# Przetwarzanie danych w chmurach obliczeniowych – projekt

Adam Młyńczak

## 1. Projekt koncepcji, założenia

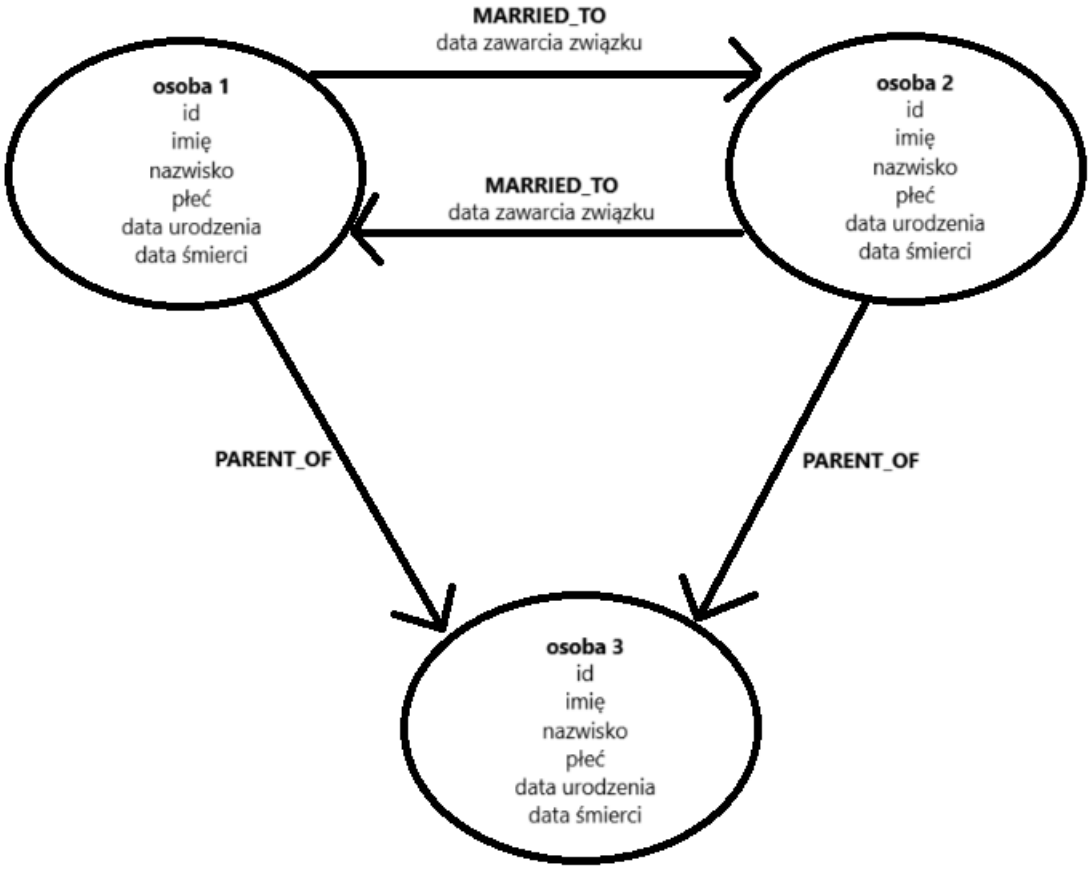
### 1.1. Temat

Tematem projektu jest prosta aplikacja (*Proof of Concept*) obsługująca drzewo genealogiczne.

### 1.2. Funkcjonalności

- Dodawanie osoby do bazy (imię, nazwisko, płeć, data urodzenia, ew. data śmierci)
- Usuwanie osoby z bazy danych
- Modyfikowanie danych osoby w bazie
- Dodawanie relacji pomiędzy osobami (rodzicielstwo, małżeństwo)
- Usuwanie relacji pomiędzy osobami
- Wyświetlenie listy osób – również z przykładowymi filtrami (małżeństwa, rodzeństwa, rodzice)

## 2. Schemat bazy danych




*Rysunek 1: Diagram bazy danych*

## 3. Implementacja

Baza danych została zaimplementowana jako grafowa baza danych *Neo4j* (*Neo4j AuraDB*). Backend aplikacji został napisany przy pomocy Pythona i bibliotek FastAPI, uvicorn oraz neo4j. Frontend aplikacji to szablony HTML, które przy pomocy skryptów napisanych w JavaScript komunikują się z odpowiednimi endpointami.

## 4. Architetkura aplikacji

W aplikacji zaimplementowane są odpowiednie endpointy, za pomocą których możemy otrzymać potrzebne informacje. Są to:

4.1. GET('/') – strona startowa
4.2. POST('api/person') – dodanie osoby do bazy danych
4.3. GET('api/person/{person_id}') – wyszukiwanie osoby po id
4.4. GET('api/people') – pobieranie całej listy
4.5. DELETE('api/person/{person_id}') – usunięcie osoby z bazy danych
4.6. PUT('api/person/{person_id}') – modyfikacja danych o osobie
4.7. POST('api/relationship') – dodanie relacji pomiędzy dwoma osobami
4.8. DELETE('api/relationship') – usunięcie relacji
4.9. GET('api/marriages') – pobranie listy małżeństw
4.10. GET('api/siblings') – pobieranie listy rodzeństw

## 5. Wykorzystane operacje w grafowej bazie danych:

### 5.1. Tworzenie osoby w bazie:

```
CREATE (p:Person {firstName: $firstName, lastName: $lastName, birthDate:
$birthDate, deathDate: $deathDate})
```

### 5.2. Wyszukiwanie osoby po id:

```
    MATCH (p:Person)
    WHERE ID(p) = $person_id
    RETURN p, ID(p) AS id
```

### 5.3. Wyciągnięcie wszystkich osób z bazy:

```
MATCH (p:Person)
    OPTIONAL MATCH (p)<-[:PARENT_OF]-(father:Person {gender: 'male'})
    OPTIONAL MATCH (p)<-[:PARENT_OF]-(mother:Person {gender: 'female'})
    OPTIONAL MATCH (p)-[:MARRIED_TO]-(spouse:Person)
    WITH p,
         ID(p) AS id,
         COALESCE(father.firstName + ' ' + father.lastName, '') AS father,
         COALESCE(mother.firstName + ' ' + mother.lastName, '') AS mother,
         COALESCE(spouse.firstName + ' ' + spouse.lastName, '') AS spouse
    RETURN DISTINCT p, id, father, mother, spouse
```

### 5.4. Usuwanie osoby z bazy danych:

```
MATCH (p:Person)
    WHERE ID(p) = $person_id
    OPTIONAL MATCH (p)-[r]-()
    DELETE r, p
```

### 5.5. Ustawienie rodzicielstwa:

```
MATCH (p1:Person), (p2:Person)
        WHERE ID(p1) = $person1_id AND ID(p2) = $person2_id
        CREATE (p1)-[r:PARENT_OF]->(p2)
        RETURN r
```

### 5.6. Ustawienie małżeństwa:

```
MATCH (p1:Person), (p2:Person)
        WHERE ID(p1) = $person1_id AND ID(p2) = $person2_id
        CREATE (p1)-[r:MARRIED_TO {marriageDate: $marriage_date}]->(p2),
(p2)-[r2:MARRIED_TO {marriageDate: $marriage_date}]->(p1)
        RETURN r, r2
```

### 5.7. Usunięcie relacji:

```
MATCH (p1:Person)-[r]-(p2:Person)
    WHERE ID(p1) = $person1_id AND ID(p2) = $person2_id
    DELETE r
```

### 5.8. Modyfikacja danych o osobie w bazie:

```
MATCH (p:Person)
    WHERE ID(p) = $person_id
    SET p.firstName = $firstName, p.lastName = $lastName, p.birthDate =
$birthDate, p.deathDate = $deathDate, p.gender = $gender
    RETURN p
```

### 5.9. Filtrowanie samych małżeństw:

```
MATCH (p1:Person)-[r:MARRIED_TO]->(p2:Person)
    RETURN p1.firstName + ' ' + p1.lastName AS person1,
           p2.firstName + ' ' + p2.lastName AS person2,
           r.marriageDate AS marriageDate
```

### 5.10. Filtrowanie samych rodzeństw:

```
MATCH (parent:Person)-[:PARENT_OF]->(sibling1:Person)
    MATCH (parent)-[:PARENT_OF]->(sibling2:Person)
    WHERE sibling1 <> sibling2
    RETURN DISTINCT sibling1.firstName + ' ' + sibling1.lastName AS
sibling1,
                    sibling2.firstName + ' ' + sibling2.lastName AS
sibling2
```