总结:

第一步都是创建引擎:

```
engine = create_engine('mysql+pymysql://root:mysql@myubuntu:3306/test_soon')
```

第二步创建基类，有两种方式:

- 使用自动匹配基类---from sqlalchemy.ext.automap import

automap_base

```
# 启动自动匹配，取出表类
    # 匹配引擎中的表类
    Base.prepare(engine, reflect=True)

    # 从引擎中取出映射tblteacher表的数据类
    teachers = Base.classes.tblteacher
```

- 使用映射基类--from sqlalchemy.ext.declarative import

declarative_base

  - 需要自己写映射类，可以创建新表

```
Base.metadata.create_all(engine)
```

第三步创建会话session

```
session = sessionmaker(bind=engine)()
# 查询
session.query(类--teacher/Teacher)
```

实例

```python
import sqlalchemy


def first_way():
    #   第一种方式

    from sqlalchemy.ext.automap import automap_base
    from sqlalchemy.orm import sessionmaker
    from sqlalchemy import create_engine

    engine = create_engine('mysql+pymysql://root:mysql@myubuntu:3306/test_soon')
    # 自动匹配基类
    Base = automap_base()
    # 放入连接引擎
    Base.prepare(engine, reflect=True)

    # 从引擎中取出映射tblteacher表的数据类
    teachers = Base.classes.tblteacher

    # 做一个实例会话出来
    session = sessionmaker(bind=engine)()
    # 把模型放进会话中
    rets = session.query(teachers).all()

    for i in rets:
        print(i)
        # <sqlalchemy.ext.automap.tblteacher object at 0x0000026EC1E60DD8>
        # <sqlalchemy.ext.automap.tblteacher object at 0x0000026EC1E60E48>
        # ....
    # print(rets.__table__)

# 第二种方法
```

```python
def second_way():
    # 导入的是映射包
    from sqlalchemy.ext.declarative import declarative_base
    from sqlalchemy import Column,INT,String,create_engine
    from sqlalchemy.orm import sessionmaker

    # 引擎
    engine = create_engine('mysql+pymysql://root:mysql@myubuntu:3306/test_soon')
    # 映射基础表
    Base = declarative_base(bind=engine)

    # 映射的教师表
    class Teacher(Base):
        __tablename__ = 'tblteacher'
        name = Column('TeaName',String(),nullable=False,index=True, unique=True) # 不允许
        id = Column('TeaId',INT(),primary_key=True)

        def __repr__(self):
            return self.id + ":" + self.name
    # metadata = Base.metadata
    # metadata.create_all(engine),创建新表的意思

    # 创建会话，实例化
    session = sessionmaker(bind=engine)()
    # 添加新的教师对象
    teacher = Teacher(id=105,name="liukaitao")
    session.add(teacher)
    session.commit()

    # 查询所有
    teachers = session.query(Teacher).all()
    for i in teachers:
        print(i)
        print(i.id+":"+i.name)


if __name__ == '__main__':
```

```
    second_way()
```