data_base_demo

October 17, 2018

1 简单数据库操作

我们在之前的《树莓派入坑系列》就给大家介绍过如果安装和配置数据库。在一下的这一部分内容，将给大家演示如何使用 Python 去操作数据库。

在准备这一部分的内容的时候，我花了不少时间去考虑，想怎么样去以一种简单的方式来介绍使用 Python 以及数据库。因为，在这么多期的视频反馈中，我了解到我们的观众朋友们已经掌握的知识以及长处差别还是比较大的。针对一些初学、对这方面刚入门的朋友，可能还不清楚什么是关系型数据库，什么是 UML 什么是 E-R 图；而对于一些有目的性想要开发某种软件或硬件应用的朋友，又更渴望看到具体实务相关的数据库设计内容；一些有相当基础和经验的朋友，可能又希望能看到有关进阶数据库操作、性能、部署运维方面的东西。

但是，我们的视频却又有相当的局限性，我能从后台数据上明显感觉到，只要视频的长度稍微增加，观看的数量以及留存率都会锐减，这就形成了一个矛盾，**大家都渴望得到干货，也同样并不是已经全知全能，而且还没时间去广撒网捕获新知**。这也许是这个时代的问题，我也不可能有解决方案。

所以，我的视频以及周边资料，希望能达到两个目标：
- 简化观众朋友们的学习平台困难 让你对一件事感兴趣之后想要深入的第一步变得稍微轻松一点 - 有一定的启发 让你对自已想做的事情更有灵感

因此，当你看完视频后，希望你能感觉 —— “我的 XX 想法其实也可以 XXX 试试，也许不难，能行”，而不是，“Wow，这个 Up 主好 Cool！”。

1.1 本期目标

本期将以《Flask Web 开发：基于 Python 的 Web 应用开发实战》一书的第五章作为参考，使用 SQLAlchemy 库对 SQLite 数据库进行：
创建表 插入数据 查询数据 修改数据 删除数据
的操作。《Flask Web》一书非常详尽的介绍如何从零开始构建一个项目，非常值得阅读。这里的内容仅仅参考了其中的一小部分数据库表的设计，如果想要了解更多，建议自行阅读。

1.2 SQLAlchemy 数据库安装

```
pip install sqlalchemy
```

1.2.1 数据库连接配置

```
In [1]: # 引入包
        from sqlalchemy import Column, String, Integer, create_engine, ForeignKey
        from sqlalchemy.orm import sessionmaker, relationship
        from sqlalchemy.ext.declarative import declarative_base
```

```
In [2]: # 初始化数据库配置
        Base = declarative_base()
        engine = create_engine('sqlite:///demo.db', echo=True)
        Database = sessionmaker(bind=engine)
```

定义表 Role User
primary_key 主键
unique 是否唯一
index 创建索引（查询效率有关）
repr() 返回格式化的字符串，方便测试用

```
In [3]: class Role(Base):
            __tablename__ = 'roles'
            id = Column('id', Integer, primary_key=True)
            name = Column('name', String(64), unique=True)
            users = relationship('User', backref='role')

            def __repr__(self):
                return '<Role %r>' % self.name

        class User(Base):
            __tablename__ = 'users'
            id = Column('id', Integer, primary_key=True)
            username = Column('name', String(64), unique=True, index=True)
            role_id = Column(Integer, ForeignKey('roles.id'))

            def __repr__(self):
                return '<User %r>' % self.username
```

初始化数据库 db

```
db = Database()
```

创建表（第一次执行时使用）

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

```
In [4]: db = Database()
        Base.metadata.create_all(engine)
```

```
2018-10-17 17:46:07,113 INFO sqlalchemy.engine.base.Engine SELECT CAST('test plain returns' AS
2018-10-17 17:46:07,126 INFO sqlalchemy.engine.base.Engine ()
2018-10-17 17:46:07,140 INFO sqlalchemy.engine.base.Engine SELECT CAST('test unicode returns'
2018-10-17 17:46:07,147 INFO sqlalchemy.engine.base.Engine ()
2018-10-17 17:46:07,169 INFO sqlalchemy.engine.base.Engine PRAGMA table_info("roles")
2018-10-17 17:46:07,175 INFO sqlalchemy.engine.base.Engine ()
2018-10-17 17:46:07,193 INFO sqlalchemy.engine.base.Engine PRAGMA table_info("users")
2018-10-17 17:46:07,198 INFO sqlalchemy.engine.base.Engine ()
2018-10-17 17:46:07,205 INFO sqlalchemy.engine.base.Engine
```

```
CREATE TABLE roles (
        id INTEGER NOT NULL,
        name VARCHAR(64),
        PRIMARY KEY (id),
        UNIQUE (name)
)


2018-10-17 17:46:07,210 INFO sqlalchemy.engine.base.Engine ()
2018-10-17 17:46:07,252 INFO sqlalchemy.engine.base.Engine COMMIT
2018-10-17 17:46:07,257 INFO sqlalchemy.engine.base.Engine
CREATE TABLE users (
        id INTEGER NOT NULL,
        name VARCHAR(64),
        role_id INTEGER,
        PRIMARY KEY (id),
        FOREIGN KEY(role_id) REFERENCES roles (id)
)


2018-10-17 17:46:07,261 INFO sqlalchemy.engine.base.Engine ()
2018-10-17 17:46:07,284 INFO sqlalchemy.engine.base.Engine COMMIT
2018-10-17 17:46:07,289 INFO sqlalchemy.engine.base.Engine CREATE UNIQUE INDEX ix_users_name O
2018-10-17 17:46:07,293 INFO sqlalchemy.engine.base.Engine ()
2018-10-17 17:46:07,324 INFO sqlalchemy.engine.base.Engine COMMIT
```

查询当前数据库表

Base.metadata.tables

```
In [5]: tables = Base.metadata.tables
        print(tables)
```

```
immutabledict({'roles': Table('roles', MetaData(bind=None), Column('id', Integer(), table=<rol
```

1.2.2 插入新数据操作

向数据库中插入新数据 - 实例化一个数据模型 - db.add() - db.commit()

```
In [6]: admin_role = Role(name='Admin')
        mem_role = Role(name='Menber')
        user_role = Role(name='User')
        user_xiaoming = User(username='xiaoming', role=admin_role)
        user_xiaohong = User(username='xiaohong', role=user_role)
        user_xiaolan = User(username='xiaolan', role=user_role)
```

```
In [7]: db.add_all([admin_role, mem_role, user_role, user_xiaohong, user_xiaolan, user_xiaoming
In [8]: db.commit()
```

```
2018-10-17 17:48:00,853 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2018-10-17 17:48:00,873 INFO sqlalchemy.engine.base.Engine INSERT INTO roles (name) VALUES (?)
2018-10-17 17:48:00,881 INFO sqlalchemy.engine.base.Engine ('Admin',)
2018-10-17 17:48:00,890 INFO sqlalchemy.engine.base.Engine INSERT INTO roles (name) VALUES (?)
2018-10-17 17:48:00,894 INFO sqlalchemy.engine.base.Engine ('Menber',)
2018-10-17 17:48:00,898 INFO sqlalchemy.engine.base.Engine INSERT INTO roles (name) VALUES (?)
2018-10-17 17:48:00,902 INFO sqlalchemy.engine.base.Engine ('User',)
2018-10-17 17:48:00,911 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, role_id) V
2018-10-17 17:48:00,916 INFO sqlalchemy.engine.base.Engine ('xiaoming', 1)
2018-10-17 17:48:00,922 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, role_id) V
2018-10-17 17:48:00,926 INFO sqlalchemy.engine.base.Engine ('xiaohong', 3)
2018-10-17 17:48:00,932 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, role_id) V
2018-10-17 17:48:00,937 INFO sqlalchemy.engine.base.Engine ('xiaolan', 3)
2018-10-17 17:48:00,944 INFO sqlalchemy.engine.base.Engine COMMIT
```

查询数据库

- db.query()
- 遍历查询结果

```
In [9]: # 查询 User 表
        for row in db.query(User).all():
            print(row.id, row.username, row.role)
```

```
2018-10-17 17:48:41,210 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2018-10-17 17:48:41,221 INFO sqlalchemy.engine.base.Engine SELECT users.name AS users_name, us
FROM users
2018-10-17 17:48:41,230 INFO sqlalchemy.engine.base.Engine ()
2018-10-17 17:48:41,254 INFO sqlalchemy.engine.base.Engine SELECT roles.id AS roles_id, roles.
FROM roles
WHERE roles.id = ?
2018-10-17 17:48:41,261 INFO sqlalchemy.engine.base.Engine (1,)
1 xiaoming <Role 'Admin'>
2018-10-17 17:48:41,274 INFO sqlalchemy.engine.base.Engine SELECT roles.id AS roles_id, roles.
FROM roles
WHERE roles.id = ?
2018-10-17 17:48:41,278 INFO sqlalchemy.engine.base.Engine (3,)
2 xiaohong <Role 'User'>
3 xiaolan <Role 'User'>
```

```
In [10]: # 查询 Role 表
         for row in db.query(Role).all():
             print(row)
```

```
2018-10-17 17:48:52,548 INFO sqlalchemy.engine.base.Engine SELECT roles.id AS roles_id, roles.
FROM roles
```

```
2018-10-17 17:48:52,558 INFO sqlalchemy.engine.base.Engine ()
<Role 'Admin'>
<Role 'Menber'>
<Role 'User'>
```

```
In [11]: # 过滤查询
         for row in db.query(User).filter_by(role=user_role).all():
             print(row)
```

```
2018-10-17 17:49:22,257 INFO sqlalchemy.engine.base.Engine SELECT users.name AS users_name, use
FROM users
WHERE ? = users.role_id
2018-10-17 17:49:22,264 INFO sqlalchemy.engine.base.Engine (3,)
<User 'xiaohong'>
<User 'xiaolan'>
```

1.2.3 修改行数据

- 重新定义数据实例
- db.add()
- db.commit()

```
In [12]: # 修改行
         user_role.name = 'VIP'
         db.add(user_role)
         db.commit()
```

```
2018-10-17 17:50:30,090 INFO sqlalchemy.engine.base.Engine UPDATE roles SET name=? WHERE roles
2018-10-17 17:50:30,101 INFO sqlalchemy.engine.base.Engine ('VIP', 3)
2018-10-17 17:50:30,112 INFO sqlalchemy.engine.base.Engine COMMIT
```

```
In [13]: for row in db.query(Role).all():
             print(row)
```

```
2018-10-17 17:50:35,263 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2018-10-17 17:50:35,273 INFO sqlalchemy.engine.base.Engine SELECT roles.id AS roles_id, roles.n
FROM roles
2018-10-17 17:50:35,281 INFO sqlalchemy.engine.base.Engine ()
<Role 'Admin'>
<Role 'Menber'>
<Role 'VIP'>
```

```
In [14]: for row in db.query(User).all():
             print(row.id, row.username, row.role)
```

```
2018-10-17 17:51:51,617 INFO sqlalchemy.engine.base.Engine SELECT users.name AS users_name, use
FROM users
2018-10-17 17:51:51,628 INFO sqlalchemy.engine.base.Engine ()
1 xiaoming <Role 'Admin'>
2 xiaohong <Role 'VIP'>
3 xiaolan <Role 'VIP'>
```

1.2.4 删除行数据

- db.delete()
- db.commit()

```
In [15]: # 删除行
         db.delete(mem_role)
         db.commit()
```

```
2018-10-17 17:52:49,783 INFO sqlalchemy.engine.base.Engine SELECT users.name AS users_name, use
FROM users
WHERE ? = users.role_id
2018-10-17 17:52:49,795 INFO sqlalchemy.engine.base.Engine (2,)
2018-10-17 17:52:49,811 INFO sqlalchemy.engine.base.Engine DELETE FROM roles WHERE roles.id =
2018-10-17 17:52:49,819 INFO sqlalchemy.engine.base.Engine (2,)
2018-10-17 17:52:49,832 INFO sqlalchemy.engine.base.Engine COMMIT
```

```
In [16]: for row in db.query(Role).all():
             print(row)
```

```
2018-10-17 17:52:53,064 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2018-10-17 17:52:53,073 INFO sqlalchemy.engine.base.Engine SELECT roles.id AS roles_id, roles.
FROM roles
2018-10-17 17:52:53,082 INFO sqlalchemy.engine.base.Engine ()
<Role 'Admin'>
<Role 'VIP'>
```