



dr inż. Michał Malinowski
michal.malinowski@uth.edu.pl



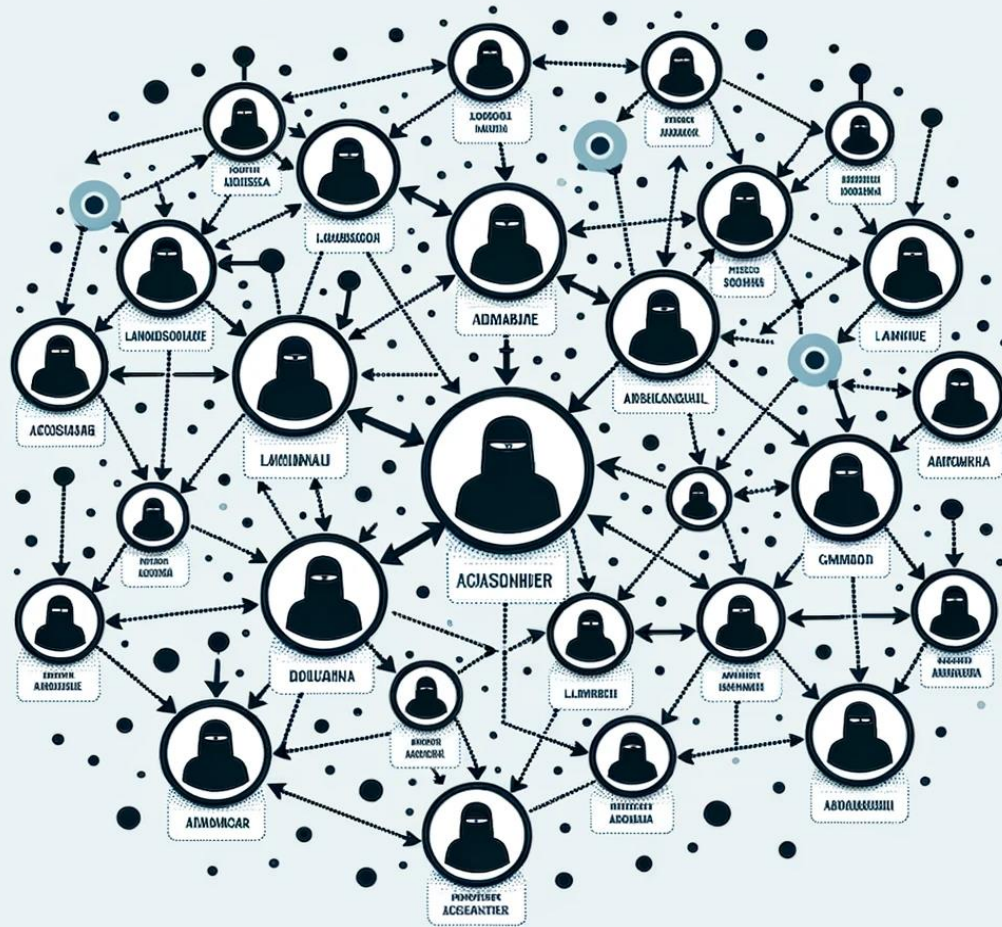
[https://www.linkedin.com/in/
michał-malinowski-malkon/](https://www.linkedin.com/in/michał-malinowski-malkon/)



Algorytmy grafowe

Ćwiczenie 12

ALGORYTMY
I STRUKTURY DANYCH



Zagadnienia (2h)

- [Algorytmy grafowe - Analiza danych](#)
- [Algorytmy wyszukiwania ścieżek](#)
- [Algorytmy wspólnoty](#)
- [Algorytmy detekcji anomalii](#)
- [Algorytmy centralności](#)
- [Algorytmy podobieństwa](#)
- [Algorytmy predykcji relacji](#)
- [Detekcja Liderów](#)
- [Mapowanie zależności zadań](#)

Algorytmy grafowe



Algorytmy wyszukiwania ścieżek

Wyszukują optymalne ścieżki pomiędzy węzłami dla określonych kryteriów



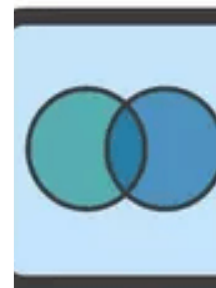
Algorytmy centralności

Pozwalają ocenić, jakie węzły w sieci są najważniejsze, czyli mogą mieć największy wpływ



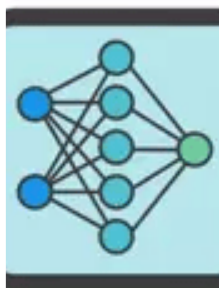
Algorytmy wspólnoty

Identyfikują skupiska węzłów, które są gęściej połączone między sobą niż z resztą sieci



Algorytmy podobieństwa

Obliczają stopień podobieństwa między elementami



Algorytmy detekcji anomalii

Wykorzystywane są do identyfikowania nieoczekiwanych wzorców lub obserwacji w danych



Algorytmy predykcji relacji

Prognozują czy dwa węzły w grafie mogą w przyszłości stworzyć połączenie

Algorytmy wyszukiwania ścieżek

Algorytmy wyszukiwania ścieżek

Algorytmy wyszukiwania ścieżek znajdują ścieżkę między dwoma lub więcej węzłami lub oceniają dostępność i jakość ścieżek.

- Znalezienie ścieżki: Lokalizowanie jednej lub kilku ścieżek spełniających określone kryteria, takie jak minimalny koszt, najkrótszy czas przejścia lub najmniejsza liczba przejść między wierzchołkami.
- Ocena ścieżki: Ocenianie dostępności i jakości ścieżek na podstawie różnych parametrów, jak dystans, przepustowość czy koszt.

- Nawigacja do wyznaczania najkrótszej trasy między punktami.
- Planowanie tras w logistyce do optymalizacji kosztów dostaw.



Algorytmy wspólnoty

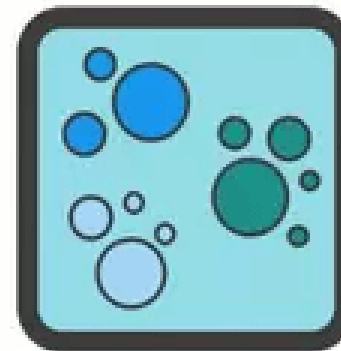
Algorytmy wspólnoty

Algorytmy wykrywania społeczności są wykorzystywane do oceny, w jaki sposób grupy węzłów są grupowane lub dzielone, a także ich tendencji do wzmacniania się lub rozpadania.

Te grupy, często nazywane wspólnotami, charakteryzują się względnie gęstszym połączeniem między węzłami wewnątrz grupy niż z węzłami zewnętrznymi.

Proces wykrywania społeczności pozwala zrozumieć nie tylko strukturę i organizację sieci, ale także może dostarczyć wglądu w sposób, w jaki informacje lub wpływy rozprzestrzeniają się w sieci.

- Analiza sieci społecznych do wykrywania grup zainteresowań lub wspólnot.
- Biologia obliczeniowa do identyfikowania funkcjonalnych klastrów w sieciach interakcji białkowych.

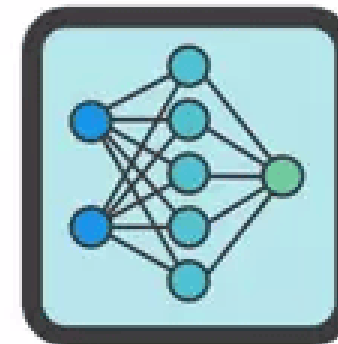


Algorytmy detekcji anomalii

Algorytmy detekcji anomalii

są narzędziami służącymi do identyfikacji nieoczekiwanych wzorców lub obserwacji w danych, które mogą wskazywać na błędy, oszustwa, awarie systemowe lub inne ważne zdarzenia.

Takie anomalie mogą wskazywać na różnorodne kwestie, od błędów po podejrzaną celową aktywność.



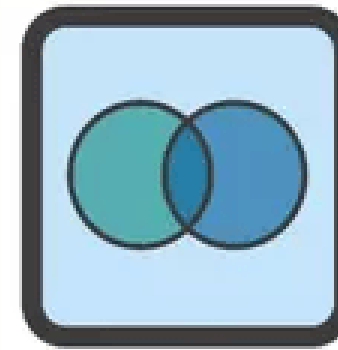
- Wykrywanie oszustw finansowych poprzez identyfikację nietypowych wzorców transakcji.
- Monitoring sieci komputerowych w celu wychwycenia nietypowego ruchu wskazującego na potencjalne ataki cybernetyczne.

Algorytmy podobieństwa

Algorytmy podobieństwa

służą do oceny, jak bardzo podobne są do siebie węzły na podstawie określonych cech lub wzorców połączeń.

Te algorytmy wykorzystują różne metryki, takie jak współczynniki korelacji, odległość euklidesowa, czy wspólne sąsiedztwo, aby kwantyfikować podobieństwo.



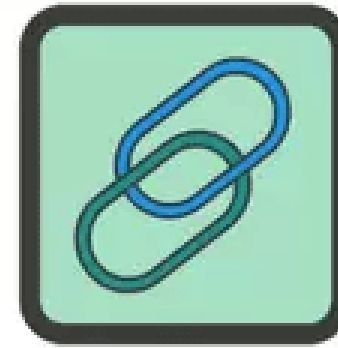
- Systemy rekomendacyjne, które sugerują produkty podobne do tych, które klient już przeglądał.
- Analiza dokumentów prawnych lub naukowych w celu znalezienia podobnych przypadków lub publikacji.

Algorytmy predykcji relacji

Algorytmy predykcji relacji

Na podstawie struktury grafu (relacji) prognozują czy dwa węzły w grafie mogą w przyszłości stworzyć połączenie (krawędź).

- Wspólni sąsiedzi
- Ta sama wspólnota
- Całkowita liczba sąsiadów



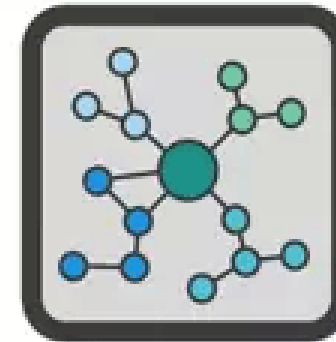
- Prognozowanie rozwoju sieci społecznych, np. powstawanie nowych połączeń na platformach społecznościowych.
- Przewidywanie interakcji między białkami w genomice, co może mieć zastosowanie w projektowaniu leków.

Algorytmy centralności

Algorytmy centralności

Pozwalają ocenić, jakie węzły w sieci są najważniejsze, czyli mają mogą mieć największy wpływ.

- Centralność stopnia (Degree Centrality)
- Centralność bliskości (Closeness Centrality)
- Centralność pośrednictwa (Betweenness Centrality)
- Centralność wektora własnego (Eigenvector Centrality)
- PageRank

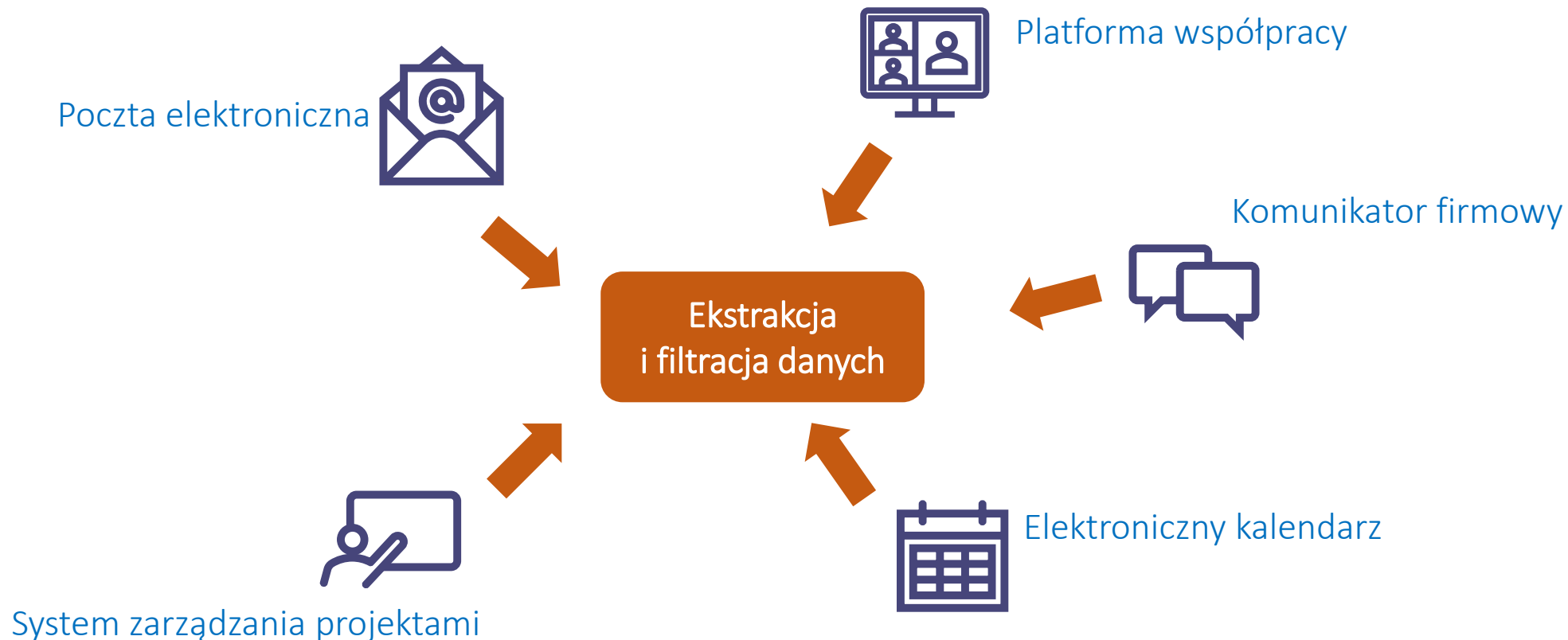


- Analiza struktury organizacji w celu identyfikacji kluczowych pracowników.
- Badanie sieci społecznych, aby znaleźć influencerów lub kluczowe węzły rozprzestrzeniania informacji.

Detekcja Liderów

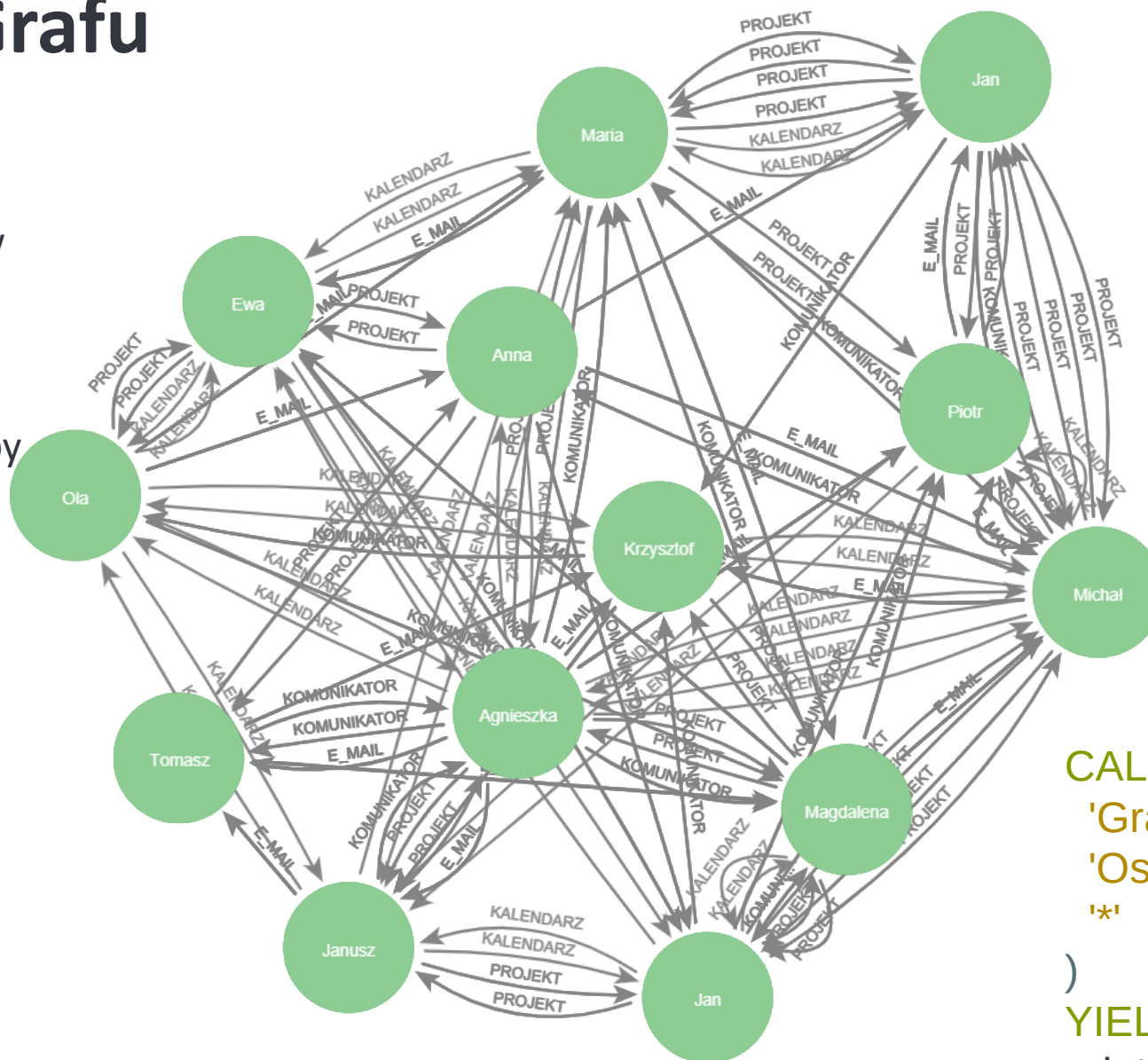
Eksploracja Danych (Data Mining)

Zebranie danych o interakcjach i komunikacji między pracownikami organizacji.



Tworzenie Grafu

Utworzenie grafu, w którym węzły reprezentują osoby (pracowników), a krawędzie różne typy interakcji i komunikacji między pracownikami organizacji.



```
CALL gds.graph.project(
  'GrafOsoba',
  'Osoba',
  '*'
)
YIELD graphName, nodeCount,
relationshipCount
RETURN *
```

Centralność stopnia

(Degree Centrality)

to miara w analizie sieci społecznościowych, która określa, jak wiele połączeń (krawędzi) prowadzi do danego wężła (osoby, wężła sieci).

Definiuje się jako:

$$C_D(v) = \frac{\deg(v)}{n-1}$$

gdzie:

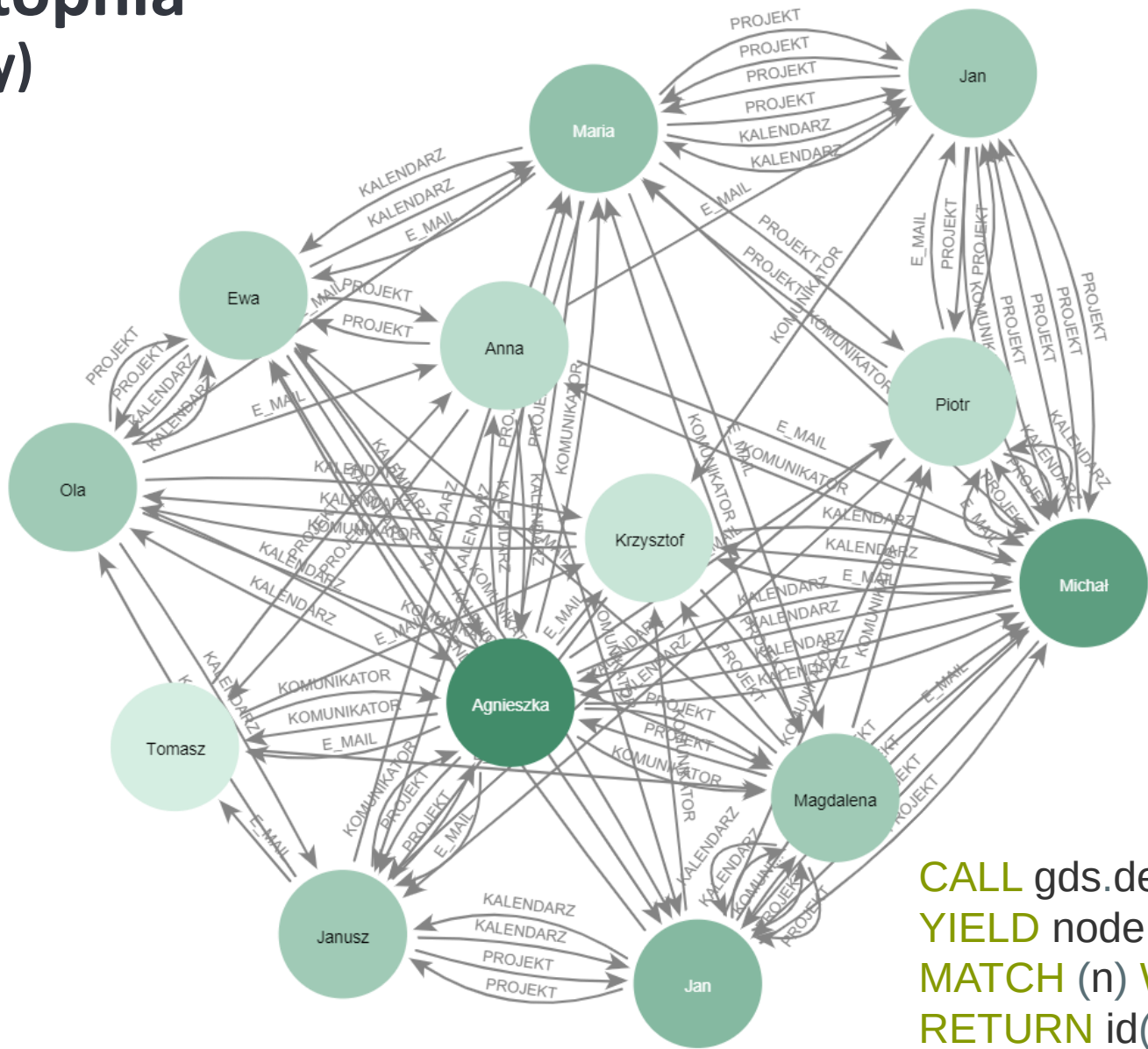
- $\deg(v)$ - oznacza stopień wężła v , czyli liczbę bezpośrednich połączeń (krawędzi) wężła v z innymi wężłami,
- n - to całkowita liczba wężłów w grafie, minus 1, ponieważ nie uwzględnia się połączeń z samym sobą.

Wskazuje na to, jak „popularna” jest dana osoba

Osoby z wysokim stopniem centralności są często uważane za wpływowe lub strategicznie położone w sieci, ponieważ mogą bezpośrednio komunikować się z wieloma innymi osobami lub pełnić rolę ważnego pośrednika w przepływie informacji.

Centralność stopnia (Degree Centrality)

to miara, która określa, jak wiele połączeń (krawędzi) prowadzi do danego wężła (osoby). Wskazuje na to, jak „popularna” jest dana osoba.



Pozycja	ID	Osoba	Wynik
1	25	"Agnieszka"	15,00 1,25000
2	24	"Michał"	13,00 1,08333
3	41	"Jan"	10,00 0,83333
4	34	"Maria"	9,00 0,75000
5	21	"Magdalena"	8,00 0,66667
6	26	"Janusz"	8,00 0,66667
7	31	"Jan"	8,00 0,66667
8	40	"Ola"	8,00 0,66667
9	23	"Ewa"	7,00 0,58333
10	32	"Anna"	6,00 0,50000
11	33	"Piotr"	6,00 0,50000
12	22	"Krzysztof"	5,00 0,41667
13	35	"Tomasz"	4,00 0,33333

```
CALL gds.degree.stream('GrafOsoba')
YIELD nodeld, score
MATCH (n) WHERE id(n) = nodeld
RETURN id(n), n.imie, score
ORDER BY score DESC
```

Centralność pośrednictwa

(Betweenness centrality)

to miara centralności w grafie, która wskazuje na stopień, w jakim dany węzeł leży na ścieżkach między innymi węzłami.

Definiuje się jako:

$$C_B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

gdzie:

- σ_{st} - to liczba najkrótszych ścieżek od węzła s do węzła t ,
- $\sigma_{st}(v)$ - to liczba tych ścieżek, które przechodzą przez węzeł v ,
- suma rozciąga się na wszystkie pary węzłów s i t w sieci.

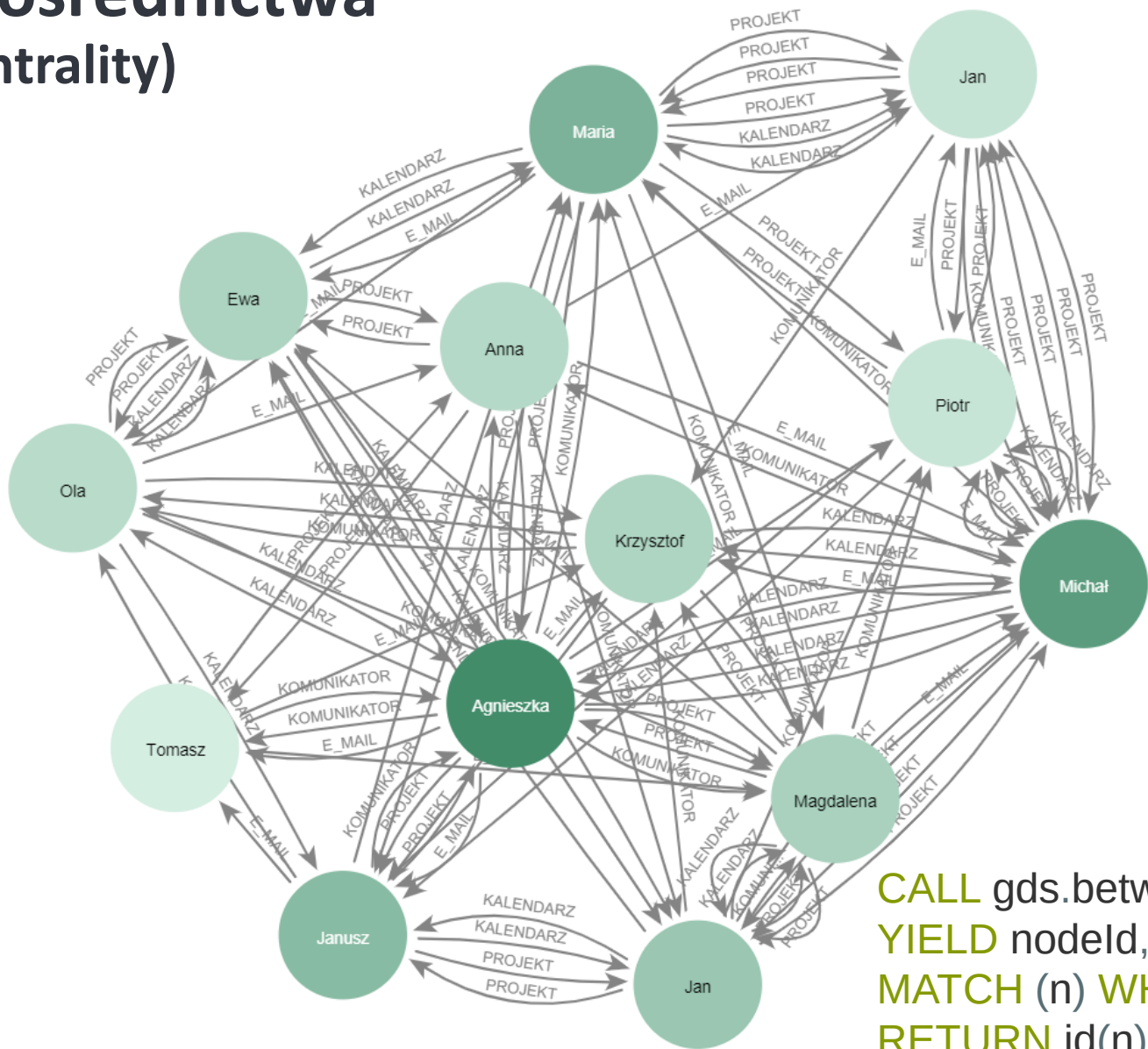
Wskazuje na to jak ważna jest osoba dla ciągłości kanałów przepływu informacji

Osoba z wysoką centralnością pośrednictwa działa jako punkt pośredni (HUB) i ma duże znaczenie dla przepływu informacji. Ponadto ma ona strategiczną rolę, jeśli chodzi o kontrolę nad przepływem informacji lub zasobów.

Centralność pośrednictwa

(Betweenness centrality)

to miara, która wskazuje na stopień, w jakim dany węzeł leży na ścieżkach między innymi węzłami. Wskazuje na to jak ważna jest osoba dla ciągłości kanałów przepływu informacji



Pozycja	ID	Osoba	Wynik
1	25	"Agnieszka"	15,40300
2	24	"Michał"	12,95922
3	34	"Maria"	9,73657
4	26	"Janusz"	8,45818
5	41	"Jan"	6,71694
6	21	"Magdalena"	5,15317
7	23	"Ewa"	4,81508
8	22	"Krzysztof"	4,69628
9	32	"Anna"	4,00580
10	40	"Ola"	3,70001
11	31	"Jan"	2,33810
12	33	"Piotr"	2,19072
13	35	"Tomasz"	0,82692

```
CALL gds.betweenness.stream('GrafOsoba')
YIELD nodeId, score
MATCH (n) WHERE id(n) = nodeId
RETURN id(n), n.imie, score
ORDER BY score DESC
```

PageRanking

(PageRank)

to miara wpływu węzła w sieci ze względu na jakość jego połączeń.

Definiuje się jako:

$$P(U) = \frac{1 - d}{N} + d \sum_{V \in B(U)} \frac{P(V)}{L(V)}$$

gdzie:

- d jest współczynnikiem tłumienia, zazwyczaj ustawianym na około 0.85
- N jest całkowitą węzłów w grafie
- $B(U)$ jest zbiorem węzłów wskazujących na węzeł U
- $P(V)$ jest PageRank węzła V
- $L(V)$ jest liczbą krawędzi wychodzących z węzła V

Pokazuje jak ważna jest dana osoba ze względu na wagę połączeń z innymi ważnymi osobami/

Osoba z wysokim PageRanking jest uważana za szczególnie wpływową. To oznacza, że taka osoba może skutecznie rozpowszechniać informacje i wpływać na opinie w obrębie swojej społeczności.

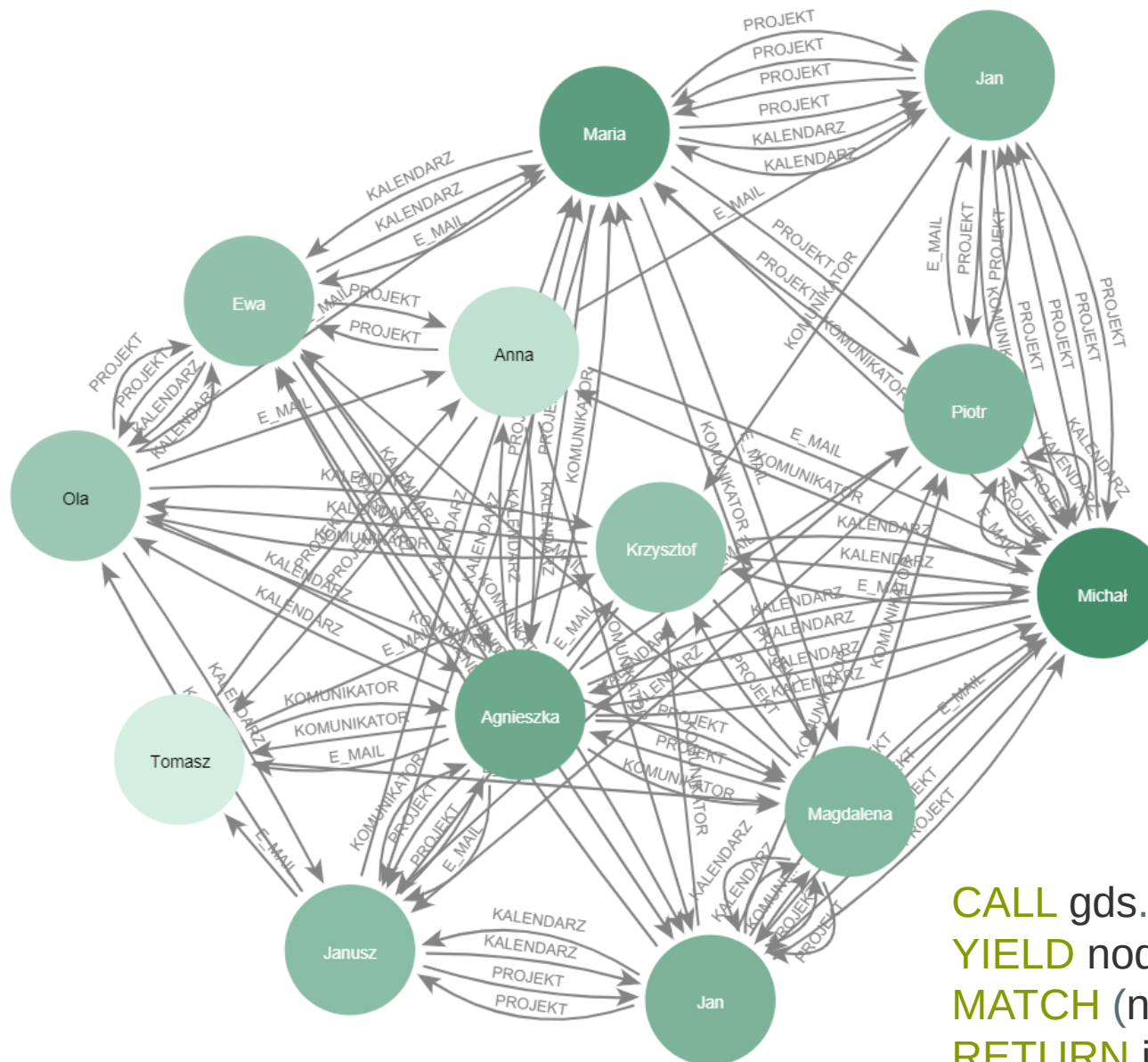
Algorytm iteracyjny

to proces obliczeniowy, który powtarza zestaw instrukcji (zwany iteracją) do osiągnięcia określonego wyniku lub stanu.

Proces iteracyjny kontynuowany jest do spełnienia warunku zakończenia, takiego jak osiągnięcie zadanej dokładności, wyczerpanie zestawu danych wejściowych lub wykonanie określonej liczby iteracji.

PageRank (PageRank)

to miara wpływu węzła w sieci ze względu na jakość jego połączeń.
Pokazuje jak ważna jest dana osoba ze względu na wagę połączeń z innymi ważnymi osobami.



Pozycja	ID	Osoba	Wynik
1	24	"Michał"	1,40721
2	34	"Maria"	1,24306
3	25	"Agnieszka"	1,13165
4	31	"Jan"	1,04978
5	41	"Jan"	1,04315
6	33	"Piotr"	1,00760
7	21	"Magdalena"	0,99239
8	26	"Janusz"	0,94848
9	23	"Ewa"	0,89871
10	22	"Krzysztof"	0,88480
11	40	"Ola"	0,82921
12	32	"Anna"	0,59842
13	35	"Tomasz"	0,46166

```
CALL gds.pageRank.stream('GrafOsoba')
YIELD nodeId, score
MATCH (n) WHERE id(n) = nodeId
RETURN id(n), n.imie, score
ORDER BY score DESC
```

Modułowość Louvain

(Louvain Modularity)

to miara, która ocenia jak dobrze graf dzieli się na moduły, czyli gęsto połączone grupy węzłów z rzadszymi połączeniami pomiędzy grupami.

Definiuje się jako:

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)$$

gdzie:

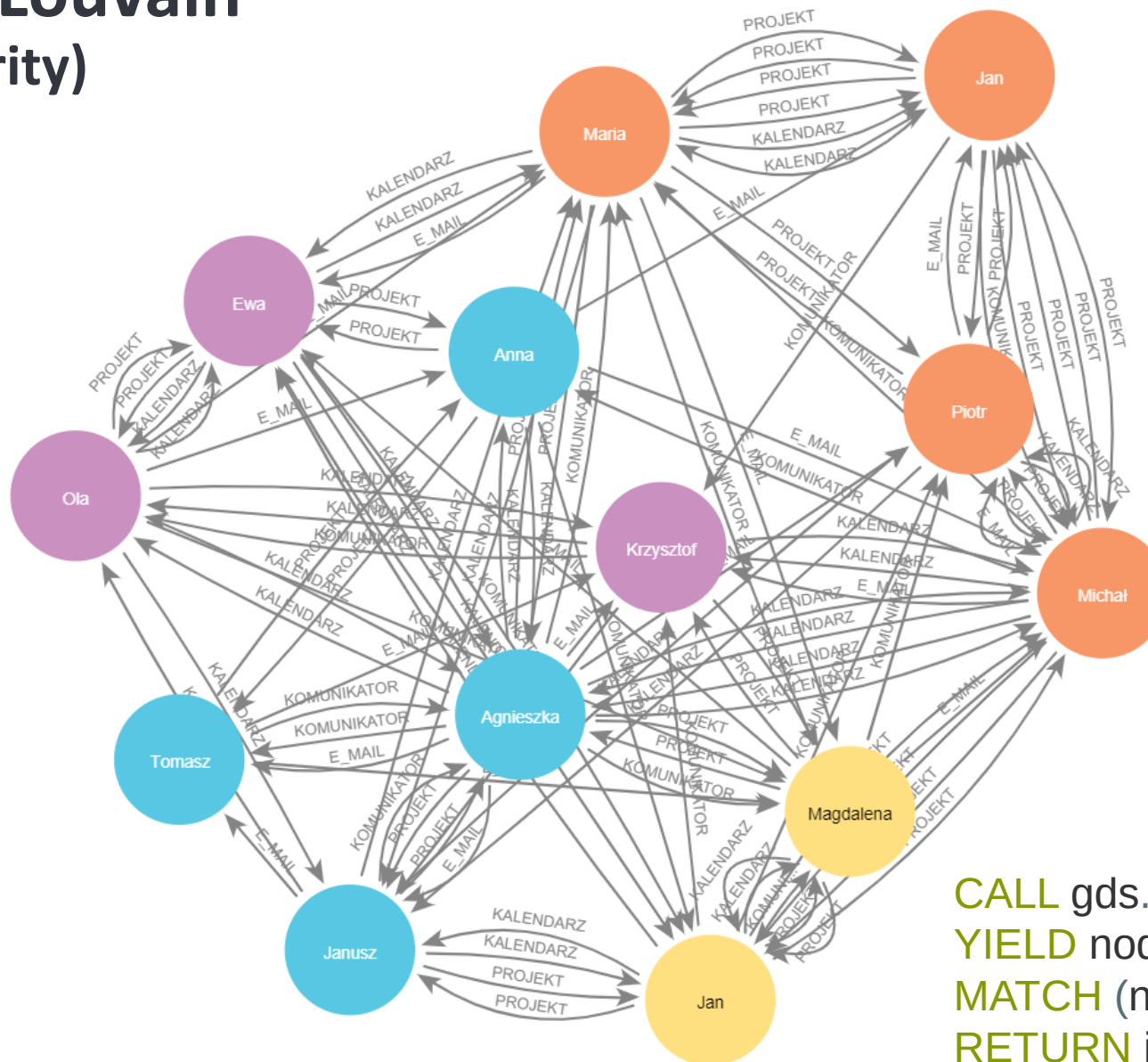
- A_{ij} reprezentuje wagę krawędzi między węzłami i i j
- k_i i k_j to sumy wag krawędzi przyłączonych do węzłów i i j
- m to suma wszystkich wag krawędzi w sieci
- c_i i c_j to społeczności, do których należą węzły i i j
- δ to funkcja Kroneckera, która jest równa 1, gdy $c_i = c_j$ i 0 w przeciwnym przypadku.

Jest stosowany do wykrywania społeczności w dużych sieciach. Identyfikuje grupy o podobnych zainteresowaniach lub wzorcach interakcji.

Modułowość Louvain

(Louvain Modularity)

jest stosowany do wykrywania społeczności w dużych sieciach. Identyfikuje grupy o podobnych zainteresowaniach lub wzorcach interakcji.



ID	Osoba	Grupa
22	"Krzysztof"	11
23	"Ewa"	11
40	"Ola"	11
25	"Agnieszka"	10
26	"Janusz"	10
32	"Anna"	10
35	"Tomasz"	10
24	"Michał"	9
31	"Jan"	9
33	"Piotr"	9
34	"Maria"	9
21	"Magdalena"	0
41	"Jan"	0

```
CALL gds.louvain.stream('GrafOsoba')
YIELD nodeId, communityId
MATCH (n) WHERE id(n) = nodeId
RETURN id(n), n.imie, communityId
ORDER BY communityId DESC
```

Czy i kogo należy się bać?

Ranking osób pod
względem
rozpatrywanych
miar

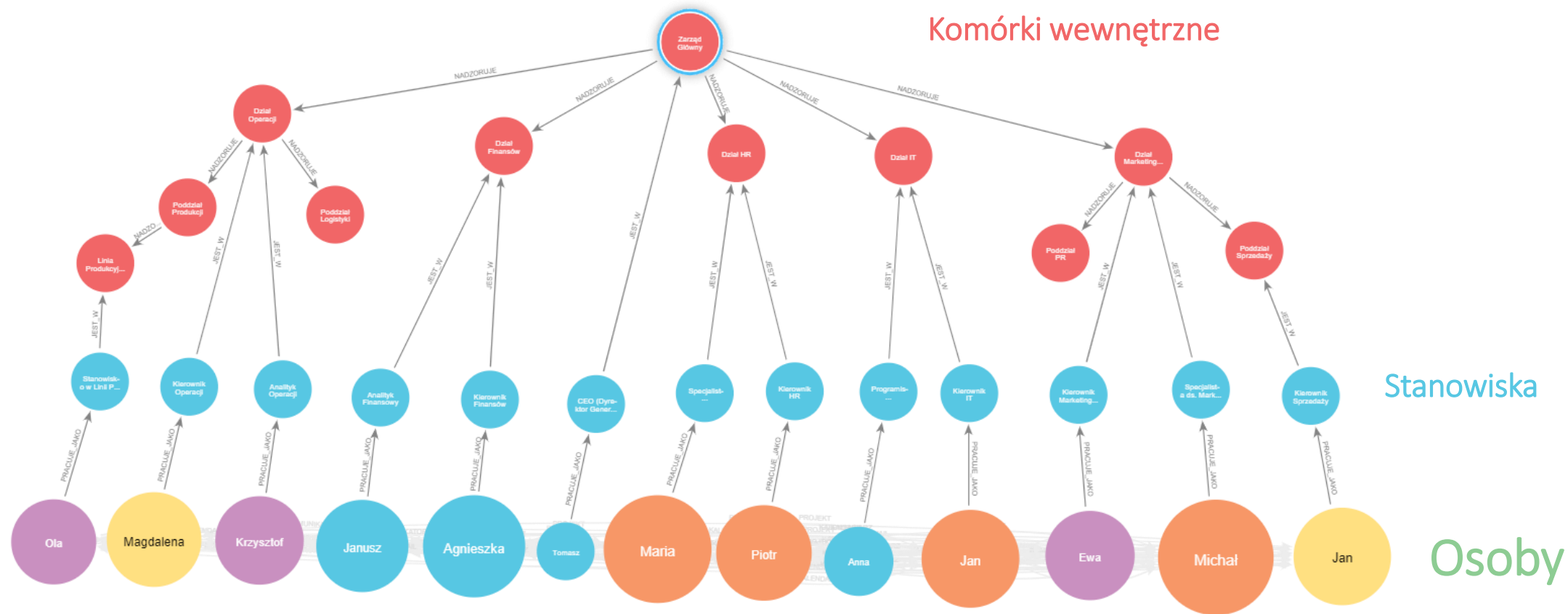
	Pozycja			
Louvain	Degree	Betweenness	PageRank	Średnia
"Agnieszka"	1	1	3	1,67
"Michał"	2	2	1	1,67
"Maria"	4	3	2	3,00
"Jan"	3	5	5	4,33
"Magdalena"	5	6	7	6,00
"Janusz"	6	4	8	6,00
"Jan"	7	11	4	7,33
"Ewa"	9	7	9	8,33
"Ola"	8	10	11	9,67
"Piotr"	11	12	6	9,67
"Krzysztof"	12	8	10	10,00
"Anna"	10	9	12	10,33
"Tomasz"	13	13	13	13,00

Czy i kogo należy się bać?

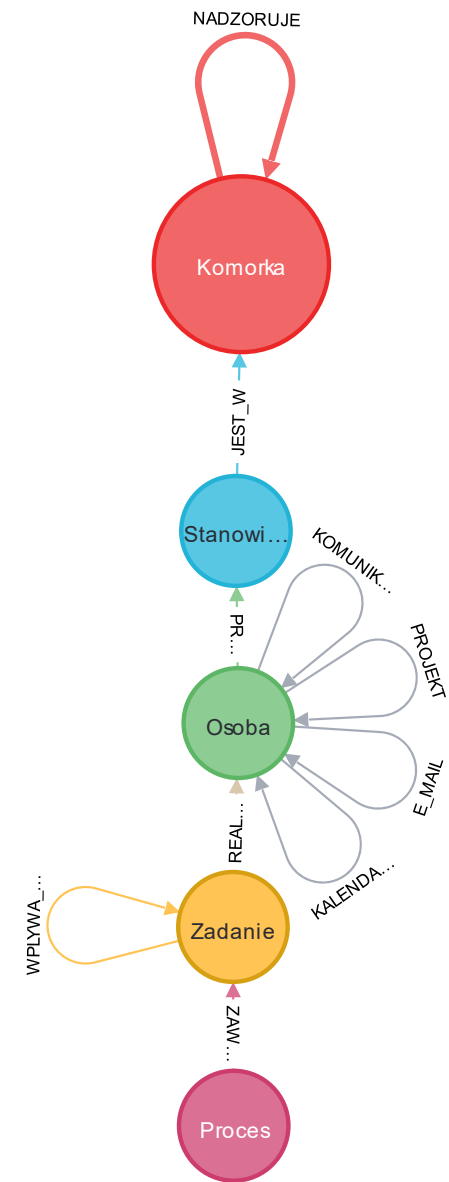
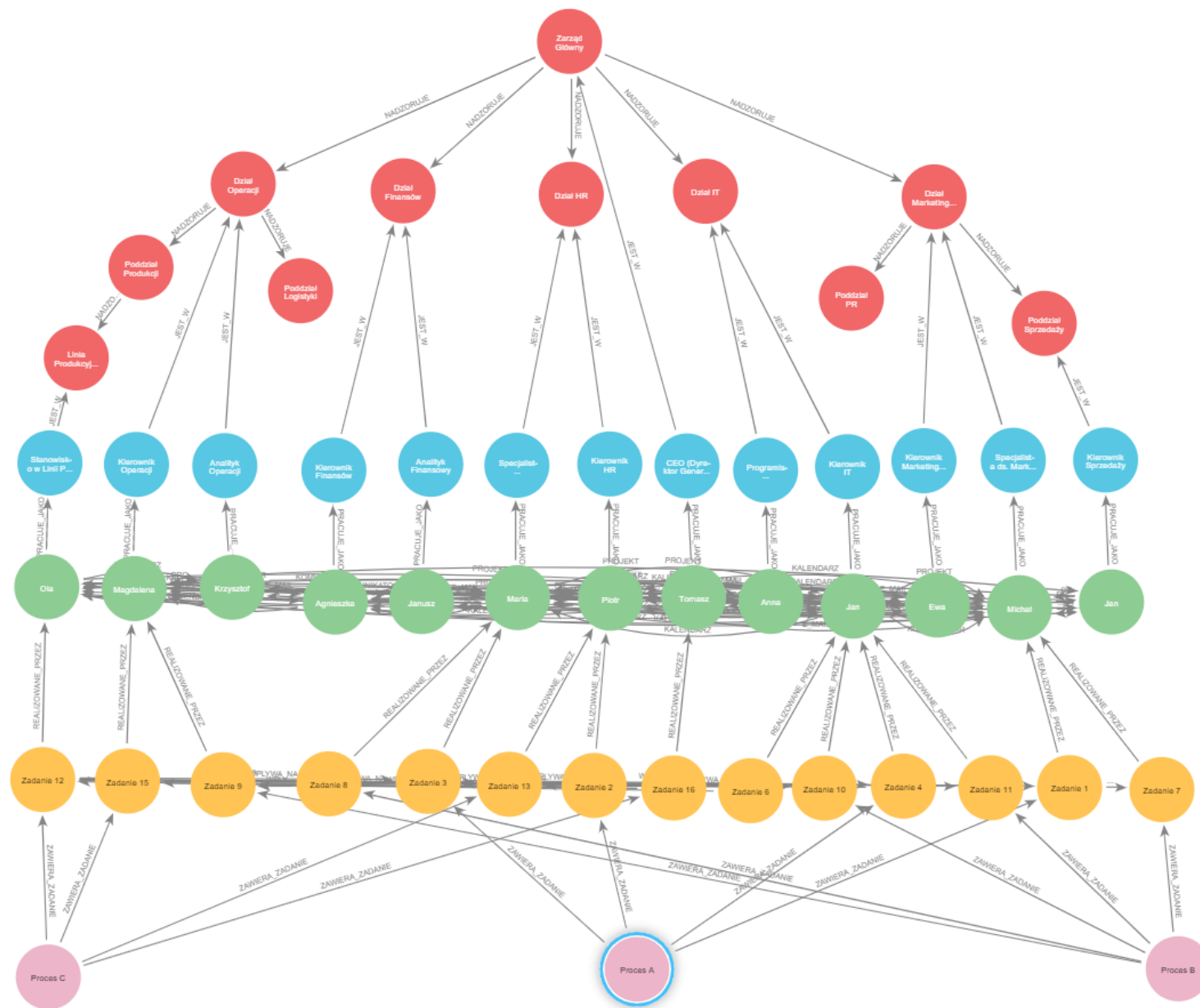
Połączenie wyników Algorytmów PageRankong i Louvain



Graf liderów organizacji



Heterogeniczny graf organizacji



Przypadek użycia - Relacyjne Zapytania

Mapowanie zależności zadań

Bazy grafowe pozwalają na łatwą wizualizację i zrozumienie złożonych zależności między zadaniami, co jest nieocenione przy planowaniu projektów i alokacji zasobów.



Cypher - Relacyjne Zapytania

Cypher

to język zapytań deklaratywny, stworzony przez Neo4j dla baz danych grafowych. Umożliwia on efektywne tworzenie, modyfikowanie oraz przeszukiwanie grafów w bazie danych Neo4j.

Cypher skupia się na tym, co chcemy osiągnąć, a nie na tym, jak to osiągnąć

```
MATCH (o:Osoba)-[p:PRACUJE_DLA]->(f:Firma)
WHERE f.nazwa = 'Alfa' AND o.wiek > 30
RETURN o.imie, o.wiek, f.nazwa
ORDER BY o.wiek DESC LIMIT 10
```

CREATE, MERGE

- Tworzenie danych

MATCH

- Wyszukiwanie danych

SET, REMOVE

- Aktualizacja danych

DELETE

- Usuwanie danych

WITH, RETURN

- Kontrola przepływu

Mapowanie zależności zadań

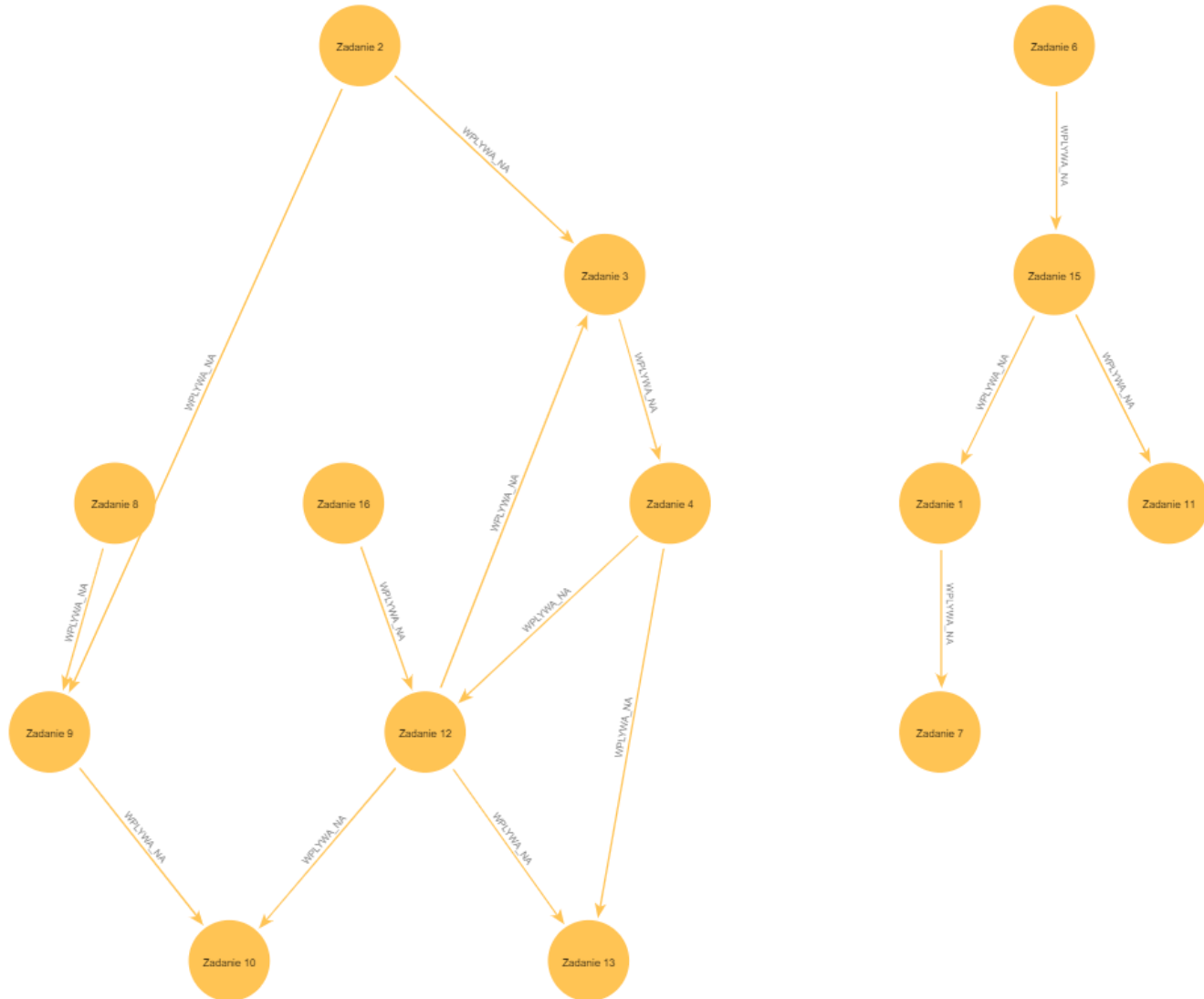
Eksploracja Danych

Zebranie danych o zadaniach realizowanych w organizacji w ramach procesów i projektów.



Tworzenie Grafu

Utworzenie grafu, w którym węzły reprezentują zadania a krawędzie ich wzajemny wpływ na siebie (zależności).



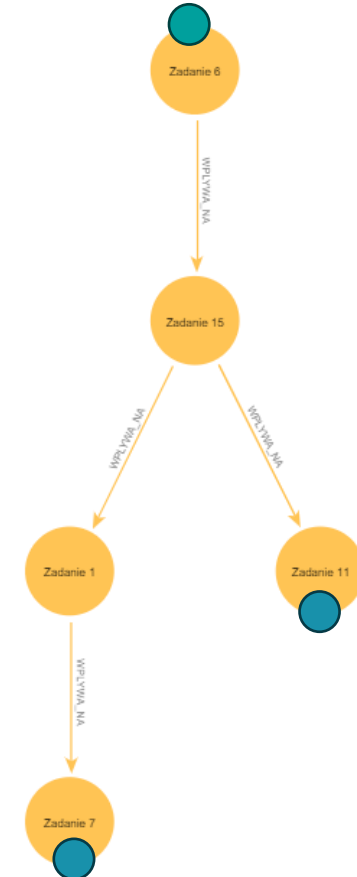
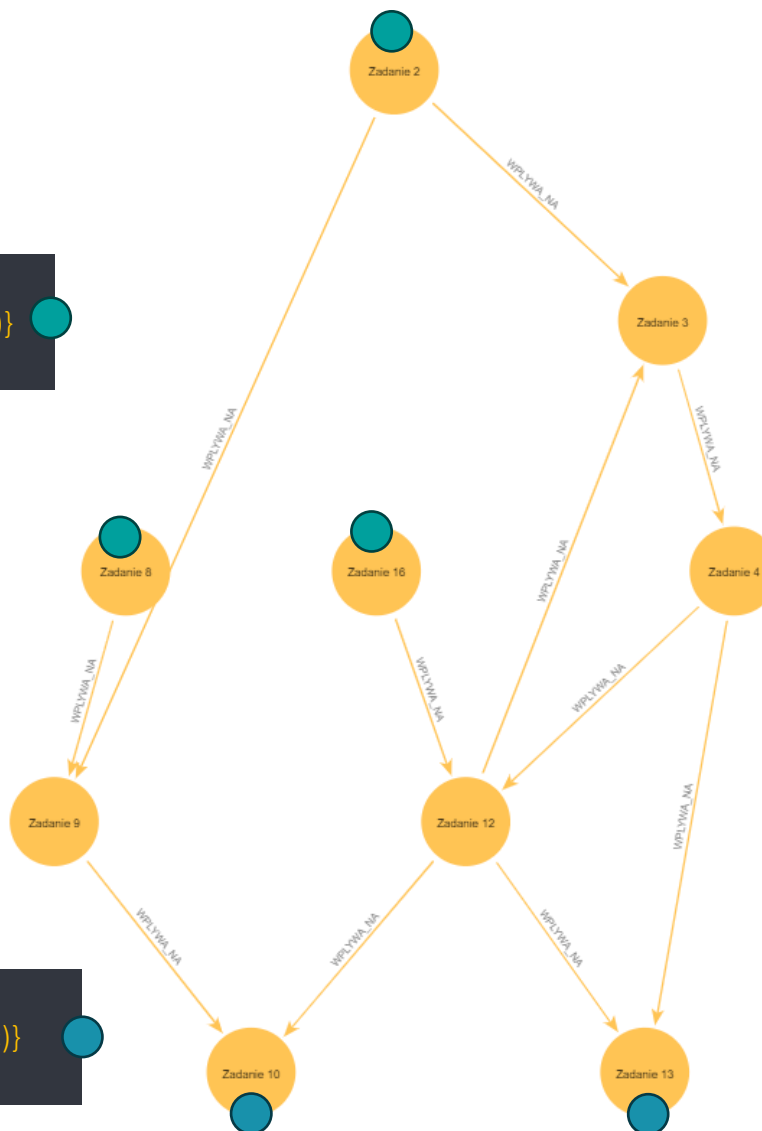
Relacyjne zapytania (Cypher)

Zadania startowe (inicjujące)
(korzenie w drzewie)

```
MATCH (korzen:Zadanie)
WHERE NOT EXISTS {MATCH (korzen)-[:WPLYWA_NA]-(rodzic)}
RETURN korzen
```

Zadania końcowe (cele)
(liście w drzewie)

```
MATCH (lisc:Zadanie)
WHERE NOT EXISTS {MATCH (dziecko)-[:WPLYWA_NA]-(lisc)}
RETURN lisc
```

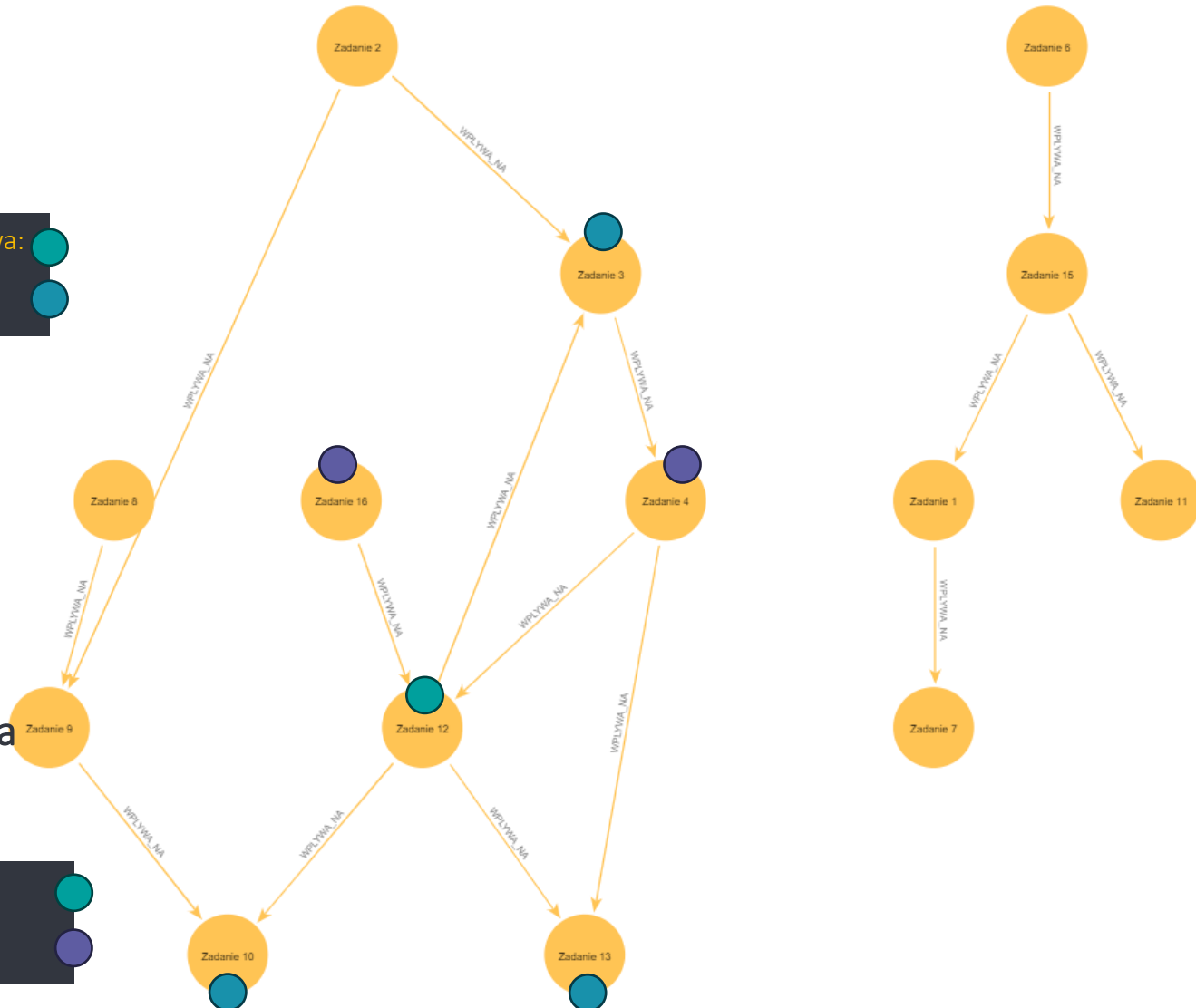


Relacyjne zapytania (Cypher)

Na jakie zadania wpływa bezpośrednio zadanie X
(wszystkie następniki zadania X)

Jakie zadania wpływają bezpośrednio na zadanie X
(wszystkie poprzedniki zadania X)

```
MATCH (x:Zadanie {nazwa: 'Zadanie 12'})<-[:WPLYWA_NA]-
(poprzednik:Zadanie)
RETURN x, poprzednik
```



Relacyjne zapytania (Cypher)

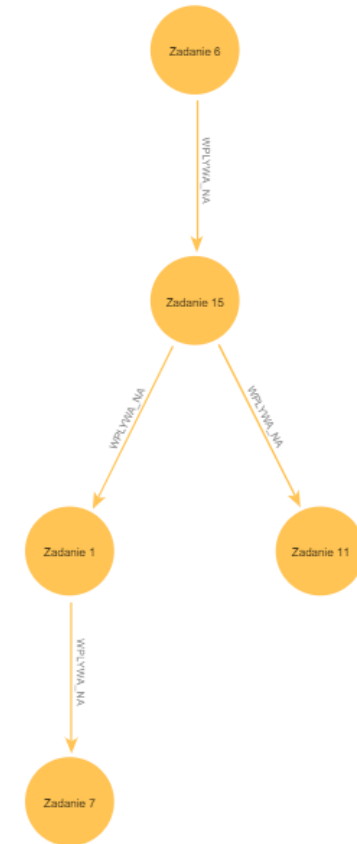
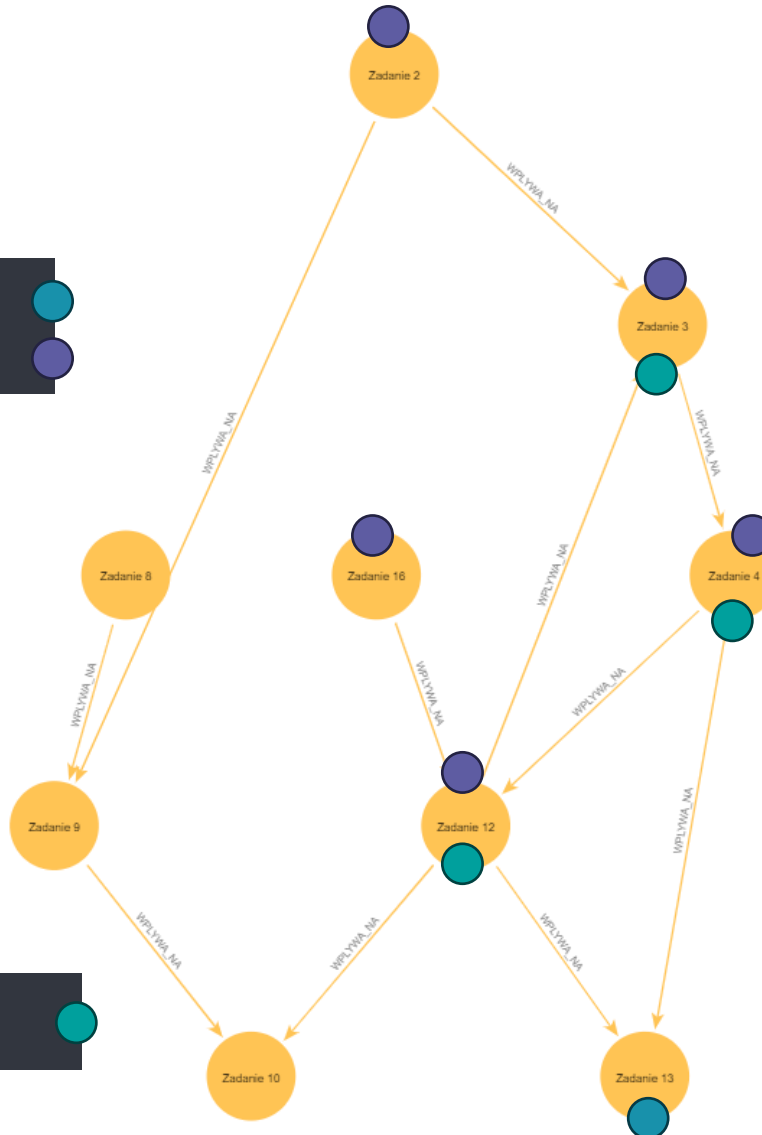
Harmonogram osiągnięcia celu X
(drogi dochodzące do zadania X)

```
MATCH (nastepnik:Zadanie)-[:WPLYWA_NA*]-(x:Zadanie  
{nazwa: 'Zadanie 13'})  
RETURN x, nastepnik
```

Cykl Sprzężenia Zwrotnego Zadań
(cykle w grafie)

Aspekt **pozytywny** i **negatywny**

```
MATCH path = (n:Zadanie)-[:WPLYWA_NA*]->(n:Zadanie)  
RETURN path
```



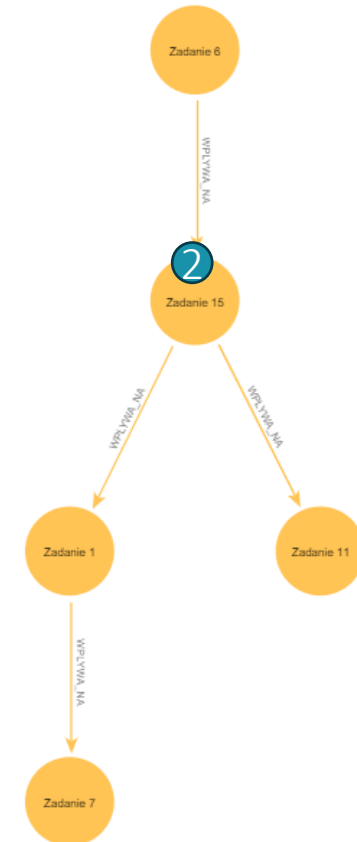
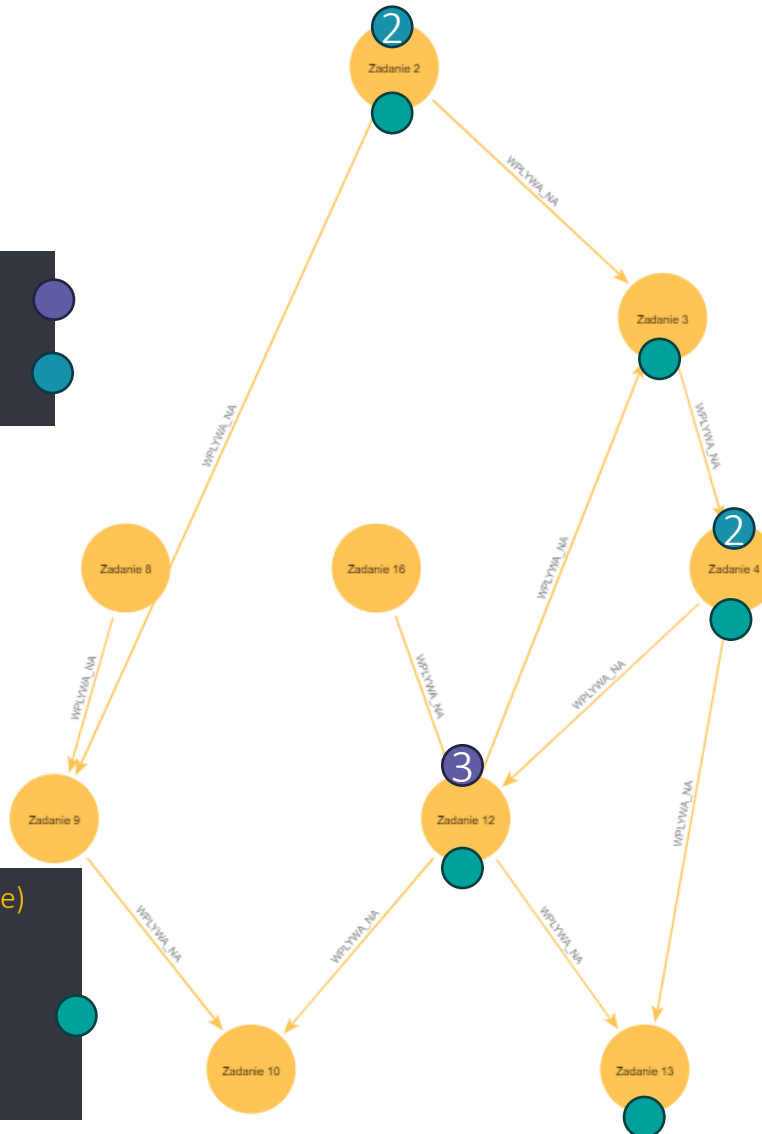
Relacyjne zapytania (Cypher)

Kluczowe zadania / krytyczne
(stopnie wychodzące węzłów)

```
MATCH ()<-[:WPLYWA_NA]-(n:Zadanie)
WITH n, COUNT(r) as outDegree
RETURN n, outDegree
ORDER BY outDegree DESC
```

Ścieżka krytyczna
(bezpośredni wpływ na całkowity
czas trwania projektu)

```
MATCH path=(start:Zadanie)-[:WPLYWA_NA*]->(end:Zadanie)
WITH path, REDUCE(s = 0, n IN nodes(path) | s +
n.czas_trwania) AS totalDuration
RETURN path AS CriticalPath, totalDuration
ORDER BY totalDuration DESC
LIMIT 1
```



Porównanie Cypher z SQL

Powiązanie **zadań** z **procesem A** i realizującymi je **osobami** pracującymi na **stanowiskach** w **komórkach** wewnętrznych organizacji

Cypher

```
MATCH d=(n:Proces {nazwa: 'Proces A'})-[:ZAWIERA_ZADANIE]->()-[:REALIZOWANE_PRZEZ]->()-[:PRACUJE_JAKO]->()-[:JEST_W]->(k)
RETURN d
```

SQL

```
SELECT * FROM Proces
INNER JOIN Zadanie ON Proces.id = Zadanie.idProces
INNER JOIN Osoba ON Zadanie.idOsoba = Osoba.id
INNER JOIN Stanowisko ON Osoba.idStanowisko = Stanowisko.id
INNER JOIN Komorka ON Stanowisko.idKomorka = Komorka.id
WHERE Proces.nazwa = 'Proces A';
```

Warunek związku tylko 1:n



Narzędzia



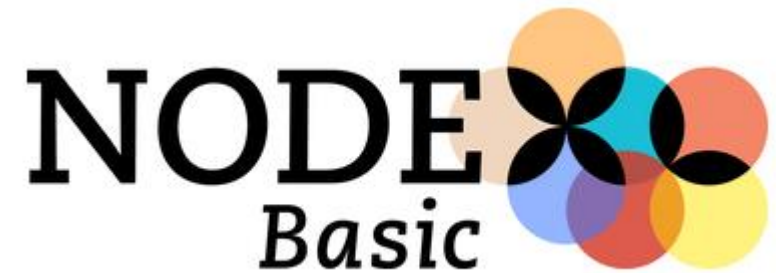
<https://neo4j.com/>



<https://cytoscape.org/>



<https://gephi.org/>



<https://nodexl.com/>



COMFORT ZONE



GROWTH ZONE