UNIVERZA V LJUBLJANI · FRI

Tehnologija upravljanja podatkov

Neo4j / Cypher · Vektorske baze · RAG · Transakcije

Vaje · 90 minut · november 2026

Opombe diapozitivov vsebujejo rešitve za izvajalca.

POTEK

Kaj bomo danes izvedli?

Demonstracija → samostojna naloga → razlaga

10 min	**SQL ↔️ MongoDB ↔️ Neo4j**
30 min	Cypher, graf Jadralec–Čoln
20 min	Vektorji: KNN in kosinus
15 min	RAG: pridobi kontekst
15 min	Transakcije: vse ali nič

Kje se lahko vaje izvajajo?

Ni treba namestiti treh novih strežnikov.

BREZ NAMESTITVE

Neo4j GraphAcademy
Cypher s pravo bazo v brskalniku

BREZ NAMESTITVE

Google Colab
Notebooki za vektorje, RAG in SQLite

GraphAcademy ↗️

Google Colab ↗️

W3Schools MongoDB ↗️

Neo4j AuraDB ↗️

W3Schools: za MongoDB in SQL; za Cypher uporabimo uradno GraphAcademy.

TUP / FRI

Neo4j in Cypher

Grafni podatkovni model in poizvedbe po povezavah.

≈ 30 minut

Iz treh tabel v en graf

Isti podatki, drugačen način predstavitve.

RELACIJSKI MODEL

Jadralec (jid, ime, rating)
Coln (cid, ime, barva)
Rezervacija (jid, cid, dan)

GRAFNI MODEL

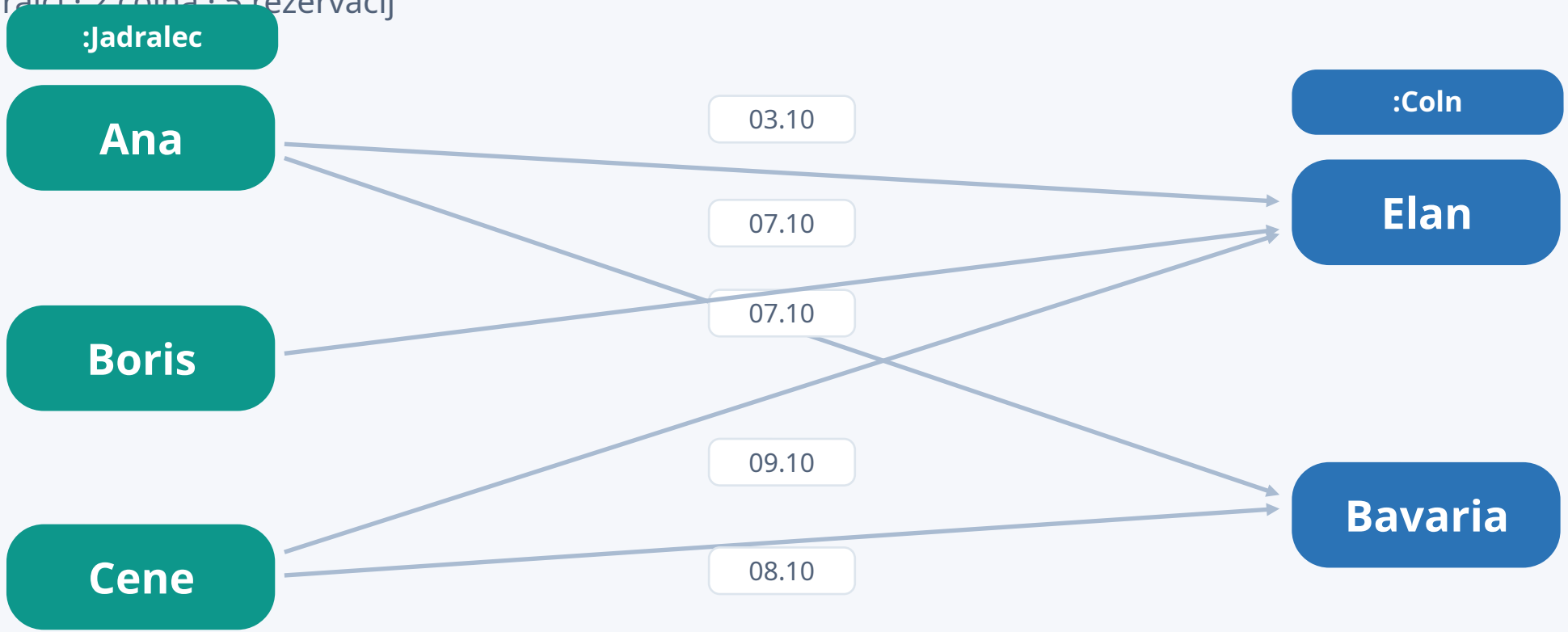
(:Jadralec) -[:REZERVIRAL]-> (:Coln)

Datum je lastnost povezave: r.dan

SQL JOIN ↔ Cypher MATCH

Naš mali graf

3 jadralci · 2 čolna · 5 rezervacij



Priprava testnega grafa (1/2)

Izvedi v prazni testni instanci AuraDB ali lokalnem Neo4j.

```
MERGE (j1:Jadralec {jid:1}) SET j1.ime='Ana', j1.rating=8
MERGE (j2:Jadralec {jid:2}) SET j2.ime='Boris', j2.rating=6
MERGE (j3:Jadralec {jid:3}) SET j3.ime='Cene', j3.rating=9
MERGE (c1:Coln {cid:10}) SET c1.ime='Elan', c1.barva='bela'
MERGE (c2:Coln {cid:20}) SET c2.ime='Bavaria', c2.barva='modra'
```

MERGE najprej poskusi ujemanje vzorca; CREATE vedno ustvarja.

Priprava testnega grafa (2/2)

Nadaljevanje istega skripta: pet rezervacij

```
WITH j1,j2,j3,c1,c2
MERGE (j1)-[:REZERVIRAL {dan:date('2026-10-03')}]->(c1)
MERGE (j1)-[:REZERVIRAL {dan:date('2026-10-07')}]->(c2)
MERGE (j2)-[:REZERVIRAL {dan:date('2026-10-07')}]->(c1)
MERGE (j3)-[:REZERVIRAL {dan:date('2026-10-08')}]->(c2)
MERGE (j3)-[:REZERVIRAL {dan:date('2026-10-09')}]->(c1)
```

```
MATCH (j:Jadralec)-[r:REZERVIRAL]->(c:Coln)
RETURN j.ime,c.ime,r.dan ORDER BY j.ime,r.dan;
```

DEMONSTRACIJE

Tri osnovne poizvedbe v jeziku Cypher

Primerjaj s SQL JOIN, GROUP BY in HAVING/podpoizvedbo.

```
MATCH (j:Jadralec)-[r:REZERVIRAL]->(c:Coln)
RETURN j.ime, c.ime, r.dan ORDER BY j.ime, r.dan;
```

```
MATCH (j:Jadralec)
OPTIONAL MATCH (j)-[r:REZERVIRAL]->(:Coln)
RETURN j.ime, count(r) AS stevilo ORDER BY stevilo DESC;
```

```
MATCH (j:Jadralec) WITH avg(j.rating) AS avgR
MATCH (j:Jadralec) WHERE j.rating > avgR
RETURN j.ime, j.rating ORDER BY j.rating DESC;
```

Cypher: prvi dve nalogi

N1 + N2

N1

Izpiši vse čolne in njihove barve. Kaj je razlika med MATCH in CREATE?

N2

Poišči vse čolne, ki jih je rezervirala Ana; uredi jih po datumu.

Cypher: manjkajoče povezave in skupni sosedi

N3 + N4

N3

Poišči jadralce BREZ rezervacij. Zakaj uporabimo OPTIONAL MATCH?

N4

Poišči pare jadralcev, ki so kdaj rezervirali isti čoln.

SAMOSTOJNA NALOGA

Cypher: spreminjanje grafa

N5 · SET in MERGE

NALOGA N5

Spremeni barvo čolna Bavaria v zeleno.

Dodaj Borisovo rezervacijo čolna Bavaria za 10. oktober 2026.

Za preverjanje napiši še poizvedbo, ki izpiše vse Borisove rezervacije.

RAZMISLEK

Kdaj je rezervacija povezava in kdaj vozlišče?

Povezava ima lahko lastnosti, vendar nima vedno dovolj bogate identitete.

SAMO DATUM

(:Jadralec)-[:REZERVIRAL {dan}]->(:Coln)

STATUS, PLAČILO, UDELEŽENCI

(:Jadralec)-[:USTVARIL]->(:Rezervacija)
(:Rezervacija)-[:ZA_COLN]->(:Coln)

Vprašanje: katera rešitev je bolj razširljiva, če pridejo še računi in plačila?

Neo4j iz Pythona / Jupyterja

Priporočen uradni paket `neo4j`; lokalna ali oblačna baza.

```
python -m pip install neo4j

from neo4j import GraphDatabase
from getpass import getpass
uri = 'bolt://localhost:7687'
with GraphDatabase.driver(uri, auth=('neo4j', getpass())) as d:
  d.verify_connectivity()
  rows, _, _ = d.execute_query(
    'MATCH (j:Jadralec) RETURN j.ime AS ime',
    database_='neo4j')
  print([r['ime'] for r in rows])
```

V Colabu uporabite AuraDB: lokalni localhost ni dostopen iz oblaka.

TUP / FRI

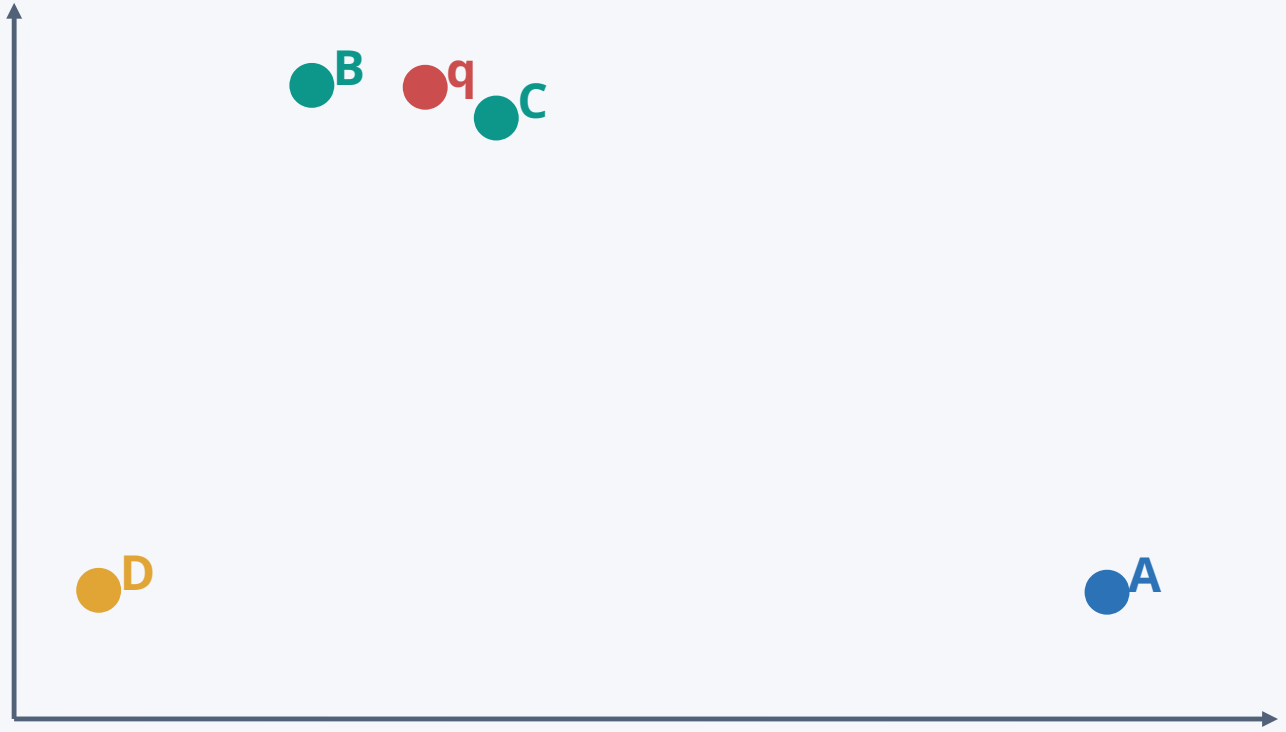
Vektorske baze

Najbližji sosedi, kosinus in razlika med vektorjem ter besedilom.

≈ 20 minut

Vektorji kot točke v prostoru

Za vprašanje q poiščemo k najbolj podobnih vektorjev.



2D prikaz prvih dveh koordinat (tretja ni narisana)

ODGOVORI

Kaj pomeni k = 2?

Zakaj morajo biti vektorji enake dimenzije?

Kosinusna podobnost ali evklidska razdalja?

Merita različni lastnosti dveh vektorjev.

KOSINUS

Podobnost smeri (kota):

$$\cos(a,b) = (a \cdot b) / (||a||\ ||b||)$$

L2 RAZDALJA

Geometrijska razdalja:

$$d(a,b) = \sqrt{\Sigma(ai - bi)^2}$$

Vprašanje: ali bosta razvrstitvi vedno enaki?

UČNI PRIMER

Štirje dokumenti in eno vprašanje

Ročno določeni 3D vektorji – ne pravi embeddingi.

A · Grafne baze	(0.95, 0.10, 0.05)
B · Vektorsko iskanje	(0.05, 0.98, 0.10)
C · RAG in iskanje	(0.20, 0.90, 0.10)
D · Transakcije	(0.05, 0.10, 0.97)

q = (0.15, 0.95, 0.15)

Kateri je najbližji?

Vektorji: nalogi V1 in V2

V1 + V2

V1

Spremeni q v (0.95, 0.05, 0.10). Kateri dokument je prvi?

V2

Izračunaj kosinusno podobnost in L2-razdaljo za B in prvotni q. Ali metodi vedno enako razvrstita?

SAMOSTOJNA NALOGA

Vektorji: naloga V3

Pomen dimenzij in parametra k

V3

Kaj bi se zgodilo, če bi primerjali vektorja različnih dolžin?

V3

Kaj pomeni k = 3 pri metodi KNN?

Dodatno: zakaj cos(a, 0) ni definiran?

RAG

Pridobivanje relevantnih odlomkov pred ustvarjanjem odgovora.

≈ 15 minut

Kako deluje RAG?

Dva dela: priprava znanja in odgovor na vprašanje.



Vprašanje → embedding vprašanja → top-k odlomkov → kontekst za LLM

RAG zmanjša tveganje izmišljanja, ne daje pa garancije pravilnosti.

Mini RAG demonstracija brez strežnika

V našem Notebooku je to iskanje po prekrivanju besed, ne vektorski RAG.

```
dokumenti = [
 ("D1", "Cypher uporablja MATCH ..."),
 ("D2", "Vektorske baze poiščejo ..."),
 ("D3", "Transakcija zagotavlja ..."),
 ("D4", "RAG najprej pridobi...")
]
viri = retrieve("Kaj zagotavlja transakcija?", top_k=2)
print(viri)
```

Kaj izgubimo, če iščemo samo identične besede?

RAG: nalogi R1 in R2

R1 + R2

R1

Kateri dokument vrne `retrieve("Kaj zagotavlja transakcija?")`?

R2

Primerjaj natančno iskanje po ključnih besedah s semantičnim iskanjem.

RAG: zanesljivost in navedbe virov

R3 + R4

R3

Zakaj RAG še vedno lahko ustvari napačen odgovor?

R4

Kako bi hranili vir in številko strani posameznega odlomka?

Transakcije

BEGIN, COMMIT, ROLLBACK in pojem »vse ali nič«.

≈ 15 minut

ACID: štiri zahteve

Transakcija je logična enota dela.

A · ATOMARNOST

Vse operacije ali nobena.

C · KONSISTENTNOST

Ohranjanje pravil podatkov.

I · IZOLACIJA

Sočasne transakcije se ustrezno ločijo.

D · TRAJNOST

Potrjeni podatki ostanejo zapisani.

Primer prenosa 30 enot

Začetno stanje: račun A = 100, račun B = 60




```
BEGIN;
UPDATE racun SET saldo=saldo-30 WHERE id=1;
UPDATE racun SET saldo=saldo+30 WHERE id=2;
COMMIT;
```

Invariant: vsota sredstev 100 + 60 = 70 + 90 = 160

Transakcije brez strežnika: SQLite

Podatki so lahko v pomnilniku; `sqlite3` je del Pythona.

```
import sqlite3
con = sqlite3.connect(':memory:', isolation_level=None)
con.execute('CREATE TABLE racun (id INTEGER, saldo INTEGER)')
con.executemany('INSERT INTO racun VALUES (?,?)', [(1,100),(2,60)])
con.execute('BEGIN')
con.execute('UPDATE racun SET saldo=saldo-30 WHERE id=1')
con.execute('UPDATE racun SET saldo=saldo+30 WHERE id=2')
con.execute('ROLLBACK')
```

Kaj vrne SELECT id, saldo FROM racun ORDER BY id?

Transakcije: nalogi T1 in T2

T1 + T2

T1

Namesto ROLLBACK izvedi COMMIT. Kakšno je končno stanje?

T2

Kaj se zgodi, če potrdimo samo odtegljaj, druge posodobitve pa ne izvedemo?

Transakcije: nalogi T3 in T4

T3 + T4

T3

Kako preprečiti negativen saldo? Kaj preverimo pred COMMIT?

T4

Zakaj sta dve sočasni transakciji zahtevnejši od ene zaporedne?

Hitri kviz: graf in vektorji

Ustno ali v parih – 4 vprašanja

- 1 Kaj sta vozlišče in povezava?
- 2 Kateri Cypher najde jadralce, ki so rezervirali Elan?
- 3 Kdaj uporabimo OPTIONAL MATCH?
- 4 Ali je pomen besedila vedno enak kosinusni podobnosti vektorjev?

Hitri kviz: RAG in transakcije

Še 4 vprašanja

- 5 Kaj je embedding in kdo ga izračuna?
- 6 V RAG pride najprej retrieval ali generiranje?
- 7 Kaj naredi ROLLBACK pri prenosu?
- 8 Ali Python sqlite3 potrebuje zunanji strežnik?

POVEZAVE

Za študente: viri in naslednji koraki

Za praktično delo uporabite spremljevalni Jupyter Notebook.

- Neo4j – Cypher Fundamentals ↗️
- Neo4j – Cypher Manual ↗️
- Google Colab – Notebook ↗️
- Neo4j – Python Driver ↗️
- W3Schools – MongoDB ↗️
- SQLite – Transactions ↗️

Ponovi naloge N1–N5, V1–V3, R1–R4 in T1–T4.