**后端数据库表的设计：**

后端采用流行的Python框架Django，用来处理各种请求的响应，URL的匹配以及后台数据库的相关增删查改工作。

1. **user\_data 表——对应User类**

该表用于存储用户数据，包括用户账号、密码等，在后面的开发中我们会逐步加入用户账号的其它个人信息，完善相关字段，以满足网站的需求。

每个用户对表中一条记录，而这一条记录在Django的封装接口下以一个基于Python类创建得到的对象呈现，因此我们可以直接对该User对象访问来实现对数据库和表中数据的操作。

下面是目前User的设计要点：

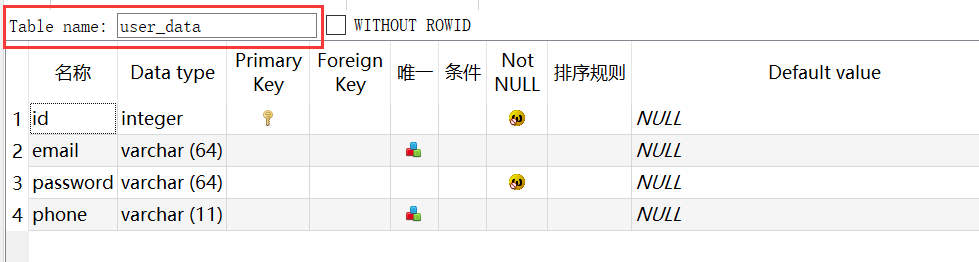
**字段：**

(ID)：隐式生成的字段；自增类型的整数；该表的主键。由phone和email的联合主键选项自动生成。

phone：用户的手机号码。最大长度为11位的字符串；Unique属性为真，意味着整个表中只能有唯一的手机号码，不允许重复；默认缺省值为null。

email：用户的邮箱地址。对大长度为64的字符串，并采用Django的EmailField限制；Unique值为真，不允许重复；默认缺省值为null。

password：用户账号的密码。最大长度为64的字符串。



**约束：**

联合主键：phone 与 email联合唯一，并且由此产生一个ID字段，以ID作为表的主键。意味着用户可以使用手机号和邮箱地址的其中一个作为注册账号，但后面补充信息时不能填写已存在的手机号或者邮箱地址。

唯一：phone、email字段都有唯一限制，不允许重复出现的值。

1. **comment\_data 表——对应Comment类**

该表用于存储在文章中的评论数据，包括评论内容、作者、发布时间等。

**字段：**

(ID)：隐式生成的字段；自增类型的整数；该表的主键。由author和content的联合主键选项自动生成。

content：该条评论的内容。采用Django提供的TextField限制域，本质上是一个字符串。

author：该条评论的作者。此处使用外键属性连接到上面的User中。

time：评论的发布时间。使用Django提供的DateTimeField限制域；使用auto\_now\_add选项，自动提供当前系统时间为默认缺省值。



**特殊关系映射：**

外键：将User链接到author作为外键。意味着每条记录中的author都对应user\_data

中的一条记录，是一种特殊的多对一关系（多条评论对应一个用户）。

隐含的多对多关系：后面将会提到，一个评论被对应到一篇文章，而一篇文章可以对应很多条评论记录。借助这个隐含的多对多关系，我们可以实现反向查询，通过评论找到它属于的文章。

**约束**：

联合主键：author 与 content 联合唯一，并且由此产生一个ID字段，以ID作为表的主键。意味着同一个作者发布的两个评论不能含有一模一样的内容。

1. **article\_data 表——对应Article类**

该表用于存储文章数据，包括文章的标题、作者、内容、评论、发布时间、浏览数和点赞数等等。

**字段：**

(ID)：隐式生成的字段；自增类型的整数；该表的主键。由author和title的联合主键选项自动生成。

title：文章的标题。最大长度为64的的字符串。

author：文章的作者。使用外键链接到User。

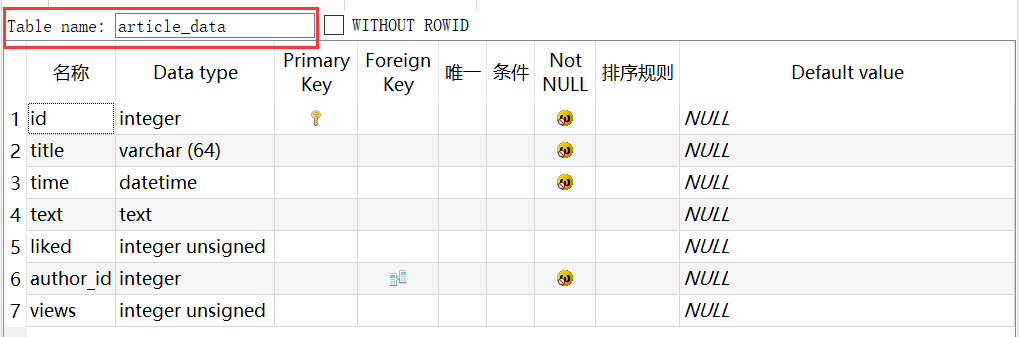
time：文章的发布时间。使用DateTimeField限制域；使用auto\_now\_add选项，自动提供当前系统时间为默认缺省值。

text：文章的主体内容。使用TextField限制域；默认缺省值为null。

comments：文章对应的评论。使用ManyToManyField 链接到Comment，产生多对多关系。

views：文章的阅读量。非负整数；默认初始值为0。

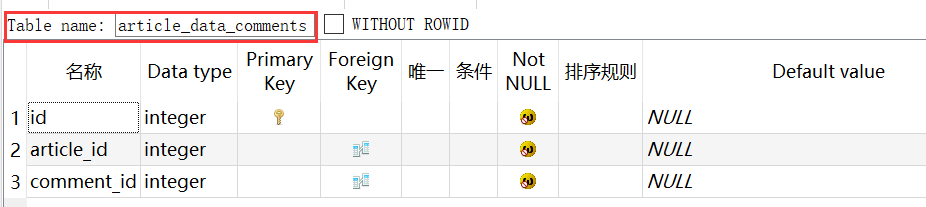
liked：文章的点赞数。非负整数；默认初始值为0。



**特殊关系映射：**

外键：将User链接到author作为外键。实现一篇文章对应外部表中一个用户记录。

多对多关系：借助ManyToMany关系将comment\_data表中的记录链接到comments字段。在Python中comments成员对应的是一个集合，该集合中包含该篇文章对应的所有评论。意味着定义了多对多关系后，我们可以通过文章记录直接得到其下面的评论，无需进行二次条件筛选查询。实际上这种多对多关系的映射是通过另外一个中间表存储各自的主键来实现表A中的一条记录与表B中一条记录的关联关系的存储。



**约束**：

联合主键：author 与 title联合唯一，并且由此产生一个ID字段，以ID作为表的主键。意味着同一个作者发布的两篇文章不能有一样的标题。