数据库ORM操作

1. 数据库ORM操作实现

为了实现ORM方式操作数据库，而不是使用SQL语句，就需要通过安装`Flask-SQLAlchemy`扩展库来实现。同时通过安装`Flask-Migrate`扩展库实现数据库更新的跟踪，并实现数据迁移。

```
pip install Flask-SQLAlchemy
pip install Flask-Migrate
```

2. 配置sqlite数据库

对于小型应用来说，使用sqlite数据库更为便利，每一个数据库都单独存放在一个文件中。使用sqlite数据库需要在`app/config.py`配置文件进行相关配置。

```
import os

# 配置SQLITE数据库信息
BASE_DIR = os.path.abspath(os.path.dirname(__file__))
# 数据库文件存放路径
SQLALCHEMY_DATABASE_URI = 'sqlite:///' + os.path.join(BASE_DIR, '..', 'instance',
'flask.sqlite')
```

3. 编写数据库定义函数

新建`app/database.py`文件，并编写数据库定义方法。

```
from sqlalchemy import create_engine
from sqlalchemy.orm import scoped_session, sessionmaker
from sqlalchemy.ext.declarative import declarative_base
from flask_migrate import Migrate

from app.config import SQLALCHEMY_DATABASE_URI

# 创建数据库引擎
engine = create_engine(SQLALCHEMY_DATABASE_URI, convert_unicode=True)
# 创建数据库会话
db_session = scoped_session(sessionmaker(
    autocommit=False,
    autoflush=False,
    bind=engine
))
# 创建数据库模型对象
BaseModel = declarative_base()
# 为数据库模型对象添加Query对象，用于数据库的查询操作
BaseModel.query = db_session.query_property()


def init_db():
    """初始化数据库，用于创建数据库表"""
    # 在这里导入定义模型所需要的所有模块，这样它们就会正确的注册在元数据上。
    # 否则你就必须在调用 init_db() 之前导入它们。
    import app.models
    BaseModel.metadata.create_all(bind=engine)
    return BaseModel


def shutdown_session(exception=None):
    """关闭数据session"""
    db_session.remove()


def init_app(application):
    """初始化app，同时注册数据库模型"""
    # Flask 会自动在请求结束时或者应用关闭时删除数据库会话
    application.teardown_appcontext(shutdown_session)
```

```
    # 注册数据库模型
    # 在这里导入定义模型所需要的所有模块，这样它们就会正确的注册在数据库迁移上。
    import app.models
    Migrate(application, BaseModel)
```

5. 初始化数据库

修改`app/__init__.py`初始化文件，初始化数据库

```
......
def create_app():
    """应用工厂函数"""
    app = Flask(__name__)
    # 加载config配置文件
    app.config.from_pyfile('config.py', silent=True)

    # 初始化数据库
    from app.database import init_app
    init_app(app)
......
```

6. 创建数据库模型

新建`app/models.py`文件，创建用户信息的数据库模型。

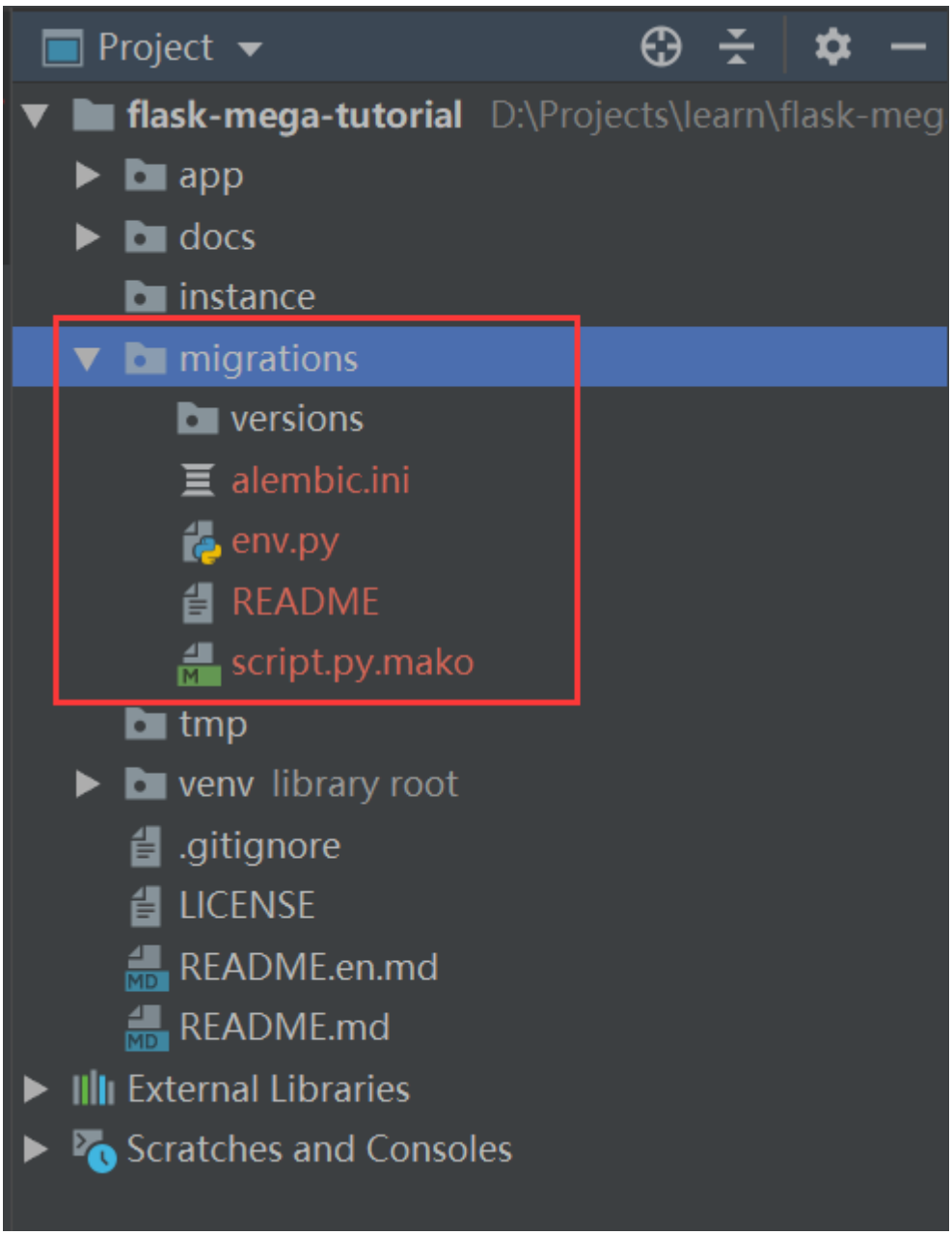
```
from sqlalchemy import Column, String, Integer

from app.database import BaseModel


class User(BaseModel):
    __tablename__ = 'users'
    id = Column(Integer, primary_key=True)
    nickname = Column(String(64), index=True, unique=True)
    email = Column(String(120), index=True, unique=True)

    def __init__(self, nickname=None, email=None):
        self.nickname = nickname
        self.email = email

    def __repr__(self):
        """打印类对象时的展示方式"""
        return '<User %r>' % self.nickname
```

7. 数据库模型初始化

7.1. 迁移目录初始化

通过`flask db init`命令，初始化数据库模型迁移目录`migrations`。

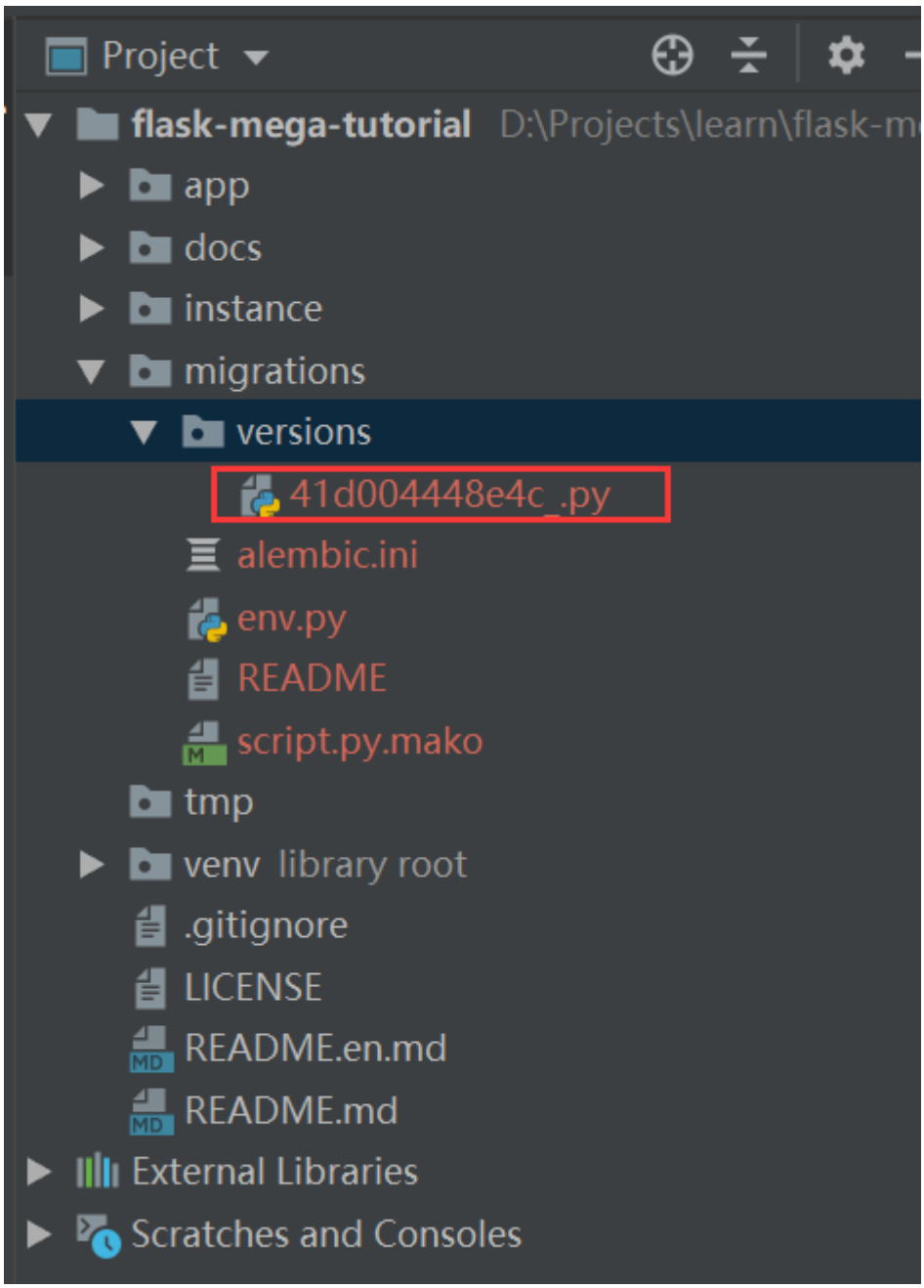
```
(venv) D:\Projects\learn\flask-mega-tutorial>flask db init
Creating directory D:\Projects\learn\flask-mega-tutorial\migrations ... done
Creating directory D:\Projects\learn\flask-mega-tutorial\migrations\versions ... done
Generating D:\Projects\learn\flask-mega-tutorial\migrations\alembic.ini ... done
Generating D:\Projects\learn\flask-mega-tutorial\migrations\env.py ... done
Generating D:\Projects\learn\flask-mega-tutorial\migrations\README ... done
Generating D:\Projects\learn\flask-mega-tutorial\migrations\script.py.mako ... done
Please edit configuration/connection/logging settings in 'D:\\Projects\\learn\\flask-mega-tutorial\\migrations\\alembic.ini' before proceeding.
```



7.2. 迁移文件创建

通过`flask db migrate`命令，将数据库迁移文件生成到迁移目录`versions`中，同时也初始化创建了`Flask-Migrate`扩展库依赖的数据库表。

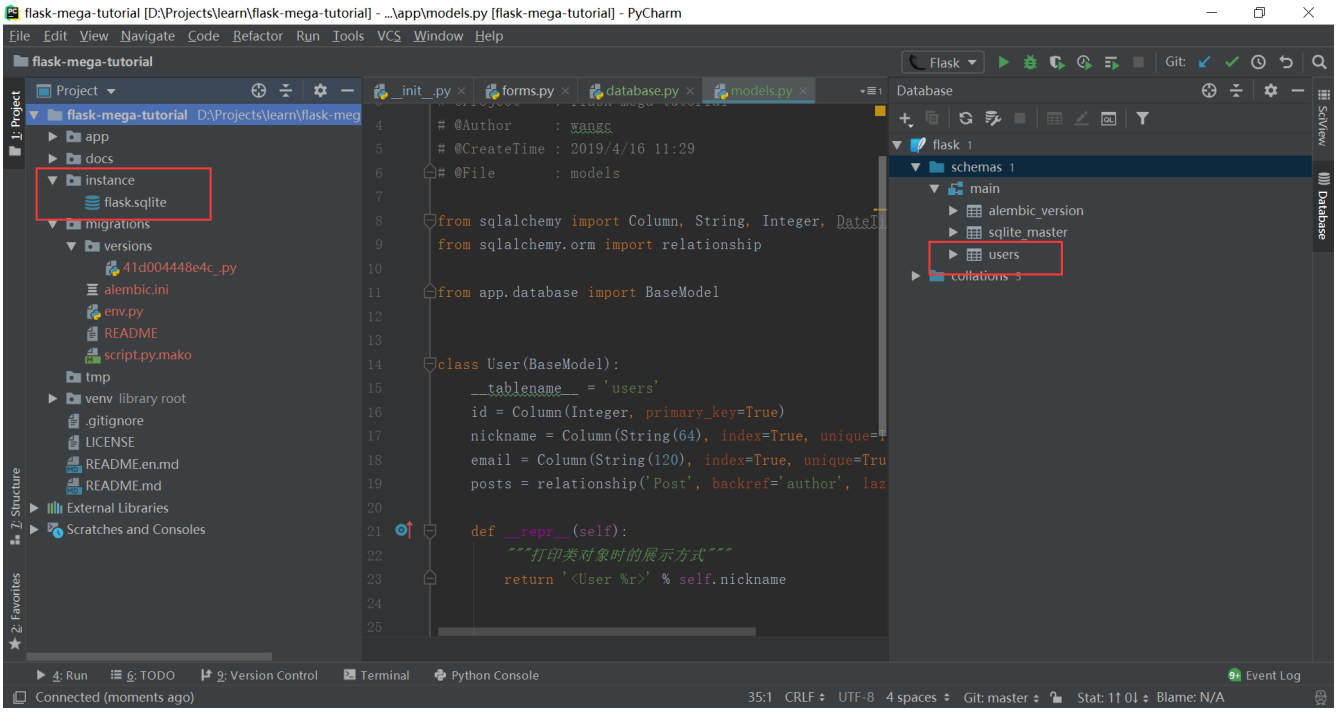
```
(venv) D:\Projects\learn\flask-mega-tutorial>flask db migrate
INFO  [alembic.runtime.migration] Context impl SQLiteImpl.
INFO  [alembic.runtime.migration] Will assume non-transactional DDL.
INFO  [alembic.autogenerate.compare] Detected added table 'users'
INFO  [alembic.autogenerate.compare] Detected added index 'ix_users_email' on '['email']'
INFO  [alembic.autogenerate.compare] Detected added index 'ix_users_nickname' on
'['nickname']'
Generating D:\Projects\learn\flask-mega-tutorial\migrations\versions\41d004448e4c_.py ...
done
```



7.3. 数据库同步

通过`flask db upgrade`命令，将数据库迁移文件的变化同步到数据库中。

```
(venv) D:\Projects\learn\flask-mega-tutorial>flask db upgrade
INFO  [alembic.runtime.migration] Context impl SQLiteImpl.
INFO  [alembic.runtime.migration] Will assume non-transactional DDL.
INFO  [alembic.runtime.migration] Running upgrade  -> 41d004448e4c, empty message
```



8. 新增数据库模型、数据库同步

8.1. 新增数据库模型定义

编辑`app/models.py`文件，新增posts表的数据库模型定义

```
from sqlalchemy import Column, String, Integer, DateTime, ForeignKey
from sqlalchemy.orm import relationship

from app.database import BaseModel


class User(BaseModel):
    __tablename__ = 'users'
    id = Column(Integer, primary_key=True)
    nickname = Column(String(64), index=True, unique=True)
    email = Column(String(120), index=True, unique=True)
    posts = relationship('Post', backref='author', lazy='dynamic')

    def __init__(self, nickname=None, email=None):
        self.nickname = nickname
        self.email = email

    def __repr__(self):
```

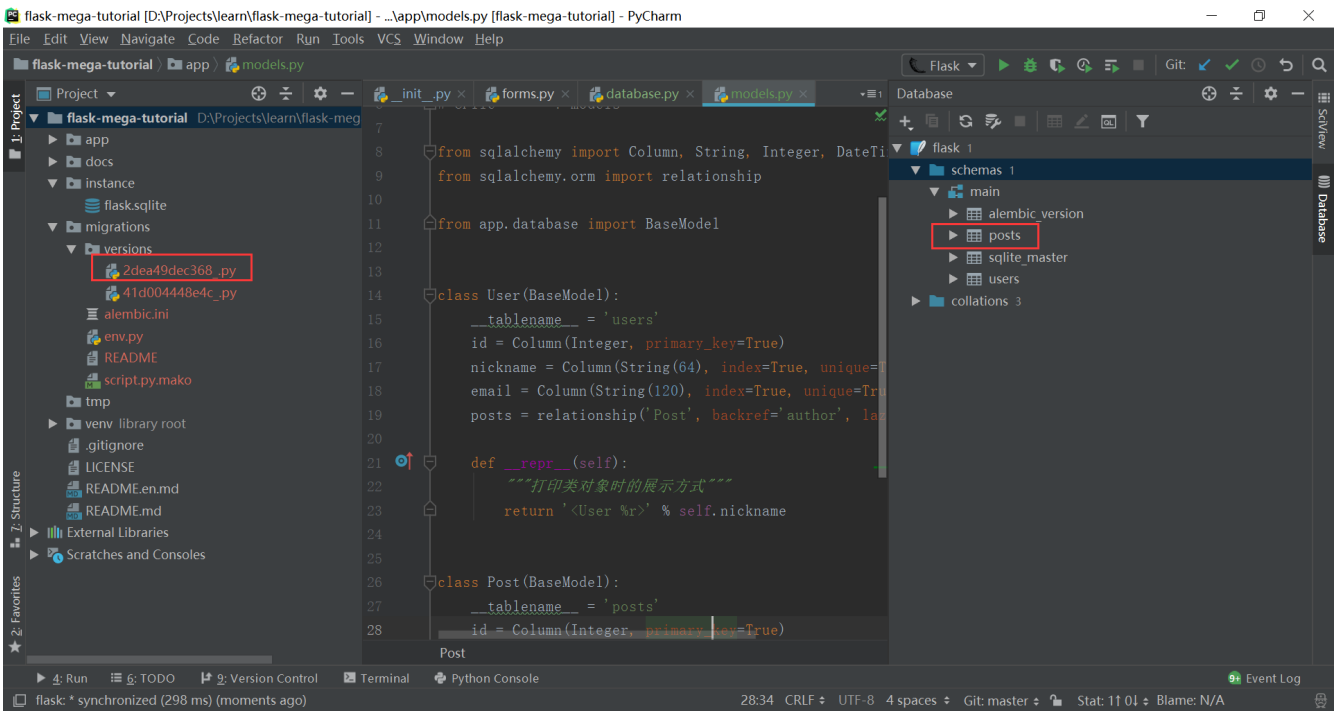
```
        """打印类对象时的展示方式"""
        return '<User %r>' % self.nickname


class Post(BaseModel):
    __tablename__ = 'posts'
    id = Column(Integer, primary_key=True)
    body = Column(String(140))
    timestamp = Column(DateTime)
    user_id = Column(Integer, ForeignKey('users.id'))

    def __repr__(self):
        return '<Post %r>'.format(self.body)
```

8.2. 迁移文件创建

```
(venv) D:\Projects\learn\flask-mega-tutorial>flask db migrate
INFO  [alembic.runtime.migration] Context impl SQLiteImpl.
INFO  [alembic.runtime.migration] Will assume non-transactional DDL.
INFO  [alembic.autogenerate.compare] Detected added table 'posts'
Generating D:\Projects\learn\flask-mega-tutorial\migrations\versions\2dea49dec368_.py ...
done
```

8.3. 数据库同步

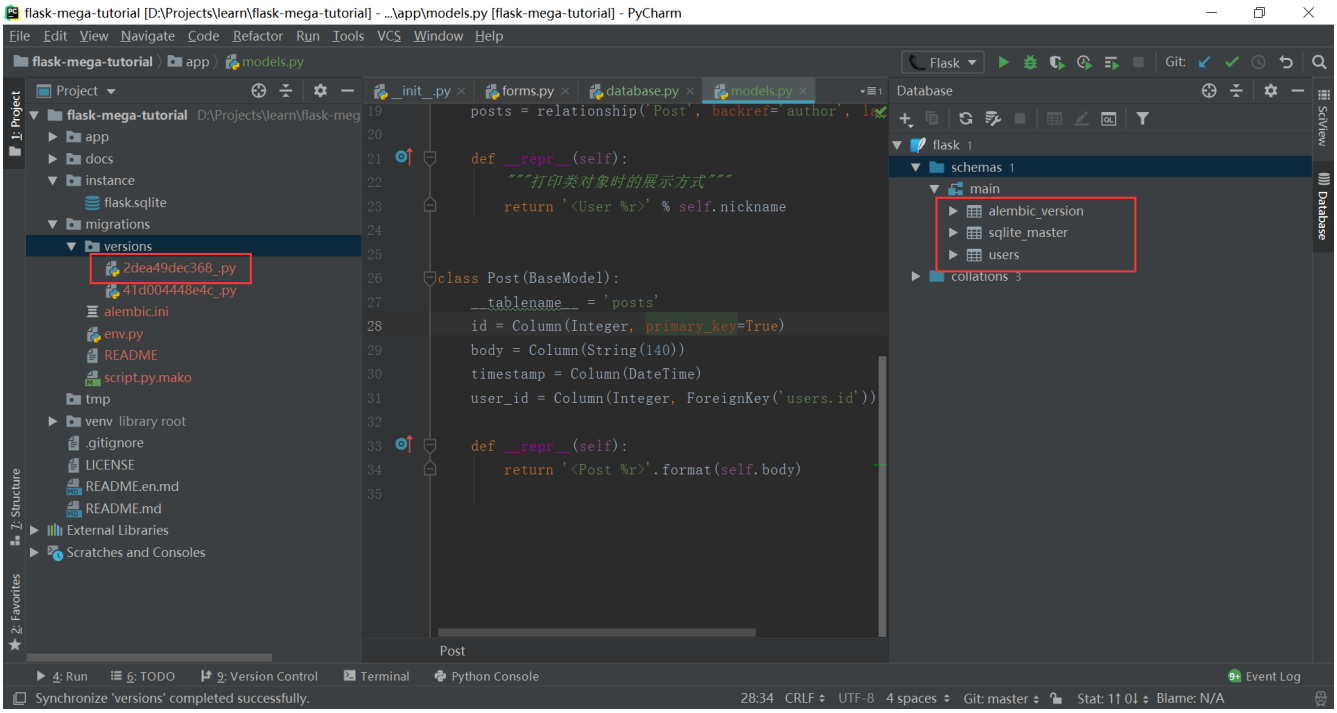
```
(venv) D:\Projects\learn\flask-mega-tutorial>flask db upgrade
INFO  [alembic.runtime.migration] Context impl SQLiteImpl.
INFO  [alembic.runtime.migration] Will assume non-transactional DDL.
INFO  [alembic.runtime.migration] Running upgrade 41d004448e4c -> 2dea49dec368, empty
message
```



9. 数据库操作回退

通过`flask db downgrade`命令，对已同步的数据库进行回退处理，此操作仅限于**数据库同步**的回退，但是迁移文件还在。

```
(venv) D:\Projects\learn\flask-mega-tutorial>flask db downgrade
INFO  [alembic.runtime.migration] Context impl SQLiteImpl.
INFO  [alembic.runtime.migration] Will assume non-transactional DDL.
INFO  [alembic.runtime.migration] Running downgrade 2dea49dec368 -> 41d004448e4c, empty
message
```



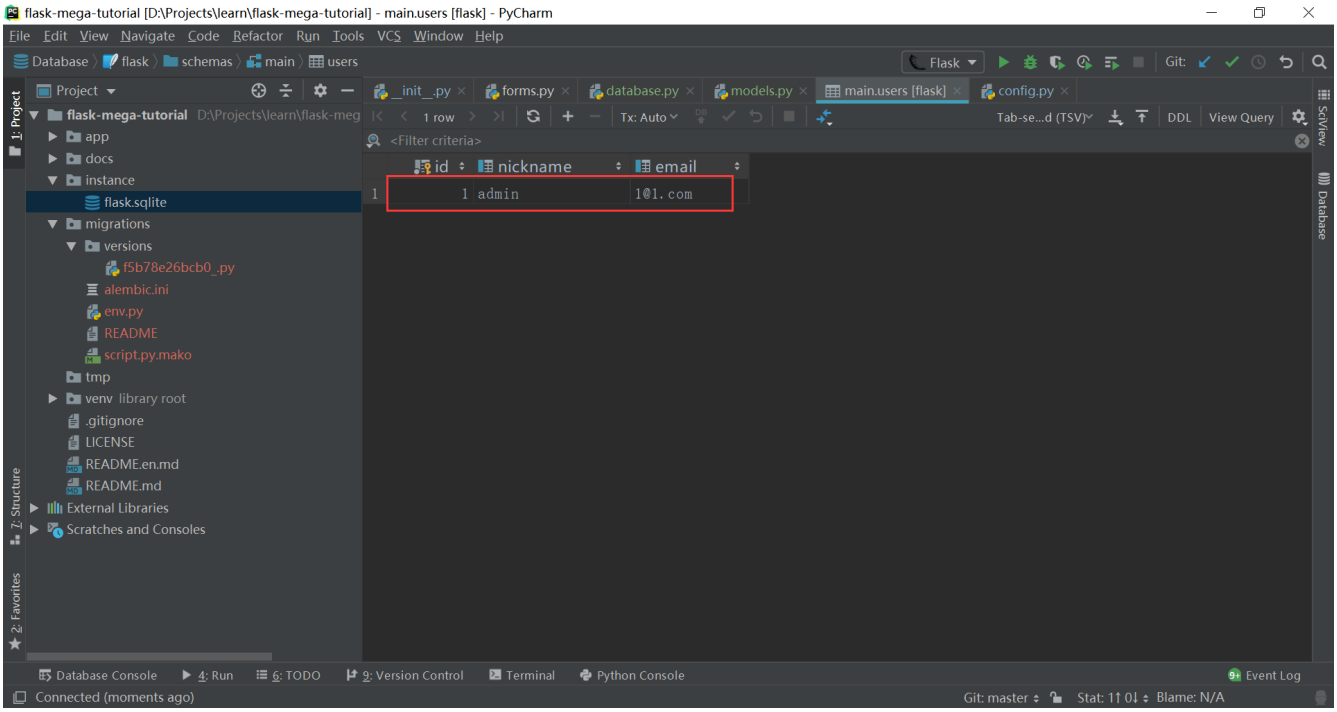
10. 数据库操作测试

10.1. 数据查询

```
>>> from app.models import User
>>> User.query.all()
[]
```

10.2. 数据插入

```
>>> from app.models import User
>>> from app.database import db_session
>>> u = User('admin', '1@1.com')
>>> db_session.add(u)
>>> db_session.commit()
>>> db_session.close()
>>> User.query.all()
[<User 'admin'>]
```



10.3. 数据删除

```
>>> User.query.all()
[<User 'admin'>]
>>> db_session.delete(User.query.filter(User.nickname=='admin').one())
>>> db_session.commit()
```