使用django + nginx + mysql搭建成绩管理系统

**需求分析：**

本课题开发一个成绩管理系统，该系统主要用于在校学生成绩的日常管理，用户可通过网络登录操作。系统的基本功能包括实现学生基本信息、课程信息和课程成绩的录入和管理；提供对相关课程成绩的查询；提供成绩统计和分析功能等

**整体设计：整体版面扁平化**

该成绩管理系统初步分为：

Admin：修改全部用户密码，浏览所有用户信息，安排课程时间，添加课程，修改课程时间

Teacher：输入修改本人信息，录入成绩，修改成绩，提交评价…

Student：输入修改本人信息，查看课程信息，查询成绩、评价…

全部成绩统计、分析（ 图表）

**环境版本：**

Python：3.5

Django：1.10.5

SQLite：3

Nginx：1.10.3

SQLite，是一款轻型的数据库，是遵守ACID的关系型数据库管理系统，它包含在一个相对小的C库中。它是D.RichardHipp建立的公有领域项目。它的设计目标是嵌入式的，而且目前已经在很多嵌入式产品中使用了它，它占用资源非常的低，在嵌入式设备中，可能只需要几百K的内存就够了。它能够支持Windows/Linux/Unix等等主流的操作系统，同时能够跟很多程序语言相结合，比如 Tcl、C#、PHP、Java等，还有ODBC接口，同样比起Mysql、PostgreSQL这两款开源的世界著名数据库管理系统来讲，它的处理速度比他们都快。

**功能设计：**

登陆功能：登陆页

登陆成功，失败，忘记密码（使用预留的邮箱修改）

个人信息：个人信息页

查看个人信息，修改个人信息，修改个人密码

课程信息：分三个identity

Student：显示已选课程，查看课程成绩排名，（对课程进行评价、评分，查看 教师评价）

Teacher：显示所教课程，查看学生成绩、排名、评价，对学生进行评价（可选）

统计分析：

**数据库设计：**

User（用户表）:

Id

Password

Name

Gender

Address

phone

Identity(admin/teacher/student)

Teacher（教师表）:

Id

Job\_number

Course\_name

Student(学生表):

Id

Student\_number

Course\_name

Course(课程表):

Course\_number

Course\_name

Course\_intro

Job\_number

Student\_number

Credit(学分)

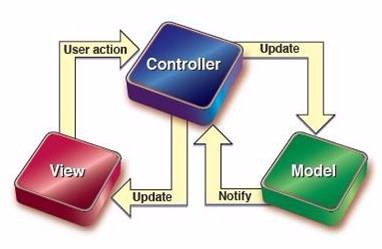
Achievement(成绩表)：

Course\_number

Student\_number

Course\_score

MVC就是把Web应用分为模型(M)，控制器(C)和视图(V)三层，他们之间以一种插件式的、松耦合的方式连接在一起，模型负责业务对象与数据库的映射(ORM)，视图负责与用户的交互(页面)，控制器接受用户的输入调用模型和视图完成用户的请求，其示意图如下所示：

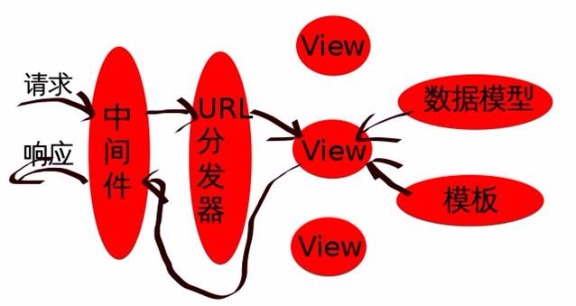


Django的MTV模式本质上和MVC是一样的，也是为了各组件间保持松耦合关系，只是定义上有些许不同，Django的MTV分别是值：

M 代表模型（Model）：负责业务对象和数据库的关系映射(ORM)。

T 代表模板 (Template)：负责如何把页面展示给用户(html)。

V 代表视图（View）：负责业务逻辑，并在适当时候调用Model和Template。

除了以上三层之外，还需要一个URL分发器，它的作用是将一个个URL的页面请求分发给不同的View处理，View再调用相应的Model和Template，MTV的响应模式如下所示：

1，Web服务器（中间件）收到一个http请求

2，Django在URLconf里查找对应的视图(View)函数来处理http请求

3，视图函数调用相应的数据模型来存取数据、调用相应的模板向用户展示页面

4，视图函数处理结束后返回一个http的响应给Web服务器

5，Web服务器将响应发送给客户端

这种设计模式关键的优势在于各种组件都是松耦合的。这样，每个由 Django驱动的Web应用都有着明确的目的，并且可独立更改而不影响到其它的部分。

比如，开发者更改一个应用程序中的 URL 而不用影响到这个程序底层的实现。设计师可以改变 HTML页面的样式而不用接触Python代码。

数据库管理员可以重新命名数据表并且只需更改模型，无需从一大堆文件中进行查找和替换。

落到实处，Django的MTV模式相对应的python文件如下：

