

# Advanced databases

## Dokumentacja projektu

Bartosz Bywalec

Mateusz Motyka

### Opis programu:

Rozpoczynamy od dołączenia do projektu niezbędnych bibliotek, takich jak sqlalchemy oraz pandas. Następnie ustawiamy połączenie z bazą danych znajdującą się na lokalnym serwerze oraz uruchamiamy automatyczny proces mapowania istniejących tabel na klasy:

```
from sqlalchemy import create_engine, Column, Integer, String, Float, ForeignKey, Sequence, CheckConstraint, UniqueConstraint, Date, Table, MetaData, select
from sqlalchemy.ext.declarative import declarative_base
import pandas as pd

db_string = "postgres://postgres:postgres@localhost:5432/slogans"

engine = create_engine(db_string)

Base = declarative_base()
```

### Wczytujemy dane wejściowe:

```
data = pd.read_csv('input_data.csv', delimiter=';')
data = data.drop('Unnamed: 4', 1)
data
```

Wyszukujemy wszystkie kategorie, podkategorie, nazwy firm oraz slogany:

```
all_category = data['CATEGORY'].unique()
print("category array: {0}".format(all_category))
```

```
all_subcategory = data['SUBCATEGORY'].unique()
print("subcategory array: {0}".format(all_subcategory))
```

```
all_brand = data['BRAND'].unique()
print("brand array: {0}".format(all_brand))
```

```
all_slogan = data['SLOGAN'].unique()
print("slogan array: {0}".format(all_slogan))
```

Stworzenie listy podkategorii z odniesieniem od id odpowiedniej kategorii:

```
subcategory_list = data[['CATEGORY', 'SUBCATEGORY']].drop_duplicates().reset_index().drop(columns = ['index']);
subcategory_list.index.name = 'id'

subcategory_list = subcategory_list.rename(columns = {'CATEGORY': 'category_id'})
subcategory_list = subcategory_list.rename(columns = {'SUBCATEGORY': 'subcategory'})

subcategory_list['category_id'] = subcategory_list['category_id'].map(lambda x: category_list[category_list['category'] == x].index.values.astype(int)[0])

pd.set_option('display.max_rows', 100)
subcategory_list
```

Tworzymy 4 tabele:

- category – tabela zawierająca wszystkie kategorie
- subcategory – tabela, w której znajdują się wszystkie podkategorie z odnośnikiem do odpowiedniej kategorii
- brand – tabela z wszystkimi firmami, brak odnośników do podkategorii spowodowany jest tym, że jedna firma może posiadać slogany odnoszące się do kilku podkategorii
- slogan – główna tabela, w której znajdują się wszelkie slogany oraz odnośniki do firm i podkategorii

Ostatnia linijka odpowiada za stworzenie pełnego schematu tabel w bazie danych.

```
class category(Base):
    __tablename__ = 'category'
    __table_args__ = (
        UniqueConstraint('category'),
    )
    id = Column(Integer, Sequence('seq_category_id'), primary_key = True)
    category = Column(String(50))

class subcategory(Base):
    __tablename__ = 'subcategory'
    __table_args__ = (
        UniqueConstraint('subcategory'),
    )
    id = Column(Integer, Sequence('seq_subcategory_id'), primary_key=True)
    category_id=Column(Integer, ForeignKey('category.id'))
    subcategory = Column(String(100))

class brand(Base):
    __tablename__ = 'brand'
    __table_args__ = (
        UniqueConstraint('brand'),
    )
    id = Column(Integer, Sequence('seq_brand_id'), primary_key = True)
    brand = Column(String(200))

class slogan(Base):
    __tablename__ = 'slogan'
    id = Column(Integer, Sequence('seq_slogan_id'), primary_key=True)
    slogan =Column(String(200))
    subcategory_id=Column(Integer, ForeignKey('subcategory.id'))
    brand_id=Column(Integer, ForeignKey('brand.id'))

Base.metadata.create_all(engine)
```

Dodajemy stworzone wcześniej listy do odpowiednich tabel w bazie danych:

```
category_list.to_sql('category',engine, if_exists='append')
subcategory_list.to_sql('subcategory',engine, if_exists='append')
brand_list.to_sql('brand',engine, if_exists='append')
slogan_list.to_sql('slogan',engine, if_exists='append')
```

W celu sprawdzenia poprawności stworzenia bazy danych wykonujemy dwa proste selecty:

- zapytanie o wszystkie występujące kategorie oraz sprawdzenie ich ilości

```
mapper_stmt = select([dic_table['category'].columns.id, dic_table['category'].columns.category])
print('Mapper select: ')
print(mapper_stmt)
mapper_results = engine.execute(mapper_stmt).fetchall()
print(mapper_results)
print('\nIlosc kategorii:')
len(mapper_results)
```

```
Mapper select:
SELECT category.id, category.category
FROM category
[(0, 'Apparel slogans'), (1, 'Automotive slogans'), (2, 'Beauty slogans'), (3, 'Beverage slogans'), (4, 'Business slogans'),
(5, 'Construction slogans'), (6, 'Dining slogans'), (7, 'Educational slogans'), (8, 'Financial service slogans'), (9, 'Casino
slogans'), (10, 'Computers slogans'), (11, 'Condoms slogans'), (12, 'Magazines slogans'), (13, 'Motorcycle slogans'), (14, 'N
ewspapers slogans'), (15, 'Pet food slogans'), (16, 'Radio stations slogans'), (17, 'Real estate slogans'), (18, 'Tobacco slo
gans'), (19, 'Vitamins slogans'), (20, 'Watch slogans')]
```

```
Ilosc kategorii:
21
```

- zapytanie o wszystkie slogany z kategorii automotive:

```
mapper_stmt = select([dic_table['slogan'].columns.slogan]).\
    where(dic_table['slogan'].columns.subcategory_id.in_(select([dic_table['subcategory'].columns.id]).\
        where(dic_table['subcategory'].columns.category_id.in_(select([dic_table['category'].columns.id]).\
            where(dic_table['category'].columns.category == 'Automotive slogans' )))))
print('Mapper select: ')
print(mapper_stmt)
mapper_results = engine.execute(mapper_stmt).fetchall()
print(mapper_results)
```

```
Mapper select:
SELECT slogan.slogan
FROM slogan
WHERE slogan.subcategory_id IN (SELECT subcategory.id
FROM subcategory
WHERE subcategory.category_id IN (SELECT category.id
FROM category
WHERE category.category = :category_1))
[['Blaupunkt. The advantage in your car.'], ('Kenwood. Listen to the Future.'], ('Rockford Fosgate. Car audio for fanatic s.'], ('Volfenhag. A German Concept.'], ('Kicker. Fuel for Livin' Loud!'], ('Not just oil, Pennzoil.'], ('Havoline. Add M ore Life to Your Car.'], ('Mobil 1. The oil that's changing oil.'], ('Unlike any other.'], ('Mercedes-Benz. The Future of the Automobile.'], ('Engineered to move the human spirit.'], ('The Power of Dreams.'], ('It must be love.'], ('Honda. Fir st man, then machine.'], ('Technology you can enjoy.'], ('The True Definition of Luxury. Yours.'], (' Acura. Precision Cra fted Performance.'], ('Driven by passion. FIAT.'], ('Alfa Romeo. Beauty is not enough.'], ('Power for your control.'], (' Volvo. For life.'], ('Subaru. Think. Feel. Drive.'], ('Driven By What's Inside.'], ('When You Get It, You Get It.'], ('Th e Beauty of All-Wheel Drive.'], ('Ford. Feel the difference.'], (' Ford. Bold moves.(USA'), (' Built for life in Canad a.(Canada)'), (' Built for the road ahead.'], (' Ford. Designed for living. Engineered to last.'], (' Have you driven a Ford lately?'], ('Infiniti. Accelerating the Future.'], ('BMW. The Ultimate Driving Machine.'], ('BMW. Sheer Driving Pleas
```