

# Baza danych medium społecznościowego

Łukasz Fabia 272724  
Mikołaj Kubś 272662  
Martyna Łopianiak 272682  
Piotr Schubert 272659  
Projektowanie baz danych wt 18:55

11 listopada 2024

## Spis treści

<b>1</b>	<b>Etap 1: Faza koncepcyjna</b>	<b>1</b>
1.1	Analiza świata rzeczywistego	1
1.1.1	Streszczenie - Zarys wymagań projektu	1
1.1.2	Potrzeby informacyjne	1
1.1.3	Czynności, wyszukiwania	2
1.1.4	Cele projektu	2
1.1.5	Zakres projektu	2
1.2	Wymagania funkcjonalne	2
1.3	ERD	3
<b>2</b>	<b>Etap 2: Faza logiczna</b>	<b>3</b>
<b>3</b>	<b>Etap 3: Faza fizyczna</b>	<b>4</b>
<b>4</b>	<b>Faza Fizyczna cd.</b>	<b>5</b>

## 1 Etap 1: Faza koncepcyjna

### 1.1 Analiza świata rzeczywistego

#### 1.1.1 Streszczenie - Zarys wymagań projektu

Celem projektu jest stworzenie bazy danych do obsługi medium społecznościowego. Ma ona przechowywać informacje o użytkownikach, ich treściach, relacjach i aktywnościach. Baza powinna być zaprojektowana w sposób wydajny i skalowalny.

#### 1.1.2 Potrzeby informacyjne

- Rejestracja i logowanie użytkowników z różnymi poziomami dostępu.
- Publikowanie i interakcje z treściami użytkowników (posty, polubienia, komentarze).
- Zarządzanie relacjami społecznymi (znajomości).
- Przechowywanie wiadomości prywatnych i historii aktywności.
- Tworzenie konwersacji z innymi użytkownikami
- Reportowanie postów z nieodpowiednimi treściami
- Tworzenie i zarządzanie stronami organizacji, firm, fanpage itd.
- Tworzenie i zarządzanie wydarzeniami

### 1.1.3 Czynności, wyszukiwania

- Wyszukiwanie użytkowników.
- Wyszukiwanie treści po hashtagach lub słowach kluczowych.
- Filtrowanie aktywności użytkownika, np. przeglądanie polubień i komentarzy.
- Wyszukiwanie relacji (np. znajomi użytkownika, osoby obserwujące daną osobę).
- Dodawanie innych użytkowników do znajomych i interakcja z treściami - dodawanie komentarzy i reakcji
- Tworzenie postów, wydarzeń, grup
- Konwersacja grupowa, pisanie wiadomości

### 1.1.4 Cele projektu

- S (Specific):** Zaprojektowanie bazy danych dla medium społecznościowego.
- M (Measurable):** Baza musi być wydajna, tzn. musi być w stanie obsługiwać dużą ilość użytkowników, co najmniej 20 000.
- A (Achievable):** Projekt zostanie zrealizowany przy użyciu PostgreSQL. Do stworzenia struktury tabel wykorzystany zostanie mechanizm ORM (Object Relational Mapping). Na koniec baza zostanie wypełniona danymi, aby przetestować jej wydajność.
- R (Relevant):** Przechowywanie profili użytkowników oraz interakcje między nimi są kluczowe dla funkcjonowania medium społecznościowego.
- T (Time-bound):** Praca nad projektem powinna zająć 2 miesiące.

### 1.1.5 Zakres projektu

- Multimedia:** W bazie przechowywane będą wyłącznie linki do plików na zewnętrznym serwerze.
- Obsługa haseł:** Wszystkie hasła w bazie będą hashowane.

## 1.2 Wymagania funkcjonalne

### Streszczenie

Użytkownikom przypisany jest jeden z tych poziomów dostępu: admin, user lub guest.

#### Guest(Gość)

- Może przeglądać wybrane dane.

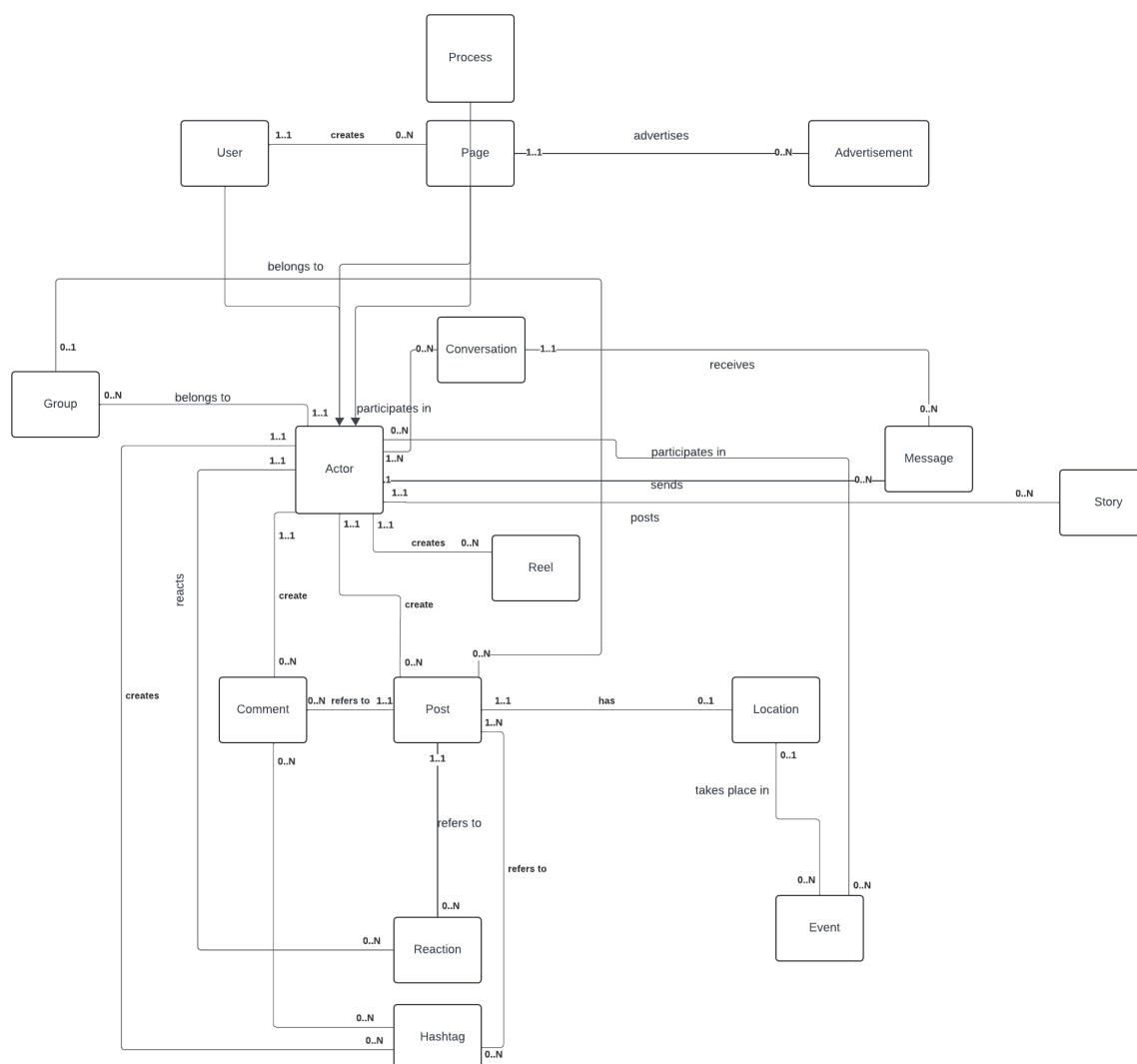
#### Admin

- Może przeglądać, edytować, usuwać, dodawać i przeglądać wszystkie treści, zarządza bazą i nadaje uprawnienia

#### User(Użytkownik)

- System umożliwia rejestrację oraz logowanie.
- Rejestracja wymaga imienia, nazwiska, daty urodzenia, hasła oraz maila.
- Logowanie wymaga maila i hasła.

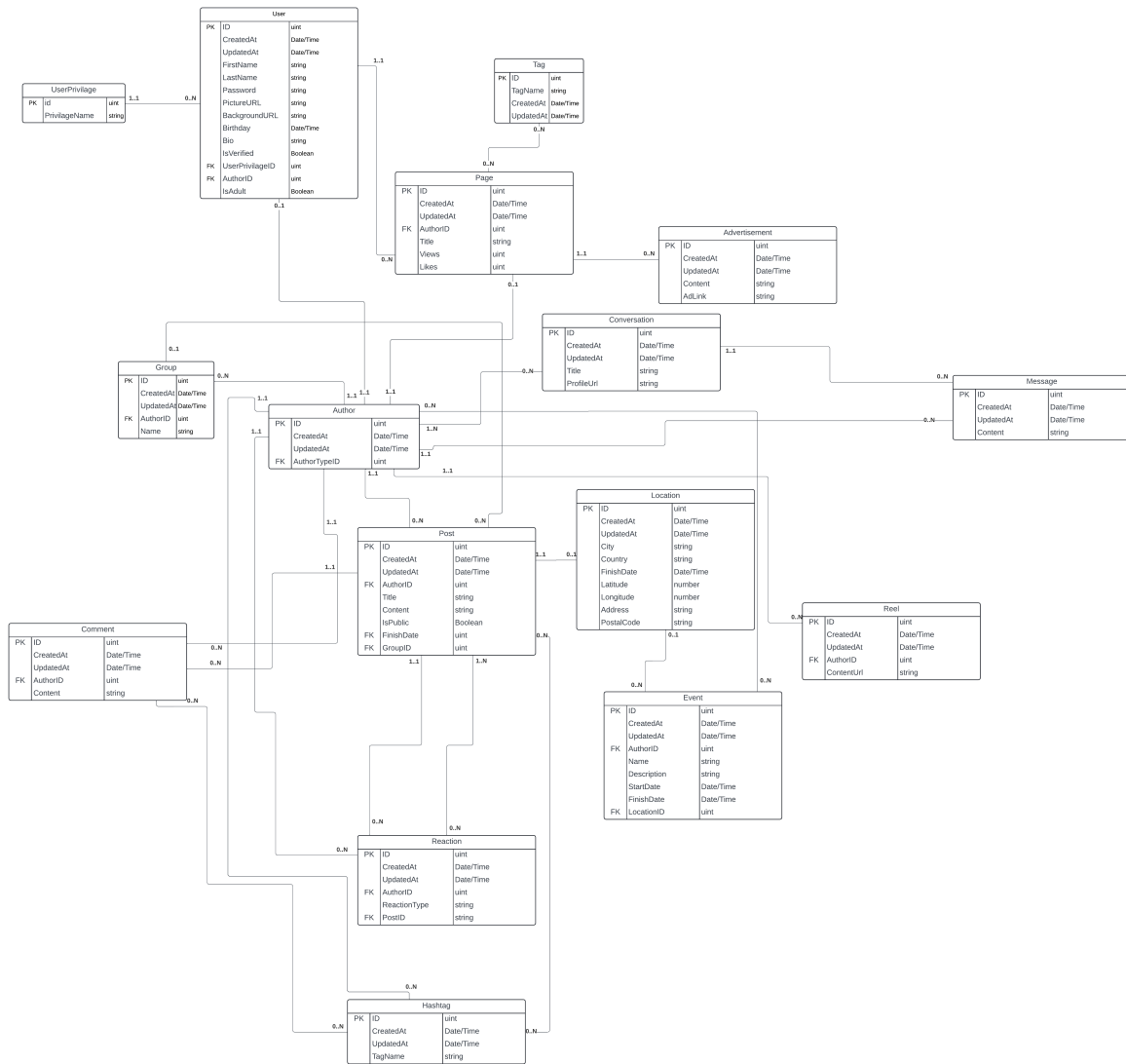
## 1.3 ERD



Rysunek 1: Diagram obiektowo-relacyjny

## 2 Etap 2: Faza logiczna

Do tabel zostały dodane atrybuty, tabele **Page** oraz **User** zostały uogólnione przez **Author**, który bierze udział w innych czynnościach. Baza została także sprowadzona do *III postaci normalnej*, przez co wydzielono kilka nowych tabel.



Rysunek 2: Diagram relacji

### 3 Etap 3: Faza fizyczna

Encje, które uległy zmianie:

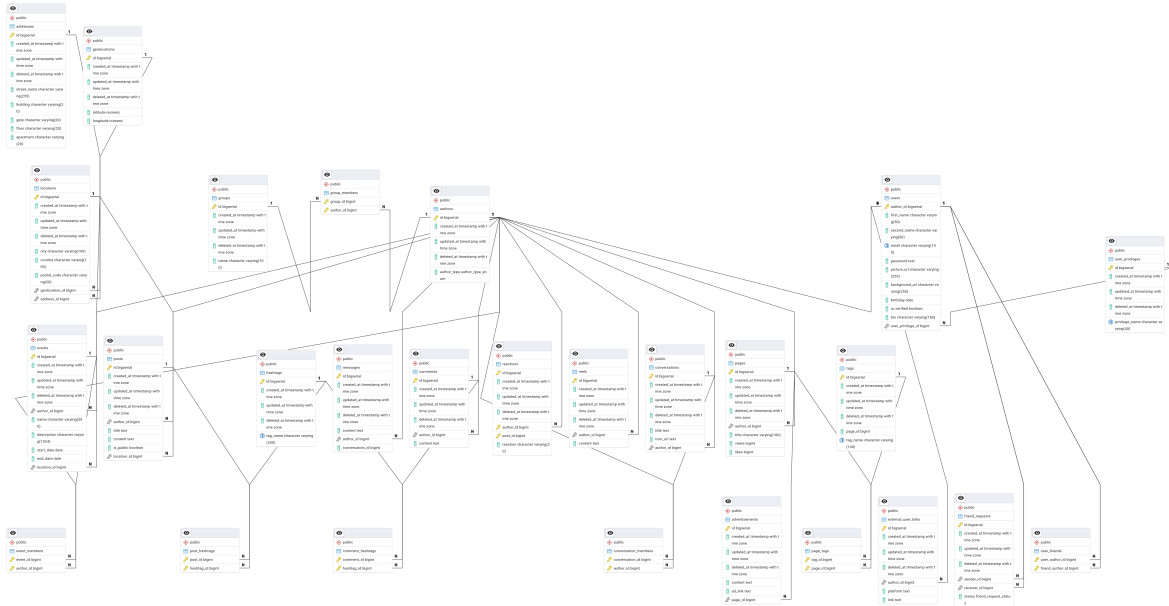
- Rozbicie lokalizacji na pomniejsze tabele
- Zamiana na enumy: typu autora oraz statusu zaproszenia do znajomych
- Dodanie do encji Conversation pole Members, które przechowuje id użytkowników biorących udział w konwersacji

Wykorzystano instrukcję CHECK do potwierdzenia poprawności danych w paru encjach. Przykładowo:

- 'A' nie może być przyjacielem 'A'
- 'A' nie może wysłać zaproszenia do znajomych do 'A'
- Czas rozpoczęcia wydarzenia musi być wcześniejszy niż czas zakończenia wydarzenia
- Grupa musi mieć od 1 do 10000000 członków.

Do stworzenia struktury bazy wykorzystano mechanizm ORM - ([gorm](#)). Sama baza wymagała dopracowania jeśli chodzi o enumeracje oraz reguły usuwania już bezpośrednio w systemie PostgreSQL (pgadmin).

**Usuwanie:** Każdy model posiada pole *DeletedAt* z indeksem. Podczas usuwania danego wiersza pole *DeletedAt* jest ustawiane na znacznik czasu (timestamp) wskazujący moment, w którym dane zostały usunięte. Dzięki temu baza obsługuje soft deleting. Oznacza to, że gdy użytkownik usunie swoje konto, będzie można je przywrócić. Taką operację obsługują na przykład serwisy takie jak Facebook.



Rysunek 3: Diagram relacji z PostgreSQL

## 4 Faza Fizyczna cd.

Mając napisany ORM poprzemio, w tym etapie pozostało napisać funkcje generujące dane do bazy. Rozwiązanie można podzielić na 3 części (od szczegółu do ogółu):

1. Dekorator, który będzie wykonywać określony blok count razy.
2. Funkcja, generująca dany typ danych np. generator użytkowników.
3. Odpowienie ułożenie wywołań.

Zapełnianie bazy danych dużą ilością danych zajmuje stosunkowo dużo czasu, może to być spowodowane przez unikalność niektórych atrybutów(biblioteka "męczy" się z generowaniem unikalnych sensownych danych), ale także przez wywołania które tworzą listy autorów dla np. eventów, ostatnim podejrzeniem może być nieoptymalnie napisany kod.

**Język:** [Go](#)

**Generowanie sztucznych danych:** [gofakeit](#)

**Object relational mapping (ORM):** [gorm](#)

Reszta rzeczy, które zostały wykonane:

- Funkcja usuwająca dane z tabel
- Własne implementacje niektórych danych np. **Title, Birthday**
- Konfiguracja loggera
- Funkcja haszująca hasło