***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 3](#_Toc168501537)

[Środowisko 3](#_Toc168501538)

[Opis środowiska 3](#_Toc168501539)

[Instrukcja do przygotowania bazy danych 4](#_Toc168501540)

[Wykaz tabel 4](#_Toc168501541)

[Schemat bazy danych 5](#_Toc168501542)

[Węzły 6](#_Toc168501543)

[Relacje 8](#_Toc168501544)

[Ograniczenia 10](#_Toc168501545)

[Indeksy 10](#_Toc168501546)

[Zapytania 11](#_Toc168501547)

[Podsumowanie 18](#_Toc168501548)

# 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:

Gdzie:

[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
   1. Uruchomienie wszystkiego na raz – skrypt „script.cypher”
   2. Uruchomienie elementowo:
      1. Stworzenie węzłów – skrypt „nodes.cypher”
      2. Stworzenie indeksów – skrypt „indexes.cypher”
      3. Stworzenie ograniczeń – skrypt „constrains.cypher”
      4. 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”.

A diagram of a model

Description automatically generated

# Węzły

Zbiór węzłów (etykieta) na przykładzie Klientów

A group of dots with names

Description automatically generated

Podgląd atrybutów danego węzła

A screenshot of a phone

Description automatically generated

# Relacje

Zbiór relacji na przykładzie relacji Samolot -> Miejsce o nazwie „MA\_MIEJSCE”

A group of purple circles with white text

Description automatically generated

Pogląd relacji na przykładzie relacji ROLA\_W\_LOCIE, która poza łączeniem pracownika z lotem mówi również jaką rolę pełni

A screenshot of a computer

Description automatically generated

# Ograniczenia

Zastosowane ograniczenia, pokazane przy pomocy polecenia „SHOW CONSTRAINTS”

A screenshot of a computer

Description automatically generated

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”.

A screenshot of a computer

Description automatically generated

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

# Zapytania

### 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

A screenshot of a computer

Description automatically generated

### 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

A screenshot of a computer

Description automatically generated

### 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).

A diagram of a network

Description automatically generated

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

A network of circles and dots

Description automatically generated

### 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

A diagram of a network

Description automatically generated

### 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.

A network of circles and numbers

Description automatically generated

### 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

A screenshot of a computer

Description automatically generated

# 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.