

Temat 10.

Projekty praktyczne modelowania danych

dr inż. Michał Malinowski

<https://www.drmalinowski.edu.pl>

Bazy Grafowe – część I
projektowanie Baz
Grafowych w systemie
zarządzania Neo4j



Zagadnienia (5h)

- Model forum dyskusyjnego
- Model produktów bankowych
- Model sklepu internetowego
- Obszary zastosowań baz grafowych

Projekt

Model forum dyskusyjnego



Ćwiczenie 1

Model forum dyskusyjnego

Opis

Forum dyskusyjne jest złożoną strukturą społecznościową, która umożliwia użytkownikom dzielenie się myślami, pytaniem i odpowiedziami na różnorodne tematy w postaci postów w ramach tematycznych wątków.

Cele

1. Zrozumienie wspieranego zagadnienia i zdefiniowanie konkretnych przypadków użycia (pytań) dla aplikacji.
2. Opracowanie wstępnego modelu grafu.
3. Przetestowanie przypadków użycia względem danych testowych i modelu grafu.

Zrozumienie wspieranego zagadnienia i zdefiniowanie konkretnych przypadków użycia (pytań) dla aplikacji

Przypadki użycia (pytania)

- Jak wątki są najczęściej obserwowane przez użytkowników?
- Kto jest najbardziej twórczym autorem postów?
- Które, wątki posiadają najmniej postów?

Narysować model na kartce papieru !

Opracowanie wstępnego modelu grafu

Węzły

- Uzytkownik {id, nazwa, email}: Użytkownicy (autorzy), którzy biorą udział w dyskusjach na forum.
- Watek {id, tytul, data utworzenia}: Indywidualne tematy lub pytania, które są na forum.
- Post {id, tresc, data dodania}: Odpowiedzi lub komentarze do wątków.

Relacje

- obserwuje (Uzytkownik->Watek): Użytkownik obserwuje określony wątek.
- jest autorem (Uzytkownik-> Post): Użytkownik jest autorem określonego posta.
- posiada (Watek->Post): Wątek posiada określony post.

Opracowanie wstępnego modelu grafu

Węzły

MERGE (:Uzytkownik {id:1, nazwa: 'Jan', email: 'jan@example.com'})

MERGE (:Watek {id:1, tytul: 'Jak używać Neo4j?', data_utworzenia: date('2023-08-23')})

MERGE (:Post {id:1, tresc: 'To jest odpowiedź na pytanie.', data_dodania: date()})

Relacje

MATCH (u:Uzytkownik {id:1}), (w:Watek {id:1}) CREATE (u)-[:obserwuje]->(w)

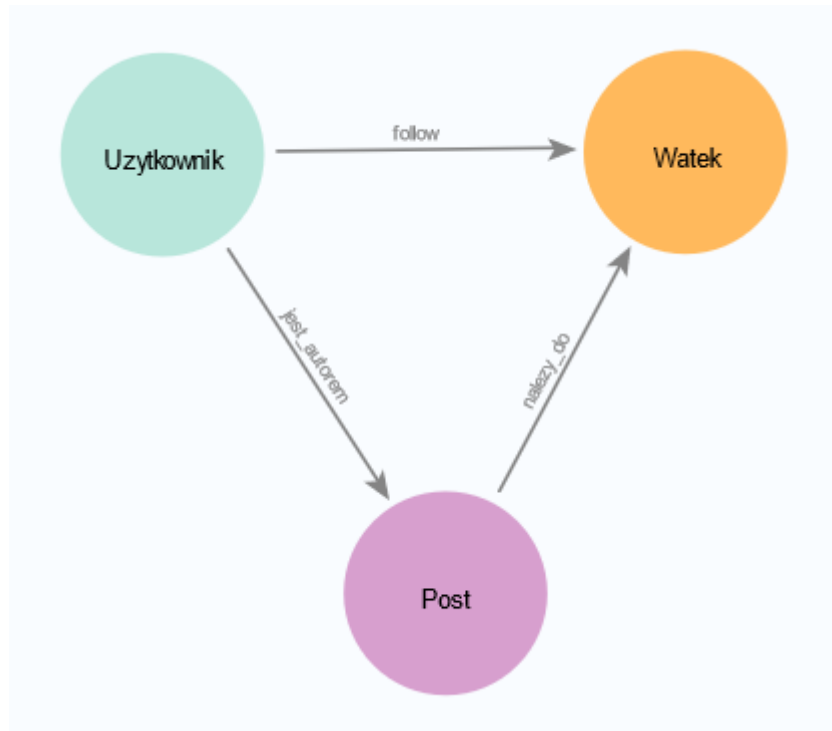
MATCH (p:Post {id:1}), (w:Watek {id:1}) CREATE (w)-[:posiada]->(p)

MATCH (u:Uzytkownik {id:1}), (p:Post {id:1}) CREATE (u)-[:jest_autorem]->(p)

Opracowanie wstępnego modelu grafu

Sprawdzenie modelu

CALL db.schema.visualization()



Przetestowanie przypadków użycia względem danych testowych i modelu grafu

Przypadki użycia (pytania)

- Jakie wątki są najczęściej obserwowane przez użytkowników?
MATCH (w:Watek)<-[:obserwuje]-(u:Uzytkownik)
RETURN w.tytul AS tytul_watku, COUNT(u) AS liczba_obserwujacych
ORDER BY COUNT(u) DESC LIMIT 5
- Kto jest najbardziej popularnym autorem postów?
MATCH (u:Uzytkownik)-[:jest_autorem]->(p:Post)
RETURN u.nazwa AS nazwa_autora, COUNT(p) AS liczba_postow
ORDER BY COUNT(p) DESC LIMIT 1
- Które wątki posiadają najmniej postów?
MATCH (w:Watek)-[:posiada]->(p:Post)
RETURN w.tytul AS tytul_watku, COUNT(p) AS liczba_postow
ORDER BY COUNT(p) ASC LIMIT 5



Jak klient opisał
swoje potrzeby



Jak to zrozumiał
projektant



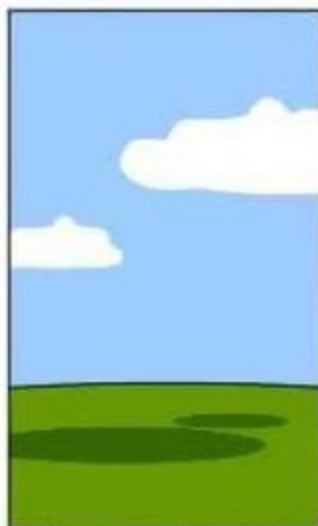
Jak to wymodelował
analityk



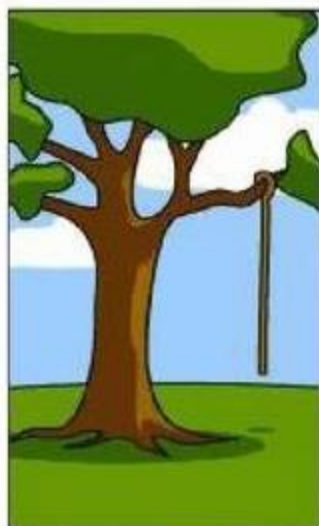
Jak to programista
zakodował



Jak to marketer
przedstawił



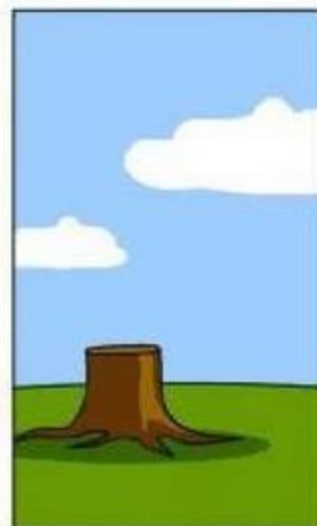
Jak to zostało opisane
w dokumentacji



Jakie funkcje zostały
przewidziane



Za co zapłacił klient



Czego dotyczy pomoc
techniczna



Czego klient naprawdę
potrzebował

Projekt

Model produktów bankowych



Ćwiczenie 2

Model produktów bankowych

Opis

Banki oferują różnorodne produkty finansowe dla swoich klientów. Model produktów bankowych został stworzony, aby odzwierciedlić różne relacje pomiędzy klientami a produktami bankowymi w bazie grafowej. Oto opis modelowanej rzeczywistości:

Klienci są podstawowymi podmiotami w tym modelu, reprezentującymi osoby fizyczne lub firmy korzystające z usług bankowych. Mogą oni posiadać różne produkty bankowe, takie jak konta, kredyty, lokaty czy karty. Każdy klient ma unikalne identyfikatory oraz inne dane demograficzne, takie jak imię, nazwisko, data urodzenia, adres czy numer telefonu. Firmy, jako klienci, mają dodatkowo unikatowe identyfikatory, takie jak NIP.

- Konta bankowe: Są to wirtualne "pudełka", gdzie klienci przechowują swoje pieniądze. Konta mają określony numer, saldo, datę otwarcia oraz typ (np. osobiste, oszczędnościowe, firmowe).
- Kredyty: Reprezentują pożyczone pieniądze, które klienci muszą zwrócić w określonym czasie. Każdy kredyt ma określoną kwotę, oprocentowanie, datę udzielenia i czas trwania.
- Lokaty: Są to specjalne konta, na których klienci przechowują pieniądze przez określony czas, zazwyczaj o wyższym oprocentowaniu niż standardowe konta. Lokaty mają określoną kwotę, oprocentowanie, datę rozpoczęcia i czas trwania.
- Karty bankowe: Umożliwiają klientom dostęp do ich kont i płatności bezgotówkowych. Każda karta ma numer, typ, datę wydania i datę ważności. Karty są zazwyczaj przypisane do konkretnego konta.

Model produktów bankowych

Opis

Banki oferują różnorodne produkty finansowe dla swoich *klientów*. Model produktów bankowych został stworzony, aby odzwierciedlić różne relacje pomiędzy klientami a produktami bankowymi w bazie grafowej. Oto opis modelowanej rzeczywistości:

Klienci są podstawowymi podmiotami w tym modelu, reprezentującymi osoby fizyczne lub firmy korzystające z usług bankowych. Mogą oni posiadać różne produkty bankowe, takie jak *konta*, *kredyty*, *lokaty* czy *karty*. Każdy klient ma unikalne identyfikatory oraz inne dane demograficzne, takie jak imię, nazwisko, data urodzenia, adres czy numer telefonu. Firmy, jako klienci, mają dodatkowo unikatowe identyfikatory, takie jak NIP.

- Konta bankowe: Są to wirtualne "pudełka", gdzie klienci przechowują swoje pieniądze. Konta mają określony numer, saldo, datę otwarcia oraz typ (np. osobiste, oszczędnościowe, firmowe).

- Kredyty: Reprezentują pożyczone pieniądze, które klienci muszą zwrócić w określonym czasie. Każdy kredyt ma określoną kwotę, oprocentowanie, datę udzielenia i czas trwania.

- Lokaty: Są to specjalne konta, na których klienci przechowują pieniądze przez określony czas, zazwyczaj o wyższym oprocentowaniu niż standardowe konta. Lokaty mają określoną kwotę, oprocentowanie, datę rozpoczęcia i czas trwania.

- Karty bankowe: Umożliwiają klientom dostęp do ich kont i płatności bezgotówkowych. Każda karta ma numer, typ, datę wydania i datę ważności. Karty są zazwyczaj *przypisane do konkretnego konta*.

Zrozumienie wspieranego zagadnienia i zdefiniowanie konkretnych przypadków użycia (pytań) dla aplikacji

Cele

1. Zrozumienie wspieranego zagadnienia i zdefiniowanie konkretnych przypadków użycia (pytań) dla aplikacji.
2. Opracowanie wstępnego modelu grafu.
3. Przetestowanie przypadków użycia względem danych testowych i modelu grafu.

Przypadki użycia (pytania)

- Wyszukaj klientów, którzy mają konto z saldem większym niż 10 000 PLN
- Wyszukaj klientów, którzy zaciągnęli kredyt o oprocentowaniu wyższym niż 5%
- Znajdź wszystkie karty przypisane do kont posiadanych przez danego klienta
- Znajdź najbogatszych klientów z uwzględnieniem sald ich kont oraz kwot na lokatach i kredytach

Opracowanie wstępnego modelu grafu

Węzły



Relacje



Zrozumienie wspieranego zagadnienia i zdefiniowanie konkretnych przypadków użycia (pytań) dla aplikacji

Cele

1. Zrozumienie wspieranego zagadnienia i zdefiniowanie konkretnych przypadków użycia (pytań) dla aplikacji.
2. Opracowanie wstępnego modelu grafu.
3. Przetestowanie przypadków użycia względem danych testowych i modelu grafu.

Przypadki użycia (pytania)

- Wyszukaj **klientów**, którzy **mają konto** z saldem większym niż 10 000 PLN
- Wyszukaj **klientów**, którzy **zaciągnęli kredyt** o oprocentowaniu wyższym niż 5%
- Znajdź wszystkie **karty przypisane do kont posiadanych** przez danego **klienta**
- Znajdź najbogatszych **klientów** z uwzględnieniem sald ich **kont** oraz kwot na **lokatach** i **kredytach**

Opracowanie wstępnego modelu grafu

Węzły

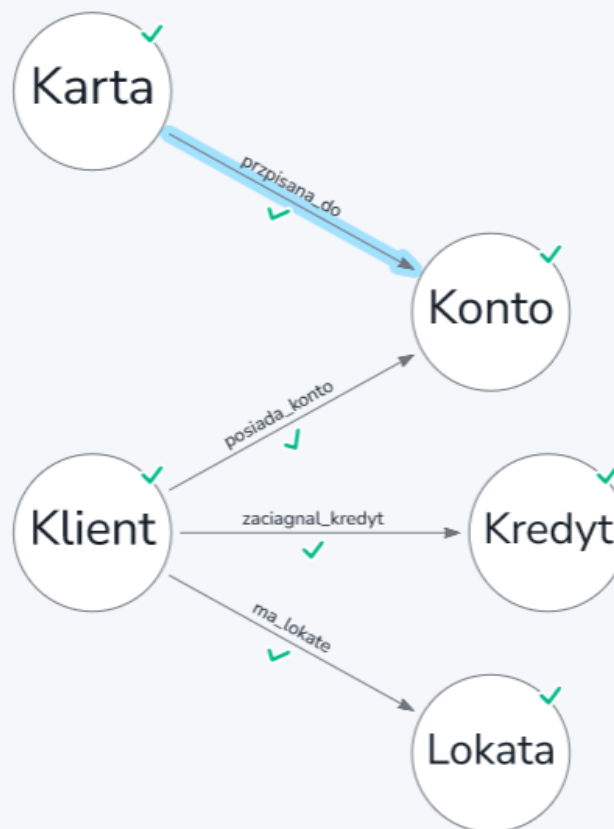
- Klient {id, imie, nazwisko, data_urodzenia, adres, telefon}: osoba korzystająca z produktów bankowych
- Konto {numer, saldo, data_otwarcia}: podstawowy produkt bankowy umożliwiający przechowywanie pieniędzy.
- Kredyt {id, kwota, oprocentowanie, data_udzielenia, czas_trwania}: produkt bankowy umożliwiający zaciągnięcie pożyczki.
- Lokata {id, kwota, oprocentowanie, data_rozporoczenia, czas_trwania}: produkt umożliwiający dobre ulokowanie pieniędzy
- Karta {numer, data_wydania, data_waznosci}: produkt umożliwiający płatności bezgotówkowe

Relacje

- posiada_konto (Klient->Konto): symbolizująca posiadanie konta przez klienta.
- zaciagnal_kredyt (Klient->Kredyt): wskazująca, że dany klient zaciągnął określony kredyt.
- ma_lokate (Klient->Lokata): reprezentująca lokatę założoną przez klienta.
- przypisana_do (Karta->Konto): wskazująca, do jakiego konta jest przypisana dana karta.

[Browse](#)

^ ! b_lokata.csv ...

☐ Filter file

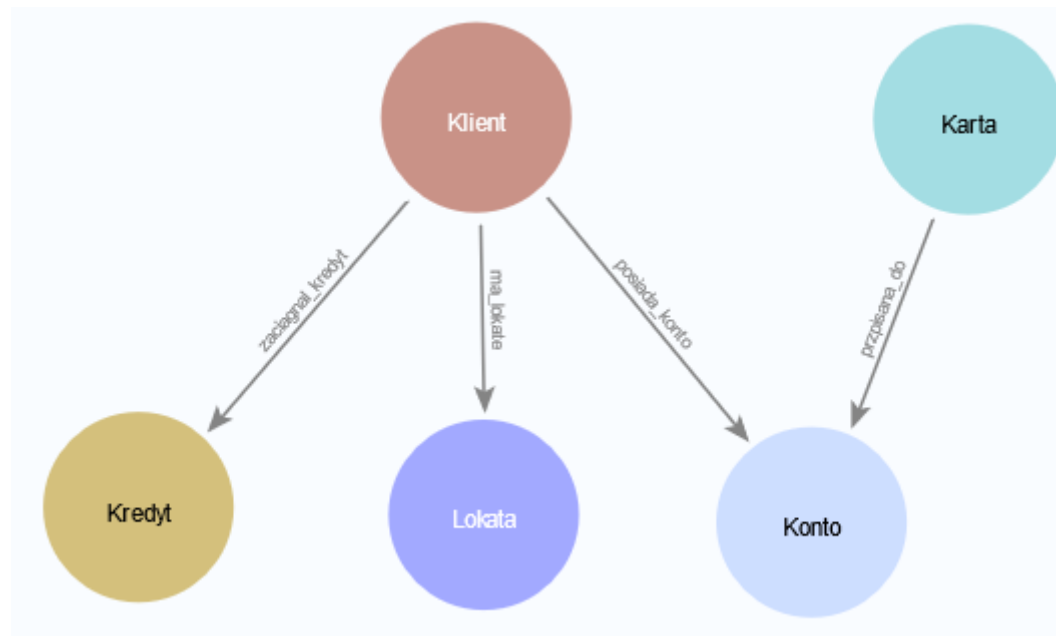
numer_k... ▾

numer_... ▾

Opracowanie wstępnego modelu grafu

Sprawdzenie modelu

CALL db.schema.visualization()



Explore

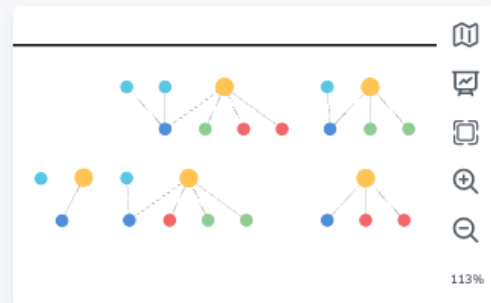
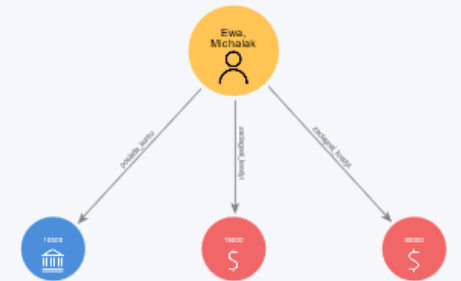
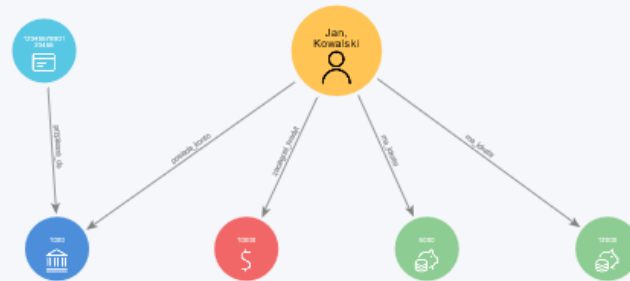
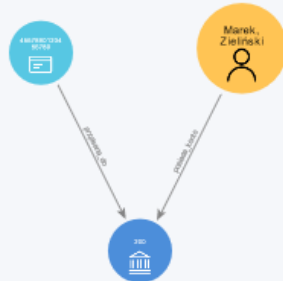
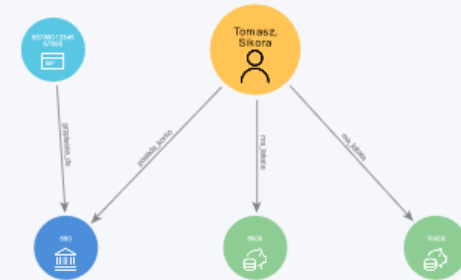
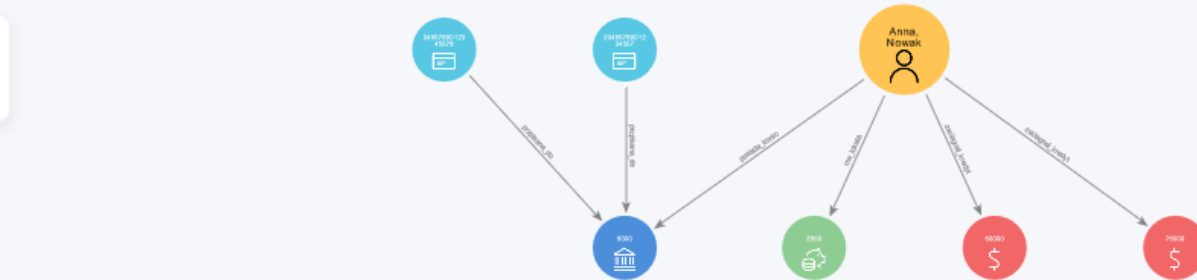
Query

Import

7285dc39.databases.neo4j.io:7687 / neo4j

Perspective

Search graph



Przetestowanie przypadków użycia względem danych testowych i modelu grafu

Przypadki użycia (pytania)

- Wyszukaj **klientów**, którzy **mają konto** z saldem większym niż 10 000 PLN

```
MATCH (k:Klient)-[:posiada_konto]->(kon:Konto)
WHERE kon.saldo > 10000
RETURN k.imie, k.nazwisko, kon.saldo ORDER BY kon.saldo DESC;
```

- Wyszukaj **klientów**, którzy **zaciągnęli kredyt** o oprocentowaniu wyższym niż 5%

```
MATCH (k:Klient)-[:zaciagnal_kredyt]->(kr:Kredyt)
WHERE kr.oprocentowanie > 5
RETURN k.imie, k.nazwisko, kr.oprocentowanie ORDER BY kr.oprocentowanie DESC;
```

- Znajdź wszystkie **karty przypisane do kont posiadanych** przez danego **klienta**

```
MATCH (k:Klient{imie:"Anna",nazwisko:"Nowak"})-[:posiada_konto]->(ko:Konto)<-[:przpisana_do]-(ka:Karta)
RETURN k.imie, k.nazwisko, ka.numer
```

Przetestowanie przypadków użycia względem danych testowych i modelu grafu

Przypadki użycia (pytania)

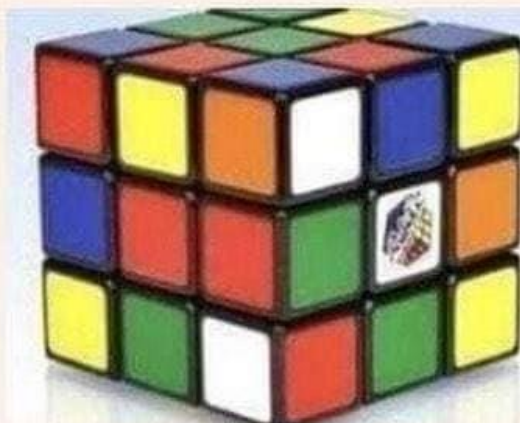
- Znajdź najbogatszych **klientów** z uwzględnieniem sald ich **kont** oraz kwot na **lokatach** i **kredytach**

```
MATCH (k:Klient)
OPTIONAL MATCH (k)-[:posiada_konto]->(ko:Konto)
OPTIONAL MATCH (k)-[:ma_lokate]->(lo:Lokata)
OPTIONAL MATCH (k)-[:zaciagnal_kredyt]->(kr:Kredyt)
RETURN k.imie, k.nazwisko,
SUM(coalesce(ko.saldo, 0)) + SUM(coalesce(lo.kwota, 0)) - SUM(coalesce(kr.kwota, 0)) as majatek
ORDER BY majatek DESC
```

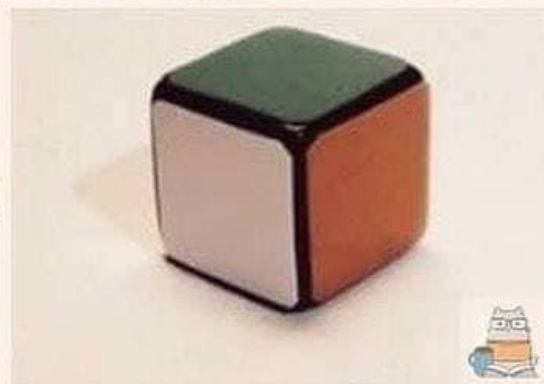
COALESCE - zwraca pierwszą napotkaną wartość nie będącą nullem

```
RETURN COALESCE(NULL, 'druga', 'trzecia');
```

Co omówił
wykładowca



Co
zapamiętałem



Co było
na zaliczeniu



Projekt

Model sklepu internetowego



Ćwiczenie 3

Model sklepu internetowego

Opis

Sklep internetowy oferuje różnorodne produkty w różnych kategoriach. Klienci dokonują zakupów, tworząc zamówienia, które składają się z różnych produktów.

Cele

1. Zrozumienie wspieranego zagadnienia i zdefiniowanie konkretnych przypadków użycia (pytań) dla aplikacji.
2. Opracowanie wstępnego modelu grafu.
3. Przetestowanie przypadków użycia względem danych testowych i modelu grafu.

Zrozumienie wspieranego zagadnienia i zdefiniowanie konkretnych przypadków użycia (pytań) dla aplikacji

Przypadki użycia (pytania)

- Znajdź 10 najpopularniejszych produktów (liczby produktów w zamówieniach)
- Znajdź klientów, którzy dokonali zakupu w ostatnich 4 miesiącach
- Rekomendacje produktów na podstawie historii zakupów produktów z danych kategorii

Opracowanie wstępnego modelu grafu

Węzły



Relacje



Opracowanie wstępnego modelu grafu

Węzły

- Kupiec {id, imie, nazwisko, email, data_rejestracji}: osoba składająca zamówienia w sklepie
- Produkt {id, nazwa, cena, magazyn}: towar sprzedawany w sklepie.
- Zamowienie {id, data_zamowienia, status}: zbiór produktów zakupionych w ramach jednej transakcji.
- Kategoria {id, nazwa}: zbiór produktów logicznie powiązanych ze sobą.

Relacje

- kupil (Kupiec->Zamowienia): opisuje, który klient dokonał danego zamówienia.
- zawiera (Zamowienie->Produkt, liczba): opisuje, jakie produkty są w danym zamówieniu.
Może zawierać dodatkowe atrybuty, takie jak liczba.
- należy do (Produkt->Kategoria): opisuje, do jakiej kategorii należy dany produkt.

Files

[Browse](#)

^ c_kategoria.csv ...

^ c_kupiec.csv ...

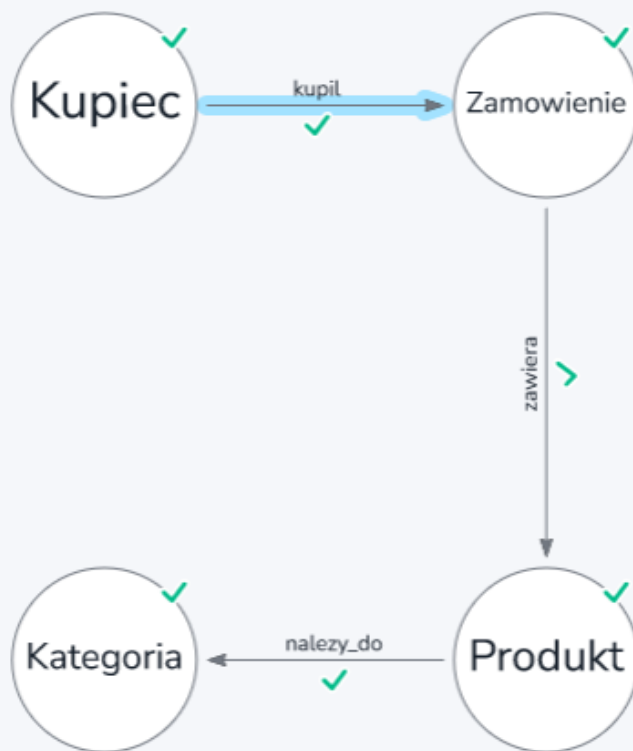
^ c_kupiec_zamowieni... ...

^ c_produkt.csv ...

^ c_produkt_kategoria... ...

^ c_zamowienie.csv ...

^ c_zamowienie_prod... ...

[Show results](#)[Run import](#)

Relationship type ⓘ

[Reverse direction](#)

Name

kupil

File ⓘ

Name

c_kupiec_zamowienie.csv

☐ Filter file

Node ID mapping

Node

ID

ID column

From Kupiec

id

Kupiec_id

To

Zamo...

id

Zamowi...

[Need help?](#)

Properties

[Map from file](#)

Name

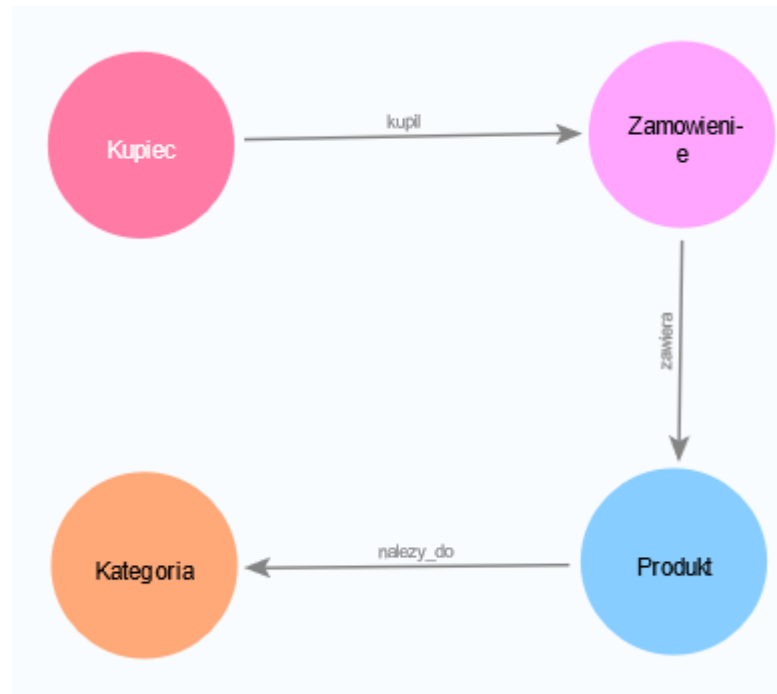
Type

Column

Opracowanie wstępnego modelu grafu

Sprawdzenie modelu

CALL db.schema.visualization()



Przetestowanie przypadków użycia względem danych testowych i modelu grafu

Przypadki użycia (pytania)

- Znajdź 10 najpopularniejszych **produktów** (liczby **produktów** w **zamówieniach**)

```
MATCH (z:Zamowienie)-[x:zawiera]->(p:Produkt)
RETURN p.nazwa, sum(x.ilosc) AS liczba_zamowien
ORDER BY liczba_zamowien DESC LIMIT 10
```

- Znajdź **klientów**, którzy dokonali **zakupu** w ostatnich 4 miesiącach

```
MATCH (k:Kupiec)-[:kupil]->(z:Zamowienie)
WHERE date(z.data_zamawienia)> date()- duration('P4M')
RETURN DISTINCT k.imie, k.nazwisko
```

Przetestowanie przypadków użycia względem danych testowych i modelu grafu

Przypadki użycia (pytania)

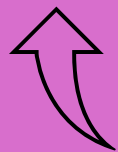
- Rekomendacje **produktów** na podstawie historii **zakupów**

```
MATCH (k:Kupiec {id: 1})-[:kupil]->(Zamowienie)-[:zawiera]->(p:Produkt)-[:nalezy_do]->(cat:Kategoria)
<-[:nalezy_do]-(rec:Produkt)
WHERE NOT (k)-[:kupil]->(Zamowienie)-[:zawiera]->(rec)
RETURN DISTINCT rec.nazwa AS RekomendowaneProdukty, cat.nazwa AS Kategoria
```

1. Wyszukuje Kupca o konkretnym id.
2. Następnie przeszukuje zamówienia tego kupca, aby znaleźć produkty, które kupił.
3. Na podstawie kategorii produktów, które Kupiec już zakupił, szukamy innych produktów z tej samej kategorii (które reprezentują rekomendacje).
4. Używamy klauzuli WHERE NOT aby upewnić się, że rekomendowane produkty nie zostały już wcześniej zakupione przez Kupca.
5. Zwracamy unikatowe nazwy rekomendowanych produktów.

LIFE OF AN IT PERSON





Temat 11.

dr inż. Michał Malinowski

<https://www.drmalinowski.edu.pl>

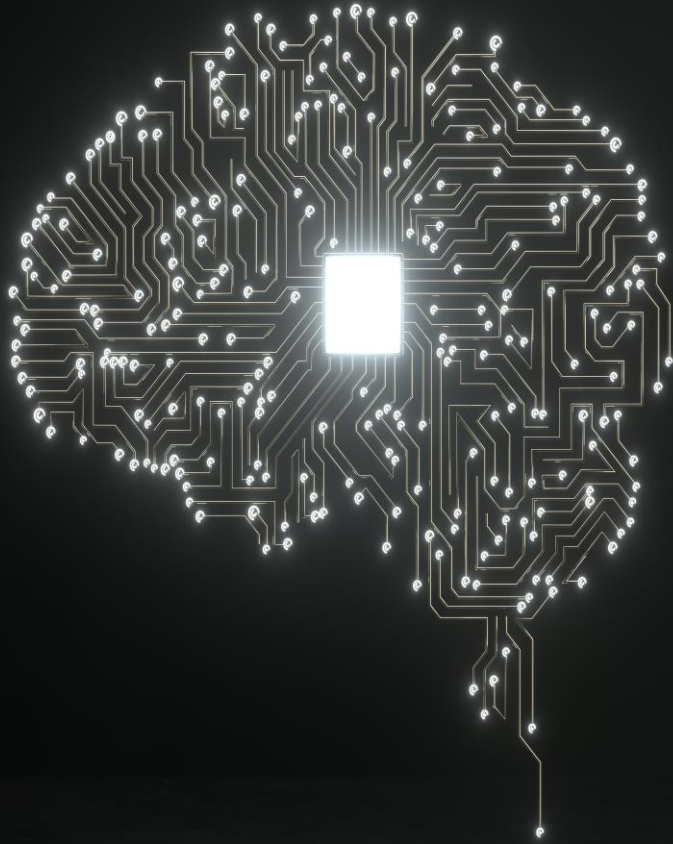
Egzamin praktyczny

Bazy Grafore – część I
projektowanie Baz
Grafore w systemie
zarządzania Neo4j



Zagadnienia (2h)

- Samodzielna realizacja projektu
- Sprawdzenie wyników pracy



Model biblioteki

Opis

Biblioteka pragnie zarządzać swoim zasobem książek oraz śledzić wypożyczenia dokonywane przez czytelników. Kluczowe jest monitorowanie, które książki są obecnie wypożyczone, kto je posiada oraz kiedy mają zostać zwrócone.

Cele

1. Zrozumienie wspieranego zagadnienia i zdefiniowanie konkretnych przypadków użycia (pytań) dla aplikacji.
2. Opracowanie wstępnego modelu grafu.
3. Przetestowanie przypadków użycia względem danych testowych i modelu grafu.

Zrozumienie wspieranego zagadnienia i zdefiniowanie konkretnych przypadków użycia (pytań) dla aplikacji

Przypadki użycia (pytania)

- Wyszukiwanie wszystkich książek wypożyczonych przez konkretnego czytelnika.
- Znalezienie czytelników, którzy wypożyczyli książki w ciągu ostatnich 120 dni.
- Wyszukiwanie książek, które nie zostały jeszcze nigdy wypożyczone.

Opracowanie wstępnego modelu grafu

Węzły



Relacje



Przetestowanie przypadków użycia względem danych testowych i modelu grafu

Przypadki użycia (pytania)

- Wyszukiwania wszystkich **książek wypożyczonych** przez konkretnego **czytelnika**.

?

- Znalezienie **czytelników**, którzy **wypożyczyli książki** w ciągu ostatnich 120 dni.

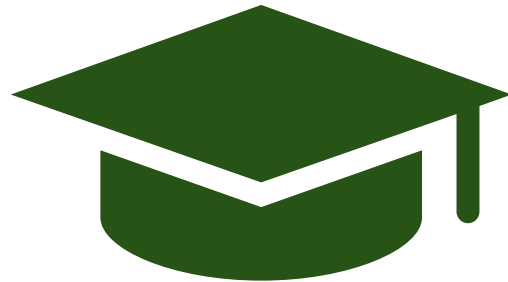
?

- Wyszukiwania **książek**, które nie zostały jeszcze nigdy **wypożyczone**.

?

Egzamin

**Sprawdzenie
wyników**



**Omówienie
egzaminu**

Opracowanie wstępnego modelu grafu

Węzły

Czytelnik {id, imie, nazwisko}

Ksiazka {id, tytul}

Relacje

- wypozyczyl (Czytelnik->Ksiazka, data_wypozyczenia, data_oddania)

Przetestowanie przypadków użycia względem danych testowych i modelu grafu

Przypadki użycia (pytania)

- Wyszukiwania wszystkich **książek wypożyczonych** przez konkretnego **czytelnika**.

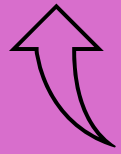
```
MATCH (c:Czytelnik {id:2})-[:wypozyczyl]->(k:Ksiazka)
RETURN k.tytul AS Tytul_Ksiazki;
```

- Znalezienie **czytelników**, którzy **wypożyczyli książki** w ciągu ostatnich 120 dni.

```
MATCH (c:Czytelnik)-[r:wypozyczyl]->(k:Ksiazka)
WHERE date(r.data_wypozyczenia) > date() - duration('P120D')
RETURN c.imie, c.nazwisko, r.data_wypozyczenia, k.tytul as Wypozyczone_ksiazki;
```

- Wyszukiwania **książek**, które nie zostały jeszcze nigdy **wypożyczone**.

```
MATCH (k:Ksiazka)
WHERE NOT (k)<-[:wypozyczyl]-()
RETURN k.tytul AS NieWypozyczoneKsiazki;
```



Temat 12.

Publikacje

dr inż. Michał Malinowski

<https://www.drmalinowski.edu.pl>

Bazy Grafowe – część I
projektowanie Baz
Grafowych w systemie
zarządzania Neo4j

Publikacje

„Porównanie grafowych i relacyjnych baz danych”

<https://www.researchgate.net/publication/387556927> Porownanie baz grafowych i relacyjnych

„Wykorzystanie nowoczesnych metod z obszaru inżynierii w postaci baz grafowych w zarządzaniu organizacją”

<https://www.researchgate.net/publication/386045671> Wykorzystanie nowoczesnych metod z obszaru inzynieri i w postaci baz grafowych w zarzadzaniu organizacja

„Zastosowanie grafów i sieci w systemach rekomendacji”

<https://www.researchgate.net/publication/356253594> Zastosowanie grafow i sieci w systemach rekomendacji



Dziękuję za uwagę 😊

dr inż. Michał Malinowski

<https://www.drmalinowski.edu.pl>

