

## Feladat leírása:

Adott relációs adatbázissal egyenértékű NoSQL adatbázis létrehozása. A relációs adatbázis legalább 3 táblás legyen, a táblák között legyenek kapcsolatok. A NoSQL adatbáziskezelő rendszer a tanultak bármelyike lehet. Beadandó a relációs adatbázis leírása, a NoSQL megfelelőjének leírása + 5 db lekérdezés a NoSQL adatbázison.

## Átalakítandó relációs adatbázis:

Tábla 1: Felhasználók

user\_id int (egyedi azonosító, elsődleges kulcs)

név nvarchar

email nvarchar

jelszó nvarchar

Tábla 2: Posztok

post\_id int (egyedi azonosító, elsődleges kulcs)

cím nvarchar

tartalom nvarchar

user\_id int (külső kulcs, kapcsolat a Felhasználók táblával)

Tábla 3: Kommentek

comment\_id int (egyedi azonosító, elsődleges kulcs)

tartalom nvarchar

user\_id int (külső kulcs, kapcsolat a Felhasználók táblával)

post\_id int (külső kulcs, kapcsolat a Posztok táblával)

Kapcsolatok: Felhasználók (1) - (n) Posztok (1) - (n) Kommentek

Átalakítás: skálázható gráf technológia

Használt Adatbázis: Neo4j

Felhasználók tábla:

Csúcsként jelenik meg, ahol a user\_id azonosító a kulcsa.

Tulajdonságok: név, email, jelszó.

Posztok tábla:

Csúcsként jelenik meg, ahol a post\_id azonosító a kulcsa.

Tulajdonságok: cím, tartalom.

Kapcsolat az Felhasználók táblával: Az Felhasználók táblával való kapcsolatot egy éllel jelöljük, ahol az él azonosítója a user\_id.

Kommentek tábla:

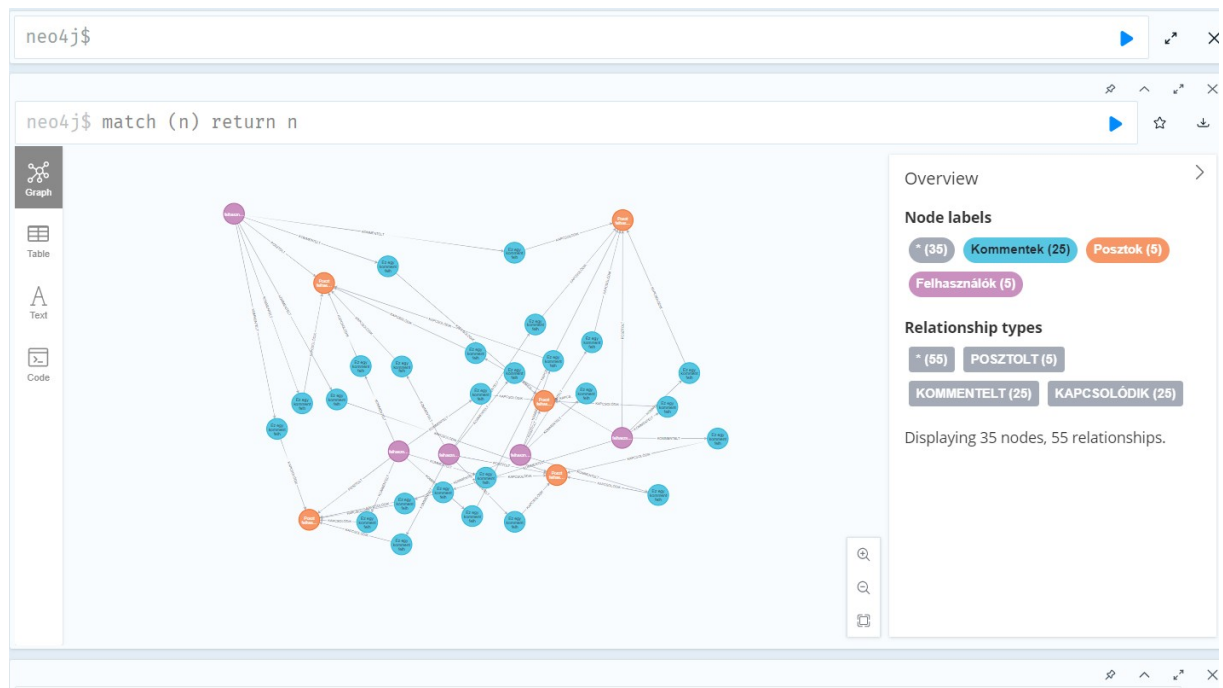
Csúcsként jelenik meg, ahol a comment\_id azonosító a kulcsa.

Tulajdonság: tartalom.

Kapcsolat a Felhasználók táblával: Az Felhasználók táblával való kapcsolatot egy éllel jelöljük ki, ahol az él azonosítója a user\_id.

Kapcsolat a Posztok táblával: A Posztok táblával való kapcsolatot egy másik éllel jelöljük ki, ahol az él azonosítója a post\_id.

A gráf adatbázist „cypherscript.txt”-ben található kóddal lehet létrehozni.



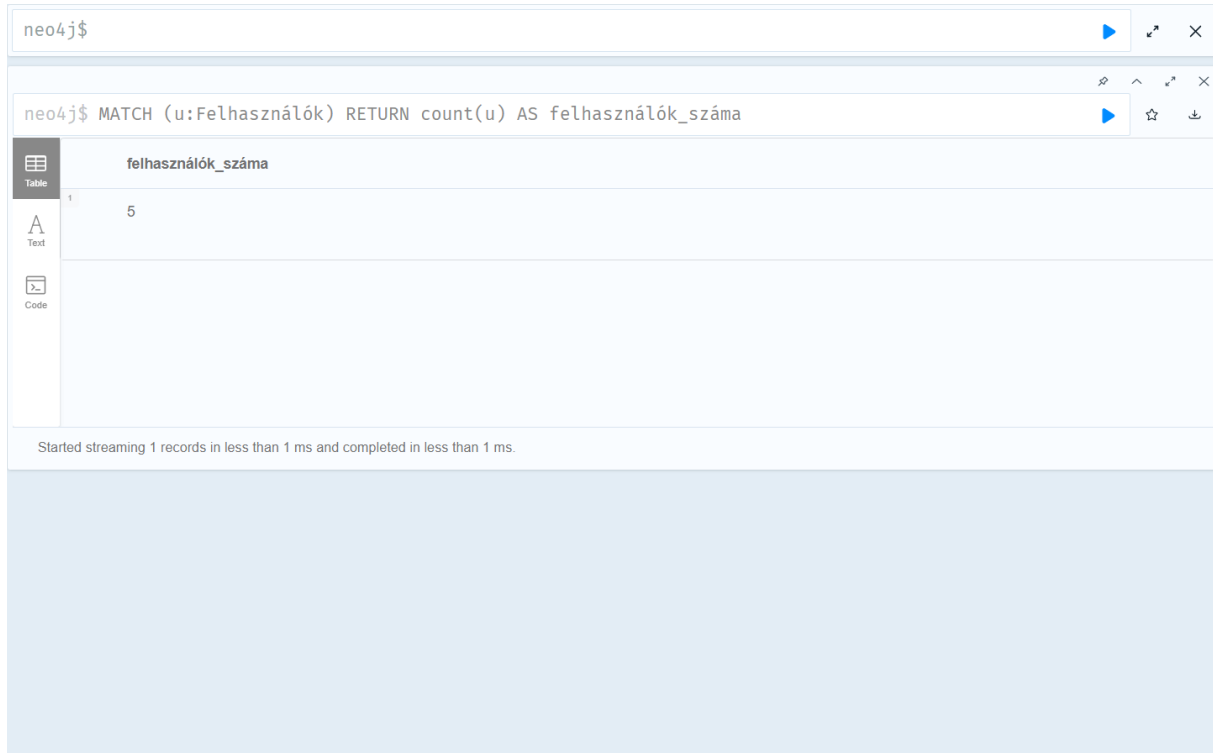
## 1, lekérdezés:

a,

//Ez a lekérdezés megszámolja a felhasználók és posztok számát az adatbázisban.

MATCH (u:Felhasználók)

RETURN count(u) AS felhasználók\_száma



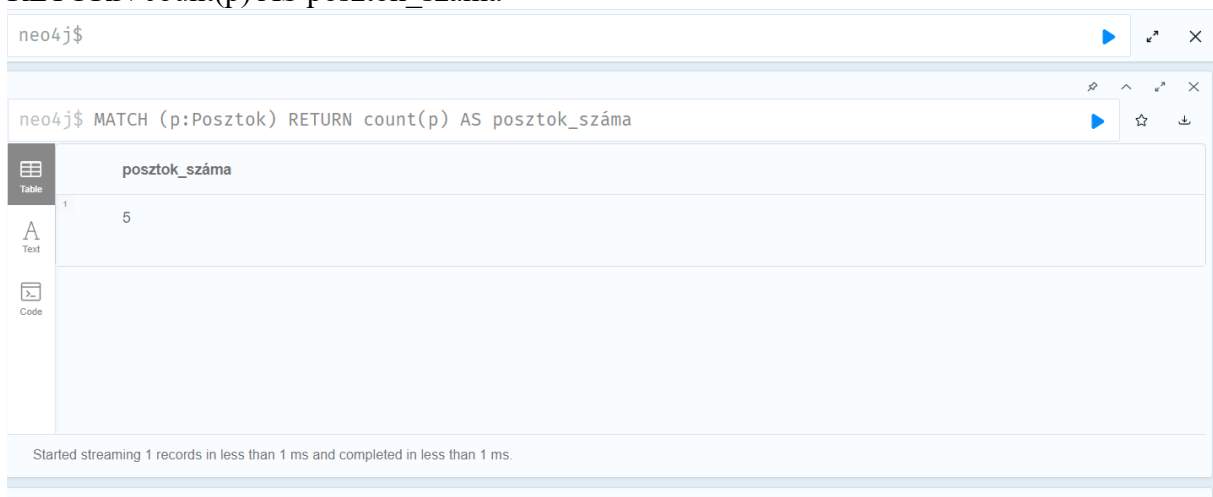
The screenshot shows the Neo4j query execution interface. The query entered is `neo4j$ MATCH (u:Felhasználók) RETURN count(u) AS felhasználók_száma`. The result is displayed in a table with one column, `felhasználók_száma`, and one row containing the value `5`. The interface includes a sidebar with icons for Table, Text, and Code, and a status bar at the bottom indicating that the query completed in less than 1 ms.

felhasználók_száma
5

b,

MATCH (p:Posztok)

RETURN count(p) AS posztok\_száma



The screenshot shows the Neo4j query execution interface. The query entered is `neo4j$ MATCH (p:Posztok) RETURN count(p) AS posztok_száma`. The result is displayed in a table with one column, `posztok_száma`, and one row containing the value `5`. The interface includes a sidebar with icons for Table, Text, and Code, and a status bar at the bottom indicating that the query completed in less than 1 ms.

posztok_száma
5

## 2, lekérés:

//Egy adott felhasználó posztjainak lekérése:

```
MATCH (u:Felhasználók {user_id: 'felhasznalo_1'})-[:POSZTOLT]->(p:Posztok)
```

```
RETURN u.név AS felhasználó, collect(p.cím) AS posztok
```



neo4j\$

```
neo4j$ MATCH (u:Felhasználók {user_id: 'felhasznalo_1'})-[:POSZTOLT]->(p:Posztok) RETURN u.név AS felhasználó, collect(p.cím) AS posztok
```

	felhasználó	posztok
1	"Aladár Péter"	["Poszt felhasznalo_1"]

Started streaming 1 records in less than 1 ms and completed after 1 ms.

## 3, lekérdezés:

//A legaktívabb felhasználók lekérése:

```
MATCH (u:Felhasználók)-[:POSZTOLT]->(p:Posztok)
```

```
WITH u, count(p) AS posztok_száma
```

```
ORDER BY posztok_száma DESC
```

```
RETURN u.név AS felhasználó, posztok_száma
```

```
LIMIT 5
```



neo4j\$

```
neo4j$ MATCH (u:Felhasználók)-[:POSZTOLT]->(p:Posztok) WITH u, count(p) AS posztok_száma ORDER BY posztok_száma DESC LIMIT 5
```

	felhasználó	posztok_száma
1	"Aladár Péter"	1
2	"Kovács Judit"	1
3	"Péter Árpád"	1
4	"János Ágnes"	1
5	"Ferenc István"	1

Started streaming 5 records after 1 ms and completed after 2 ms.

## 4, lekérdezés:

//a lekérdezés visszaadja azokat a posztokat, amelyekhez legalább 3 komment kapcsolódik.

MATCH (p:Posztok)←[:KAPCSOLÓDIK]-(k:Kommentek)

WITH p, count(k) AS kommentek\_száma

WHERE kommentek\_száma >= 3

RETURN p.cím AS poszt\_címe, kommentek\_száma

neo4j\$

neo4j\$ MATCH (p:Posztok)←[:KAPCSOLÓDIK]-(k:Kommentek) WITH p, count(k) AS kommentek\_száma WHERE ...

	poszt_címe	kommentek_száma
1	"Poszt felhasználó_5"	5
2	"Poszt felhasználó_1"	5
3	"Poszt felhasználó_2"	5
4	"Poszt felhasználó_3"	5
5	"Poszt felhasználó_4"	5

Started streaming 5 records in less than 1 ms and completed after 1 ms.

## 5.lekérdezés:

//Ez a lekérdezés visszaadja az összes felhasználó nevét, akik rendelkeznek poszttal, valamint //a posztok számát

MATCH (u:Felhasználók)-[:POSZTOLT]→(p:Posztok)

RETURN u.név AS felhasználó\_neve, COUNT(p) AS posztok\_száma

neo4j\$

neo4j\$ MATCH (u:Felhasználók)-[:POSZTOLT]→(p:Posztok) RETURN u.név AS felhasználó\_neve, COUNT(p)...

	felhasználó_neve	posztok_száma
1	"Aladár Péter"	1
2	"Kovács Judit"	1
3	"Péter Árpád"	1
4	"János Ágnes"	1
5	"Ferenc István"	1

Started streaming 5 records in less than 1 ms and completed in less than 1 ms.