

# Zaawansowane programowanie w Pythonie

## Design Proposal

### Implementacja nowego ORMa dla języka Python

#### Zespół 10

Maria Polakowska

Marta Sobol

Magda Twarowska

## 1. Funkcjonalności

Naszym zadaniem jest stworzenie silnika zapytań. Wyróżniłyśmy następujące podzadania:

### 1. Tworzenie modelu

- implementacja trzech klas: Field, Model, QuerySet
- Field jest odpowiednikiem atrybutu / kolumny
  - obsługa kluczy obcych
  - obsługa różnych typów danych
    - AutoField
    - BigIntegerField
    - BooleanField
    - CharField
    - DateField DateTimeField
    - DecimalField
    - FloatField
    - IntegerField
    - PositiveIntegerField
    - PositiveBigIntegerField
    - PositiveSmallIntegerField
    - SmallIntegerField
    - TextField
    - ForeignKey
  - obsługa: null, not null, primary key, unique, max. długość, default value, nazwa kolumny z bazy danych
- Model odpowiada obiektowi / wierszowi z bazy danych
- QuerySet odpowiada zbiorowi obiektów / zapytań
  - metoda get()

- metoda all()
- metoda filter()
  - groupby
  - orderby
  - count
  - różne metody filtrowania (==, !=, >, <, <=, >=, null, not null)

## 2. Implementacja CRUDa

- metoda create()
- metoda delete()
- zmiana wartości (UPDATE)
- metoda save() / commit()

## 3. Połączenie z bazą oraz testy

- utworzymy interfejs do połączenia z wieloma bazami danych
- pierwszą bazą jaką do niego podepnimy będzie SQLite
- zadamy o skalowalność

## 2. Stack technologiczny

- autoformatter - black
- linter - flake8
- środowisko wirtualne - venv
- oskryptowane budowanie, testowanie, uruchamianie aplikacji
- dokumentacja - sphinx
- instrukcja użytkownika - w pliku .md
- semantic versioning
- budowa paczki - pip
- testy automatyczne - pytest, tox

## 3. Harmonogram

Testy jednostkowe będą tworzone równolegle z kodem, przez cały czas trwania projektu.

18.03 - 24.03	<ul style="list-style-type: none"> <li>- szkielet projektu</li> <li>- szkielet modelu i zaczątki implementacji</li> <li>- omówienie szczegółów z prowadzącym</li> </ul>
25.03 - 31.03	<ul style="list-style-type: none"> <li>- praca nad klasą Field</li> <li>- praca nad kluczami obcymi</li> <li>- praca nad różnymi typami danych dla klasy Field</li> </ul>
01.04 - 07.04	<ul style="list-style-type: none"> <li>- praca nad różnymi typami danych dla klasy Field</li> <li>- prace nad kluczami obcymi</li> </ul>
08.04 - 14.04 (PROTOTYP)	<ul style="list-style-type: none"> <li>- prace nad CRUDem (+ klasami Model, QuerySet)</li> <li>- próba integracji silnika zapytań z silnikiem migracji</li> </ul>

15.04 - 21.04	- prace nad CRUDem (+ klasami Model, QuerySet)
22.04 - 28.04	- prace nad CRUDem (+ klasami Model, QuerySet) - utworzenie interfejsu do operacji na bazach danych
29.04 - 05.05	- MAJÓWKA
06.05 - 12.05	- połączenie z SQLite - testy kompatybilności z konkretną bazą
13.05 - 19.05	TYDZIEŃ ZAPASOWY 1
20.05 - 26.05	TYDZIEŃ ZAPASOWY 2
27.05 - 02.06	TYDZIEŃ ZAPASOWY 3

## 4. Bibliografia

Django - <https://docs.djangoproject.com/en/5.0/>  
<https://github.com/django/django/tree/main/django/db/models>  
<https://docs.djangoproject.com/en/5.0/ref/models/fields/>  
<https://docs.djangoproject.com/en/5.0/ref/models/instances/#validating-objects>

SQLite - <https://www.sqlite.org/docs.html>

SQLAlchemy - <https://docs.sqlalchemy.org/en/20/>

SQLModel - <https://sqlmodel.tiangolo.com/>

Tortoise - <https://tortoise.github.io/>

Pydantic - <https://docs.pydantic.dev/latest/>

Inspect - <https://docs.python.org/3/library/inspect.html>

Python - <https://docs.python.org/3/>