



POLITECHNIKA LUBELSKA
WYDZIAŁ ELEKTROTECHNIKI I INFORMATYKI
KIERUNEK STUDIÓW
INFORMATYKA

System biblioteczny w Neo4J

Przedmiot: Zarządzanie bazami SQL i NoSQL

Autorzy:

Ewelina Musińska

Wiktoria Matacz

Agnieszka Marzęda

1. Motywacja projektu

Biblioteka publiczna dąży do wdrożenia nowoczesnego, zintegrowanego systemu zarządzania zasobami, który usprawni codzienne funkcjonowanie instytucji, podniesie jakość obsługi czytelników oraz umożliwi lepszą organizację pracy personelu. Głównym celem jest stworzenie kompleksowego rozwiązania, które obejmie zarówno zarządzanie zasobami bibliotecznymi, jak i procesy administracyjne związane z pracą w wielu lokalizacjach (filiach biblioteki).

System powinien umożliwiać efektywne zarządzanie różnorodnymi zasobami biblioteki, w tym książkami tradycyjnymi, audiobookami oraz e-bookami. Ważne jest, aby możliwe było przypisywanie pozycji do odpowiednich kategorii, co ułatwi wyszukiwanie i porządkowanie zbiorów. Użytkownicy systemu – zarówno bibliotekarze, jak i czytelnicy – powinni mieć łatwy dostęp do informacji o dostępności poszczególnych zasobów w różnych filiach biblioteki.

System ma również wspierać procesy rezerwacji i wypożyczeń online, umożliwiając użytkownikom zdalne przeglądanie katalogu, rezerwowanie interesujących ich pozycji oraz śledzenie statusu wypożyczeń. Funkcja ta musi działać w kontekście wielu lokalizacji, pozwalając użytkownikom na wybór dogodnej filii do odbioru lub zwrotu materiałów.

Kolejnym istotnym elementem systemu jest możliwość zarządzania zaległościami, np. poprzez automatyczne naliczanie kar za przetrzymane pozycje.

Z perspektywy administracyjnej, system powinien także wspierać zarządzanie personelem biblioteki. Dotyczy to m.in. ewidencji pracowników, planowania i rejestrowania urlopów, zarządzania pensjami oraz zamówieniami towarów potrzebnych do funkcjonowania placówek.

Nowe rozwiązanie ma na celu nie tylko unowocześnienie dotychczasowych procesów, ale również poprawę jakości obsługi użytkownika końcowego i zwiększenie efektywności pracy biblioteki jako instytucji.

2. Szczegółowy opis wymagań funkcjonalnych systemu bibliotecznego opartego na Neo4j

1. Przeszukiwanie pozycji

Opis funkcjonalności:

System umożliwia użytkownikom przeszukiwanie pozycji bibliotecznych (książek, audiobooków i e-booków) po tytule, autorze, typie, wydawnictwie, języku lub kategorii. Dzięki zastosowaniu Neo4j, możliwe jest szybkie filtrowanie wyników na podstawie relacji między węzłami.

Struktura danych:

- **Węzeł „Pozycja”:**
 - Atrybuty:
 - id(string) - unikalny identyfikator pozycji
 - tytuł(string) - tytuł pozycji
 - opis(string) - opis treści
 - rok_wydania(integer) - rok publikacji
 - język(string) - język pozycji (np. "polski", "angielski")
 - typ(string) - typ pozycji ("książka", "audiobook", "ebook")
 - isbn(string) - numer ISBN (opcjonalnie)
 - liczba_stron(integer) - dla książek i e-booków
 - czas_trwania(integer) - dla audiobooków (w minutach)
 - typ_okładki(string)- dla książek
 - narrator-dla audiobooków
 - Format-dla e-booków
 - Relacje jednokierunkowe:
 - „NAPISAŁ” - łączy z węzłem Autor (wyszukiwanie pozycji danego autora)
 - „NALEŻY_DO” - łączy z węzłem Kategoria (pozycje z danej kategorii)
 - „WYDAŁO” - łączy z węzłem Wydawnictwo (kto wydał pozycję)
 - „JEST_EGZEMPLARZEM” - określa które egzemplarze są powiązane z daną pozycją
- **Węzeł „Autor”:**
 - Atrybuty:
 - id(string) - unikalny identyfikator autora

- imie(string) - imię autora
 - nazwisko(string) - nazwisko autora
 - data_urodzenia(date) - data urodzenia
 - narodowosc(string) - narodowość autora
 - biografia(string) - krótka biografia (opcjonalnie)
- Relacja jednokierunkowa:
 - „NAPISAŁ” (relacja z węzłem „Pozycja”) - łączy z węzłem Pozycja (wyszukiwanie autorów, którzy napisali daną pozycję)
- **Węzeł „Kategoria”:**
 - Atrybuty:
 - id(string) - unikalny identyfikator kategorii
 - nazwa(string) - nazwa kategorii (np. "thriller", "nauka", "beletrystyka")
 - Relacje:
 - „NALEŻY_DO” (relacja z węzłem „Pozycja”) - pozycje należące do tej kategorii
- **Węzeł „Wydawnictwo”:**
 - Atrybuty:
 - id(string) - unikalny identyfikator wydawnictwa
 - nazwa(string) - nazwa wydawnictwa
 - kraj(string) - kraj siedziby wydawnictwa
 - rok_zalozenia(integer) - rok założenia
 - nip(integer)-numer identyfikacji podatkowej (dla polskich firm)
 - regon-numer statystyczny (dla polskich firm)
 - krs-numer w krajowych rejestrze sądowym (dla polskich firm)
 - Adres-pelny adres siedziby wydawnictwa
 - Relacja jednokierunkowa:
 - „WYDAŁO” (relacja z węzłem „Pozycja”) - dane wydawnictwo opublikowało konkretną pozycję

2. Integracja czytelnika z biblioteka

Opis funkcjonalności:

System umożliwia użytkownikom wypożyczanie, rezerwowanie oraz zarządzanie egzemplarzami bibliotecznymi. Każdy egzemplarz jest powiązany z odpowiednią filią, a procesy takie jak rezerwacje, wypożyczenia oraz zwroty są monitorowane w czasie rzeczywistym. Dzięki relacjom w grafie Neo4j, system pozwala na efektywne zarządzanie dostępnością egzemplarzy oraz lokalizacją w filiach, a także naliczanie ewentualnych kar za nieterminowy zwrot.

Powiązania w grafie:

- **Węzeł „Czytelnik”:**
 - Atrybuty:
 - id(string) - unikalny identyfikator czytelnika
 - imie(string) - nazwa czytelnika
 - nazwisko (string) - nazwisko czytelnika.
 - e-mail (string) - adres e-mail czytelnika.
 - login (string) - nazwa użytkownika używana do logowania.
 - hasło (string) - zaszyfrowane hasło.
 - adres (string) - adres zamieszkania (miejscowość, ulica, nr domu, nr mieszkania).
 - telefon (string) - numer telefonu kontaktowego.
 - data_urodzenia (date) - data urodzenia czytelnika.
 - płeć (string) - płeć czytelnika.
 - Relacje jednokierunkowe:
 - „WYPOŻYCZA” (relacja z węzłem „Egzemplarz”). Wskazuje, które egzemplarze zostały wypożyczone przez czytelnika
 - Atrybuty:
 - data_wypożyczenia(date)-data wypożyczenia egzemplarza
 - data_przewidywanego_zwrotu(date)-planowana data zwrotu
 - data_zwrotu(date)-faktyczna data zwrotu
 - status(string) – np. "wypożyczony", "zwrócony", "po terminie"
 - „REZERWUJE” (relacja z węzłem „Pozycja”) - łączy czytelnika z pozycją, którą rezerwuje
 - data_rezerwacji(date)

- status(string) – np. "oczekująca", "anulowana", "zrealizowana"
 - data_wygasniecia(date) – jest wyliczane automatycznie 7 dni od data_rezerwacji
- „WYSTAWIA_RECENZJE” (relacja z węzłem „Pozycja”) - czytelnik wystawia recenzję do danej pozycji
 - Atrybuty:
 - data_recenzji(date)
 - ocena(integer) – np. w skali 1-5
 - komentarz(string) – tekst recenzji
- “POSIADA_KARE” (relacja z węzłem „Egzemplarz”) - czytelnik ma karę związaną z danym egzemplarzem
 - Atrybuty:
 - data_nalozenia(date) – data nałożenia kary
 - kwota(float) – wysokość kary (np. w złotych)
 - opis(string) – powód nałożenia kary (np. "Opóźniony zwrot")
 - czy_oplacona(boolean) – status opłacenia kary
- **Węzeł „Egzemplarz”:**
 - Atrybuty:
 - id (string) – unikalny identyfikator egzemplarza
 - numer_egzemplarza (string) – numer konkretnej kopii (np. „1”, „2”, „A23”)
 - status (string) – dostępny, wypożyczony, zarezerwowany itd.
 - Relacje jednokierunkowe:
 - „ZNAJDUJE_SIE_W” (relacja z węzłem „Filia”). Egzemplarz znajduje się w danej filii bibliotecznej.
- **Węzeł „Filia”:**
 - Atrybuty:
 - id (string) - unikalny identyfikator filii
 - nazwa (string) - nazwa danej filii
 - adres (string) - adres gdzie znajduje się dana filia

3. Zarządzanie danymi o pracownikach i ich aktywnościach

Opis funkcjonalności:

System wspiera zarządzanie pracownikami, ich pensjami, urlopami oraz zamawianiem towarów.

Powiązania w grafie:

- **Węzeł „Pracownik”:**
 - Atrybuty:
 - id(string) – unikalny identyfikator pracownika.
 - imie(string) – imię pracownika.
 - nazwisko(string) – nazwisko pracownika.
 - stanowisko(string) – np. "bibliotekarz", "manager".
 - data_zatrudnienia(date) – data rozpoczęcia pracy.
 - email(string) – adres e-mail pracownika.
 - telefon(string) – numer kontaktowy pracownika.
 - Relacje:
 - „MA_URLOP” (relacja z węzłem „Urlop”).
 - Atrybuty:
 - data_przypisania (date) – data, kiedy urlop został przydzielony
 - „OTRZYMUJE” (relacja z węzłem „Pensja”).
 - „ZAMAWIA” (relacja z węzłem „Towar”).
 - Atrybuty:
 - data_zamowienia(date) – data zamówienia.
 - ilosc(integer) – ilość zamówionych sztuk.
 - status(string) – np. "złożone", "zrealizowane".
 - „PRACUJE_W” (relacja z węzłem „Towar”).
 - Atrybuty:
 - data_rozpoczecia(date) – data rozpoczęcia pracy w danej filii.
 - etat(float) – wymiar etatu w filii.
- **Węzeł „Pensja”:**
 - Atrybuty:
 - id(string) – unikalny identyfikator pensji.

- Kwota_podstawowa(float) – stała kwota wynagrodzenia podstawowego.
 - premia(float) – kwota premii (opcjonalnie).
 - data_wypłaty(date) – data wypłaty wynagrodzenia.
 - kwota_calkowita(float) – kwota łączna (podstawowa + premia).
- **Węzeł „Urlop”:**
 - Atrybuty:
 - id(string) – unikalny identyfikator urlopu.
 - data_roz poczenia(date) – data rozpoczęcia urlopu.
 - data_zakonczenia(date) – data zakończenia urlopu.
 - typ(string) – typ urlopu (np. "macierzyński", "wypoczynkowy", "chorobowy").
 - status(string) – np. "zatwierdzony", "oczekujący".
- **Węzeł „Towar”:**
 - Atrybuty:
 - id(string) – unikalny identyfikator towaru.
 - nazwa(string) – nazwa towaru (np. "książka", "papier").
 - kategoria(string) – np. "biurowe", "techniczne".
 - ilosc(integer) – ilość dostępnych sztuk.
 - cena(float) – cena jednostkowa.
 - data_dostawy(date) – data dostarczenia towaru.
 - dostawca(string) – nazwa dostawcy.

4. Wizualizacja danych w panelu administratora

Opis funkcjonalności:

Administrator może korzystać z wizualizacji grafowych w panelu, co pozwala analizować powiązania między użytkownikami, pracownikami, książkami, rezerwacjami i filiami. Panel pokazuje również statystyki, takie jak:

- Najczęściej wypożyczane pozycje.
- Liczba aktywnych rezerwacji.
- Wartość kar za opóźnienia.

Podsumowanie

- System oparty na Neo4j pozwala na łatwe modelowanie skomplikowanych relacji między danymi, takich jak powiązania między użytkownikami, książkami, pracownikami czy filiami.
- Dzięki strukturze grafowej, możliwe jest szybkie przetwarzanie zapytań dotyczących zarówno danych szczegółowych, jak i statystyk globalnych.
- Wysoka elastyczność modelu umożliwia łatwą rozbudowę systemu w przyszłości, np. o dodatkowe funkcjonalności rekomendacji książek na podstawie historii wypożyczeń.

3. Prezentacja grafu

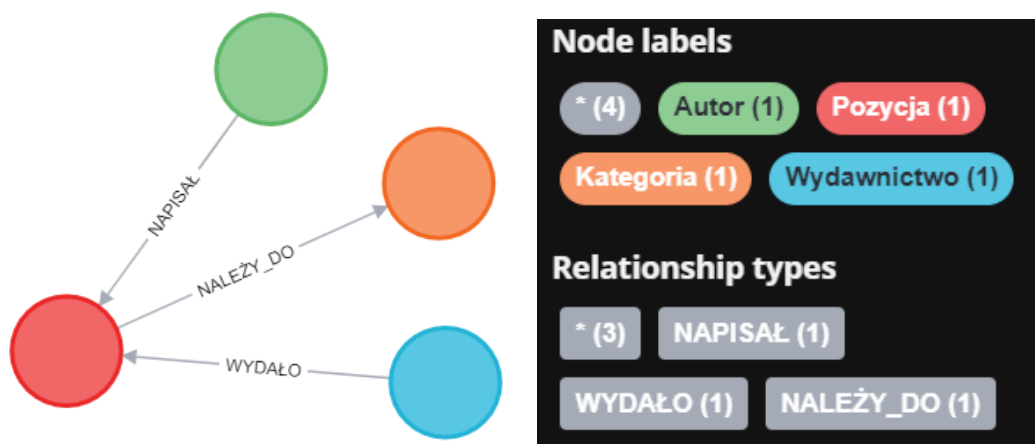
Opis struktury:

Strukturę grafu systemu bibliotecznego można podzielić na trzy kluczowe części, które reprezentują różne aspekty działania systemu:

1. **Książki i metadane:** Ta część obejmuje informacje o książkach, ich autorach, kategoriach oraz wydawnictwach. Relacje między tymi węzłami pozwalają na śledzenie, kto napisał książkę, do jakiej kategorii należy oraz które wydawnictwo jest odpowiedzialne za jej publikację. Dzięki temu możliwe jest zbudowanie spójnego modelu zarządzania treścią literacką.
2. **Czytelnicy i wypożyczenia:** Druga część skupia się na interakcji użytkowników z systemem. Reprezentuje węzły dotyczące czytelników, filii bibliecznych, egzemplarzy książek oraz relacje takie jak wypożyczenia, rezerwacje czy recenzje. Umożliwia to zarządzanie stanem egzemplarzy książek i rejestrowanie aktywności czytelników.
3. **Pracownicy i zarządzanie operacyjne:** Trzecia część modeluje aspekty zarządzania personelem i operacjami biblioteki. Zawiera węzły dla pracowników, ich pensji, urlopów oraz towarów związanych z działalnością biblioteki. Relacje w tej części opisują, gdzie pracownik jest zatrudniony, jakie ma obowiązki i w jaki sposób przyczynia się do funkcjonowania biblioteki.

Każda z tych części jest ze sobą powiązana, tworząc kompleksowy system umożliwiający efektywne zarządzanie biblioteką, obsługę użytkowników oraz śledzenie relacji między wszystkimi elementami.

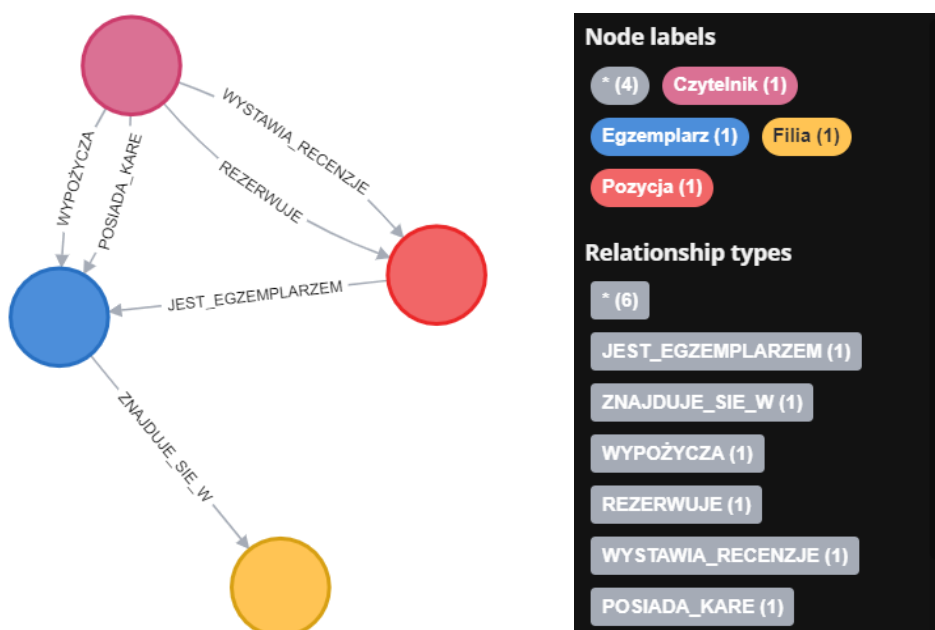
1. Pozycja, Autor, Kategoria, Wydawnictwo - wizualizacja relacji między książkami a metadanymi:



Rys. 1 Graf przedstawiający relację między książkami a metadanymi oraz opis elementów grafu

Kod odpowiedzialny za tworzenie węzłów i relacji związanych z książkami, autorami, kategoriami oraz wydawnictwami znajduje się w pliku czesc_1.cypher.

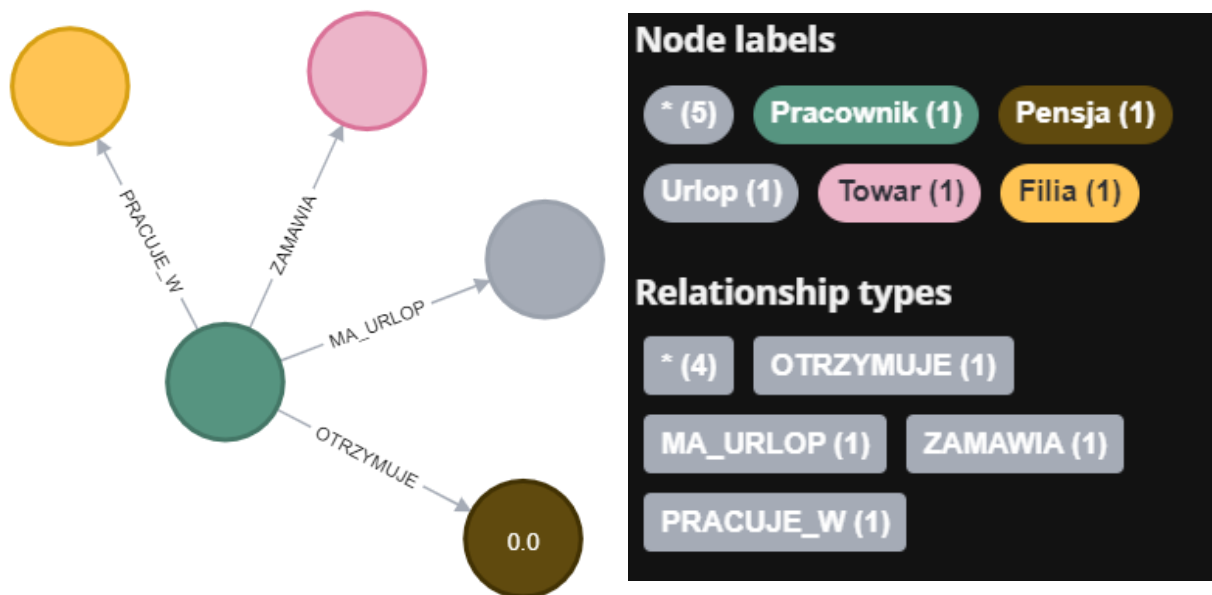
2. Pozycja, Czytelnik, Egzemplarz, Filia - wizualizacja wypożyczeń i egzemplarzy:



Rys 2 Graf przedstawiający relację wypożyczeń i egzemplarzy oraz opis elementów grafu

Kod modelujący relacje między czytelnikami, egzemplarzami książek, filią oraz aktywnościami użytkowników, takimi jak wypożyczenia i recenzje, został zapisany w pliku czesc_2.cypher.

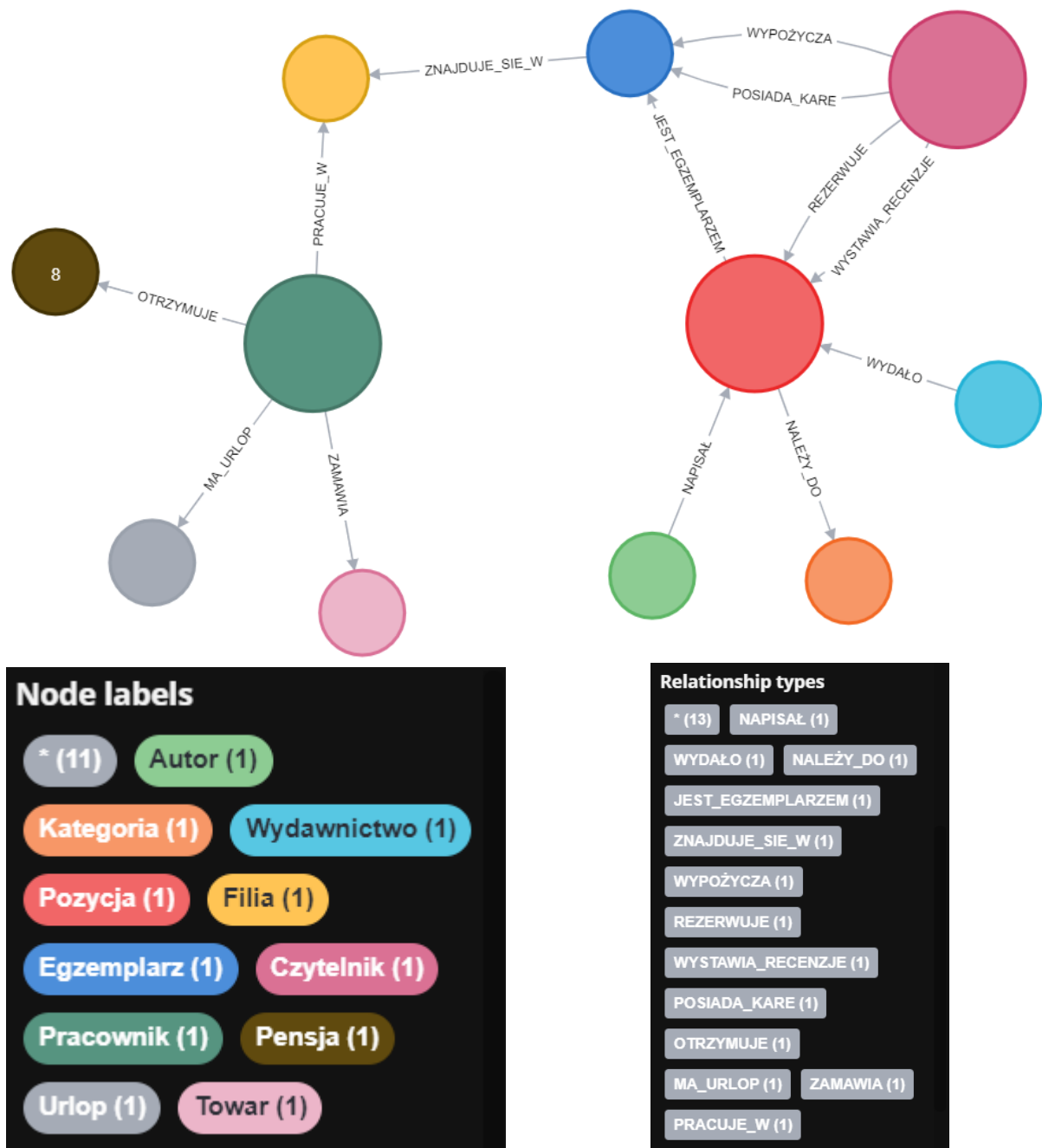
3. Filia, Pracownik, Pensja, Towar, Urlop - wizualizacja pracowników, ich wynagrodzeń, urlopów i zamówień:



Rys 3 Graf przedstawiający relację między pracownikami, urlopami i zamówieniami oraz opis elementów grafu

Kod definiujący węzły i relacje związane z pracownikami, wynagrodzeniami, urlopami oraz zarządzaniem towarami znajduje się w pliku `czesc_3.cypher`.

4. Struktura całego grafu:



Rysunek 4 Graf przedstawiający strukturę całej bazy oraz opis elementów grafu

Szkielet grafu definiujący wszystkie węzły i relacje znajduje się w pliku `calosc-szkielet.cypher`

Graf z wypełnionymi danymi znajduje się w pliku `biblioteka.cypher`

4. Zapytania w systemie bibliotecznym

Wykorzystanie Neo4j pozwala na efektywne zarządzanie danymi w sposób intuicyjny i wydajny dzięki grafowej strukturze bazy danych. Zapytania Cypher umożliwiają przeszukiwanie relacji między węzłami, takimi jak „Czytelnik”, „Egzemplarz” czy „Filia”, w celu realizacji kluczowych funkcji systemu. Przykładowo, system pozwala szybko sprawdzić dostępność egzemplarzy w wybranej filii, śledzić historię wypożyczeń czy naliczać kary za nieterminowe zwroty. Ponadto, możliwe jest monitorowanie aktywności pracowników oraz zarządzanie zamówieniami towarów, co czyni system wszechstronnym narzędziem wspierającym funkcjonowanie nowoczesnej biblioteki.

Zapytania pobierające:

1. Znajdź wszystkie książki napisane przez Stephena Kinga:

```
1 MATCH (a:Autor {nazwisko: "King"})-[:NAPISAŁ]→(p:Pozycja)
2 RETURN p.tytuł, p.typ, p.rok_wydania;
```

p.tytuł	p.typ	p.rok_wydania
"To"	"książka"	1986
"To"	"audiobook"	2017

Rys 5 Wynik zapytania wyszukującego książki napisane przez Stephena Kinga

2. Znajdź wszystkie dostępne egzemplarze książki "To" w filii głównej:

```
1 MATCH (p:Pozycja {tytuł: "To", typ: "książka"})-[:JEST_EGZEMPLARZEM]→(e:Egzemplarz {status:
"dostępny"}),(e)-[:ZNAJDUJE_SIE_W]→(f:Filia {nazwa: "Biblioteka Główna"})
2 RETURN e.numer_egzemplarza;
```

e.numer_egzemplarza
"TO-001"

Rys 6 Wynik zapytania wyszukującego dostępne egzemplarze książki „To”

3. Znajdź wszystkich czytelników, którzy mają aktualnie wypożyczone książki:

```
1 MATCH (c:Czytelnik)-[w:WYPOŻYCZA {status: "wypożyczony"}]→(e:Egzemplarz),
2 (e)←[:JEST_EGZEMPLARZEM]-(p:Pozycja)
3 RETURN c.imie, c.nazwisko, p.tytuł, w.data_wypożyczenia, w.data_przewidywanego_zwrotu;
```

c.imie	c.nazwisko	p.tytuł	w.data_wypożyczenia	w.data_przewidywanego_zwrotu
"Jan"	"Kowalski"	"To"	"2023-05-10"	"2023-06-10"
"Anna"	"Nowak"	"Harry Potter i Kamień Filozoficzny"	"2023-06-01"	"2023-07-01"

Rys 7 Wynik zapytania wyszukującego czytelników, którzy mają aktualnie wypożyczone książki

4. Znajdź wszystkie rezerwacje, które wygasają w ciągu najbliższych 3 dni:

```
1 MATCH (c:Czytelnik)-[r:REZERWUJE {status: "oczekująca"}]→(p:Pozycja)
2 WHERE r.data_wygasnięcia ≤ date() + duration({days: 3})
3 RETURN c.imie, c.nazwisko, p.tytuł, r.data_wygasnięcia;
```

c.imie	c.nazwisko	p.tytuł	r.data_wygasnięcia
"Jan"	"Kowalski"	"Księgi Jakubowe"	"2023-06-12"

Rys 8 Wynik zapytania wyszukującego rezerwacje, które wygasają w ciągu najbliższych 3 dni

5. Znajdź wszystkich autorów, których książki należą do kategorii "fantastyka":

```
1 MATCH (a:Autor)-[:NAPISAŁ]→(p:Pozycja)-[:NALEŻY_DO]→(k:Kategoria {nazwa: "fantastyka"})
2 RETURN DISTINCT a.imie, a.nazwisko;
```

a.imie	a.nazwisko
"Andrzej"	"Sapkowski"
"J.K."	"Rowling"

Rys 9 Wynik zapytania wyszukującego autorów, których książki należą do kategorii „fantastyka”

6. Znajdź pracowników pracujących w filii głównej na pełen etat:

```
1 MATCH (p:Pracownik)-[pw:PRACUJE_W {etat: 1.0}]->(f:Filia {nazwa: "Biblioteka Główna"})
2 RETURN p.imie, p.nazwisko, p.stanowisko, pw.data_rozpoczecia;
```

p.imie	p.nazwisko	p.stanowisko	pw.data_rozpoczecia
"Maria"	"Zielińska"	"bibliotekarz"	"2015-03-10"
"Tomasz"	"Wójcik"	"manager"	"2010-07-15"

Rys 10 Wynik zapytania wyszukującego pracowników pracujących w filii głównej na pełen etat

7. Znajdź wszystkie audiobooki w bibliotece:

```
1 MATCH (p:Pozycja {typ: "audiobook"})
2 RETURN p.tytuł, p.narrator, p.czas_trwania;
```

p.tytuł	p.narrator	p.czas_trwania
"To"	"Janusz Zadura"	1440

Rys 11 Wynik zapytania wyszukującego audiobooki w bibliotece

8. Znajdź wszystkie książki wydane przez wydawnictwo "Albatros":

```
1 MATCH (w:Wydawnictwo {nazwa: "Albatros"})-[:WYDAŁO]->(p:Pozycja)
2 RETURN p.tytuł, p.typ, p.rok_wydania;
```

p.tytuł	p.typ	p.rok_wydania
"To"	"książka"	1986
"Księgi Jakubowe"	"książka"	2014
"To"	"audiobook"	2017

Rys 12 Wynik zapytania wyszukującego książki wydane przez wydawnictwo „Albatros”

9. Znajdź czytelników z nieopłaconymi karami:

```
1 MATCH (c:Czytelnik)-[k:POSIADA_KARE {czy_oplacona: false}]->(e:Egzemplarz),
2 (e)-[:JEST_EGZEMPLARZEM]-(p:Pozycja)
3 RETURN c.imie, c.nazwisko, p.tytul, k.kwota, k.opis;
```

c.imie	c.nazwisko	p.tytul	k.kwota	k.opis
"Piotr"	"Wiśniewski"	"Wiedźmin: Ostatnie życzenie"	15.5	"Opóźniony zwrot o 5 dni"

Rys 13 Wynik zapytania wyszukującego czytelników z nieopłaconymi karami

10. Znajdź wszystkie recenzje dla książki "Wiedźmin: Ostatnie życzenie" z oceną 4 lub wyższą:

```
1 MATCH (c:Czytelnik)-[r:WYSTAWIA_RECENZJE]->(p:Pozycja {tytuł: "Wiedźmin: Ostatnie życzenie"})
2 WHERE r.ocena ≥ 4
3 RETURN c.imie, c.nazwisko, r.ocena, r.komentarz, r.data_recenzji;
```

c.imie	c.nazwisko	r.ocena	r.komentarz	r.data_recenzji
"Anna"	"Nowak"	5	"Fantastyczna książka, polecam każdemu miłośnikowi fantasy!"	"2023-06-13"

Rys 14 Wynik zapytania wyszukującego recenzje dla książki z oceną 4 lub wyższą

11. Znajdź pracowników na urlopie w lipcu 2023:

```
1 MATCH (p:Pracownik)-[:MA_URLOP]->(u:Urlop)
2 WHERE u.data_rozpoczecia ≤ date("2023-07-31") AND u.data_zakonczenia ≥ date("2023-07-01")
3 RETURN p.imie, p.nazwisko, u.data_rozpoczecia, u.data_zakonczenia, u.typ;
```

p.imie	p.nazwisko	u.data_rozpoczecia	u.data_zakonczenia	u.typ
"Maria"	"Zielińska"	"2023-07-01"	"2023-07-14"	"wypoczynkowy"

Rys 15 Wynik zapytania wyszukującego pracowników na urlopie w lipcu 2023

12. Znajdź wszystkie zamówienia towarów złożone w maju 2023:

```
1 MATCH (p:Pracownik)-[z:ZAMAWIA]-(t:Towar)
2 WHERE z.data_zamowienia >= date("2023-05-01") AND z.data_zamowienia <= date("2023-05-31")
3 RETURN p.imie, p.nazwisko, t.nazwa, z.ilosc, t.cena, z.data_zamowienia;
```

p.imie	p.nazwisko	t.nazwa	z.ilosc	t.cena	z.data_zamowienia
"Maria"	"Zielińska"	"długopisy"	50	1.99	"2023-05-25"
"Tomasz"	"Wójcik"	"papier ksero"	10	25.99	"2023-05-10"

Rys 16 Wynik zapytania wyszukującego zamówienia towarów złożone w maju 2023

13. Znajdź pracowników i ich filie pracy:

```
1 MATCH (p:Pracownik)-[pw:PRACUJE_W]-(f:Filia)
2 RETURN p.imie, p.nazwisko, p.stanowisko, f.nazwa, pw.etat
```

p.imie	p.nazwisko	p.stanowisko	f.nazwa	pw.etat
"Maria"	"Zielińska"	"bibliotekarz"	"Biblioteka Główna"	1.0
"Tomasz"	"Wójcik"	"manager"	"Biblioteka Główna"	1.0
"Agnieszka"	"Lewandowska"	"bibliotekarz"	"Filia Nr 1"	0.5
"Agnieszka"	"Lewandowska"	"bibliotekarz"	"Filia Nr 2"	0.5

Rys 17 Wynik zapytania wyszukującego pracowników i filie, w których pracują

14. Znajdź wszystkie pozycje w kategorii "horror" lub "thriller":

```

1 MATCH (p:Pozycja)-[:NALEŻY_DO]→(k:Kategoria)
2 WHERE k.nazwa IN ["horror", "thriller"]
3 RETURN p.tytuł, p.typ, collect(k.nazwa) as kategorie;

```

p.tytuł	p.typ	kategorie
"To"	"książka"	["horror", "thriller"]
"To"	"audiobook"	["horror", "thriller"]

Rys 18 Wynik zapytania wyszukującego pozycje w kategorii „horror” i „thriller”

15. Znajdź pracowników z pensją podstawową powyżej 5000 zł:

```

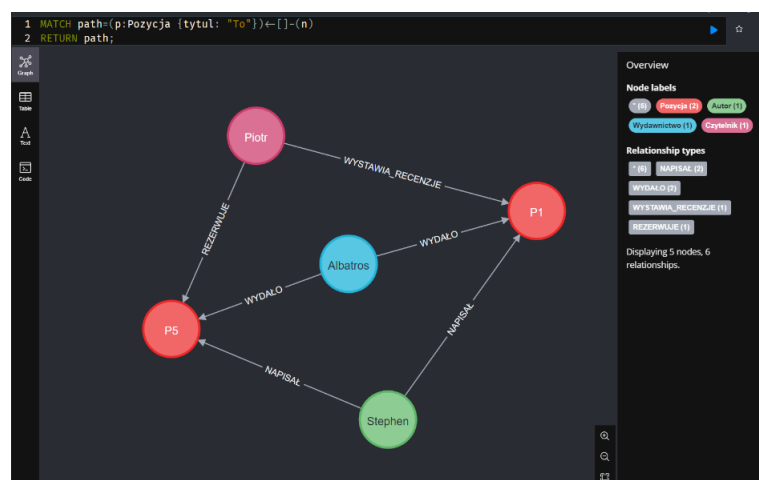
1 MATCH (p:Pracownik)-[:OTRZYMUJE]→(pe:Pensja)
2 WHERE pe.Kwota_podstawowa > 5000
3 RETURN p.imię, p.nazwisko, p.stanowisko, pe.Kwota_podstawowa, pe.kwota_calkowita;

```

p.imię	p.nazwisko	p.stanowisko	pe.Kwota_podstawowa	pe.kwota_calkowita
"Tomasz"	"Wójcik"	"manager"	6000.0	7000.0

Rys 19 Wynik zapytania wyszukującego pracowników z pensją podstawową powyżej 5000 zł

16. Znajdź wszystkie relacje między węzłami dla konkretnej pozycji (np. "To"):

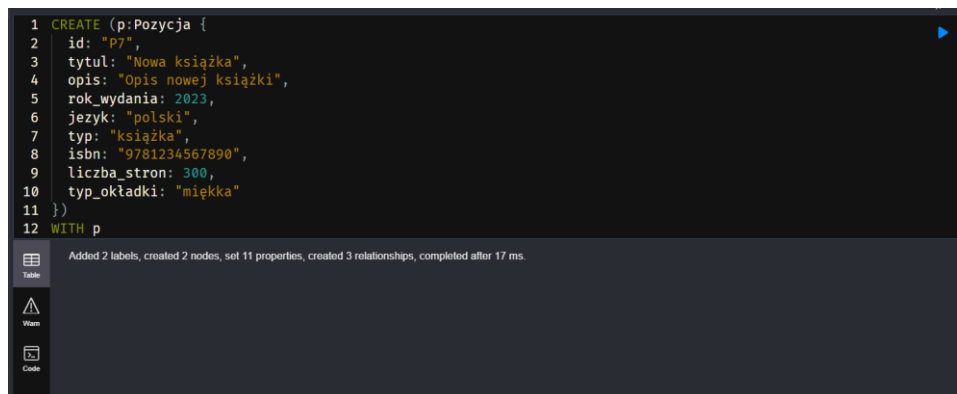


Rys 20 Wynik zapytania wyszukującego wszystkie relacje dla konkretnej pozycji

Zapytania edytujące:

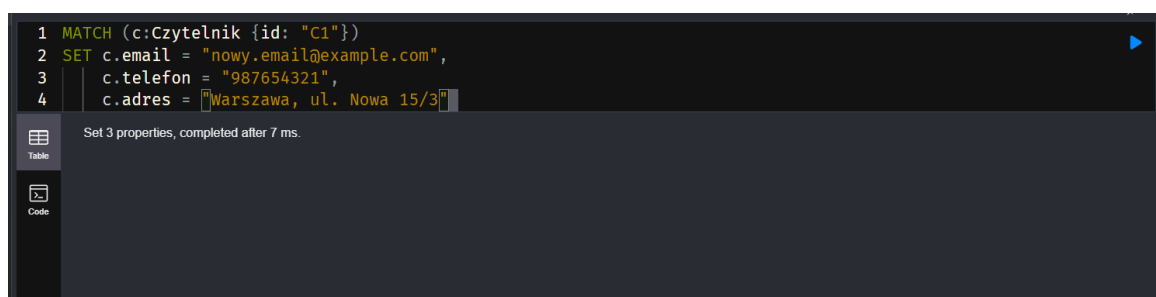
1. Dodanie nowej pozycji książkowej i powiązanie z autorem i wydawnictwem:

```
1 CREATE (p:Pozycja {
2   id: "p7",
3   tytuł: "Nowa książka",
4   opis: "Opis nowej książki",
5   rok_wydania: 2023,
6   język: "polski",
7   typ: "książka",
8   isbn: "9781234567890",
9   liczba_stron: 300,
10  typ_okładki: "miękka"
11 })
12 WITH p
13 MATCH (a:Autor {nazwisko: "Sapkowski"}), (w:Wydawnictwo {nazwa: "SuperNOWA"})
14 CREATE (a)-[:NAPISAŁ]->(p),
15        (w)-[:WYDAŁO]->(p),
16        (p)-[:NALEŻY_DO]->(:Kategoria {id: "K6", nazwa: "nowości"})
```



Rys 21 Dodanie nowej pozycji książkowej i powiązanie z autorem i wydawnictwem

2. Aktualizacja danych czytelnika:



Rys 22 Aktualizacja danych czytelnika

3. Dodanie nowego egzemplarza istniejącej pozycji:

```
1 MATCH (p:Pozycja {tytuł: "Nowa książka"})
2 CREATE (e:Egzemplarz {
3   id: "E11",
4   numer_egzemplarza: "NK-001",
5   status: "dostępny"
6 }),
7 (p)-[:JEST_EGZEMPLARZEM]-(e),
8 (e)-[:ZNAJDUJE_SIE_W]-(f:Filia {id: "F4", nazwa: "Nowa Filia", adres: "Warszawa, ul. Testowa 1"})
```

Added 2 labels, created 2 nodes, set 6 properties, created 2 relationships, completed after 9 ms.

Rys 23 Dodanie nowego egzemplarza istniejącej pozycji

4. Zmiana statusu wypożyczenia na "zwrócony":

```
1 MATCH (c:Czytelnik {nazwisko: "Kowalski"})-[w:WYPOŻYCZA]-(e:Egzemplarz {numer_egzemplarza: "TO-001"})
2 SET w.status = "zwrócony",
3   w.data_zwrotu = date(),
4   e.status = "dostępny"
```

Set 3 properties, completed after 7 ms.

Rys 24 Zmiana statusu wypożyczenia na "zwrócony"

5. Dodanie nowej rezerwacji:

```
1 MATCH (c:Czytelnik {nazwisko: "Nowak"}, (p:Pozycja {tytuł: "Nowa książka"})
2 CREATE (c)-[:REZERWUJE {
3   data_rezerwacji: date(),
4   status: "oczekująca",
5   data_wygasnięcia: date() + duration({days: 7})
6 }]->(p)
```

Set 3 properties, created 1 relationship, completed after 11 ms.

Rys 25 Dodanie nowej rezerwacji

6. Usunięcie nieaktywnych rezerwacji (które wygasły):

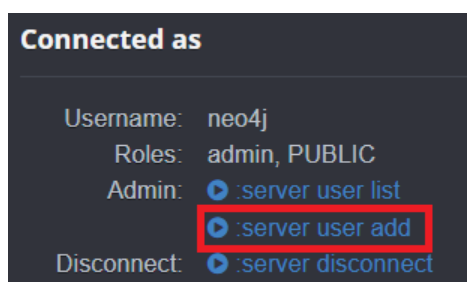
```
1 MATCH (c:Czytelnik)-[r:REZERWUJE]->(p:Pozycja)
2 WHERE r.data_wygasniecia < date() AND r.status = "oczekująca"
3 DELETE r
```

Deleted 1 relationship, completed after 8 ms.

Rys 26 Usunięcie nieaktywnych rezerwacji (które wygasły)

5. Zarządzanie użytkownikami

Użytkownika można utworzyć poprzez interfejs graficzny. Tworzy się go po podłączeniu się do serwera. Możemy go stworzyć za pomocą narzędzia Neo4j Browser. W tym celu należy w sekcji *Database Information* wybrać *Connected as -> server user add* i przyporządkować mu rolę



Rys 27 Dodanie nowego użytkownika

Użytkownik z username: user1 oraz rolą reader

Add user
Add a user to the current database

Username:

Password: Confirm password:

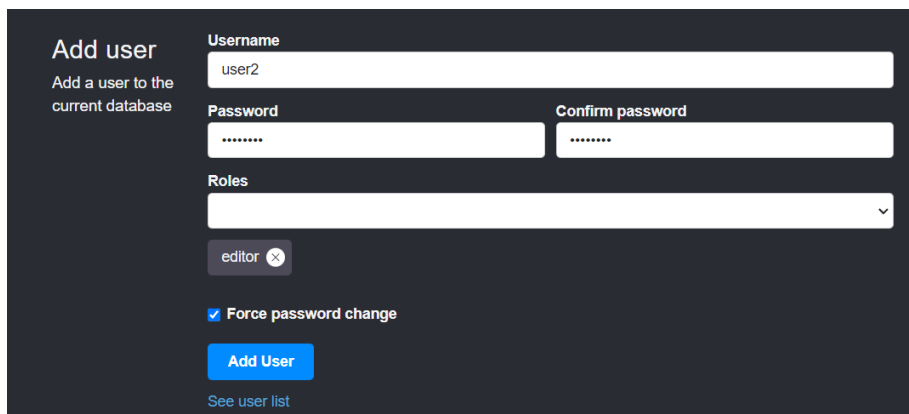
Roles:

☒ Force password change

[See user list](#)

Rys 28 Utworzenie nowego użytkownika z rolą reader

Użytkownik z username: user2 oraz rolą editor



Add user
Add a user to the current database

Username: user2

Password: Confirm password:

Roles: editor

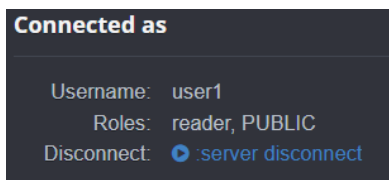
☒ Force password change

Add User

[See user list](#)

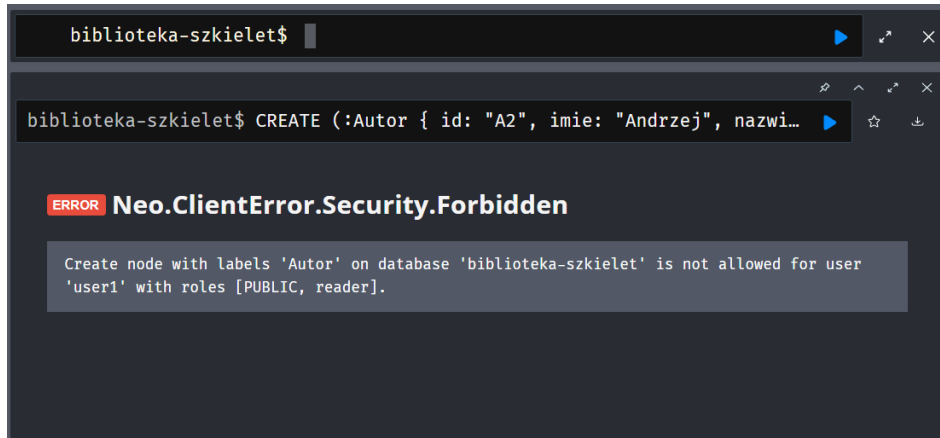
Rys 29 Utworzenie nowego użytkownika z rolą editor

Połączenie się do serwera jako użytkownik user1.



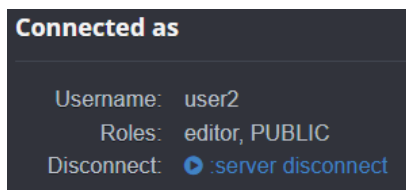
Rys 30 Połączenie z serwerem użytkownika user1

Błąd przy próbie dodania danych ponieważ użytkownik ma tylko rolę reader.



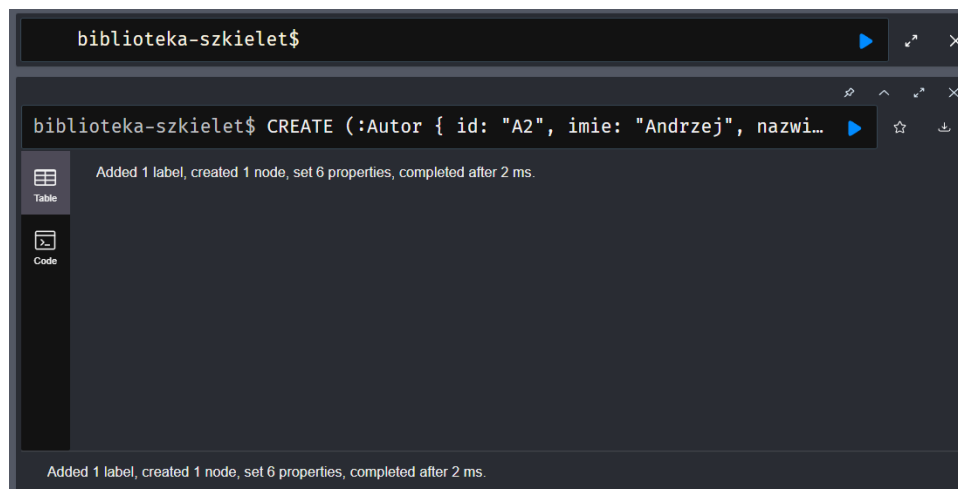
Rys 31 Błąd przy próbie dodania danych (użytkownik ma rolę reader)

Połączenie się do serwera jako użytkownik user2.



Rys 32 Połączenie z serwerem użytkownika user2

Sukces przy próbie dodanie danych do węzła autor ponieważ użytkownik posiada rolę editor.



Rys 33 Sukces przy próbie dodania danych (użytkownik ma rolę editor)

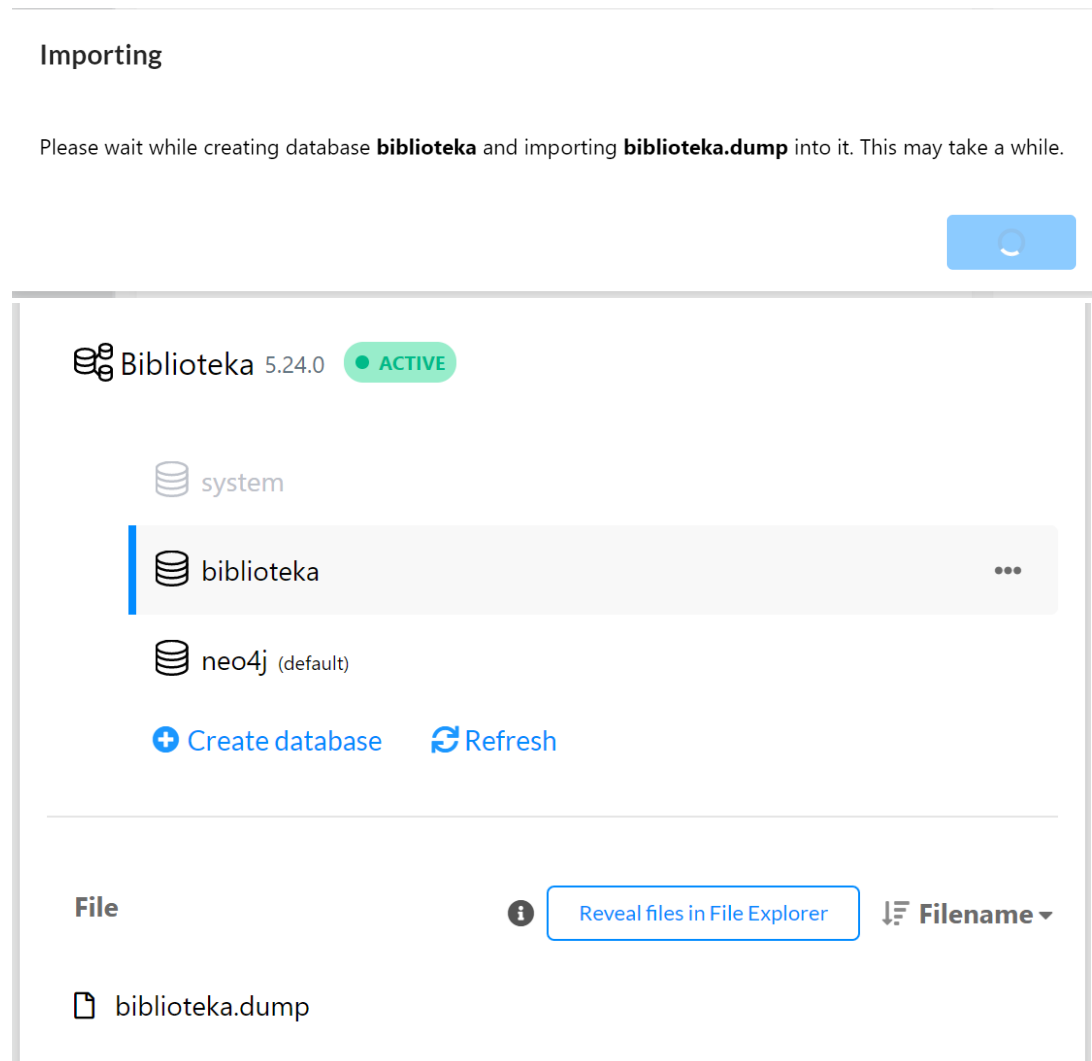
6. Kopia zapasowa, import i eksport

Kopie bazy danych biblioteka trzeba utworzyć, kiedy nie jesteśmy połączeni z serwerem. W terminalu podajemy polecenie **dump**, które pozwala utworzyć nam kopię bazy i zapisać ją w podanych miejscu poprzez podanie ścieżki do odpowiedniego folderu.

```
C:\Users\Dell\Neo4jDesktop\relate-data\dbmss\dbms-7f55ad3e-3026-45c9-9c9a-4cfb29962133>.\bin\neo4j-admin database dump biblioteka --to-path="C:\Users\Dell\Desktop\biblioteka_neo4j"
2025-05-28 06:27:44.340+0000 INFO [c.n.c.d.EnterpriseDumpCommand] Starting dump of database 'biblioteka'
Done: 54 files, 970.5KiB processed.
2025-05-28 06:27:45.772+0000 INFO [c.n.c.d.EnterpriseDumpCommand] Dump completed successfully
```

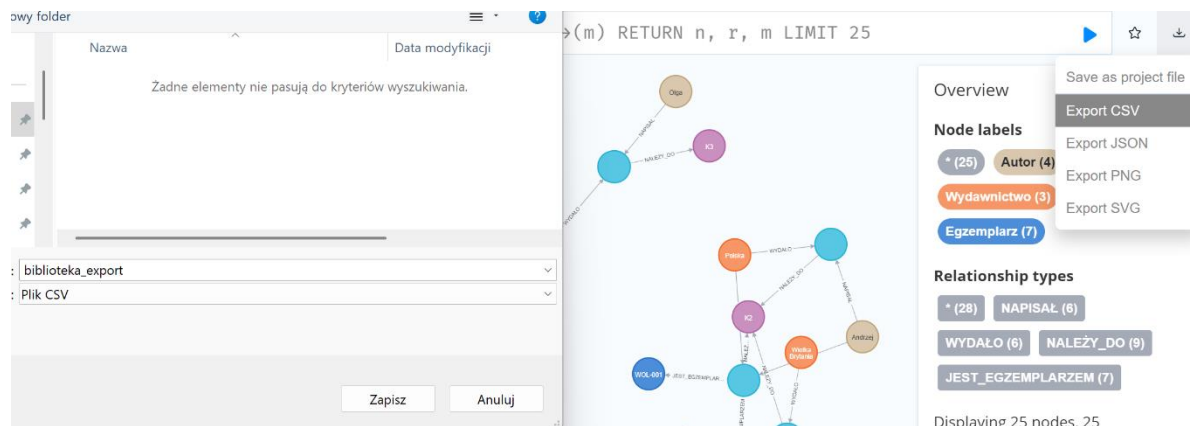
Rys 34 Utworzenie kopii zapasowej bazy biblioteka

Import bazy do Neo4j robimy za pomocą przycisku *Add* i wybieramy opcję *File*. Potem możemy zdecydować, czy tworzymy nowy DBMS (*Create new DBMS from dump*) lub importujemy do istniejącego (*Import dump into existing DBMS*), po wybraniu *Import dump...* musimy wskazać, w którym DBMS ma zostać zaimportowana baza.

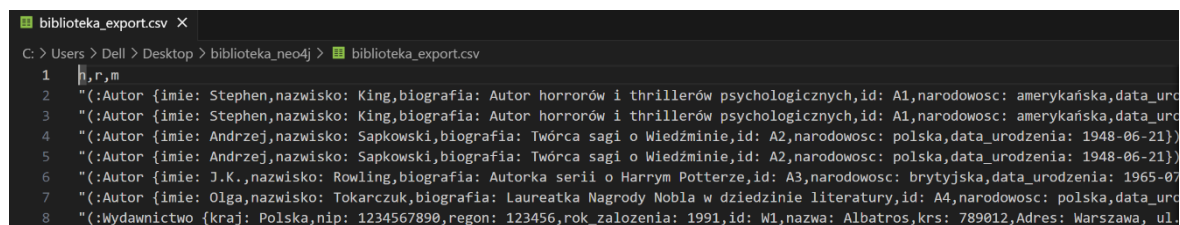


Rys 35 Import bazy kopii bazy biblioteka do Neo4j

Po wykonaniu zapytania mamy możliwość **eksportu danych**. Dane możemy wyeksportować jako rysunek lub plik z rozszerzeniem CSV lub JSON. Po wybraniu opcji wskazuje miejsce, gdzie plik ma zostać zapisany.



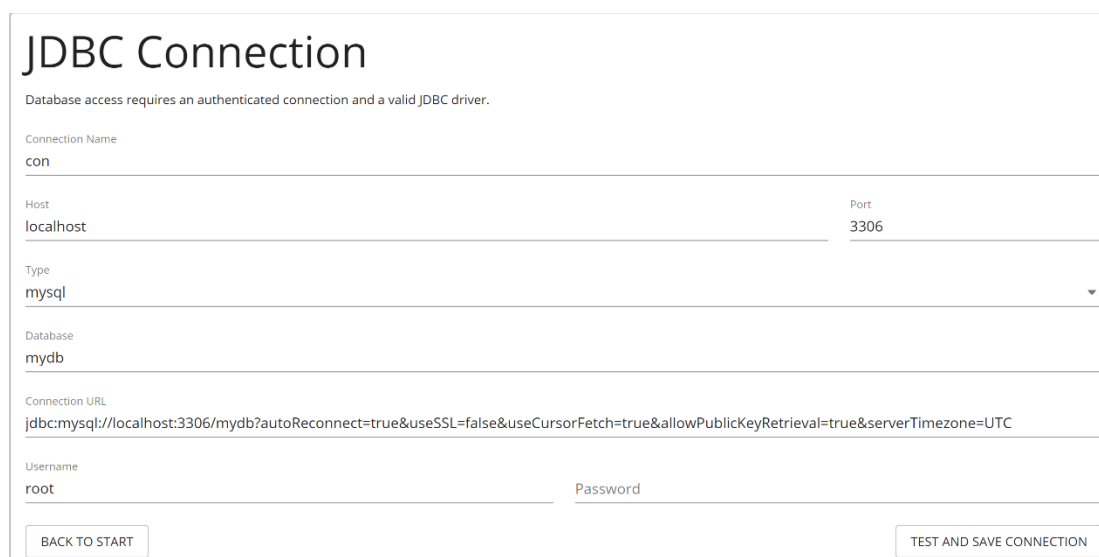
Rys 36 Eksport danych do pliku CSV



Rys 37 Plik CSV z wyeksportowanymi danymi

7. Mapowanie bazy relacyjnej na bazę grafową

Narzędzie *Neo4j ETL Tool* służy do stworzenia bazy grafowej na podstawie bazy relacyjnej. Narzędzie wybieramy z zakładki *Graph Apps*. Przed połączeniem z serwerem *mysql* na którym jest baza *mydb*, należy go uruchomić. Tworzymy połączenie (*JDBC Connection*), musimy podać nazwę połączenie, typ bazy, nazwę bazy, użytkownika i jego hasło. Do utworzonego URL trzeba dodać na końcu *&serverTimezone=UTC* inaczej mogą wystąpić błędy w połączeniu. Potem wybieramy *Test and save connection*.

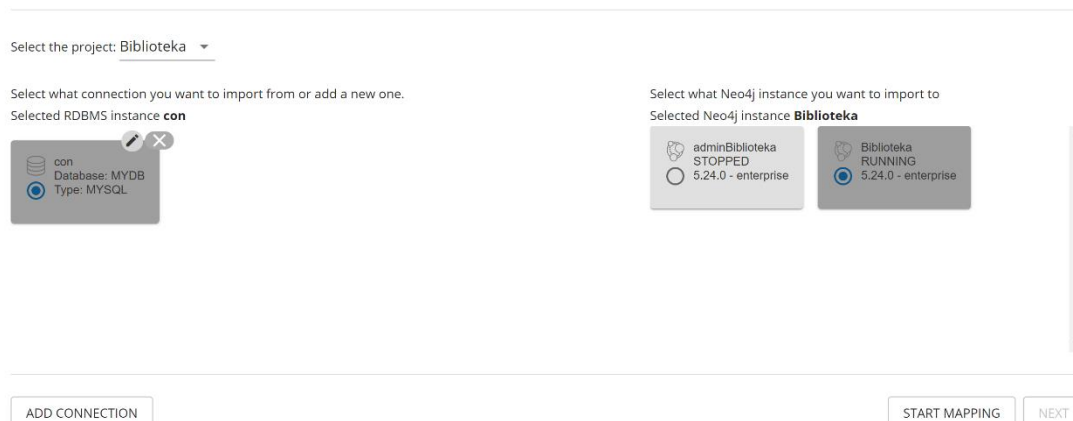


The screenshot shows the 'JDBC Connection' configuration window. It includes fields for Connection Name (con), Host (localhost), Port (3306), Type (mysql), Database (mydb), Connection URL (jdbc:mysql://localhost:3306/mydb?autoReconnect=true&useSSL=false&useCursorFetch=true&allowPublicKeyRetrieval=true&serverTimezone=UTC), Username (root), and Password. At the bottom, there are two buttons: 'BACK TO START' and 'TEST AND SAVE CONNECTION'.

Rys 38 Utworzenie połączenia z MySQL

Po wybraniu połączenia i Grapt DBMS wybieramy opcję *Start mapping*.

Load Data Model



The screenshot shows the 'Load Data Model' configuration screen. It has two main sections: 'Select the project: Biblioteka' and 'Select what connection you want to import from or add a new one.' Under the first section, there is a dropdown menu. Under the second section, there is a list of connections, including 'con' (Database: MYDB, Type: MYSQL). On the right, there is a section 'Select what Neo4j instance you want to import to' with a list of instances, including 'adminBiblioteka STOPPED 5.24.0 - enterprise' and 'Biblioteka RUNNING 5.24.0 - enterprise'. At the bottom, there are three buttons: 'ADD CONNECTION', 'START MAPPING', and 'NEXT'.

Rys 39 Rozpoczęcie mapowania

Otrzymamy relacyjny schemat przekształcony na graf i klikamy *Save mapping*.

Explore and change your metadata

NODES

Search by entity name

Entity Name	Actions	Skip
Egzemplarz		<input type="checkbox"/>
Zamowienie		<input type="checkbox"/>
Ksiazka		<input type="checkbox"/>
Pensja		<input type="checkbox"/>
Kategorium		<input type="checkbox"/>

RELATIONSHIPS

Rys 40 Relacyjny schemat przekształcony na graf

Po pomyślnym zapisaniu wybieramy nazwę bazy, do której ma być zaimportowana baza i klikamy *import data*. Potem sprawdzamy, czy baza została zaimportowana.

The screenshot shows the Neo4j web interface. On the left, the 'Database Information' sidebar is visible, showing the database name 'biblioteka' and a list of node labels: '(245) Audiobook', 'Autor', 'Czytelnik', 'DaneLogowanie', 'Ebook', 'Egzemplarz', 'EgzemplarzWypożyczeniu', 'Filia', 'Fillum', 'Kategoria', 'Kategorium', 'Klasa', 'Lokalizacja', and 'Reprezentacja'. The main area displays a Cypher query: `biblioteka$ MATCH (n)-[r]→(m) RETURN n, r, m LIMIT 25`. The query results are shown as a graph visualization with nodes and relationships. On the right, the 'Overview' panel shows the distribution of node labels: '(25) Autor (4) Pozycja (6)', 'Wydawnictwo (3) Kategoria (5)', and 'Egzemplarz (7)'. Below this, the 'Relationship types' panel shows '(28) NAPISAŁ (6)'.

Rys 41 Wynik mapowania bazy relacyjnej