# peewee-orm

pip install peewee

# 说明

char类型，要设置最大长度max\_length

对于无法确定最大长度字段要设置成textfiled

# 基本使用方法

from peewee import \*

# 连接数据库

db = MySQLDatabase(“数据库名称”,host=”11”,port=”3306”,user=”root”,passwd)

# 没有设置主键，系统会默认添加id字段，并设为主键

class Name(Model): #这个model是peewee中的model

name = CharFiled(max\_length=255,verbose\_name=”注释”,index=True)

class Mate:

database= db # 连接的数据库

table\_name = “自定义” # 表名

if \_\_name\_\_ ==”\_\_main\_\_”:

# 生成表结构，如果有就不会新建

# Name是之前定义的类名，自动生成数据表

# 可以自定义一个生成表的文件

db.create\_tables([Name])

# sql语法

## 插入

# 导入model,要操作的表

from appmodel import Superline

def add\_model ():

superline = Superline()

superline.name = “hangzhou”

superline.save() #保存数据可修改，可新建

return superline.id #数据库生成的id，保存后返回的id值

#insert()

Superline.insert(ID=id).execute()

***insert返回的是插入成功的ID，失败返回空***

## 查询

1. **获取一条数据**

def get\_model ()：

# ModelName.id是model中定义的字段

# .get() 取不到数据会抛出异常

ModelName.get(ModelName.id==”1”)

ModelName.get\_by\_id(1) # 通过id获取

#会抛异常，

ModelName.select().where(ModelName.id == “1”).get()

#不使用get()会返回空

ModelName.select().where(ModelName.id == “1”)

1. **获取所有数据**

def get\_model ():

# 类似生成器，在for循环时才调用，迭代协议

# select \* from Superline，不执行，调用时才执行

ModelName.select()

# 查询指定字段

ModelName.select(Superline.name)

# 根据条件获取

ModelName.select().where(Superline.name == “11”)

#多个条件

ModelName.select().where((ModelName.name=“11”)&(ModelName.name!=”aa”))

#模糊查询

# select \* from Superline where name like “%飞天”

# 查询官网query Examples

ModelName.select().where(ModelName.name.**contains**(“飞天”))

1. **排序**

ModelName.select().order\_by(Superline.id.desc()降序) .asc() //升序

1. **分页**

# 2代表第3条数据 ， 4代表取几条数据

ModelName.select().order\_by(ModelName.id).paginate(2,4)

## 更新

1. save更新

# 会直接更新之前查询到的数据 ，youid更新没有进行插入

good = Good.get\_by\_id(1)

good.name = “22”

good.save()

1. 使用update

# update click\_num=100 where id=1

#必须执行execute才能执行update语句，update生成的是一个update对象

Good.update(click\_num=100).where(Good.id==1).execute() Good.update(click\_num=Good.click\_num+1).where(Good.id==1).execute()

Good.update({Good.num:1000}).execte()

***如果更新成功返回1，返回的是受影响的函数***

## 删除

Goods.delete().where(...).execute()

## join查询

res = OrderInfo.select(A.aa, B.bb).join(B, on=(A.id==B.id)).where(OrderInfo.id==1)

for item in res:

res.b.bb //打印值，必须小写