

# Gemixque

Sistem de recomandări de jocuri video

Radu Damian

Dr. Cristian Frăsinaru

Facultatea de Informatică

2022

- 1 Descrierea problemei
- 2 Neo4j
- 3 Algoritmul de recomandare
- 4 Concluzii

- 1 Descrierea problemei
- 2 Neo4j
- 3 Algoritmul de recomandare
- 4 Concluzii

# Privirea în ansamblu a problemei

Trei elemente principale:

- Utilizatorul
- Jocul video
- Recenzia

Fie  $U$  mulțimea de utilizatori și  $G$  mulțimea de jocuri.

$$\forall u \in U, \exists G_u \subseteq G \text{ a.i.}$$

$$\forall g \in G_u, s(u, g) = r_{ug}$$

$$1 \leq r_{ug} \leq 10, r_{ug} \in \mathbb{N}$$

# Obiectiv

- Fie  $u \in U$ ,  $g \notin G_u$
- $\hat{s}(u, g) = ?$

- 1 Descrierea problemei
- 2 Neo4j
- 3 Algoritmul de recomandare
- 4 Concluzii

- Bază de date NoSQL de tip graf
- Datele sunt reținute prin intermediul nodurilor și muchiilor
- Utilizează limbajul de interogare Cypher



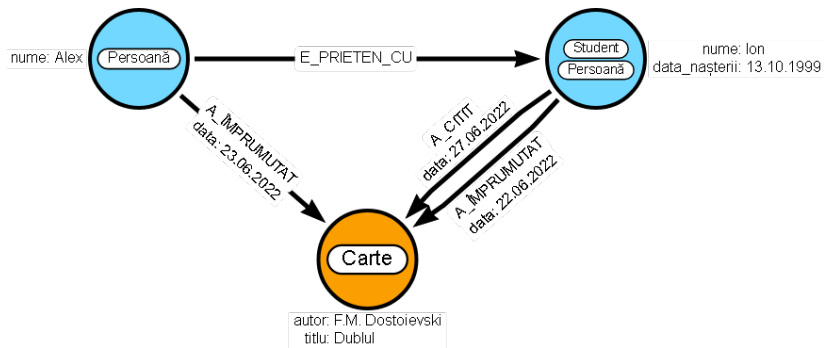
Un nod are următoarele trăsături:

- Reprezintă obiecte/entități
- Poate fi etichetat
- Poate avea proprietăți

O muchie(relație) este caracterizată de:

- Tipul acesteia
- Direcția
- Poate avea proprietăți

# Exemplu



# Limbajul de interogare Cypher

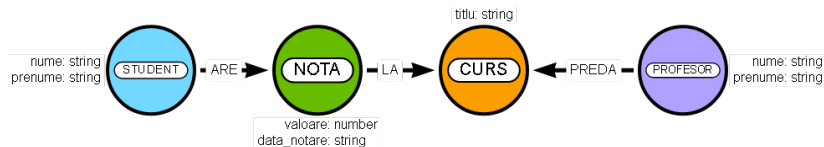
- Inspirat din SQL
- Prezintă un mod intuitiv de reprezenta noduri și relații prin intermediul unei sintaxe tip ASCII-art

## Studiu de caz:

- modelarea situației din cadrul unei facultăți (studenți, cursuri, profesori)
- exemplu interogare în SQL
- exemplu interogare în Cypher
- conceptul de *pattern-matching*

```
SELECT p.nume, p.prenume FROM NOTE n  
JOIN CURSURI c ON n.id_curs = c.id  
JOIN DIDACTIC d ON d.id_curs = c.id  
JOIN PROFESORI p ON p.id = d.id_profesor  
WHERE VALOARE = 10 AND ID_STUDENT = 36;
```

# Cypher vs. SQL

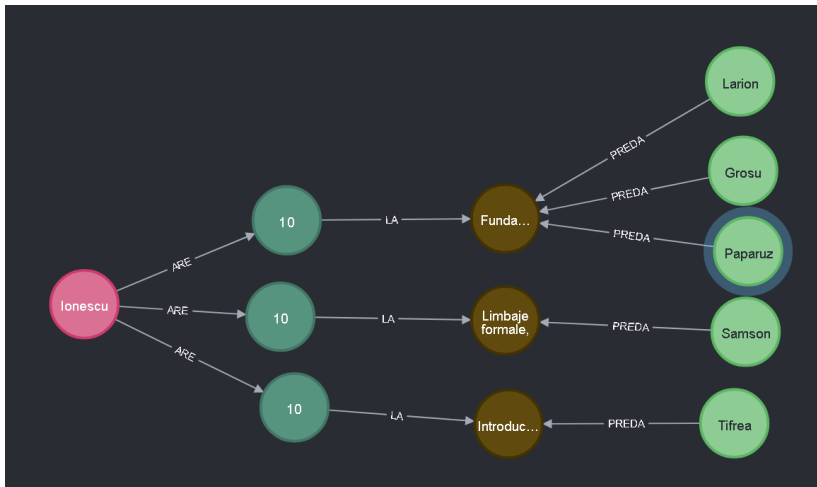


```
MATCH (s:STUDENT)-[:ARE]->(n:NOTA {valoare: 10}),  
      (n)-[:LA]->(:CURS)<-[:PREDA]-(p:PROFESOR)  
WHERE id(s) = 0  
RETURN p.num, p.prenume
```

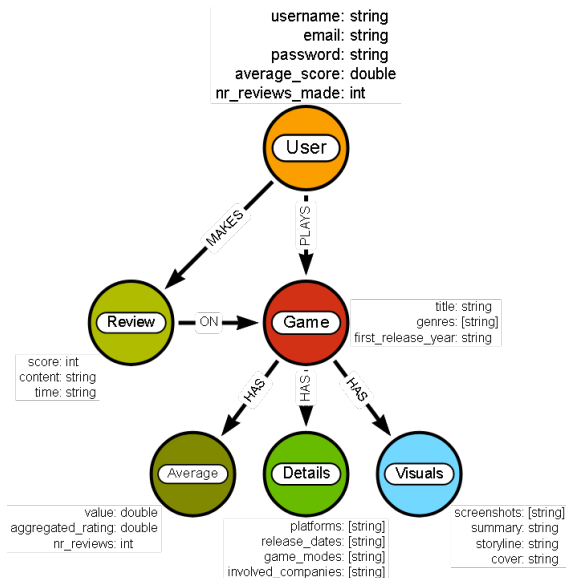


- Conceptul de *pattern-matching* este introdus prin intermediul clauzei MATCH
- În funcție de cum este definit *pattern*-ul în interogare, graful va fi parcurs într-un anumit mod

# Exemplu



# Schema bazei de date



- 1 Descrierea problemei
- 2 Neo4j
- 3 Algoritmul de recomandare
- 4 Concluzii

- Ipoteză: cantitate masivă de informații
- Problemă: filtrarea acestora
- Soluție: sistem de recomandare ce oferă conținut personalizat utilizatorilor

# Sistem de recomandări bazat pe filtrare colaborativă

- Resurse ce nu pot fi descrise prin metadate cu ușurință
- Matrice de scoruri utilizator-resursă
- Calcularea unei predicții pentru elementele lipsă din matrice

# Exemplu

	DOOM Eternal	Battlefield	Call of Duty	The Witcher 3: Wild Hunt	Dark Souls III
Ion	9	8	10	3	6
Gigel	8	9	?	4	5
Alex	5	7	4	10	9

$$s(u, g) = \frac{\sum_{u' \in \Omega_g} w_{uu'} \cdot r_{u'g}}{\sum_{u' \in \Omega_g} w_{uu'}}$$

$\Omega_g$  - mulțimea utilizatorilor care au atribuit un scor jocului  $g$

$r_{ug}$  - scorul oferit de utilizatorul  $u$  jocului  $g$

$w_{uu'}$  - gradul de similaritate între doi utilizatori



$$dev(u, g) = r_{ug} - \overline{r_u}$$

$$dev(\hat{u}, g) = \frac{1}{|\Omega_g|} \cdot \sum_{u' \in \Omega_g} dev(u', g)$$

$\overline{r_u}$  - media scorurilor utilizatorului  $u$

$$w\_dev\hat{v}(u, g) = \frac{\sum_{u' \in \Omega_g} w_{uu'} \cdot dev(u', g)}{\sum_{u' \in \Omega_g} |w_{uu'}|}$$

$$\hat{s}(u, g) = \overline{r_u} + w\_dev\hat{v}(u, g) \quad (1)$$

# Coeficientul lui Pearson

$$w_{uu'} = \frac{\sum_{g \in \Psi_{uu'}} dev(u, g) \cdot dev(u', g)}{\sqrt{\sum_{g \in \Psi_{uu'}} dev(u, g)^2} \cdot \sqrt{\sum_{g \in \Psi_{uu'}} dev(u', g)^2}}$$

$\Psi_{uu'}$  - mulțimea jocurilor în comun recenzate de utilizatorii  $u$  și  $u'$

$\Psi_u$  - mulțimea jocurilor recenzate de utilizatorul  $u$

$\Psi_{uu'} = \Psi_u \cap \Psi_{u'}$

# Algorithm - pseudocod

$u \in U, k \in \mathbb{N}$

getRecommendations( $u, k$ )

$S \leftarrow \emptyset$

*submulțime de  $k$  utilizatori similari cu  $u$  ordonați descrescător după pondere:*

$|U|^k = \{X \mid X \subseteq U \wedge |X| = k \wedge \operatorname{argmax}_X \sum_{x \in X} w_{ux}\}$

**for**  $u' \in |U|^k$  **do**

*jocuri recenzate de  $u'$  care nu sunt în comun cu jocurile recenzate de  $u$ :*

$G' = \Psi_{u'} \setminus \Psi_{uu'}$

**for**  $g' \in G'$  **do**

$S = S \cup \{g'\}$

*actualizare scor  $g'$  utilizând formula (1)*

**end for**

**end for**

- 1 Descrierea problemei
- 2 Neo4j
- 3 Algoritmul de recomandare
- 4 Concluzii

- Cu ajutorul Neo4j se pot rezolva probleme în care relațiile dintre entități au o importanță semnificativă
- Algoritmul de recomandare prezentat reprezintă un punct de pornire în dezvoltarea în amănunt a unui sistem de recomandare