**ORM层面删除数据注意事项：**

ORM层面删除数据，会无视mysql级别的外键约束。

直接会将对应的数据删除，然后将从表中的那个外键设置为NULL。

如果想要避免这种行为，应该将从表中的外键的`nullable=False`。

**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,Table  **from** sqlalchemy.dialects.mysql **import** LONGTEXT  **from** sqlalchemy.ext.declarative **import** declarative\_base  **from** sqlalchemy.orm **import** sessionmaker,relationship,backref  **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** User(Base):  \_\_tablename\_\_ = **'user'**  id = Column(Integer,primary\_key=**True**,autoincrement=**True**)  uname = Column(String(50),nullable=**False**)  **class** News(Base):  \_\_tablename\_\_ = **'news'**  id = Column(Integer, primary\_key=**True**, autoincrement=**True**)  title = Column(String(50),nullable=**False**)  uid = Column(Integer,ForeignKey(**"user.id"**),nullable=**False**)  author = relationship(**"User"**,backref=**'newss'**)  Base.metadata.drop\_all()  Base.metadata.create\_all()  user = User(uname=**'momo'**)  news= News(title=**'hello world'**)  news.author = user  session.add(news)  session.commit()  #若 uid = Column(Integer,ForeignKey("user.id")) 没有声明非空，删除父表能成功，会将子表数据 对应项置为空  #解决 若uid = Column(Integer,ForeignKey("user.id"),nullable=False) 会被阻止删除父表  user = session.query(User).first()  session.delete(user)  session.commit() |