



**Code
Academy**



1 LYGIS

14 paskaita. Duomenų bazės 2



Šiandien išmoksite

01

Susipažinsime su SQLAlchemy moduliu

02

Naudotis ORM įrankiais

03

Duomenų bazėje atlikti CRUD veiksmus



Modulis SQLAlchemy

Modulis SQLAlchemy susideda iš dviejų dalių:

- **SQLAlchemy Core** – įrankis, skirtas manipuluoti, modifikuoti ir paleisti įvairias SQL užklausas. Labai panašiai, kaip ir SQLite3, tik dirba su visomis duomenų bazėmis:
- **SQLAlchemy ORM** (ORM – Object Related Mapping) – įrankis, skirtas automatiškai susieti Python objektus su lentelėmis duomenų bazėje ir vykdyti įvairius veiksmus (CRUD), nenaudojant SQL užklausų

SQLAlchemy nėra vienintelis ORM funkcionalumą siūlantis modulis. Net ir Python pagrindu yra sukurta daugiau įrankių. Panašius modulių turi JAVA (Hibernate), kitos programavimo kalbos.



Kaip susikurti objekto klasę, iš kurios bus sukurta lentelė

```
import datetime
from sqlalchemy import Column, Integer, String, Float, DateTime, create_engine
from sqlalchemy.ext.declarative import declarative_base

engine = create_engine('sqlite:///projektai.db')
Base = declarative_base()

class Projektas(Base):
    __tablename__ = 'Projektas'
    id = Column(Integer, primary_key=True)
    name = Column("Pavadinimas", String)
    price = Column("Kaina", Float)
    created_date = Column("Sukūrimo data", DateTime, default=datetime.datetime.utcnow)

    def __init__(self, name, price):
        self.name = name
        self.price = price

    def __repr__(self):
        return f"{self.id} {self.name} - {self.price}: {self.created_date}"

Base.metadata.create_all(engine)
```

Tam, kad sukurtume reikiamos struktūros duomenų bazę ir ją naudotumės, užtenka sukurti SQLAlchemy klasę ir ją paleisti.



```
from sqlalchemy import create_engine
from sqlalchemy.orm import sessionmaker
from projektas import Projektas

engine = create_engine('sqlite:///projektai.db')
Session = sessionmaker(bind=engine)
session = Session()
```

**Kaip įrašyti, nuskaityti, atnaujinti,
ištrinti duomenys SQLAlchemy
lentelėje
(CRUD – create, read, update,
delete)**

Pavyzdys kaip sukurti ryšį su sukurta DB kitame faile



```
projektas1 = session.query(Projektas).get(1)

print(projektas1.name)
# Naujas pr.
```

```
projektas2 = session.query(Projektas).filter_by(name="2 projektas").one()
```

```
projektai = session.query(Projektas).all()

for projektas in projektai:
    print(projektas.name, projektas.price)
# Naujas pr. 20000.0
# 2 projektas 55000.0
```

Kaip gauti duomenis iš lentelės (cRud)



Kaip ieškoti duomenų pagal sąlygą ar šabloną

```
search = session.query(Projektas).filter(Projektas.name.ilike("2%"))
search2 = session.query(Projektas).filter(Projektas.price > 1000)
search3 = session.query(Projektas).filter(
    Projektas.price > 1000,
    Projektas.name.ilike("2%"))

print([i for i in search])
print([i for i in search2])
print([i for i in search3])

# [2 2 projektas - 55000.0: 2021-02-03 14:29:33.477232]
# [1 Naujas pr. - 20000.0: 2021-02-03 14:29:33.437231, 2 2 projektas - 55000.0: 2021-02-03 14:29:33.477232]
# [2 2 projektas - 55000.0: 2021-02-03 14:29:33.477232]
```



```
projektas1 = session.query(Projektas).get(1)
projektas1.price = 22000
session.commit()
```

```
projektas2 = session.query(Projektas).filter_by(name="2 projektas").one()
projektas2.name = "2 projektas tikrai"
session.commit()
```

Kaip pakeisti duomenis lentelėje (crUd)



```
projektas1 = session.query(Projektas).filter_by(name="Naujas pr.").one()  
  
session.delete(projektas1)  
session.commit()
```

Kaip ištrinti duomenis lentelėje (cruD)



```
from projektas import engine, Projektas
from sqlalchemy.orm import sessionmaker

Session = sessionmaker(bind=engine)
session = Session()

while True:
    pasirinkimas = int(input("""Pasirinkite veiksmą:
1 - atvaizduoti projektus
2 - sukurti projektą
3 - pakeisti projektą
4 - ištrinti projektą"""))

    if pasirinkimas == 1:
        projektai = session.query(Projektas).all()
        print("-----")
        for projektas in projektai:
            print(projektas)
        print("-----")

    if pasirinkimas == 2:
        name = input("Iveskite projekto pavadinimą")
        price = float(input("Iveskite projekto kainą"))
        projektas = Projektas(name, price)
        session.add(projektas)
        session.commit()
```

Programos su duomenų baze (konsolėje) pavyzdys (1 dalis)

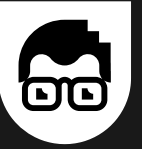


```
if pasirinkimas == 2:
    name = input("Įveskite projekto pavadinimą")
    price = float(input("Įveskite projekto kainą"))
    projektas = Projektas(name, price)
    session.add(projektas)
    session.commit()

if pasirinkimas == 3:
    projektai = session.query(Projektas).all()
    print("-----")
    for projektas in projektai:
        print(projektas)
    print("-----")
    keiciamo_id = int(input("Pasirinkite norimo pakeisti projekto ID"))
    keiciamas_projektas = session.query(Projektas).get(keiciamo_id)
    pakeitimas = int(input("Ką norite pakeisti: 1 - pavadinimą, 2 - kainą"))
    if pakeitimas == 1:
        keiciamas_projektas.name = input("Įveskite projekto pavadinimą")
    if pakeitimas == 2:
        keiciamas_projektas.price = float(input("Įveskite projekto kainą"))
    session.commit()

if pasirinkimas == 4:
    projektai = session.query(Projektas).all()
    print("-----")
    for projektas in projektai:
        print(projektas)
    print("-----")
    keiciamo_id = int(input("Pasirinkite norimo ištrinti projekto ID"))
    trinamas_projektas = session.query(Projektas).get(keiciamo_id)
    session.delete(trinamas_projektas)
    session.commit()
```

Programos su duomenų baze (konsolėje) pavyzdys (2 dalis)



Užduotis nr. 1

Sukurti programą, kuri:

- Leistų įvesti darbuotojus: vardą, pavardę, gimimo datą, pareigas, atlyginimą, nuo kada dirba (data būtų nustatoma automatiškai, pagal dabartinę datą).
- Duomenys būtų saugomi duomenų bazėje, panaudojant SQLAlchemy ORM (be SQL užklausų)
- Vartotojas galėtų įrašyti, peržiūrėti, ištrinti ir atnaujinti darbuotojus.



Užduotis nr. 2

Perdaryti programą 1 užduotyje, kad ji:

- Turėtų grafinę sąsają (su ikona ir pavadinimu). Sukurti per Tkinter
- Leistų įvesti asmenį į duomenų bazę (jų vardą, pavardę, amžių, ...)
- Parodytų visų į duomenų bazę įvestų asmenų sąrašą
- Leistų ištrinti pasirinktą asmenį iš duomenų bazės
- Leistų pagedaguoti įvesto asmens duomenis ir įrašyti atnaujinimus į duomenų bazę



Namų darbas

Užbaigti klasėje nepadarytas užduotis



Išspręsti paskaitos uždaviniai (įkelti pirmadienį)

<https://github.com/aurimas13/Python-Beginner-Course/tree/main/Programs>

SQLAlchemy

SQLAlchemy aprašymas

<https://docs.sqlalchemy.org/en/13/core/engines.html>

**Naudinga
informacija**