

СОЗДАНИЕ ТАБЛИЦ

ВВЕДЕНИЕ

Задачи урока

01 Научиться создавать таблицы

02 Изучить методы `MetaData` и `create_all`

`db.MetaData()`

это класс в библиотеке SQLAlchemy, который представляет метаданные базы данных. Он используется для описания и управления структурой базы данных, включая таблицы, столбцы, индексы, внешние ключи и другие объекты

В SQLAlchemy, метаданные - совокупность объектов **Table**, которые описывают схему базы данных. Класс **MetaData** содержит методы и атрибуты для создания и манипулирования этими объектами **Table**

Пример использования **MetaData** для создания таблицы в базе данных с помощью SQLAlchemy

```
from sqlalchemy import create_engine
import sqlalchemy as db

from sqlalchemy.orm import sessionmaker

connection_string = f"postgresql://postgres:password@localhost:5432/synergy"

engine = create_engine(connection_string)
metadata_obj = db.MetaData()

profile = db.Table(
    'profile',
    metadata_obj,
    db.Column('email', db.String, primary_key=True),
    db.Column('name', db.String),
    db.Column('contact', db.Integer),
)
```

- В приведенном выше примере мы создаем **объект метаданных с помощью `MetaData()`**
- Затем мы определяем таблицу **"users"** с помощью объекта **`Table`** и определяем столбцы этой таблицы с помощью объектов **`Column`**. Мы указываем типы данных и другие параметры для каждого столбца (например, первичный ключ или ограничения на значения)
- Затем мы вызываем метод **`create_all(engine)`** объекта метаданных, чтобы создать таблицу в базе данных, используя движок **`engine`**. Этот метод создает таблицу только в том случае, если она еще не существует

MetaData

Объект **MetaData** является важной частью SQLAlchemy, поскольку обеспечивает абстракцию над структурой базы данных и позволяет легко управлять метаданными и схемой базы данных в приложении

db.Table

это класс в библиотеке SQLAlchemy, который представляет таблицу в базе данных. Он используется для определения структуры таблицы, включая ее имя, столбцы, индексы и ограничения на поля

Класс Table

это один из базовых строительных блоков SQLAlchemy для работы с базой данных. Он предоставляет удобный способ определения таблицы и ее структуры без явного создания объектов **MetaData** и **Column**

После определения таблицы, мы вызываем метод **`create_all(engine)`** объекта метаданных, чтобы создать таблицу в базе данных, используя движок **`engine`**. Этот метод создает таблицу только в том случае, если она еще не существует

Кроме того, **Table** предоставляет ряд методов и атрибутов для работы с таблицей, таких как **`insert()`**, **`select()`**, **`update()`**, **`delete()`** и др. Это позволяет нам выполнять различные операции с данными таблицы, такие как вставка, выборка, обновление или удаление данных

```
from sqlalchemy import create_engine
import sqlalchemy as db

from sqlalchemy.orm import sessionmaker

connection_string = f"postgresql://postgres:password@localhost:5432/
synergy"

engine = create_engine(connection_string)
metadata_obj = db.MetaData()

profile = db.Table(
    'profile',
    metadata_obj,
    db.Column('email', db.String, primary_key=True),
    db.Column('name', db.String),
    db.Column('contact', db.Integer),
)

metadata_obj.create_all(engine)
```

ИТОГИ

- ✓ Для создания таблиц необходимо использовать **db.Table** и описать колонки этой таблицы
- ✓ Для внесения изменений в базу используют метод **create_all**