

SELECT/UPDATE/DELETE

ВВЕДЕНИЕ

Задачи урока

01 Научиться выбирать/обновлять/удалять данные с использованием SQLAlchemy

query в SQLAlchemy представляет собой выражение, которое используется для запроса данных из базы данных при использовании объектно-реляционной модели (ORM). Оно представляет собой интерфейс для создания структурированных запросов на основе моделей и их связей с базой данных

query позволяет выражать различные операции на базе данных, включая выборки (SELECT), фильтрацию данных, сортировку, агрегацию и другие. Он поддерживает богатый набор методов для создания сложных запросов, включая фильтры, сортировку, ограничение количества результатов и другие операции

Пример использования **query** в SQLAlchemy

```
...  
session = session.buid()  
  
users = Users(  
    Name='Андрей',  
    Age=27  
)  
session.add(users)  
session.commit()  
  
rs = session.query(Users)  
for record in rs:  
    print(record.Name)
```

В SQLAlchemy метод **add()** используется для добавления новых объектов в базу данных. Он предоставляет интерфейс для создания новых записей и сохранения их в базе данных при использовании объектно-реляционной модели (ORM)

add() позволяет добавить новый объект в сессию SQLAlchemy. Объект должен быть связан с соответствующей моделью. После вызова метода **add()**, объект будет отслеживаться сессией SQLAlchemy и будет добавлен в базу данных при следующем вызове **commit()**

Фильтрация данных в SQLAlchemy выполняется с помощью метода **filter()**. Он позволяет создавать условия для выборки данных из базы данных при использовании объектно-реляционной модели (ORM)

filter() принимает один или несколько аргументов, которые определяют условия для выборки данных. Аргументами могут быть выражения, сравнения, атрибуты модели и другие операнды, которые можно использовать для построения условий фильтрации данных

Пример использования метода filter() в SQLAlchemy:

```
...  
rs = session.query(Users).filter(Users.Age > 20).all()  
for record in rs:  
    print(record.Name)
```

В SQLAlchemy метод **get()** используется для получения объекта модели из базы данных по его первичному ключу. Он предоставляет удобный способ получить объект из базы данных по его уникальному идентификатору

get() принимает два аргумента: класс модели и значение первичного ключа. Он выполняет запрос к базе данных и возвращает объект модели, соответствующий указанному ключу. Если объект не найден, метод возвращает значение **None**

```
...  
user = session.query(Users).get(3)  
print(user.Name)
```

В SQLAlchemy метод **select_from()** используется для указания таблицы или выражения, из которых нужно выбрать данные. Этот метод является частью объектно-реляционного интерфейса SQLAlchemy (ORM) и позволяет конструировать более сложные запросы для получения данных из базы данных.

Метод **select_from()** принимает в качестве аргумента класс модели или **алиас таблицы**, из которой нужно выбрать данные. Он добавляет указанную таблицу или выражение внутри фразы **SELECT FROM**, и этот фрагмент будет использован для генерации SQL-запроса

```
result = session.query(
    Users.Name,
    FilesInUser.Name.label("FileName")
).select_from(Users)
```

ИТОГИ

- ✓ Для выбора данных используют **query**
- ✓ Для фильтрации используют **filter**
- ✓ Для обновления необходимо получить данные из БД, после чего в ORM поменять значение полей и далее закоммитить изменения используя метод **commit()**