

raport

January 8, 2022

1 Adrian Poniatowski, gr.2b, 401346

1.1 Niezbędne biblioteki

Rozpoczęcie tworzenia bazy danych standardowo rozpoczynamy od połączenia się z serwerem bazy danych.

```
[1]: from sqlalchemy import create_engine
from sqlalchemy.ext.declarative import declarative_base

try: # kod w przypadku gdy baza danych jest już utworzona
    db_string = "postgresql://postgres:postgres@localhost:5432"
    engine = create_engine(db_string)

except ProgrammingError: # kod do utworzenia bazy danych, gdy jeszcze nie
    istnieje
    db_string = "postgresql://postgres:postgres@localhost:5432"
    with create_engine(db_string, isolation_level="AUTOCOMMIT").connect() as
    connection:
        connection.execute("CREATE DATABASE lab12")

Base = declarative_base()
```

1.2 Tworzenie struktur tabel

Aby zadeklarować nowy schemat tabeli musimy dodać narzędzia do mapowania typów zmiennych z bazy danych do Pythona. Aby utworzyć nową tabelę połączoną z inną, musimy stworzyć relację między tabelą pierwszą i drugą według klucza obcego w jednej z tabel. Tabele mają postać klas. Na końcu należy utworzyć wszystkie tabele z opisanego schematu.

```
[2]: from sqlalchemy import Column, Integer, String, Date, Float, SmallInteger, Text
from sqlalchemy.types import VARCHAR
from sqlalchemy import ForeignKey

class Booking(Base):
    __tablename__ = 'bookings'
    id = Column(Integer, primary_key=True)
    user_id = Column(Integer, ForeignKey('users.id'))
```

```

place_id = Column(Integer, ForeignKey('places.id'))
start_date = Column(Date)
end_date = Column(Date)
price_per_night = Column(Float)
num_nights = Column(Integer)

class Review(Base):
    __tablename__ = 'reviews'
    id = Column(Integer, primary_key=True)
    booking_id = Column(Integer, ForeignKey('bookings.id'))
    rating = Column(SmallInteger)
    review_body = Column(Text)

class User(Base):
    __tablename__ = 'users'
    id = Column(Integer, primary_key=True)
    email = Column(VARCHAR)

class Host(Base):
    __tablename__ = 'hosts'
    id = Column(Integer, primary_key=True)
    user_id = Column(Integer, ForeignKey('users.id'))

class Place(Base):
    __tablename__ = 'places'
    id = Column(Integer, primary_key=True)
    host_id = Column(Integer, ForeignKey('hosts.id'))
    address = Column(VARCHAR)
    city_id = Column(Integer, ForeignKey('cities.id'))

class City(Base):
    __tablename__ = 'cities'
    id = Column(Integer, primary_key=True)
    name = Column(VARCHAR)
    country_id = Column(Integer, ForeignKey('countries.id'))

class Country(Base):
    __tablename__ = 'countries'
    id = Column(Integer, primary_key=True)
    country_code = Column(VARCHAR)
    name = Column(VARCHAR)

Base.metadata.create_all(engine)

```

1.3 Wyświetlenie bazy danych

Do wyświetlenia bazy danych skorzystałem z programu TablePlus.

```
[3]: from IPython import display
display.Image("./lab12_schema.jpg")
```

[3]:

#	column_name	data_type	is_nullable	column_default	foreign_key	comment
1	id	int4	NO	nextval('bookings...')	EMPTY	→ NULL
2	user_id	int4	YES	NULL	EMPTY	→ NULL
3	place_id	int4	YES	NULL	EMPTY	→ NULL
4	start_date	date	YES	NULL	EMPTY	→ NULL
5	end_date	date	YES	NULL	EMPTY	→ NULL
6	price_per_night	float8	YES	NULL	EMPTY	→ NULL
7	num_nights	int4	YES	NULL	EMPTY	→ NULL

index_name	index_algorithm	is_unique	column_name	condition
bookings_pkey	BTREE	TRUE	id	EMPTY

No row select

public + Structure + Column + Index Info Triggers