NEO4j

1. **Opis problemu:**

W ramach ćwiczenia zostanie utworzona grafowa baza danych wykrywacza podejrzanych operacji na kartach bankowych. Należy określić pewne reguły za pomocą których, będą wykrywane podejrzane operacje. Baza będzie zawierać informacje o:

Użytkownikach banku (AccountHolder):

- Imię
- Nazwisko
- Unikalne id

Adres zamieszkania (Address):

- Ulica
- Miasto
- Kod pocztowy

Karta kredytowa (Credit Card):

- Numer karty
- Limit na karcie
- Stan konta
- Kod zabezpieczenia

Niezabezpieczona pożyczka (Unsecured loan):

- Numer konta
- Stan konta
- RRSO
- Kwota pożyczki

Numer telefonu (Phone numer)

- Nr telefonu

Konto bankowe (Bank account)

- Numer konta
- Stan konta

Numer ubezpieczenia (insurance numer)

- Numer ubezpieczenia

Rodzaje relacji:

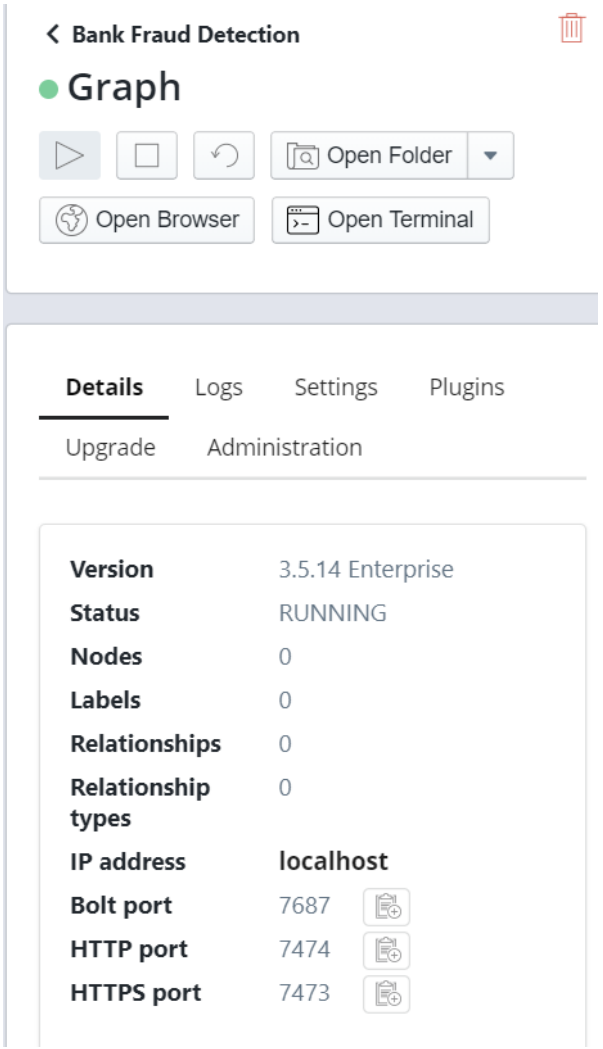
- HAS_ADDRESS (Accountholder -> Adress)
- HAS_PHONENUMBER (Accountholder ->Phone number)
- HAS_CREDITCARD (Accountholder -> Credit card)
- HAS_BANKACCOUNT (Accountholder ->Bank account)
- HAS_UNSECUREDLOAN (Accountholder ->Unsecured_loan)
- HAS_INSNUMBER (Accountholder -> InsNmber)

Problem nadaje się do użycia w grafowych bazach, ponieważ umożliwia ona łatwe wyszukiwania zależności pomiędzy label'ami, jak na przykład wyszukiwanie kręgu przestępców, którzy mogą działać w grupie. W wyszukiwaniach przestępstw bankowych, ważne są powiązania, relacje pomiędzy krawędziami (obiektami), co grafowa baza danych umożliwia do obserwowania i szukania zależności pomiędzy nimi.

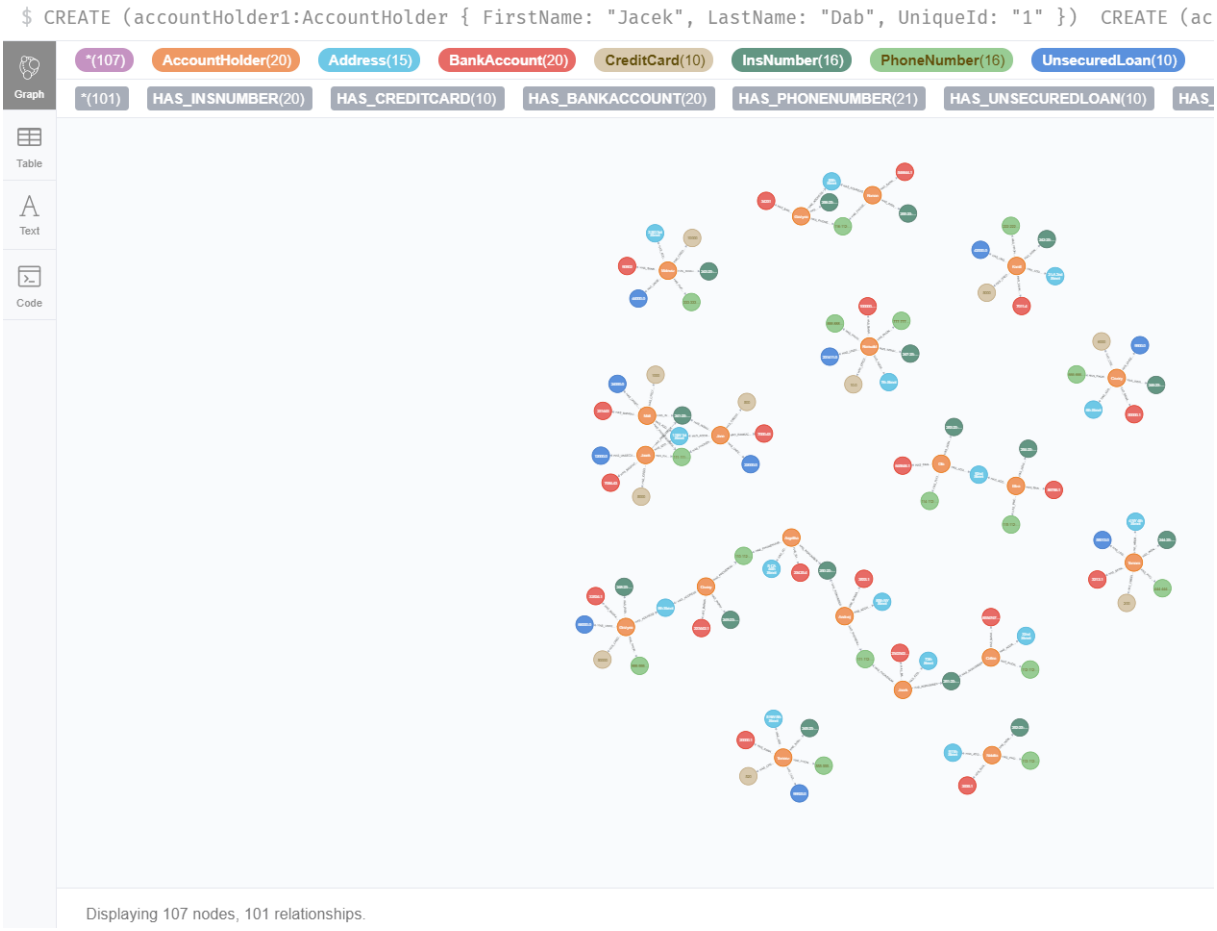
Baza ta może posłużyć do wyszukiwania tak zwanych kręgów (ring) przestępców. Działają oni w pewien określony sposób. Zakładają kilka kont na jeden adres, numer telefonu. Następnie korzystają z kont w normalny sposób. Po określonym czasie uzyskują zdolność kredytową. Po pewnym czasie wykorzystują to do zapożyczenia się w banku na jak największą sumę i następnie wyciągają pieniądza z konta jednocześnie. Grafowa baza świetnie się sprawdza w wyszukiwaniu takich powiązań pomiędzy jednostkami, wyszukując na przykład kilka kont powiązanych do tego samego adresu, numeru telefonu, numeru ubezpiecznia itp.

2. Przebieg ćwiczenia

Po instalacji narzędzia, utworzono nowy projekt „Bank Fraud Detection" Następnie utworzono graf i uruchomiono. Umożliwiło to włączenie przeglądarki w której były realizowane modyfikacje bazy.



Następnie utworzono skrypt w języku *Cypher* tworzący instancje obiektów oraz relacje pomiędzy nimi. Graficzna reprezentacja jest widoczna poniżej.:



Rysunek 1 Graficzna reprezentacja bazy danych

Utworzono w ten sposób 101 krawędzi (relacji), 107 węzłów. Liczba atrybutów po odjęciu id, wynosi 363.

Widoczne są pewne kręgi podejrzane na rysunku, które tworzą kilka połączonych ze sobą grafów. Dwa z nich składają się na 3 połączone osoby korzystające z tego samego adresu i telefonu, następnie 2 osoby korzystające z tego samego adresu oraz 5 osób tworzące ciąg połączeń adresami, numerami oraz jednym połączeniem numeru ubezpieczenia.

Stworzono zapytania służące do wyszukiwania takich kręgów, które wydają się podejrzane. Znajdź konta które dzielą więcej niż jedną informację kontaktową. Zwraca ona numery użytkowników w systemie, rozmiar kręgu oraz typ korelacji pomiędzy nimi.

```
MATCH     (accountHolder:AccountHolder)-[]->(contactInformation)
WITH    contactInformation,
      count(accountHolder) AS RingSize
MATCH     (contactInformation)<-[]-(accountHolder)
WITH    collect(accountHolder.UniqueId) AS AccountHolders,
      contactInformation, RingSize
WHERE     RingSize > 1
RETURN    AccountHolders AS FraudRing,
      labels(contactInformation) AS ContactType,
      RingSize
ORDER BY  RingSize DESC
```

Wyniki po powyższym wyszukiwaniu w Neo4j:

"FraudRing"	"ContactType"	"RingSize"
["1","3","2"]	["InsNumber"]	3
["1","3","2"]	["PhoneNumber"]	3
["3","2","1"]	["Address"]	3
["11","10"]	["Address"]	2
["11","12"]	["PhoneNumber"]	2
["12","13"]	["InsNumber"]	2
["13","14"]	["PhoneNumber"]	2
["14","15"]	["InsNumber"]	2
["18","17"]	["Address"]	2
["20","19"]	["PhoneNumber"]	2
["20","19"]	["Address"]	2

Jak widać powyżej, zapytanie znalazło parę kręgów w którym mogą znajdować się oszuści. Wyszukana została również informacja która jest taka sama w obiektach.

W następnym kroku zdefiniowano co to jest krąg podejrzanych. W tym celu stworzono kolejne zapytanie. Na początku zdefiniowano RingSize, które jest ilością podpiętych użytkowników do „contactinformation”. W naszym przypadku to nie jest tylko numer telefonu i adres, ale również inne dane takie jak numer ubezpieczenia. Następnie ustalono, jeśli posiada kartę kredytową lub (OR) pożyczkę wtedy zapisujemy sobie użytkownika jako „unsecuredAccount”. Następnie liczony jest limit na koncie – jeśli posiada kartę kredytową, lub liczony stan konta jeśli posiada pożyczkę. Następnie zwracana jest wartość zapytania. Wyeliminowano powiązane konta niespełniające warunków (pożyczka lub konto).

```
MATCH     (accountHolder:AccountHolder)-[]->(contactInformation)
WITH    contactInformation,
       count(accountHolder) AS RingSize
MATCH     (contactInformation)<-[]-(accountHolder),
       (accountHolder)-[r:HAS_CREDITCARD|HAS_UNSECUREDLOAN]->(unsecuredAccount)
WITH    collect(DISTINCT accountHolder.UniqueId) AS AccountHolders,
       contactInformation, RingSize,
       SUM(CASE type(r)
         WHEN 'HAS_CREDITCARD' THEN unsecuredAccount.Limit
         WHEN 'HAS_UNSECUREDLOAN' THEN unsecuredAccount.Balance
         ELSE 0
       END) as FinancialRisk
WHERE     RingSize > 1
RETURN    AccountHolders AS FraudRing,
       labels(contactInformation) AS ContactType,
       RingSize,
       round(FinancialRisk) as FinancialRisk
ORDER BY  FinancialRisk DESC
```

Wyniki z zapytania:

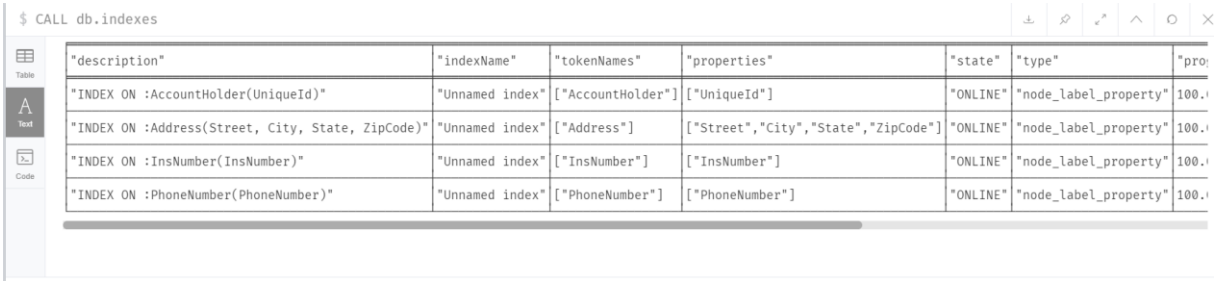
"FraudRing"	"ContactType"	"RingSize"	"FinancialRisk"
["10"]	["Address"]	2	73135.0
["1","3","2"]	["InsNumber"]	3	51888.0
["1","3","2"]	["PhoneNumber"]	3	51888.0
["3","2","1"]	["Address"]	3	51888.0

Indeksy

Cypher pozwala na utworzenie indeksów dwóch typów:

- Na pojedynczej właściwości węzła
- Indeks kompozytowy

Jednak każdy indeks to kopia pewnej ilości danych w bazie na potrzeby szybszego wyszukiwania, jednak wolniejszego zapisu: link. W tym przypadku najlepszym rozwiązaniem umiejscowienia indeksów jest unikalny id, numer ubezpieczenia, numer telefonu i kompozytowy na adresie. Wyszukiwane będą te części wspólne najczęściej, dlatego zdecydowano się na utworzenie na tych polach indeksów.



```
$ CALL db.indexes
```

"description"	"indexName"	"tokenNames"	"properties"	"state"	"type"	"pro
"INDEX ON :AccountHolder(UniqueId)"	"Unnamed index"	["AccountHolder"]	["UniqueId"]	"ONLINE"	"node_label_property"	100.
"INDEX ON :Address(Street, City, State, ZipCode)"	"Unnamed index"	["Address"]	["Street","City","State","ZipCode"]	"ONLINE"	"node_label_property"	100.
"INDEX ON :InsNumber(InsNumber)"	"Unnamed index"	["InsNumber"]	["InsNumber"]	"ONLINE"	"node_label_property"	100.
"INDEX ON :PhoneNumber(PhoneNumber)"	"Unnamed index"	["PhoneNumber"]	["PhoneNumber"]	"ONLINE"	"node_label_property"	100.

Zostały utworzone następującą składnią (kompozytowy indeks na adresie):

```
$ CREATE INDEX ON :Address(Street,City,State, ZipCode)
```