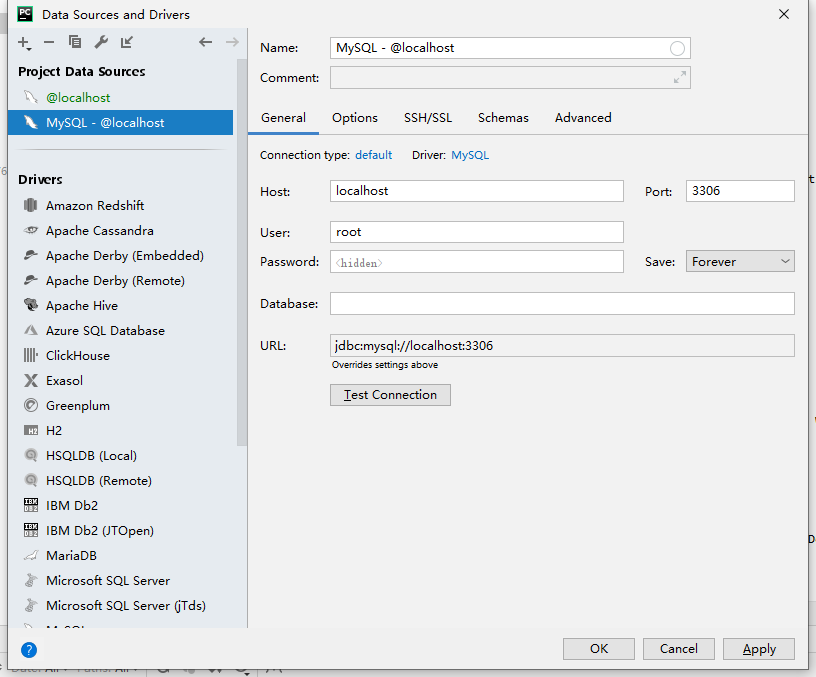
至此代码部分已全部完成。我们已上传全部代码（包括wxFormBuilder工程文件）至GitHub，见<https://github.com/liyufan/Sport>

**测试数据：**

PyCharm Professional自带SQL支持，可以为测试提供帮助。首先安装驱动：



添加连接：



在MySQL命令行中输入set global time\_zone='+8:00';便可成功连接。

此后便可在PyCharm中直接查询，避免了调用MySQL命令行和MySQL Workbench的麻烦。具体演示见随附视频。

