Wojskowa Akademia Techniczna
im. Jarosława Dąbrowskiego

Wydział Cybernetyki, kierunek informatyka - inżynieria systemów

Realizacja zadania laboratoryjnego w ramach przedmiotu:

Systemy Baz Danych

Temat laboratorium:

Grafowe Bazy Danych

Opracował: Radosław Relidzyński, **Grupa:** WCY23IX3S4

Spis treści

Wstęp teoretyczny

Baza danych – „uporządkowany zbiór danych określających wybrany fragment rzeczywistości lub problemu, które są przechowywane trwale w pamięci komputerowej do której może mieć dostęp wielu użytkowników w dowolnej chwili czasu.”

System zarządzania bazami danych – „zorganizowany zbiór narzędzi (programów komputerowych i bibliotek), które umożliwiają wykonanie podstawowych operacji na danych (CRUD) zawartych w jednej lub więcej bazach danych.”

System baz danych – jego definicja wyraża się wzorem:

$$SBD =< \{U, SO, DB, SZBD, P\}, R >$$

Gdzie:

$$U - zbiór\ urządzeń$$

$$SO - system\ operacyjny$$

$$BD - baza\ danych\ (schemat, stan, ścieżki\ dostępu)$$

$$SZBD - system\ zarządzania\ bazą\ danych$$

$$P - polecenia\ użytkownika$$

$$R - relacje\ między\ obiektami\ SBD\ a\ otoczeniem$$

[źródło: materiały z wykładu „Temporalne bazy danych” dr inż. Jarosława Koszeli]

Grafowa baza danych – „Bazy danych grafów to bazy danych NoSQL, które mogą przechowywać, mapować i odpytywać relacje między danymi. Elementy w bazie danych grafów mogą łączyć się ze sobą w każdy możliwy sposób.”
[źródło: https://appmaster.io/pl/blog/baza-danych-grafow-neo4j]

Środowisko

Opis środowiska

W ramach projektu rolę systemu zarządzania bazą danych będzie pełnić narzędzie Neo4j. Wykorzystane zostanie środowisko Neo4j Workspace.

Neo4j – „system zarządzania bazą danych grafowych (GDBMS) opracowany przez Neo4j, Inc. Elementy danych przechowywane przez Neo4j to węzły, krawędzie łączące je oraz atrybuty węzłów i krawędzi.”
[tłumaczone, źródło: https://neo4j.com/developer/neo4j-browser/]

Neo4j Browser – „narzędzie skierowane do deweloperów, które pozwala na wykonywanie zapytań Cypher i wizualizowanie wyników. Jest to domyślny interfejs deweloperski zarówno dla edycji Enterprise, jak i Community Neo4j. Jest dostępny od razu we wszystkich ofertach baz danych grafowych Neo4j, w tym Neo4j Server (edycje Community i Enterprise), Neo4j AuraDB (baza danych Neo4j jako usługa) oraz Neo4j Desktop (wszystkie wersje systemów operacyjnych).
[tłumaczone, źródło: https://neo4j.com/docs/browser-manual/current/]

Na dzień projektu aktualna wersja Neo4j to 5.20.0 z 23 maja 2024 r.

Instrukcja do przygotowania bazy danych

Wywołanie instrukcji w aplikacji przeglądarkowej odbywa się poprzez skopiowanie zawartość skryptów jako jedno duże zapytanie do konsoli w zakładce „Query”.

1. Tworzenie zawartości bazy danych, dostępne 2 opcje
    a. Uruchomienie wszystkiego na raz – skrypt „script.cypher”
    b. Uruchomienie elementowo:
        i. Stworzenie węzłów – skrypt „nodes.cypher”
        ii. Stworzenie indeksów – skrypt „indexes.cypher”
        iii. Stworzenie ograniczeń – skrypt „constrains.cypher”
        iv. Stworzenie relacji – skrypt „relations.cypher”
2. Wywoływanie zapytań, uruchamiane osobno zapytań ze skryptu „queries.cypher”, każde zapytanie oddzielone i oznaczone jest numerem z opisem.
3. (Po zakończeniu działania) usunięcie całej stworzonej zawartości – skrypt „delete.cypher”

Wykaz etykiet

Baza danych stworzona w ramach projektu pełni zadanie zbierania i zarządzania informacjami dotyczącymi wybranych lotnisk Europy oraz lotów między nimi. W ramach tego powołane są następujące etykiety:

- Samolot – informacje o samolocie
- Miejsce – informacje o miejscu siedzącym w samolocie
- Model – informacje o istniejących w bazie modelach samolotów
- Pracownik – Informacje o pracownikach
- Lotnisko – informacje o lotniskach

- Lot – informacje o locie na podstawie samolotu i lotnisk
- Klient – informacje o kliencie
- Zniżki – informacje o zniżkach klientów
- Bilet – informacje o bilecie na podstawie lotu, miejsca w samolocie oraz klienta

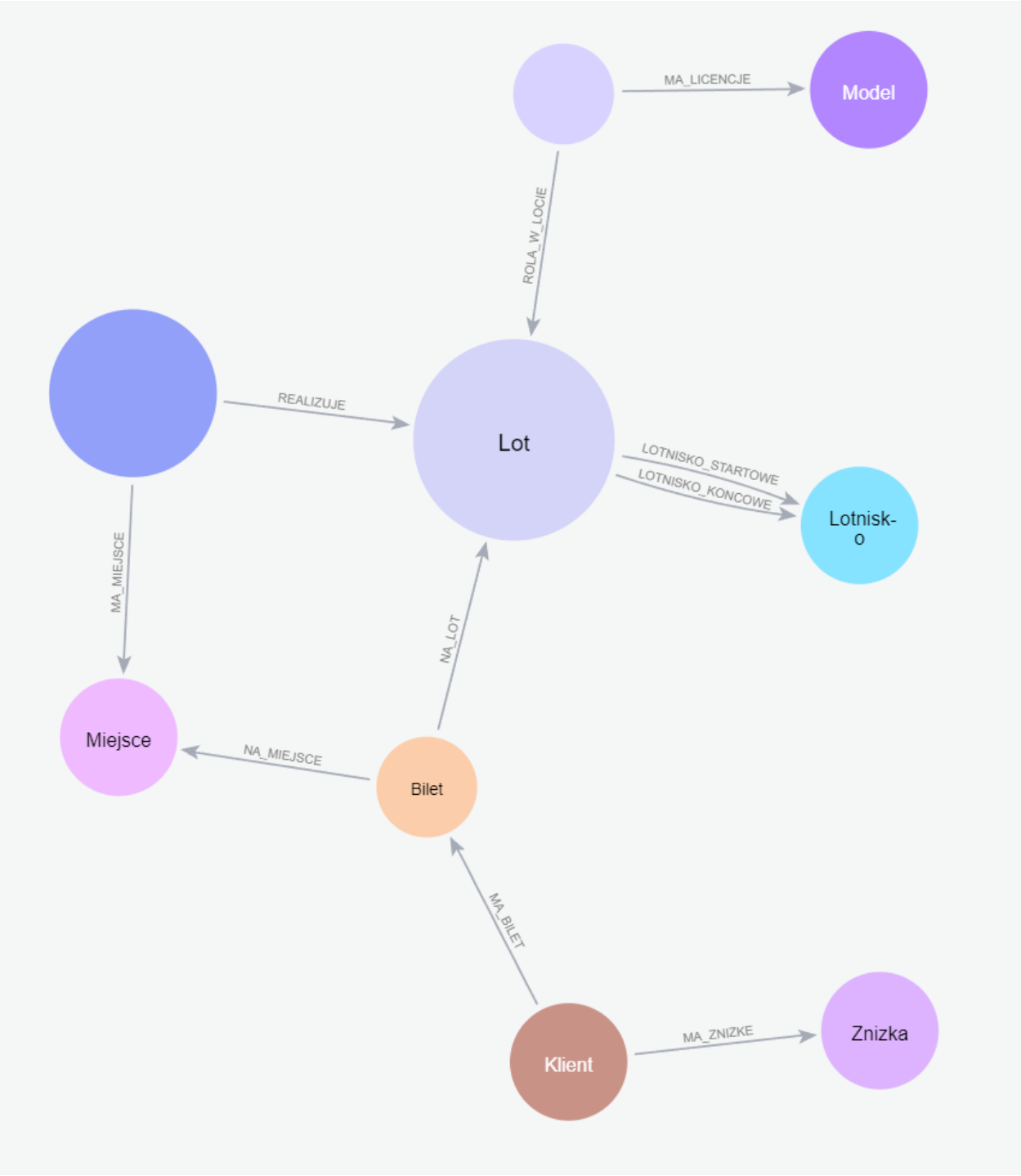
Wykaz relacji

Węzły w bazie danych połączone są relacjami pokazujące zależności między nimi oraz miejscami uzupełniające informacje o nich. W tym celu stworzone są następujące relacje:

- MA_LICENCJE – czy pracownik ma uprawnienia do pełnienia roli w ramach danego samolotu
- ROLA_W_LOCIE –jacy pracownicy biorą udział w locie oraz jaką rolę będą pełnić
- MA_ZNIZKE – jaki rodzaj zniżki (jeśli ma) posiada dany klient
- REALIZUJE – samolot realizujący dany lot
- LOTNISKO_STARTOWE – lotnisko startowe dla lotu
- LOTNISKO_KONCOWE – lotnisko końcowe dla lotu
- MA_MIEJSCE – przyporządkowanie miejsca do samolotu
- NA_LOT – lot, na który jest bilet
- NA_MIEJSCE – miejsce, na które jest bilet
- MA_BILET – do kogo należy bilet

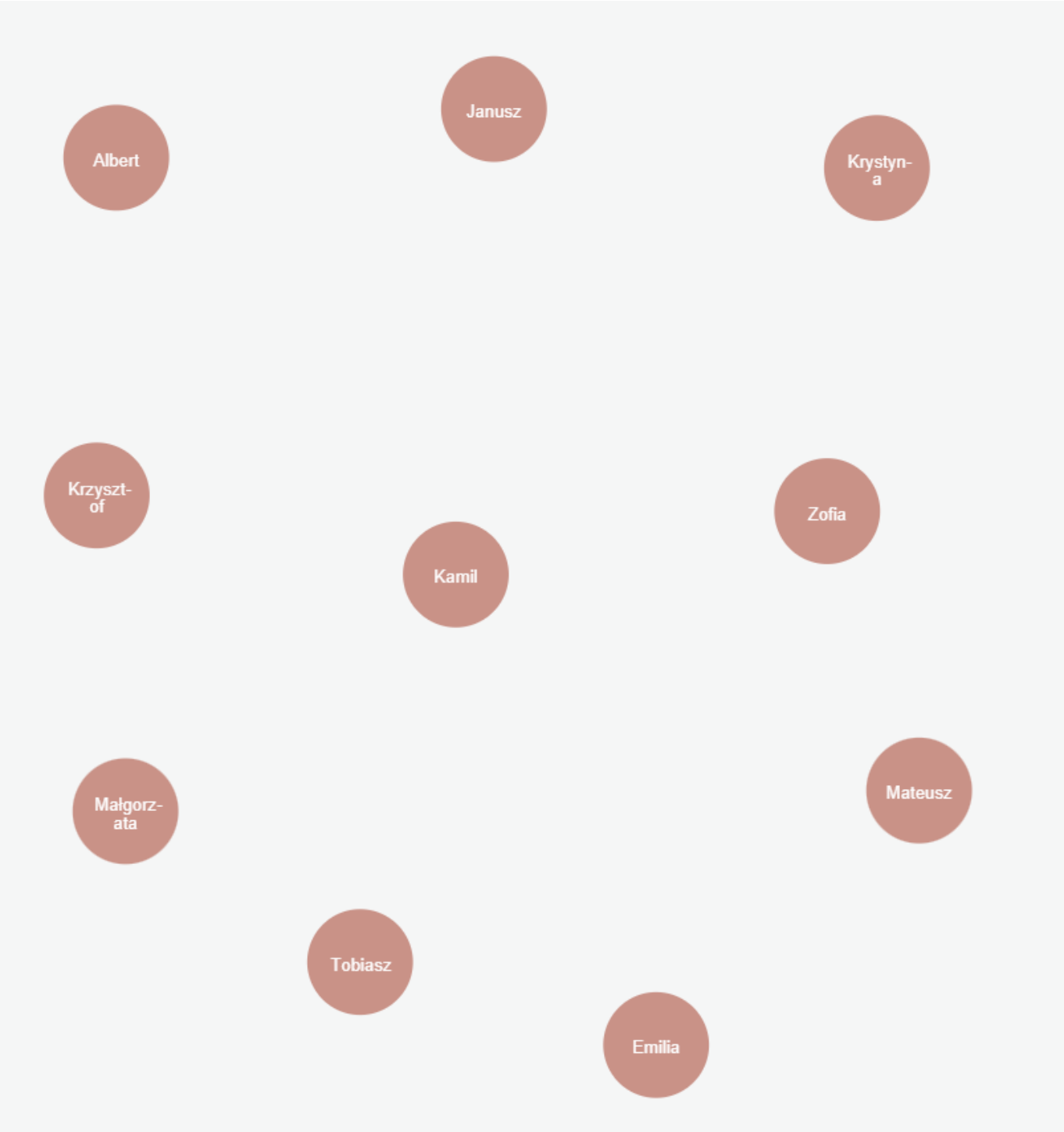
Schemat bazy danych

Wywołany przy pomocy polecenia „CALL db.schema.visualization”.

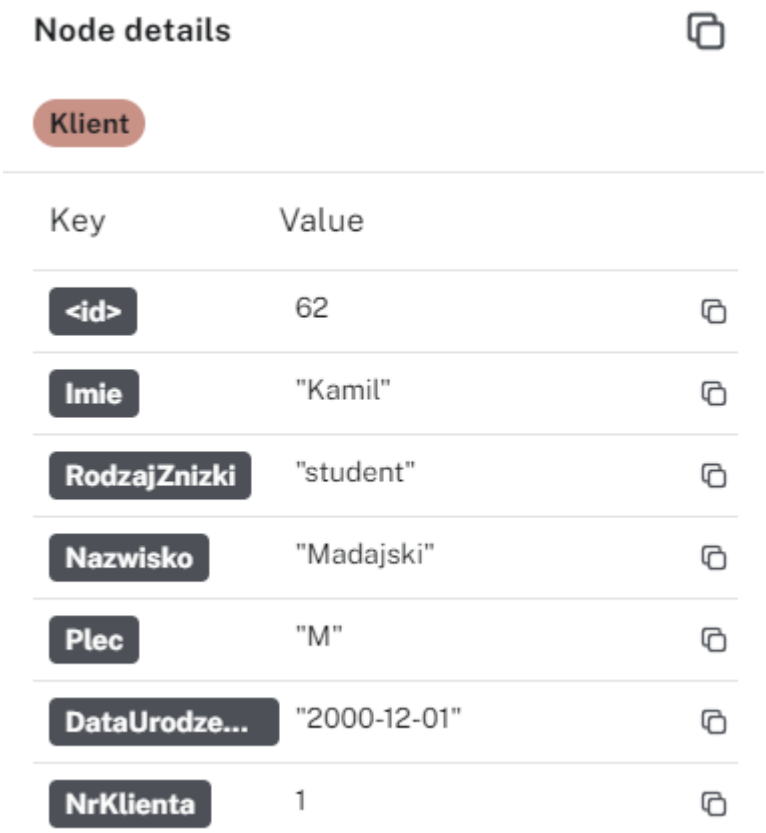


Węzły

Zbiór węzłów (etykieta) na przykładzie Klientów

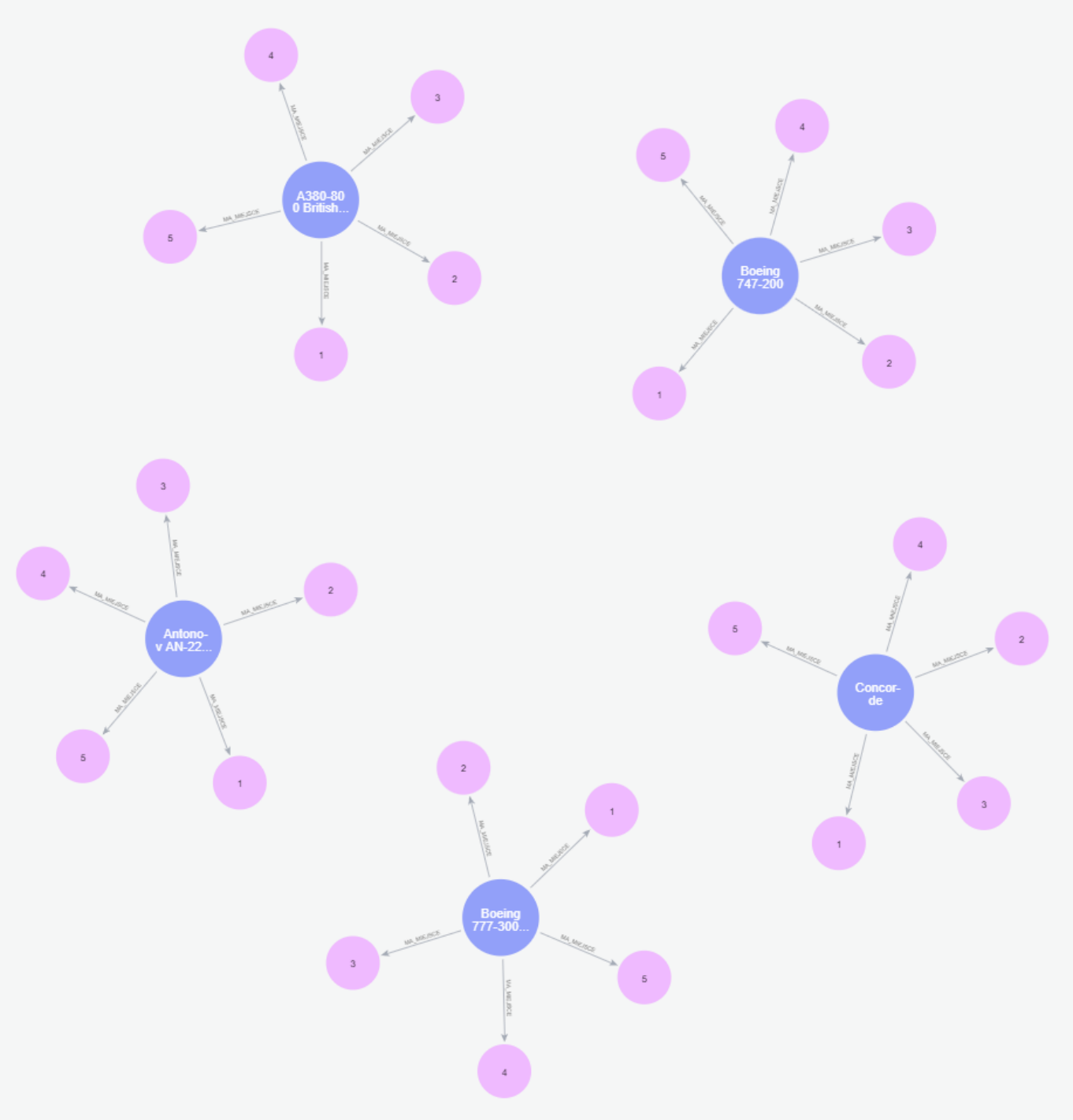


Podgląd atrybutów danego węzła



Relacje

Zbiór relacji na przykładzie relacji Samolot -> Miejsce o nazwie „MA_MIEJSCE”



Pogląd relacji na przykładzie relacji ROLA_W_LOCIE, która poza łączeniem pracownika z lotem mówi również jaką rolę pełni

Relationship details

ROLA_W_LOCIE

Key	Value
<id>	1152937997281263711
Rola	"steward"

Ograniczenia

Zastosowane ograniczenia, pokazane przy pomocy polecenia „SHOW CONSTRAINTS”

	id	name	type	entityType	labelsOrTypes	properties	ownedIndex	propertyType
1	30	"constraint_10(	"NODE_PROPERTY_	"NODE"	["Samolot"]	["NrSamolotu"]	null	null
2	59	"constraint_10{	"NODE_PROPERTY_	"NODE"	["Bilet"]	["Cena"]	null	"FLOAT"
3	35	"constraint_19(	"NODE_PROPERTY_	"NODE"	["Znizka"]	["RodzajZnizki"]	null	null
4	40	"constraint_1e4	"NODE_PROPERTY_	"NODE"	["Znizka"]	["ProcentUmazany"]	null	null
5	26	"constraint_21:	"NODE_PROPERTY_	"NODE"	["Klient"]	["Nazwisko"]	null	"STRING"
6	49	"constraint_26(	"NODE_PROPERTY_	"NODE"	["Klient"]	["DataUrodzenia"]	null	null
7	13	"constraint_2a3	"NODE_PROPERTY_	"NODE"	["Miejsce"]	["NrSamolotu"]	null	null
8	24	"constraint_2e3	"NODE_PROPERTY_	"NODE"	["Pracownik"]	["Imie"]	null	null
9	43	"constraint_357	"NODE_PROPERTY_	"NODE"	["Bilet"]	["KodBiletu"]	null	null
10	7	"constraint_425	"NODE_PROPERTY_	"NODE"	["Samolot"]	["RokProdukcji"]	null	"INTEGER"
11	36	"constraint_43(	"NODE_PROPERTY_	"NODE"	["Miejsce"]	["Klasa"]	null	"STRING"
12	15	"constraint_465	"NODE_PROPERTY_	"NODE"	["Lot"]	["DataLotu"]	null	null
13	14	"constraint_48(	"NODE_PROPERTY_	"NODE"	["Miejsce"]	["NrMiejsca"]	null	null
14	31	"constraint_52f	"NODE_PROPERTY_	"NODE"	["Lotnisko"]	["KrajPolozenia"]	null	null

Podział ograniczeń z przykładami:

- Wymóg bycia wypełnionym
  - CREATE CONSTRAINT FOR (n:Klient) REQUIRE (n.Imie) IS NOT NULL;
  - CREATE CONSTRAINT FOR (n:Klient) REQUIRE (n.Nazwisko) IS NOT NULL;
  - CREATE CONSTRAINT FOR (n:Klient) REQUIRE (n.DataUrodzenia) IS NOT NULL;
- Wymóg bycia danym typem danej
  - CREATE CONSTRAINT FOR (n:Bilet) REQUIRE (n.KodBiletu) IS TYPED STRING;
  - CREATE CONSTRAINT FOR (n:Cena) REQUIRE (n.KodBiletu) IS TYPED FLOAT;
  - CREATE CONSTRAINT FOR (p:Pracownik) REQUIRE (p.DataUrodzenia) IS TYPED DATE;

Indeksy

Możliwe do przejrzenia przy pomocy polecenia „SHOW INDEXES”.

	id	name	state	populatio	type	entityType	labelsOrTypes	properties	indexProvider	owningC	lastRead	readCount
1	0	"index_343aff4e	"ONLINE"	100.0	"LOOKUP"	"NODE"	null	null	"token-lookup-1.0"	null	2024-06-05T16:07:27.63E	2
2	1	"index_f770047	"ONLINE"	100.0	"LOOKUP"	"RELATIONS	null	null	"token-lookup-1.0"	null	2024-06-05T16:07:53.80E	2
3	2	"index_3db3386	"ONLINE"	100.0	"RANGE"	"NODE"	["Klient"]	["NrKlienta"]	"range-1.0"	null	2024-06-05T15:39:39.94E	30
4	3	"index_3555339	"ONLINE"	100.0	"RANGE"	"NODE"	["Lotnisko"]	["Nazwa"]	"range-1.0"	null	2024-06-05T15:39:24.92	14
5	4	"index_2b9ae72	"ONLINE"	100.0	"RANGE"	"NODE"	["Pracownik"]	["NrPracownika"]	"range-1.0"	null	2024-06-05T15:39:32.92	30
6	9	"index_884b013	"ONLINE"	100.0	"RANGE"	"NODE"	["Lot"]	["NrLotu"]	"range-1.0"	null	2024-06-05T15:39:39.94	67
7	17	"index_8b28ef8	"ONLINE"	100.0	"RANGE"	"NODE"	["Znizka"]	["RodzajZnizki"]	"range-1.0"	null	null	0
8	18	"index_4451d07	"ONLINE"	100.0	"RANGE"	"NODE"	["Model"]	["Nazwa"]	"range-1.0"	null	null	0
9	22	"index_d8bd57a	"ONLINE"	100.0	"RANGE"	"NODE"	["Samolot"]	["NrSamolotu"]	"range-1.0"	null	2024-06-05T15:39:33.22E	57
10	47	"index_b92336e	"ONLINE"	100.0	"RANGE"	"NODE"	["Bilet"]	["KodBiletu"]	"range-1.0"	null	2024-06-05T15:39:39.94	30

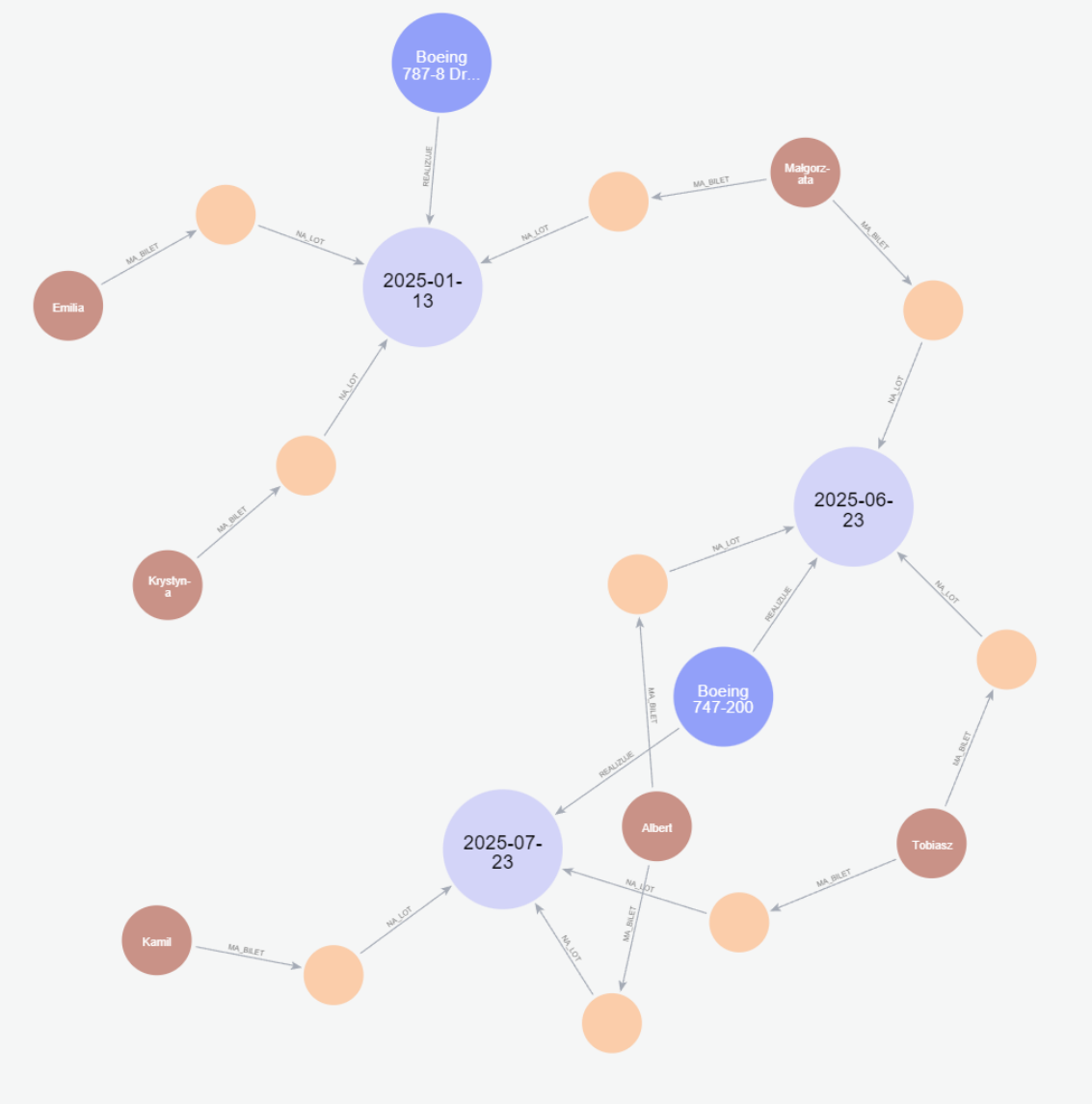
Lista pokazuje wszystkie istniejące ograniczenia co do unikalności, w tym również utworzone indeksy.

Zapytania

1. Wyświetl klientów, którzy mają kupiony bilet na lot samolotem z serii "Boeing".

```
MATCH (k:Klient)-[r1:MA_BILET]->(b:Bilet)-[r2:NA_LOT]->(l:Lot)<-[r3:REALIZUJE]-
(s:Samolot)
WHERE s.Model CONTAINS 'Boeing'
RETURN k, r1, b, r2, l, r3, s
```

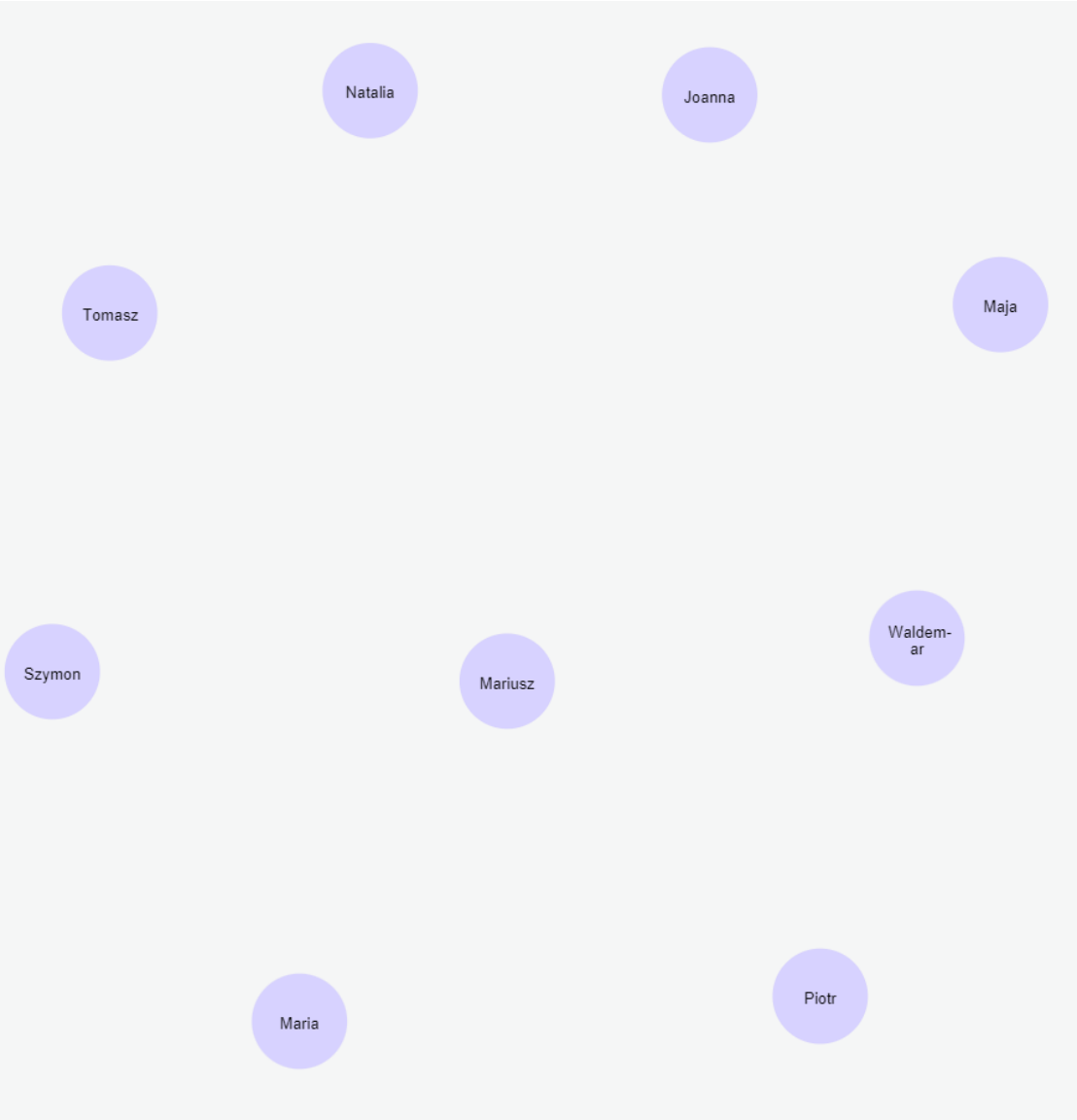
Proste zapytanie sprawdzające działanie połączeń dla większego zbioru relacji



2. Wyświetl pracowników, którzy pełnili więcej niż jedną rolę w różnych lotach.

```
MATCH (p:Pracownik)-[r1:ROLA_W_LOCIE]->(l:Lot)
WITH p, COUNT(DISTINCT l) AS num_lots
WHERE num_lots > 1
RETURN p
```

Sprawdzanie działania funkcji agregującej dla bazy grafowej

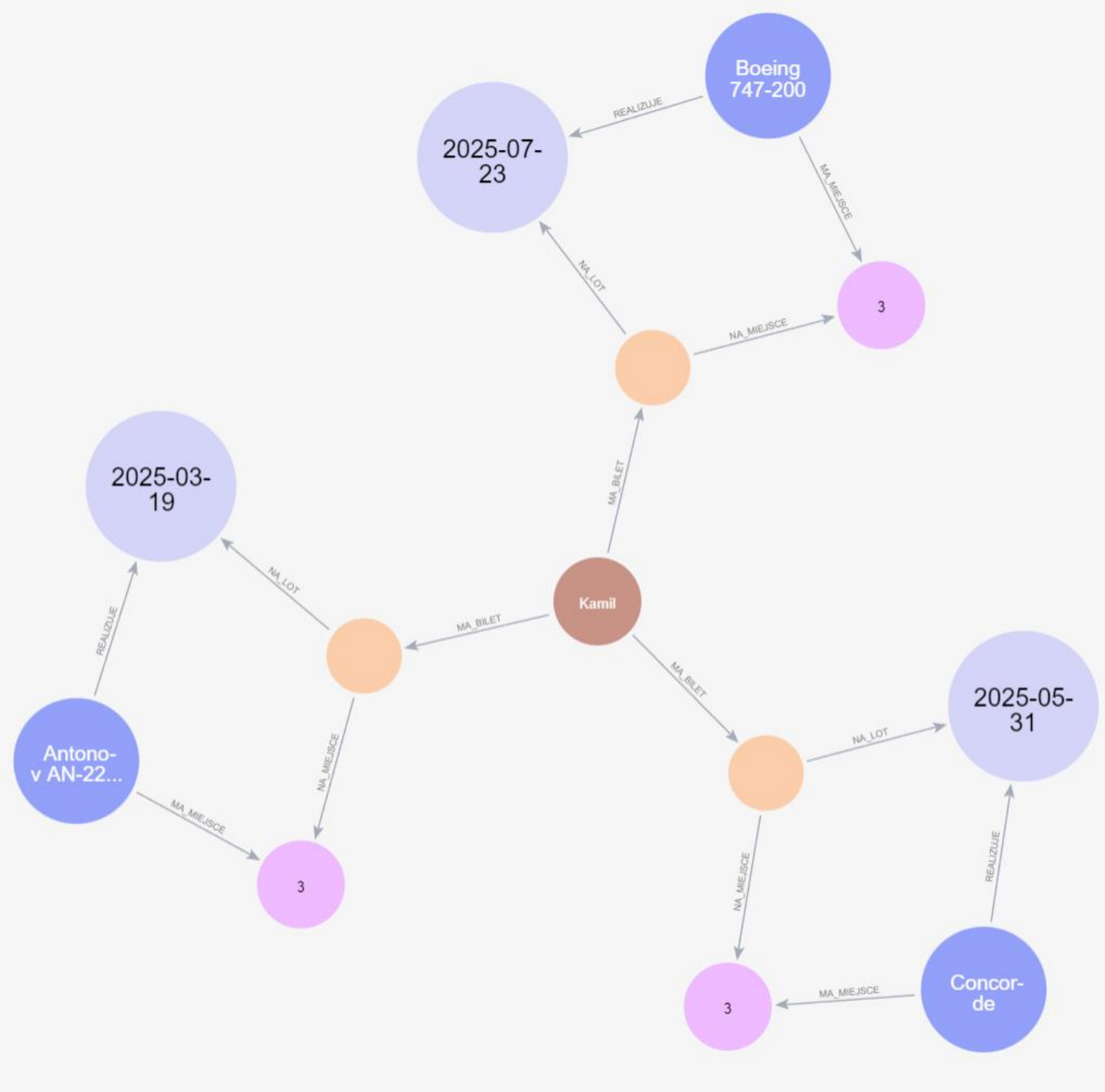


3. Wyświetl wszystkie samoloty na które bilet ma klient o numerze 1.

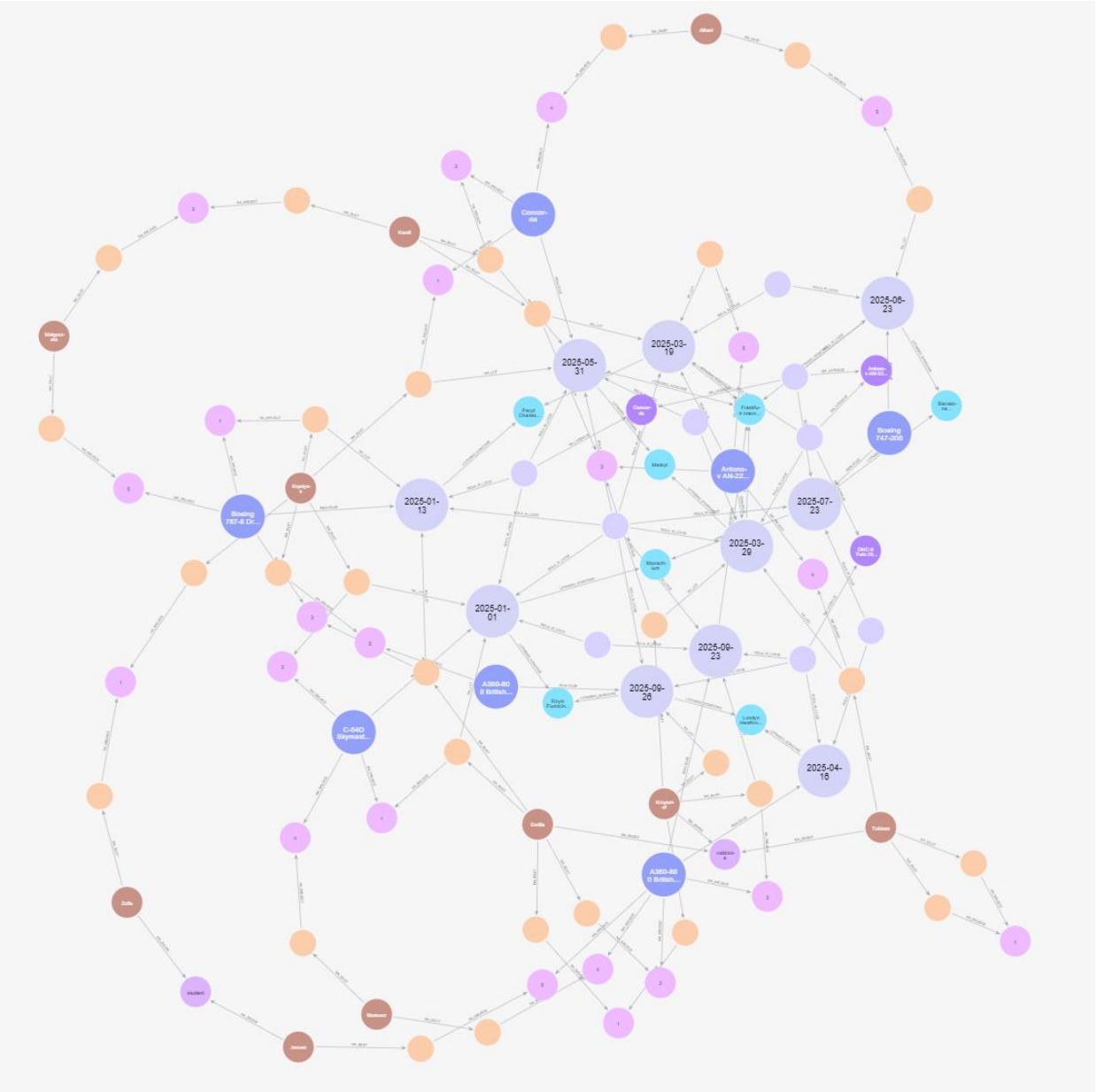
```
MATCH p = (k:Klient {NrKlienta: 1})-[*3]-(s:Samolot)
RETURN p

// Długie i błędne zapytanie, ponieważ szuka wszystkich ścieżek z samolotami, również
tymi, na które nie ma biletu
// MATCH p = (k:Klient {NrKlienta: 1})-[*]-(s:Samolot) RETURN p
```

Poszukiwanie na podstawie wiedzy o tym, że żeby znaleźć tylko samoloty dla biletów klienta to długość połączenia nie może przekraczać 3 (pomiędzy jest bilet, miejsce lub lot, a na końcu jest poszukiwany samolot).



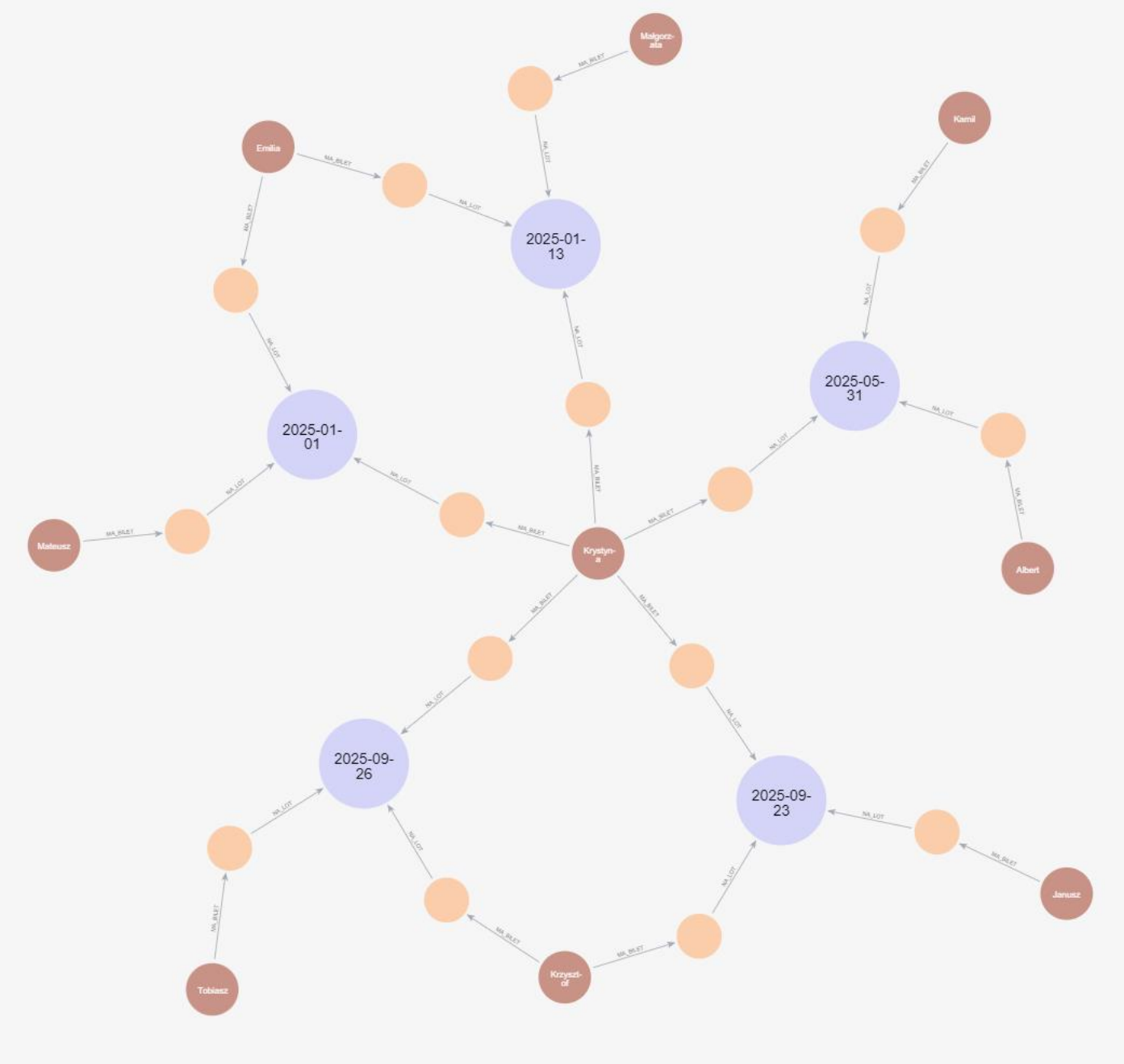
Uruchomienie bez ograniczenia długościowego (widać, że poszukując dla różnych długości zaczyna szukać po innych klientach, docelowo wypisując bardzo dużą liczbę nieoczekiwanych wyników).



4. Wyświetl połączenia dla klientów, którzy mają bilet na ten sam lot.

```
MATCH p = (k1:Klient)-[:MA_BILET]->(b1:Bilet)-[:NA_LOT]->(l:Lot)<-[:NA_LOT]-
(b2:Bilet)<-[:MA_BILET]-(k2:Klient)
WHERE k1 <> k2
RETURN p
```

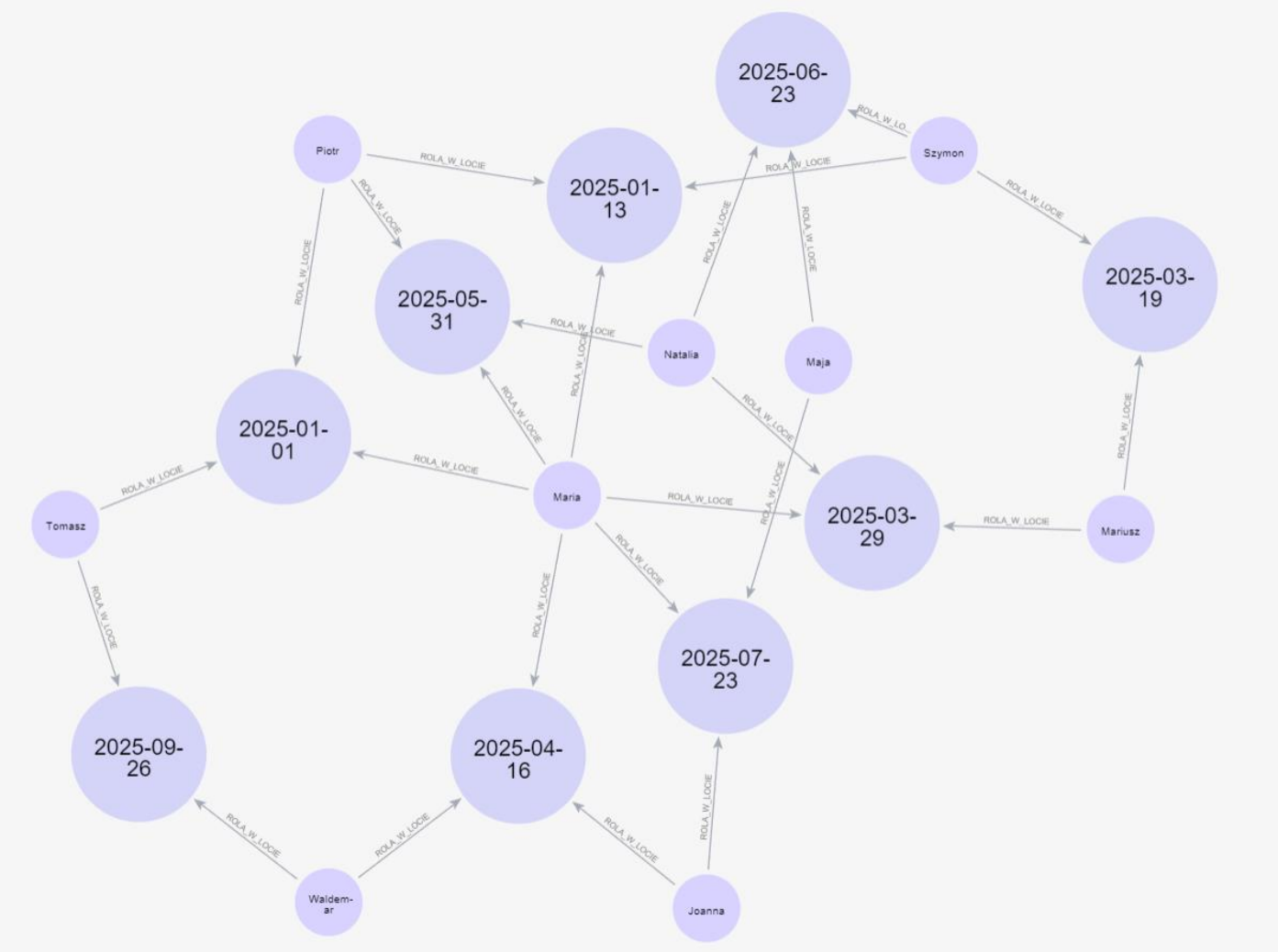
Wyświetla wszystkie poszukiwane połączenia dla danego klienta



5. Wyświetlenie dla każdego pracownika z jakimi innymi pracownikami biorą udział w locie.

```
WITH 2 AS minLength
MATCH (p1:Pracownik)-[:ROLA_W_LOCIE]->(l:Lot)<-[:ROLA_W_LOCIE]-(p2:Pracownik)
WHERE p1 <> p2
WITH p1, p2, shortestPath((p1)-[:ROLA_W_LOCIE*]-(p2)) AS path
WHERE length(path) = minLength
RETURN path
```

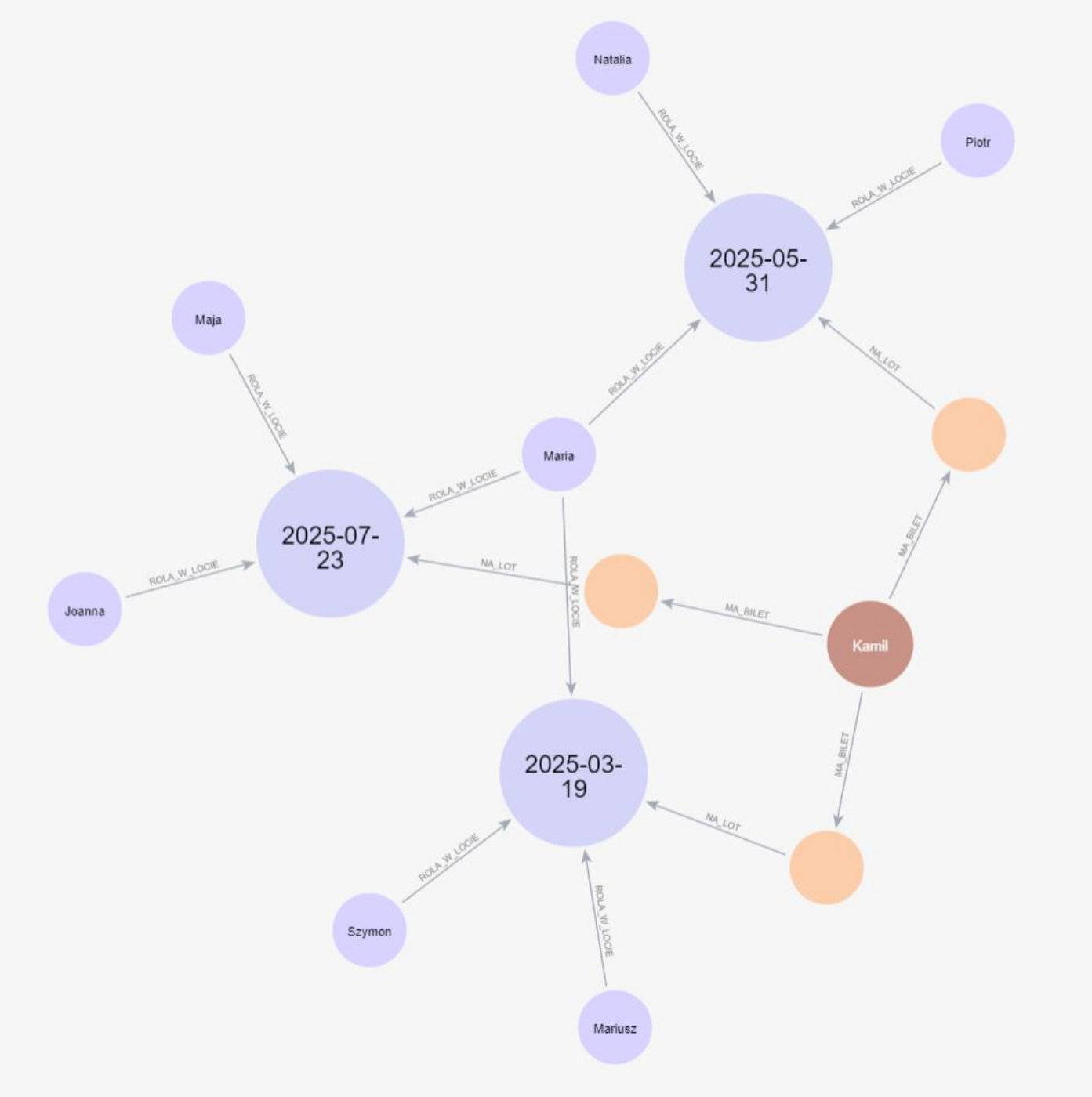
W przypadku gdy dany pracownik chciałby sprawdzić z jakimi innymi pracownikami będzie realizować dany lot, może skorzystać z podobnego zapytania. Zapytanie wykorzystuje funkcję „shortestPath”, dzięki któremu wykluczane są wszystkie dłuższe ścieżki od oczekiwanej.



6. Przejrzenie połączeń między klientem o numerze 1 a pracownikami, długości węzłów od 4 do 6.

```
MATCH p = (start:Klient {NrKlienta: 1})-[*2..4]-(end:Pracownik)
RETURN p
```

Dodatkowe przykładowe zapytanie dla górnego i dolnego ograniczenia długości ścieżki



Podsumowanie

W ramach realizacji zadania laboratoryjnego udało się stworzyć system zarządzania informacjami w zakresie lotnisk Europy.

Udało się skutecznie zastosować narzędzia Neo4j Workspace oraz Neo4j Browser do implementacji zapytań w języku Cypher i wizualizacji wyników. W projekcie wykorzystano zaawansowane funkcje takie jak indeksy, ograniczenia oraz różnorodne zapytania, które umożliwiają analizę danych oraz ich relacji w kontekście grafowym.

Zastosowane podejście umożliwia lepszą kontrolę i analizę danych w czasie rzeczywistym, co jest szczególnie istotne w kontekście dynamicznych i złożonych systemów

informacyjnych. Przeglądanie danych jest bardzo łatwe dzięki wygodnemu interfejsowi graficznemu wyświetlającym dane w postaci grafu.