**filter过滤条件：**

过滤是数据提取的一个很重要的功能，以下对一些常用的过滤条件进行解释，并且这些过滤条件都是只能通过filter方法实现的：

**1. equals ： ==**

news= session.query(News).filter(News.title == "title1").first()

**2. not equals : !=**

query(User).filter(User.name != 'ed')

**3. like & ilike [不区分大小写]：**

query(User).filter(User.name.like('%ed%'))

**4. in：**

query(User).filter(User.name.in\_(['ed','wendy','jack']))

**5. not in：**

query(User).filter(~User.name.in\_(['ed','wendy','jack']))

**6. is null：**

query(User).filter(User.name==None)

# 或者是

query(User).filter(User.name.is\_(None))

**7. is not null:**

query(User).filter(User.name != None)

# 或者是

query(User).filter(User.name.isnot(None))

**8. and：**

query(User).filter(and\_(User.name=='ed',User.fullname=='Ed Jones'))

# 或者是传递多个参数

query(User).filter(User.name=='ed',User.fullname=='Ed Jones')

# 或者是通过多次filter操作

query(User).filter(User.name=='ed').filter(User.fullname=='Ed Jones')

**9. or：**

query(User).filter(or\_(User.name=='ed',User.name=='wendy'))

如果想要查看orm底层转换的sql语句，可以在filter方法后面不要再执行任何方法直接打印就可以看到了。比如：

news = session.query(News).filter(or\_(News.title=='abc',News.content=='abc'))

print(news)

**代码演示：**

|  |
| --- |
| **from** sqlalchemy **import** create\_engine,Column,Integer,Float,Boolean,DECIMAL,Enum,\  Date,DateTime,Time,String,Text,func,or\_,and\_  **from** sqlalchemy.dialects.mysql **import** LONGTEXT  **from** sqlalchemy.ext.declarative **import** declarative\_base  **from** sqlalchemy.orm **import** sessionmaker  **import** random  HOSTNAME = **'127.0.0.1'**  PORT = **'3306'**  DATABASE = **'first\_sqlalchemy'**  USERNAME = **'root'**  PASSWORD = **'root'**  DB\_URI =**"mysql+pymysql://{username}:{password}@{host}:{port}/{db}?charset=utf8"**.format(username=USERNAME,password=PASSWORD,host=HOSTNAME,port=PORT,db=DATABASE)  engine = create\_engine(DB\_URI)  Base = declarative\_base(engine)  session = sessionmaker(engine)()  **class** Article(Base):  \_\_tablename\_\_ = **'article'**  id = Column(Integer,primary\_key=**True**,autoincrement=**True**)  title = Column(String(50),nullable=**False**)  price = Column(Float,nullable=**False**)  content = Column(Text)  **def** \_\_repr\_\_(self):  **return "<Article(title:%s)>"** % self.title  #共两种查询 过滤的方法  # r1 = session.query(News).filter(News.id == 1).first()  # print(r1)  # r2 =session.query(News).filter\_by(id = 2).first()  # print(r2)  # 1. equal  # news = session.query(News).filter(News.title == "title0").first()  # print(news)  # 2. not equal  # news= session.query(News).filter(News.title != 'title0').all()  # print(news)  # 3. like & ilike（不区分大小写）  # news= session.query(News).filter(News.title.ilike('title%')).all()  # print(news)  # 4. in：  # for xxx in xxx  # def \_in()  # news= session.query(News).filter(News.title.in\_(['title1','title2'])).all()  # print(news)  #5. not in  # news= session.query(News).filter(~News.title.in\_(['title1','title2'])).all()  # print(news)  # news= session.query(News).filter(News.title.notin\_(['title1','title2'])).all()  # print(news)  #6. is null  #修改表，添加一个列content,操作数据  # news= session.query(News).filter(News.content==None).all()  # print(news)  #7. is not null  # news= session.query(News).filter(News.content!=None).all()  # print(news)  #8. and  # news= session.query(News).filter(News.title=='title5',News.content=='abc').all()  # print(news)  # 或者  # news= session.query(News).filter(and\_(News.title=='title5',News.content=='abc')).all()  # print(news)  #9.or  # news= session.query(News).filter(or\_(News.title=='title3',News.content=='abc')).all()  # print(news) |