多对多需要3张表

可选参数

Integer 整形

String(50) 字符串50

nullable=False 不为空

default=”” 默认值

primary\_key=True 主键

autoincrement=True 自增

class Base(db.Model):

\_\_abstract\_\_ = True # 不创建表

\_\_tablename\_\_ = “” #自定义表名

\_passwd = Columu(“password”,String(64)) #更改字段名称password

# 查询

User.query.filter(User.id=name).first()

User.query.filter\_by(id=name).first() 查询一条

group\_by() 分组

.order\_by(Gift.create\_time) 排序

.order\_by(desc(gift.create\_time)) //倒序 列表推导式

.limit(30) 查询30条

.all() 查询所有

如果没有查询到数据，返回的是空值

.first\_or\_404() 查询不到数据，会抛出异常

# 表达式查询

in查询

db.session.query(with).filter(With.langer == false, Wish.isbn.in\_(isbn\_list))

db.session.query(func.count(with.id))..... 查询数量

# 添加

user = User()

user.id = 10

db.session.add(user)

db.session.commit()

# 事务

天然支持事务

try:

gift = Gift()

gift.id = 1

db.session.add(gift)

db.session.commit()

except Exception as e:

db.session.rollback() //回滚

reise e

方法二 修改SQLAlchemy

from flask\_sqlalchemy import SQAlchemy as \_SQLAlchemy

from contextlib import contextmanager

class SQLAlchemy(\_SQLAlchemy)

@contextmanager

def auto\_commit(self):

try:

yield

self.session.commit()

except Exception as e:

self.session.rollback()

调用

with db.auto\_commit():

# 正常的数据库添加

gift = Gift()

gift.id = 1

db.session.add(gift)

# 修改查询状态

class Query(BaseQuery):

def filter\_by(self, \*\*kwargs):

if ‘status’ not in kwargs.keys():

kwargs[‘status’]=1

return super(Query, self).filter\_by(\*\*kwargs) # 传入字典，必须解包

db = SQLAlchemy(query\_class=Query) # 替换查询query

# 多对多关系

1. UserTable、GiftTable、BookTable

设置外键

class GiftTable(db.Model)

#表示引用关系，引用User表，并使用user表的id

user = relationship(“User”)

uid = Column(Integer, ForeignKey(“user.id”)

# 设置多个外键

user = relationship(“User”)

uid = Column(Integer, ForeignKey(“user.id”)