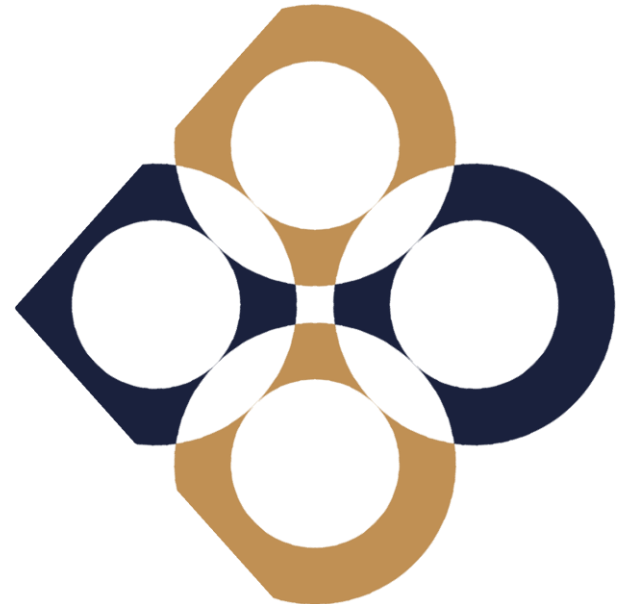


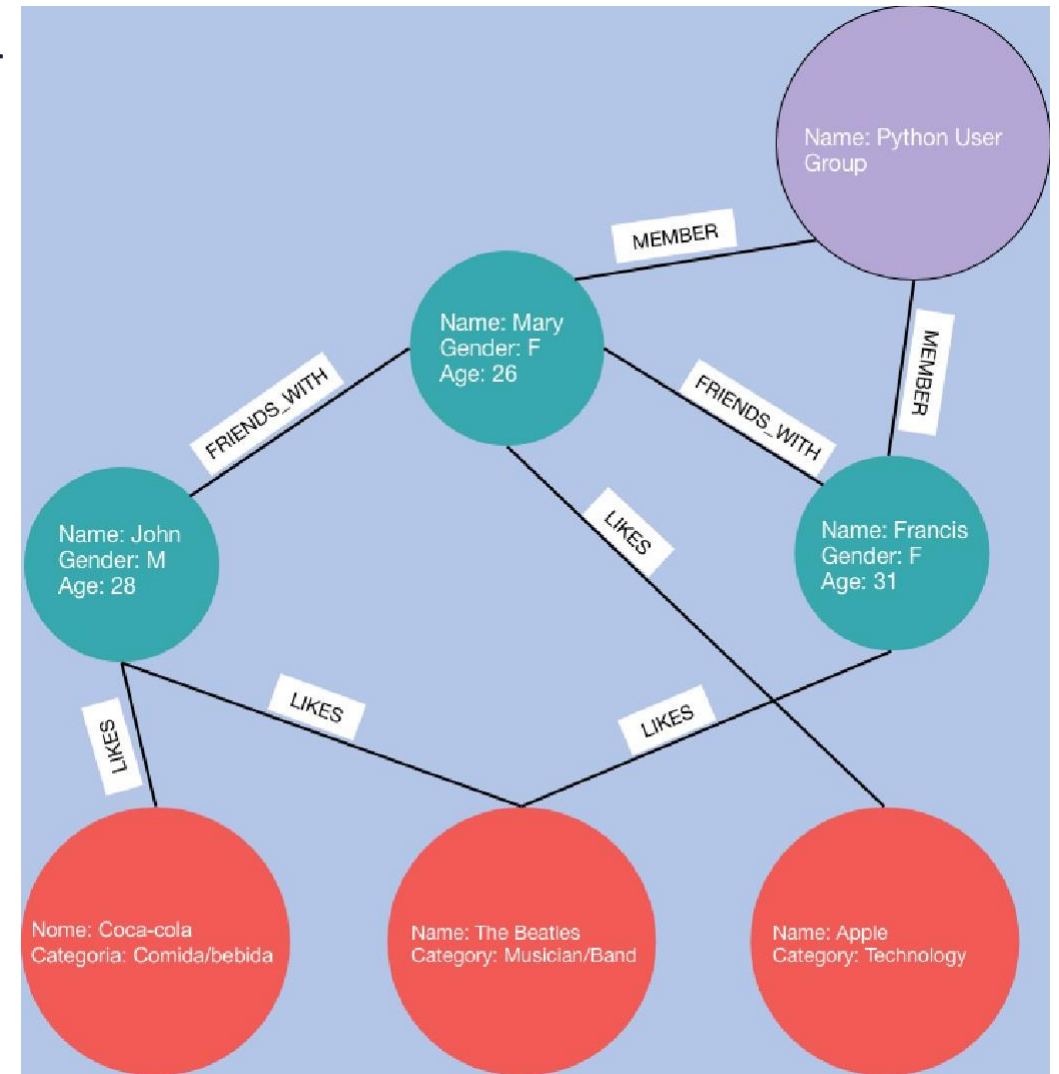
# Adatbázisok gyakorlat 09

Gráf adatbázisok



Olyan adatbázisok, amelyek az adatok tárolására és megjelenítésére gráf struktúrát alkalmaznak

- ❑ A gráf csúcsaiban vannak az adatok
  - ❑ Az adatok sémája nem rögzített
- ❑ A gráf élei jelentik a kapcsolatokat
  - ❑ Az élek irányítottak
  - ❑ Az éleknek adott név a kapcsolatra jellemző



<https://medium.com/labcodes/graph-databases-talking-about-your-data-relationships-with-python-b438c689dc89>

# Gráf adatbázisok – Neo4j

- ☐ A legismertebb gráf adatbázis
- ☐ A csúcsok ~ entitások, objektumok
  - ☐ Lehetnek tulajdonságaik (kulcs-érték párok)
    - ☐ Az értékek primitív adattípusok
    - ☐ A tulajdonságok (részben) indexelhetők
    - ☐ Megadható UNIQUE kényszer
    - ☐ Nincs NULL elem
  - ☐ Lehetnek címkéik
- ☐ A kapcsolatok
  - ☐ Van nevük
  - ☐ Lehetnek tulajdonságaik
  - ☐ Indexelhetők



The #1 Database for Connected Data

# Neo4j - Cypher

- ☐ A Neo4j preferált lekérdező nyelve
- ☐ Deklaratív (nem procedurális )nyelv
- ☐ Minta egyezéseket vizsgál
- ☐ Az emberi gondolkodáshoz közel álló nyelv
- ☐ Záradékok használata (pl: WHERE, ORDER BY)

# Fontosabb Cypher adattípusok

Típus	Példa	Megjegyzés
Integer	13	Tulajdonság típus
Float	3.14	Tulajdonság típus
String	'Hello', "World"	Tulajdonság típus
Boolean	true, false	Tulajdonság típus
Date	"2019-06-01"	Tulajdonság típus
Time	"21:40:32"	Tulajdonság típus
DateTime	"2019-09-25T06:29:39Z"	Tulajdonság típus
Node	(a:Actor)	Szerkezet típus
Relationship	[d:Directed]	Szerkezet típus
Path	(a:Actor)-[:Acted_in]->(m:Movie)	Szerkezet típus
List	[0, 1, 2]	Összetett típus
Map	{kulcs1: érték1, kulcs2: érték2 ...}	Összetett típus

# Fontosabb Cypher operátorok

Operátor típus	Példák
Matematikai	+, -, *, /, %, ^
Összehasonlító	=, <, >, <>, <=, >=, IS NULL, IS NOT NULL
Szöveg összehasonlító	STARTS WITH, ENDS WITH, CONTAINS
Logikai	NOT, AND, OR, XOR
Szöveg	+ (összefűzés), =~ (regex)
Aggregációs	DISTINCT
Tulajdonság (property)	. (csomópont vagy kapcsolat tulajdonság elérése) = (csomópont vagy kapcsolat tulajdonságok felülírása) += (csomópont vagy kapcsolat tulajdonság módosítása, hozzáadása)
Lista	IN (tartalmazást vizsgál) + (összefűz) [ ] (listaelemek elérése)

# Neo4j - lekérdezések

MATCH( ) - Csúcsok, kapcsolatok, tulajdonságok, címkék és minták keresése az adatbázisban

- ☐ A SQL SELECT-hez hasonló elven működik
- ☐ A lekérdezés által visszaadott értékeket a RETURN kulcsszó után adhatjuk meg
- ☐ A lekérdezés eredményét a WHERE kulcsszó után megadott feltételekkel szűrhetjük
- ☐ A megjelenítendő eredményt a LIMIT kulcsszóval korlátozhatjuk
- ☐ Az eredményt többféle nézetben (Graph, Table, Text, Code) is megtekinthetjük

# Neo4j – Egyszerű lekérdezések I.

```
MATCH (n)  
RETURN n
```

Listázza az összes csúcsot

```
MATCH (p:Person)  
RETURN p  
LIMIT 1
```

Megjeleníti a legelső személyt

```
MATCH (p:Person {name: 'Tom Hanks'})  
RETURN p
```

Megjeleníti Tom Hanks adatait

```
MATCH (:Person {name: 'Tom Hanks'})-[:DIRECTED]->(movie:Movie)  
RETURN movie.title
```

Megjeleníti, hogy Tom Hanks  
milyen film(ek)et rendezett



# Neo4j – Egyszerű lekérdezések II.

```
MATCH (p:Person {name:'Tom Hanks'})-[rel:DIRECTED]-(m:Movie)
RETURN p.name AS name, p.born AS `Year Born`, m.title AS title,
m.released AS `Year Released`
```

Megjeleníti Tom Hanks és az általa rendezett film egyes adatait

```
MATCH (:Person)-[:DIRECTED]->(m:Movie)
RETURN DISTINCT m.released
```

Megjeleníti azon éveket, amikor filmeket rendeztek

```
MATCH (j:Person)
WHERE j.born = 1955
RETURN j
```

Megjeleníti az 1955-ben született személyeket

```
MATCH (j:Person)
WHERE NOT j.born = 1955
RETURN j
```

Megjeleníti azokat, akik nem 1955-ben születtek

# Neo4j – Egyszerű lekérdezések III.

```
MATCH (p:Person)
WHERE p.name STARTS WITH 'M'
RETURN p.name
```

Megjeleníti az M betűvel kezdődő személyeket

```
MATCH (p:Person)
WHERE p.name CONTAINS 'a'
RETURN p.name
```

Megjeleníti azon személyeket, akik nevében van „a” betű

```
MATCH (p:Person)
WHERE p.name ENDS WITH 'n'
RETURN p.name
```

Megjeleníti azon személyeket, akik neve n-re végződik

```
MATCH (p:Person)
WHERE p.name =~ 'Jo.*'
RETURN p.name
```

Reguláris kifejezéssel szűr a személyek nevére

# Neo4j – Egyszerű lekérdezések IV.

```
MATCH (m:Movie)
WHERE ID(m) IN [0, 5, 9]
RETURN m
```

Megjeleníti a 0, 5 és 9 azonosítójú filmeket

```
MATCH (p:Person)-[d:REVIEWED]->(m:Movie)
RETURN p, d, m
```

Megjeleníti, hogy melyik személy milyen filmekről írt kritikát

```
MATCH (p:Person)-[d:WROTE]->(m:Movie)
WHERE not exists ((p)-[:ACTED_IN]->(m))
RETURN p, d, m
```

Megjeleníti azokat a személyeket és filmeket, ahol az író nem szerepelt a filmben

```
MATCH (p:Person)-[:ACTED_IN]->(m:Movie)<-[:ACTED_IN]-(p2:Person)
WHERE p.name= 'Gene Hackman'
AND exists( (p2)-[:DIRECTED]->(m) )
RETURN p, p2, m
```

Kivel és milyen filmben szerepelt együtt Gene Hackman, ha a másik szereplő egyben rendező is volt?

# Neo4j – Egyszerű lekérdezések V.

```
MATCH (p:Person)
WHERE p.name STARTS WITH 'J'
OPTIONAL MATCH (p)-[:DIRECTED]->(m)
RETURN p.name, m.title
```

Megjeleníti a személyeket és az általuk rendezett filmet (ha van olyan)

```
MATCH (p:Person)
RETURN count(*)
```

Megjeleníti, hogy hány személy van az adatbázisban

```
MATCH (p:Person)-[:FOLLOWS]->(p2:Person)
WITH p, count(*) AS db
RETURN p.name, db
```

Megjeleníti azt, hogy melyik személy hány másikat követ

```
MATCH (p:Person)-[:WROTE]->(m:Movie)
RETURN p.name, collect(m.title) AS filmek
```

Megjeleníti, hogy melyik személy milyen filmeket rendezett

# Neo4j – Egyszerű lekérdezések VI.

```
MATCH (p:Person)-[:ACTED_IN]->(m:Movie)
RETURN p.name, size(collect(m.title)) AS db
```

Megjeleníti, hogy melyik személy  
hány filmben szerepelt

```
MATCH (p:Person)-[:ACTED_IN]->(m:Movie)
RETURN p.name, size(collect(m.title)) as db
ORDER BY db DESC, p.name
LIMIT 5
```

Megjeleníti, hogy kik szerepeltek a  
legtöbb filmben – az első 5

```
MATCH (p:Person)-[r]->(p2:Person)
RETURN type(r), count(*)
```

Megjeleníti azt, hogy milyen  
típusú és hány db kapcsolat van a  
személyek között

```
MATCH (p:Person)-[r]->(m:Movie)
WHERE p.born IS NULL
RETURN p.name, type(r), m.title, avg(date().year-m.released)
```

Megjeleníti, hogy azok a személyek, akiknek  
nincs megadva a születési évük, milyen  
filmekkel vannak kapcsolatban, és a filmek  
átlagosan hány éve jelentek meg



**Köszönöm  
a figyelmet!**