**表关系：**

表之间的关系存在三种：一对一、一对多、多对多。

而SQLAlchemy中的ORM也可以模拟这三种关系。

因为一对一其实在SQLAlchemy中底层是通过一对多的方式模拟的，所以先来看下一对多的关系：

**外键：**

使用SQLAlchemy创建外键非常简单。在从表中增加一个字段，指定这个字段外键的是哪个表的哪个字段就可以了。从表中外键的字段，必须和主表的主键字段类型保持一致。

示例代码如下：

|  |
| --- |
| # 主表 / 从表  # user/news  **class** User(Base):  \_\_tablename\_\_ = **'user'**  id = Column(Integer,primary\_key=**True**,autoincrement=**True**)  uname = Column(String(50),nullable=**False**)  **def** \_\_repr\_\_(self):  **return "<User(uname:%s)>"** % self.uname  **class** News(Base):  \_\_tablename\_\_ = **'news'**  id = Column(Integer,primary\_key=**True**,autoincrement=**True**)  title = Column(String(50),nullable=**False**)  content = Column(Text,nullable=**False**)  uid = Column(Integer,ForeignKey(**"user.id"**))  **def** \_\_repr\_\_(self):  **return "<News(title:%s,content=%s)>"** % (self.title,self.content) |

**外键约束有以下几项：**

**1.** RESTRICT：若子表中有父表对应的关联数据，删除父表对应数据，会阻止删除。默认项

**2.** NO ACTION：在MySQL中，同RESTRICT。

**3.** CASCADE：级联删除。

**4.** SET NULL：父表对应数据被删除，子表对应数据项会设置为NULL。

**演示代码如下：**

|  |
| --- |
| **from** sqlalchemy **import** create\_engine,Column,Integer,Float,Boolean,DECIMAL,Enum,\  Date,DateTime,Time,String,Text,func,or\_,and\_,ForeignKey  **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)()  # 父表/从表  # user/news  **class** User(Base):  \_\_tablename\_\_ = **'user'**  id = Column(Integer,primary\_key=**True**,autoincrement=**True**)  uname = Column(String(50),nullable=**False**)  **def** \_\_repr\_\_(self):  **return "<User(uname:%s)>"** % self.uname  **class** News(Base):  \_\_tablename\_\_ = **'news'**  id = Column(Integer,primary\_key=**True**,autoincrement=**True**)  title = Column(String(50),nullable=**False**)  content = Column(Text,nullable=**False**)  # uid = Column(Integer,ForeignKey("user.id",ondelete='RESTRICT'))  # uid = Column(Integer,ForeignKey("user.id",ondelete='NO ACTION'))  # uid = Column(Integer,ForeignKey("user.id",ondelete='CASCADE'))  uid = Column(Integer,ForeignKey(**"user.id"**,ondelete=**'SET NULL'**))  **def** \_\_repr\_\_(self):  **return "<News(title:%s,content=%s)>"** % (self.title,self.content)  Base.metadata.drop\_all()  Base.metadata.create\_all()  user = User(uname=**'momo'**)  session.add(user)  session.commit()  news1= News(title=**'AAA'**,content=**'123'**,uid=1)  news2= News(title=**'BBB'**,content=**'456'**,uid=1)  session.add\_all([news1,news2])  session.commit() |