

Гак Артем, МП КН – 2

Бази Знань

Предметна область – теорія графів.

Дата сет – DBPedia

Вивантажено дата із теорії графів: 1058 елементів.

Приклад елемента:

```
"name": "(a,b)-decomposition",  
  "uri": "http://dbpedia.org/resource/(a,b)-decomposition",  
  "description": "in graph theory the a b decomposition of an undirected graph  
is a partition of its edges into a 1 sets each nducing a forest except one which  
induces a graph with maximum degree if this graph is also a forest then we call  
this a f a b",
```

Обробка дата сету – python script:

Опис розбито на токени (key_word) і далі побудовано інвертований індекс:

key_word -> Frequency (кількість у всій колекції)

Кількість унікальних key_words: 2858

```
{  
  "name": "(a,b)-decomposition",  
  "uri": "http://dbpedia.org/resource/(a,b)-decomposition",  
  "description": "in graph theory the a b decomposition of an undirected graph  
is a partition of its edges into a 1 sets each nducing a forest except one which  
induces a graph with maximum degree if this graph is also a forest then we call  
this a f a b",  
  "key_words": {  
    "graph": 1989,  
    "theory": 1969,  
    "decomposition": 27,  
    "undirected": 292,  
    "partition": 35,  
    "edges": 204,  
    "sets": 36,  
    "nducing": 2,  
    "forest": 15,  
    "except": 3,  
    "induces": 2,  
    "maximum": 49,  
    "degree": 68,  
    "also": 114,  
    "call": 2  
  }  
},
```

І потім додано параметр freq_percent – відсоток входження цього слова:

Сума всіх frequencies: 22816

$102/22816 = 0.00447054698457223$

Key_word = problem:(frequency: 102, freq_percent: 0.00447054698457223)

Завантаження в neo4j – load script, plugin from json

```
CALL apoc.load.json("file:///converted_data.json")
YIELD value
MERGE (c:GraphConcept {name: value.name})
SET c.uri = value.uri, c.description = value.description
WITH c, value
UNWIND keys(value.key_words) AS keyword
MERGE (kw:Keyword {name: keyword})
ON CREATE SET kw.frequency = value.key_words[keyword]
MERGE (c)-[:Has_KeyWord]->(kw);
```

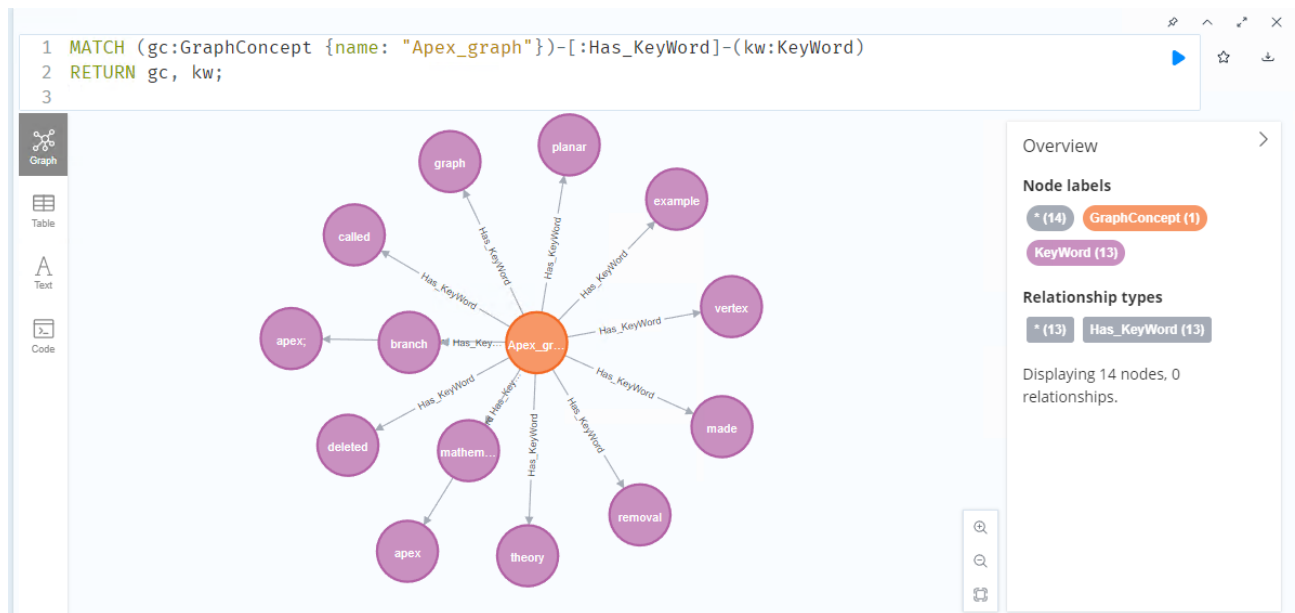


Використання, запити

1) Знайти ключові слова до заданого концепту.

Наприклад: користувач пише статтю і хоче підібрати ключові слова

```
MATCH (gc:GraphConcept {name: "Apex_graph"})-[:Has_KeyWord]-(kw:Keyword)
RETURN gc, kw;
```



2) Знайти концепти за ключовими словами

Наприклад: користувач цікавиться які Концепти мають задані «властивості»

```
MATCH (kw:Keyword)-[:Has_KeyWord]-(gc:GraphConcept)
WHERE kw.name IN ["clique", "complexity", "coloring"]
RETURN kw, gc;
```

Наприклад, користувач хоча подосліджувати речі пов'язані із кліками, складністю і розфарбуванням.

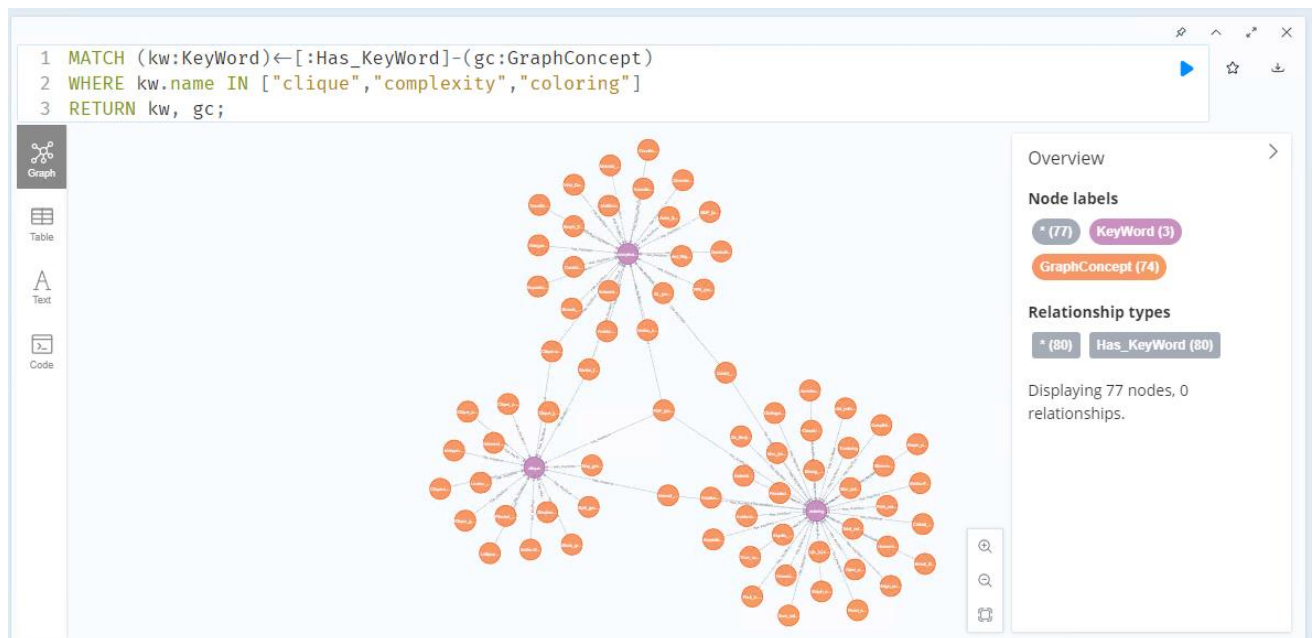
Результати:

«FNP_(complexity)» = ["clique", "complexity", "coloring"]

«Clique-width», «Tardos_function» = ["clique", "complexity"]

«Interval_graph» = ["clique", "coloring"]

«Daniel_Kráľ» = ["complexity", "coloring"]

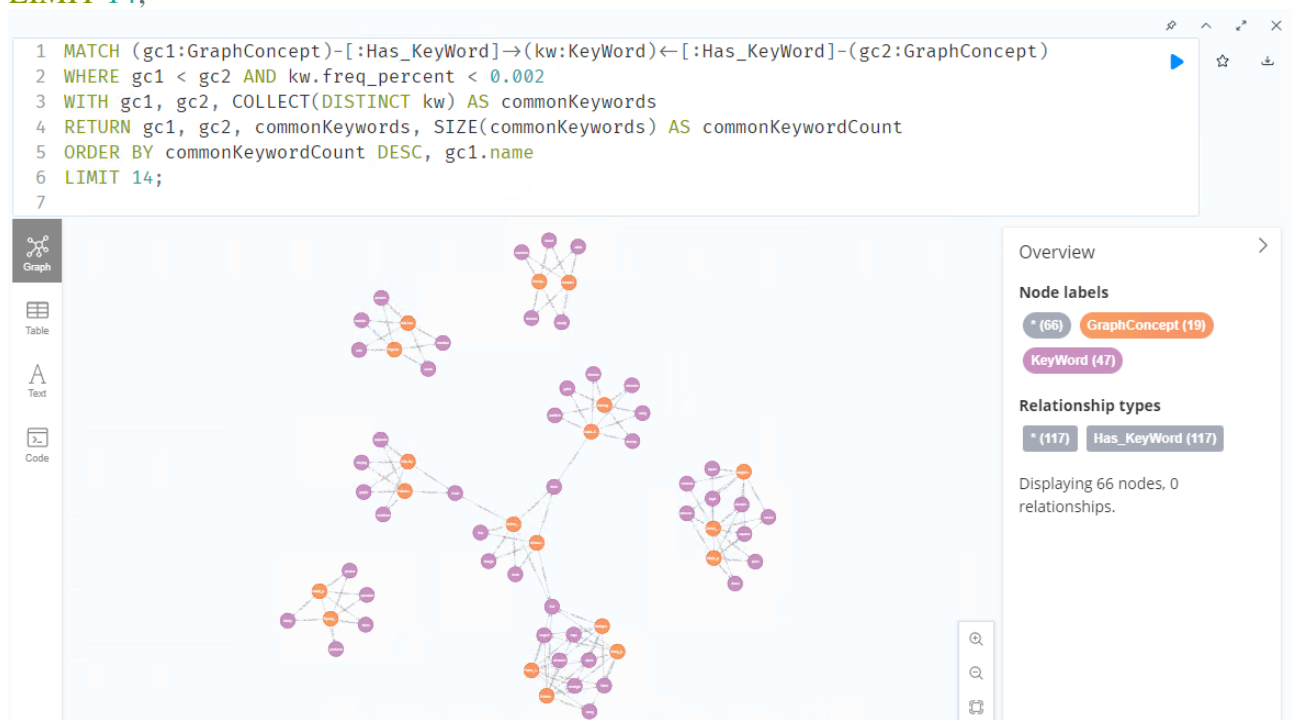


3) Найбільш unіque пов'язані графи

Знайти пари графів, які мають спільні key-words із найменшою частотою на всю базу знань.

Тобто такі-собі пари унікально пов'язаних графів.

```
MATCH (gc1:GraphConcept)-[:Has_KeyWord]->(kw:Keyword)<-[:Has_KeyWord]-(gc2:GraphConcept)
WHERE gc1 < gc2 AND kw.freq_percent < 0.002
WITH gc1, gc2, COLLECT(DISTINCT kw) AS commonKeywords
RETURN gc1, gc2, commonKeywords, SIZE(commonKeywords) AS commonKeywordCount
ORDER BY commonKeywordCount DESC, gc1.name
LIMIT 14;
```



gc1.name	gc2.name	commonKeywords	Count
"Foster_cage"	"Robertson–Wegner_graph"	["cage", "wong", "four", "robertson", "meringer", "wegner", "foster", "others"]	8
"Foster_cage"	"Meringer_graph"	["cage", "four", "robertson", "meringer", "wegner", "foster", "others"]	7
"Foster_cage"	"Wong_graph"	["cage", "wong", "four", "meringer", "wegner", "foster", "others"]	7
"King's_graph"	"Rook's_graph"	["represents", "squares", "chess", "chessboard", "piece", "moves", "legal"]	7
"Knight's_graph"	"Rook's_graph"	["square", "represents", "connects", "squares", "chessboard", "moves", "legal"]	7
"Meringer_graph"	"Robertson–Wegner_graph"	["cage", "four", "robertson", "meringer", "wegner", "foster", "others"]	7
"Robertson–Wegner_graph"	"Wong_graph"	["cage", "wong", "four", "meringer", "wegner", "foster", "others"]	7
"Dulmage–Mendelsohn_decomposition"	"Gallai–Edmonds_decomposition"	["decomposition", "partition", "using", "blossom", "gallai", "edmonds"]	6
"Meringer_graph"	"Wong_graph"	["cage", "four", "meringer", "wegner", "foster", "others"]	6
"Automotive_navigation_system"	"Turn-by-turn_navigation"	["within", "based", "identify", "examines", "shortest"]	5
"Edge-transitive_graph"	"Half-transitive_graph"	["automorphism", "transitive", "words", "transitively", "acts"]	5
"Fulkerson–Chen–Anstee_theorem"	"Gale–Ryser_theorem"	["graph;", "result", "sequence", "obeying", "conditions"]	5
"Graph_product"	"Zig-zag_product"	["product", "produces", "binary", "operation", "takes"]	5
"Grötzsch's_theorem"	"Herbert_Grötzsch"	["three", "free", "triangle", "four", "tzsch"]	5

4) Концепти які мають мало ключових слів (найменш вивчені)

MATCH (n)-[:Has_KeyWord]->(w)
RETURN n, COLLECT(w), **COUNT**(w) **as** s
ORDER BY s **DESC**
LIMIT _

n.name	
"Jinyoung_Park_(mathematician)"	2 ["graph", "theory"]
"Witold_Lipski"	2 ["graph", "theory"]
"Italo_Jose_Dejter"	2 ["graph", "theory"]
"Richard_A._Brualdi"	2 ["graph", "theory"]
"Sajal_K._Das"	2 ["graph", "theory"]
"End_(graph_theory)"	2 ["graph", "theory"]
"Henda_Swart"	2 ["graph", "theory"]
"Ray_(graph_theory)"	2 ["graph", "theory"]
"Renu_C._Laskar"	2 ["graph", "theory"]
"Arc_(graph_theory)"	2 ["graph", "theory"]

"Saidur_Rahman_(professor)"	2 ["graph", "theory"]
"Category:Graph_minor_theory"	3 ["minor", "graph", "theory"]

5) Слова які належать малій кількості концептів (найменш вивчені)

```

MATCH (n)-[:Has_KeyWord]->(w)
RETURN n, COLLECT(w), COUNT(w) as s
ORDER BY s DESC
LIMIT 7

```

w.name	s	COLLECT(n.name)
"hexagonal"	1	["26-fullerene_graph"]
"taxicab"	1	["Absolute_difference"]
"fullerene"	1	["26-fullerene_graph"]
"induces"	1	["(a,_b)-decomposition"]
"call"	1	["(a,_b)-decomposition"]
"pentagonal"	1	["26-fullerene_graph"]
"negations"	1	["2-satisfiability"]

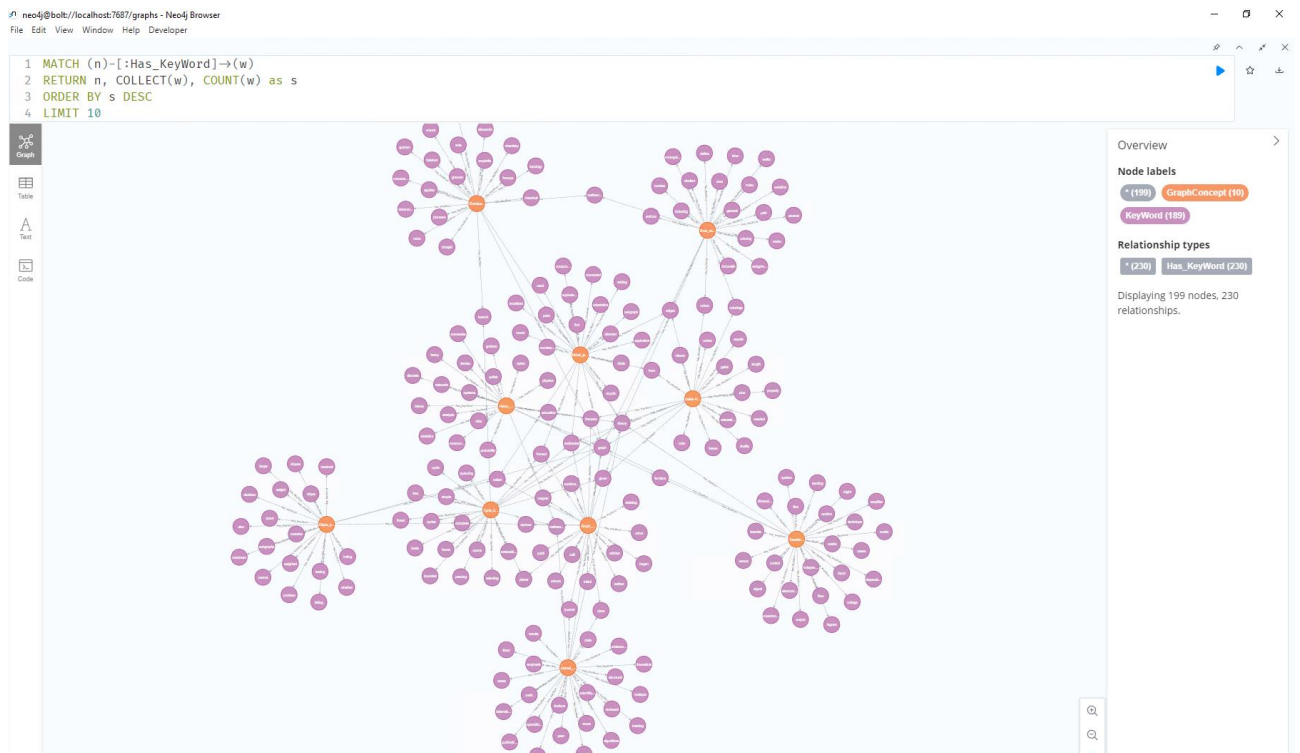
6) Знайти GraphConcepts, що мають найбільше KeyWords

```

MATCH (n)-[:Has_KeyWord]->(w)
RETURN n, COLLECT(w), COUNT(w) as s
ORDER BY s DESC
LIMIT 10

```

Concept name	Count related keywords
"Journal_of_Graph_Theory"	27
"Transfer_function"	27
"Thue_number"	24
"Moral_graph"	23
"Graph_minor"	22
"Chemical_graph_theory"	22
"Henry_O._Pollak"	22
"Cycle_basis"	21
"Clique_problem"	21
"Gallai-Hasse-Roy-Vitaver_theorem"	21



```
1 MATCH (n)-[:Has_KeyWord]->(w)
2 RETURN n.name, COUNT(w) as s, COLLECT(w.name)
3 ORDER BY s DESC
4 LIMIT 10
```

	n.name	s	COLLECT(w.name)
1	"Journal_of_Graph_Theory"	27	["algorithms", "theoretical", "mathematics", "review", "theory", "wetenschappelijk", "areas", "journal", "riodique", "structural", "peer", "specializing", "publication", "emphasis", "reviewed", "results", "cials", "ondenworpen", "scientif"]
2	"Transfer_function"	27	["electronics", "rosenbrock", "dimensional", "independent", "technique", "matrix", "output", "function", "lagrang", "signal", "dependent", "flow", "semilog", "cases", "like", "might", "unction", "control", "graph", "versus", "voltage", "
3	"Thue_number"	24	["vertex", "several", "variation", "colors", "number", "path", "general", "chromatic", "define", "coloring", "nonrepetitive", "area", "including", "assignment", "studied", "edges", "colorings", "index", "graph", "walks", "thue", "theory"]
4	"Moral_graph"	23	["equivalent", "directed", "chain", "orientation", "undirected", "called", "edges", "used", "graph", "pairs", "moral", "connected", "component", "form", "subgraph", "find", "counterpart", "adding", "equivalently", "acyclic", "formed", "
5	"Graph_minor"	22	["undirected", "called", "minor", "planar", "minors", "began", "wagner", "eymour", "initit", "dans", "graph", "theorem", "deleting", "journal", "neither", "xxii", "publi", "combinatorial", "formed", "complete", "articles", "theory"]
6	"Chemical_graph_theory"	22	["chemistry", "pioneers", "hosoya", "topology", "trinajst", "graovac", "riassumeva", "branch", "acquiste", "graph", "alexandru", "balaban", "ante", "applies", "haruo", "volumi", "chemical", "conoscenze", "theory", "gutman", "nsiv"]
7	"Henry_O_Pollak"	22	["education", "pollak", "data", "mathematics", "probability", "epian", "graham", "landau", "analysis", "networks", "graph", "discrete", "namesake", "henry", "theorem", "systems", "blems", "function", "physics", "communication", "
8	"Cycle_basis"	21	["given", "forest", "undirected", "planar", "mathematics", "tree", "simple", "cycles", "basis", "branch", "cycle", "bounded", "graph", "space", "panning", "spanning", "embedding", "formed", "theory", "forms", "selecting"]
9	"Clique_problem"	21	["larger", "cliques", "called", "subgraphs", "decision", "testing", "weight", "also", "inding", "maximal", "acent", "graph", "clique", "contains", "problem", "whether", "maximum", "cannot", "complete", "listing", "weighted"]
10	"Gallai-Hasse-Roy-Vitaver_theorem"	21	["given", "undirected", "color", "needed", "colorings", "orientations", "edges", "length", "hasse", "duality", "graph", "gallai", "colors", "plus", "property", "theorem", "form", "umber", "equals", "theory", "vitaver"]

7) Знайти KeyWords, що мають найбільше GraphConcepts

```
MATCH (n)-[:Has_KeyWord]->(w)
RETURN w.name, COUNT(n) as s, COLLECT(n.name)
ORDER BY s DESC
LIMIT 10
```

Key_word	Count related concepts
"graph"	1049
"theory"	1040
"mathematical"	199

