

SESSION В ORM

ВВЕДЕНИЕ

Задачи урока

01 Познакомиться с сессиями в SQLAlchemy

02 Разобраться в работе execute

Session в SQLAlchemy представляет собой контекст работы с базой данных, который позволяет выполнить операции **CRUD (Create, Read, Update, Delete)** с объектами, связанными с базой данных. **Session** обеспечивает отслеживание изменений, контроль целостности данных и управление транзакциями

Задачи Session



Открытие соединения с базой данных при необходимости



Создание и выполнение SQL-запросов к базе данных



Управление жизненным циклом объектов ORM, отслеживание изменений и обновление состояния объектов



Управление транзакциями, обеспечивая атомарность и целостность данных

Пример использования Session в SQLAlchemy

```
...  
  
from sqlalchemy import create_engine  
  
from sqlalchemy.orm import sessionmaker  
  
connection_string = f"postgresql://postgres:password@localhost:5432/  
synergy"  
  
engine = create_engine(connection_string)  
  
Session = sessionmaker(bind=engine)  
Session = Session()
```

В данном примере мы создаем сеанс, связанный с движком базы данных **engine**, и используем его для добавления объекта **MyObject** в базу данных и фиксации изменений

execute() может быть вызван на объекте **Connection** или на объекте **Engine**, и он принимает в качестве параметра строку, содержащую SQL-запрос

```
...  
rs = session.buid().execute(text('select * from "Users"'))
```

или

```
...  
data = (  
    {  
        "Name": "Петр",  
        "Age": 20  
    },  
    {  
        "Name": "Василий",  
        "Age": 26,  
    },  
)  
statement = text("""INSERT INTO "Users" ("Name", "Age") VALUES(:Name,  
:Age)""")  
print(statement)  
  
session = session.buid()  
for line in data:  
    session.execute(statement, line)  
    session.commit()
```

Метод **execute()** также позволяет передавать параметры в SQL-запрос с использованием привязки параметров или форматирования строк, чтобы обезопасить запросы от SQL-инъекций. Это особенно полезно при работе с пользовательским вводом или динамическими значениями

ИТОГИ

- ✓ Для получения данных можно использовать метод **execute** в который передаем SQL запрос
- ✓ Если нам нужно внести изменения в таблицу (вставить/обновить или удалить данные), это тоже можно реализовать через **execute**, после чего необходимо применить метод **commit**, который фиксирует изменения в базе