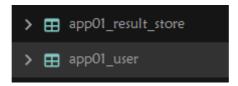
记得在setting.py中修改DATABASES的用户名和密码,连上自己的MySQL

```
1 DATABASES = {
    'default': {
2
         'ENGINE': 'django.db.backends.mysql',
         'NAME': '******', # schema名
5
         'HOST': '127.0.0.1',
         'PORT': 3306,
6
         'USER': '****', # 用户名、密码
7
   }
         'PASSWORD': '****
8
9
10 }
```

然后运行

```
python manage.py makemigrations
python manage.py migrate
```

建立了两张表,一张存储用户名密码,一张存储学生的成绩信息



0 7 id \$	userid \$	🗐 username 💠	assword \$	usertype \$
1	070? userid: varchar(32	gela 2)	123456	STUDENT
2	07E4CA3/7B0378B3	вор	123456	STUDENT
3	0AA5BDC7FBD16CC0	Creed	123456	STUDENT
4	0BBCCCEA11FF61CF	Dwight	123456	STUDENT
5	0DA11CE40B024715	Erin	123456	STUDENT
6	admin	Richard	california	ADMIN
7	teacher	ZZZ	123456	TEACHER
8	test_student	test	123456	newSTUDENT

^{*}注意设置usertype时全大写

目前系统实现的功能:

- 分角色 (老师、学生、管理员) 登录系统
- 学生登录后自动查询信息展示在页面上
- 管理员、老师点进"期末考试预测",可以查询所有学生的成绩(输入学号或姓名)
- 查询学生平时成绩和评价
- 管理员可以在增加/删除用户界面中增加、删除用户
- 删除用户加个确定 (check) 步骤
- 管理员可以在修改用户权限
- 自助成绩预测(待改进)
- 管理员、教师可以在修改用户信息
- *下载 (打印) 结果
- 修改密码

待实现的功能:

• *版本更新 (需预测成绩模型与评价生成模型)

app01/views.py 中的函数解释

- index: 登录界面,在user的表中查询是否有匹配的用户名和密码,验证通过后根据角色不同导向adminindex、studentindex和teacherindex三个页面
- logout: 登出, 只有一条语句返回
- studentindex: 在登录成功后直接在数据库查询这个学生的信息,显示在主页上,这样就不用他们进入查询页面了
- view_score: 在数据库中查询学生预测的成绩
- add_remove_user: 判断提交的按钮是add还是delete, 然后进行插入数据、删除数据的操作
- update_permission: 根据user_id查找待操作的数据, 然后修改它的身份
- self_predict: 调用预先保存的svm模型,输入数据后显示预测的结果
- update_detail: 选择user_id (学生学号) ,输入要修改的值(不需要填满),然后根据有哪些修改的值在数据库中修改对应的值
- update_password:修改密码update_system:版本更新