

# Программная инженерия. Разработка ПО (Python для продвинутых специалистов. Машинное обучение)

Модуль: Введение в Python для машинного обучения

Лекция 7: SQLAlchemy для работы с базой данных

Дата: 14.04.2025

Q&A



# Содержание лекции

- Что такое SQLAlchemy и ORM?
- SQLAlchemy для работы с SQLite и PostgreSQL
- Создание простой ORM модели
- Подключение к Pandas
- Практическая часть

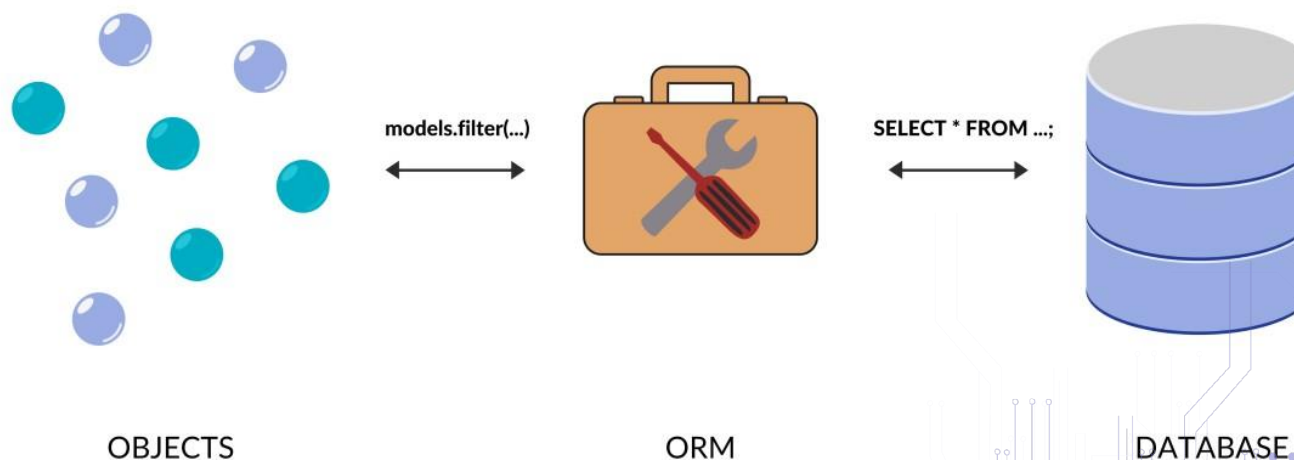


# SQLAlchemy и ORM

SQLAlchemy — программная библиотека на языке Python для работы с реляционными СУБД с применением технологии ORM.

Может использоваться в двух режимах:

- В качестве инструмента для работы с существующей базой данных
- В качестве ORM



# SQLAlchemy. Старт

Для обращения к базе данных необходимо создать engine – «движок» подключения. Принцип его работы похож на объект connection из стандартных библиотек по работе с базами данных.

```
from sqlalchemy import create_engine

# Создание «движка» подключения
engine = create_engine( ' <DB conn string> ' )

# Извлекаем данные
rows = engine.execute( "select * from <some_table>" ).scalar()

# Выводим их на экран
for row in rows:
    print(row)
```

# SQLAlchemy. «Синтаксический сахар»

SQLAlchemy можно использовать в качестве средства для обращения к таблицам базы данных в функциональном стиле, избегая SQL.

```
# Создание метаданных
metadata = MetaData()

# Определение таблицы
users_table = Table('users_new', metadata,
    Column('id', Integer, Sequence('user_id_seq'), primary_key=True),
    Column('name', String(50)),
    Column('fullname', String(50)),
    Column('age', Integer))

# Создание всех таблиц в базе данных
metadata.create_all(engine)

# Подключение к базе данных
conn = engine.connect()

# Добавление данных в таблицу
stmt = insert(users_table).values(name='Andrew', fullname='Andrew B', age=15)
conn.execute(stmt)
conn.commit(stmt)

# Получение всех пользователей из базы данных
stmt = select(users_table)
result = conn.execute(stmt)
```

# SQLAlchemy. ORM

SQLAlchemy также можно использовать для определения схемы базы данных и управления ей.

```
from sqlalchemy import ForeignKey, String
from sqlalchemy.orm import DeclarativeBase, Session, relationship

# Создаем базовый класс
class Base(DeclarativeBase):
    pass

# Создаем объект-таблицу Клиент
class Client(Base):
    __tablename__ = "clients"

    id = Column(Integer, primary_key=True)
    name = Column(String(30))
    email = Column(String(30))
    orders = relationship("Order", back_populates="client", cascade="all, delete-orphan")

# Создаем таблицу на основе класса
Base.metadata.create_all(engine)

# Создаем объект подключения к БД
session = Session(engine)
```

```
# Читаем данные из базы
```

```
stmt = select(Client)
```



Передовые  
инженерные  
школы



МИНОБРНАУКИ  
РОССИИ



УНИВЕРСИТЕТ  
ИННОПОЛИС



онлайн  
университет

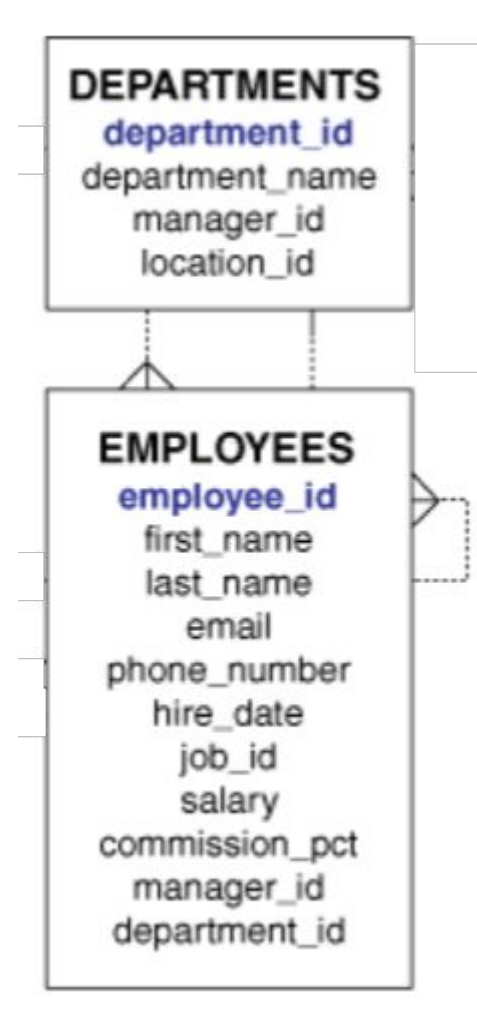
# Практическая часть



# Задание

- Создайте с помощью ORM часть схемы данных HR.
- Заполните таблицы данными.
- Проверьте, что записи появились в базе.

NB: Сами таблицы и данные нам понадобятся на следующем занятии.





Передовые  
инженерные  
школы



МИНОБРНАУКИ  
РОССИИ



УНИВЕРСИТЕТ  
ИННОПОЛИС



онлайн  
университет

# Спасибо за внимание