UNIVERZA V LJUBLJANI · FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Rešitve vaj: Neo4j, vektorske baze, RAG in transakcije

Ločeno gradivo za izvajalca vaj · odgovori na N1–N5, V1–V3, R1–R4, T1–T4 in osem vprašanj za ponavljanje

Namen: rešitve so pripravljene posebej za dokument TUP_vaje_Neo4j_vektorske_baze_RAG_transakcije.pdf in spremljevalni Jupyter Notebook. Koda uporablja enak graf Jadralec–Čoln in enake štiri dokumente/vektorje kot gradivo za študente.

Oznake: N = Neo4j/Cypher, V = vektorji/KNN, R = RAG/retrieval, T = transakcije. Pri vsaki nalogi je rešitev, pričakovani izid in kratka didaktična razlaga.

Sklop	Naloge	Prikaz
Neo4j / Cypher	N1–N5	AuraDB Free, Neo4j Desktop ali Query/Browser
Vektorji / KNN	V1–V3	Jupyter, standardni Python
RAG / retrieval	R1–R4	Jupyter; iskanje po besedah, brez LLM
Transakcije	T1–T4	Jupyter + SQLite
Kviz	8 vprašanj	ustno ali projekcija

Pred izvedbo: za Neo4j uporabi prazen testni graf in skript s strani 3 študentskega PDF-ja. Po N5 se podatki spremenijo; za točno primerjavo rezultatov N1–N4 jih je najbolje izvesti pred N5. Vektorski in transakcijski primeri delujejo brez zunanjih knjižnic.

Pričakovani rezultati niso izmišljeni: Cypher rezultate smo logično izpeljali iz petih rezervacij v danem vzorčnem grafu; numerične vektorske rezultate smo izračunali v Pythonu. Navedena koda Cypher v tem okolju ni bila izvedena na dejanskem strežniku Neo4j.

1 Neo4j in Cypher: rešitve N1–N3

N1 · Čolni in barve; MATCH in CREATE

```
MATCH (c:Coln)
RETURN c.cid AS cid, c.ime AS ime, c.barva AS barva
ORDER BY cid;
```

cid	ime	barva
10	Elan	bela
20	Bavaria	modra

Razlaga: MATCH prebere vozlišča, ki ustrezajo vzorcu; CREATE ustvari nova vozlišča oziroma povezave. Ustvarjanje z MERGE najprej išče ustrezen vzorec, nato ga po potrebi ustvari. MATCH ne spreminja podatkov sam po sebi.

N2 · Čolni, ki jih je rezervirala Ana, urejeni po datumu

```
MATCH (j:Jadralec {ime:'Ana'})-[r:REZERVIRAL]->(c:Coln)
RETURN c.ime AS coln, r.dan AS dan
ORDER BY dan;
```

coln	dan
Elan	2026-10-03
Bavaria	2026-10-07

Razlaga: iz vozlišča Ana sledimo povezavam REZERVIRAL; datum je lastnost povezave, ne čolna.

N3 · Jadralci brez rezervacij; OPTIONAL MATCH

```
MATCH (j:Jadralec)
OPTIONAL MATCH (j)-[r:REZERVIRAL]->(:Coln)
WITH j, count(r) AS st
WHERE st = 0
RETURN j.ime AS ime;
```

Pričakovani rezultat: prazna tabela (0 vrstic). V začetnem grafu imajo Ana, Boris in Cene vsi vsaj eno rezervacijo.

Razlaga: OPTIONAL MATCH ohrani jadralca tudi, če povezave ni, in za manjkajočo povezavo poda null; count(r) takih vrednosti ne šteje. Navadni MATCH bi jadralce brez povezave izločil že pred filtriranjem.

Opomba za izvajalca. Prazna tabela pri N3 je pravilen odgovor! Za bolj nazoren prikaz ustvari dodatnega jadralca brez rezervacij in znova zaženi N3; takrat se pojavi v rezultatu.

2 Neo4j in Cypher: rešitve N4–N5

N4 · Pari jadralcev, ki so rezervirali isti čoln

```
MATCH (j1:Jadralec)-[:REZERVIRAL]->(c:Coln)
      <-[:REZERVIRAL]-(j2:Jadralec)
WHERE j1.jid < j2.jid
RETURN DISTINCT j1.ime AS prvi, j2.ime AS drugi
ORDER BY prvi, drugi;
```

prvi	drugi	Skupni čoln v izhodiščnih podatkih
Ana	Boris	Elan
Ana	Cene	Elan in Bavaria
Boris	Cene	Elan

Razlaga: pot dolžine 2 najde skupnega soseda (čoln). Pogoj j1.jid < j2.jid odstrani isti par v obratnem vrstnem redu ter prepreči par osebe s samo sabo. DISTINCT odpravi podvojitve, kadar si dva jadralca delita več čolnov ali več rezervacij.

N5 · Sprememba barve in nova rezervacija

```
MATCH (c:Coln {cid:20})
SET c.barva = 'zelena';

MATCH (j:Jadralec {jid:2}), (c:Coln {cid:20})
MERGE (j)-[:REZERVIRAL {dan:date('2026-10-10')}]->(c);
```

Pričakovani rezultat: čoln Bavaria je po posodobitvi zelen; Boris ima novo rezervacijo za Bavaria z datumom 2026-10-10. Skupno število povezav je po prvem izvajanju 6 namesto 5. Če isti MERGE izvedemo še enkrat, se isti vzorec praviloma ne podvoji.

```
MATCH (j:Jadralec {jid:2})-[r:REZERVIRAL]->(c:Coln)
RETURN j.ime, c.ime, c.barva, r.dan
ORDER BY r.dan;
```

Dodatno vprašanje · Kdaj je rezervacija samostojno vozlišče?

Kadar ima rezervacija svojo identiteto, status, plačilo, več udeležencev ali povezave do dodatnih entitet, je smiselno modelirati (j)-[:USTVARIL]->(r:Rezervacija)-[:ZA_COLN]->(c). Če vsebuje zgolj datum in povezuje enega jadralca z enim čolnom, je lahko lastnost povezave :REZERVIRAL preprostejša.

Opomba za izvajalca. Za referenčne rezultate N1–N4 podatke najprej ponastavi. Po N5 se lahko N4 v tem konkretnem grafu ohrani pri treh parih, vendar se število skupnih povezav spremeni.

3 Vektorska podobnost in KNN: V1–V3

V izhodiščnem gradivu so vektorji A=(0.95, 0.10, 0.05), B=(0.05, 0.98, 0.10), C=(0.20, 0.90, 0.10), D=(0.05, 0.10, 0.97) in prvotno vprašanje q=(0.15, 0.95, 0.15). To so ročno določene koordinate, ne resnični embeddingi.

V1 · Novo vprašanje q = (0.95, 0.05, 0.10)

```
novo_q = (0.95, 0.05, 0.10)
for ime, v in sorted(vektorji.items(),
                     key=lambda x: -cosine(x[1], novo_q)):
    print(ime, round(cosine(v, novo_q), 3))
```

Mesto	Dokument	Kosinusna podobnost
1	A – Grafne baze	0.997
2	C – RAG in iskanje	0.276
3	D – Transakcije	0.160
4	B – Vektorsko iskanje	0.113

Odgovor: zmaga A; novi vektor vprašanja je najbolj usmerjen v prvo dimenzijo, kjer prevladuje A.

V2 · Kosinusna podobnost in evklidska razdalja

Za par B in prvotno vprašanje q dobimo približno:

```
from math import sqrt
B = (0.05, 0.98, 0.10)
q = (0.15, 0.95, 0.15)
def l2(a,b):
    return sqrt(sum((x-y)**2 for x,y in zip(a,b)))
print(round(cosine(B,q), 6))  # 0.993109
print(round(l2(B,q), 6))      # 0.115758
```

Za vse štiri primere pri prvotnem q je vrstni red po padajoči kosinusni podobnosti in po naraščajoči L2-razdalji enak: C, B, A, D. To ni splošno pravilo: kosinus primerja smer, L2 pa tudi dolžino vektorja. Če vse vektorje (in poizvedbo) normaliziramo na dolžino 1, sta rangiranji ekvivalentni.

Protiprimer: q=(1,0), a=(10,0), b=(0.9,0.1). Kosinus da prednost a (popolnoma ista smer), L2 da prednost b (bistveno manjša razdalja).

V3 · Različne dimenzije; pomen k=3

Vektorska primerjava po koordinatah zahteva enako število dimenzij in enak pomen koordinat. Računanje s zip() brez preverjanja dimenzij je nevarno, ker tiho odreže odvečne koordinate. k=3 pomeni, da KNN vrne tri najbližje vektorje glede na izbrano metriko.

Opomba za izvajalca. Za ničelni vektor kosinusna podobnost matematično ni definirana, saj je norma 0. Demo-koda v Notebooku vrne 0 kot praktično varovalo; to ni matematična definicija.

4 RAG in retrieval: rešitve R1–R4

R1 · »Kaj zagotavlja transakcija?«

```
viri = retrieve("Kaj zagotavlja transakcija?", top_k=2)
for docid, besedilo, score in viri:
    print(docid, score, besedilo)
```

Pričakovani izpis: D3, rezultat ujemanja 2, z besedilom: »Transakcija zagotavlja, da se spremembe v celoti potrdijo ali razveljavijo.« Ujemata se izraza transakcija in zagotavlja. Drugi rezultat ni vrnjen, ker nima pozitivnega števila ujemanj.

R2 · Ključne besede vs. semantično iskanje

Pri natančnem besednem iskanju primerjamo konkretne besede ali njihove normalizirane oblike; sinonimi in parafraze lahko ostanejo neopaženi. Semantično iskanje uporablja vektorske predstavitve iz modela, ki lahko bližino dodeli tudi podobnim pomenom brez istih besed. Vektorska podobnost ni popolna in je odvisna od modela ter domene.

R3 · Zakaj RAG ne zagotovi odsotnosti halucinacij?

Iskalni del lahko vrne nepomemben, zastarel ali napačen odlomek. Jezikovni model lahko tako vseeno poda napačen zaključek, preveč posploši, nepravilno citira ali doda informacijo, ki je v viru ni. Dobro preverjanje vključuje oceno relevantnosti, preverjanje virov, navedbe odlomkov in možnost odgovora »iz priloženih virov ni mogoče sklepati«.

R4 · Vir in številka strani

Že pri razrezu dokumenta shranimo metapodatke: identifikator dokumenta, naslov, številko strani, lokacijo/URL in identifikator odlomka. Skupaj z vsebino in vektorjem jih hranimo v indeksu ali povezani bazi; ob iskanju jih vrnemo skupaj z odlomkom.

```
odlomek = {
    "id": "D17-p4-odl2",
    "text": "... besedilo odlomka ...",
    "source": "ucbenik.pdf",
    "page": 4,
    "title": "Poglavje o transakcijah"
}
print(f"Vir: {odlomek['source']}, str. {odlomek['page']}")
```

Pomembno: v priloženem Notebooku so le ID-ji D1–D4, ker imajo dokumenti po eno poved. PDF-številke strani v tem demo-podatkovnem modelu niso na voljo; primer zgoraj prikazuje potrebno razširitev, ne obstoječ rezultat.

Opomba za izvajalca. Priloženi Notebook prikazuje samo retrieval po prekrivanju besed: nima modela vložitev, vektorskega indeksa ali generativnega LLM. Zato ga pošteno opredeli kot »poenostavljen prikaz principa RAG«, ne kot celoten sistem RAG.

5 Transakcije: rešitve T1–T4

V začetnem SQLite primeru imata račun 1 saldo 100 in račun 2 saldo 60. Prenos 30 prenese sredstva z računa 1 na račun 2.

T1 · Zamenjaj ROLLBACK s COMMIT

```
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("COMMIT")
print(con.execute("SELECT * FROM racun ORDER BY id").fetchall())
```

Rezultat: [(1, 70), (2, 90)]; skupni saldo ostane 160. COMMIT spremembe potrdi, ROLLBACK jih razveljavi. Ta rezultat velja, če zaženeš celico na sveže inicializirani zbirki.

T2 · Samo odštevanje, brez nakazila

Če transakcija izvede le odštevanje 30 in ga potrdi s COMMIT, dobimo (70, 60), vsota pade na 130 in poslovno pravilo prenosa je kršeno. Če namesto potrditve sprožimo ROLLBACK, se vrnemo na (100, 60). Če ni niti COMMIT niti ROLLBACK, sprememba ostane nepotrjena, dokler se transakcija ne zaključi; ne smemo je avtomatsko šteti za trajno.

T3 · Kako preprečiti negativen saldo

```
con.execute("BEGIN")
try:
    znesek = 30
    if znesek <= 0:
        raise ValueError("Znesek mora biti pozitiven")
    a = con.execute(
        "UPDATE racun SET saldo=saldo-? "
        "WHERE id=? AND saldo>=?", (znesek, 1, znesek))
    b = con.execute(
        "UPDATE racun SET saldo=saldo+? WHERE id=?", (znesek, 2))
    if a.rowcount != 1 or b.rowcount != 1:
        raise ValueError("Manjkajoc racun ali premalo sredstev")
    con.execute("COMMIT")
except:
    con.execute("ROLLBACK")
    raise
```

Pred potrditvijo preverimo: znesek > 0, obstoj obeh računov, zadostna sredstva in uspeh obeh posodobitev. Dodatna zaščita je CHECK (saldo >= 0) pri definiciji tabele. Preverjanje razpoložljivih sredstev in odštevanje naj bosta v istem atomarnem UPDATE ... WHERE saldo >= ?.

T4 · Zakaj sta sočasni transakciji težji?

Lahko tekmujeta za iste podatke, vrstni red operacij pa vpliva na rezultat. Možni so izgubljene posodobitve, neponovljiva branja, fantomski zapisi, blokade oziroma deadlocki (odvisno od SUPB in izolacije). Zato potrebujemo zaklepanje, nadzor sočasnosti, transakcijske izolacijske ravni in po potrebi ponovne poskuse. En zaporedni program nima medsebojnega prepletanja dveh transakcij.

Opomba za izvajalca. Priloženi SQLite-demo na eni povezavi ne prikazuje izolacijskih anomalij. Za tak prikaz je treba pripraviti ločen scenarij z vsaj dvema povezavama in ustreznimi izolacijskimi nastavitvami.

6 Hitri kviz: odgovori na osem vprašanj

1. Kaj je vozlišče in kaj povezava?

Vozlišče predstavlja entiteto (Jadralec, Coln), povezava pa njuno relacijo (REZERVIRAL). Oboje ima lahko lastnosti.

2. Cypher: vsi jadralci, ki so rezervirali Elan

```
MATCH (j:Jadralec)-[:REZERVIRAL]->(c:Coln {ime:'Elan'})
RETURN DISTINCT j.ime AS jadralec
ORDER BY jadralec;
```

Izid iz testnega grafa: Ana, Boris, Cene.

3. Kdaj OPTIONAL MATCH namesto MATCH?

Kadar želimo ohraniti tudi izhodiščna vozlišča brez povezave. V tem primeru dobimo za manjkajoče dele vzorca null (analogija z LEFT OUTER JOIN).

4. Ali je podobnost dveh besedil nujno kosinusna podobnost vektorjev?

Ne. Kosinus je ena možna mera na vektorskih predstavitvah; končni rezultat je odvisen od embedding modela, metrik in pomena naloge. Dve besedili imata lahko podobne besede, a različen pomen, ali obratno.

5. Kaj je embedding in kdo ga izdela?

Embedding je številski vektor za objekt (besedilo, sliko, izdelek ipd.). Izračuna ga embedding model oziroma drug algoritem za predstavitev podatkov; v študentskem prikazu so vektorji zgolj ročno določeni.

6. Kaj v RAG pride najprej: iskanje ali generiranje?

Pri obravnavanem RAG-potoku najprej poiščemo relevantne odlomke, nato sestavimo poziv z vprašanjem in kontekstom, šele potem LLM ustvari odgovor.

7. Kaj naredi ROLLBACK in zakaj je potreben?

Razveljavi nepotrjene spremembe trenutne transakcije. Pri bančnem prenosu prepreči, da bi ostal izveden samo odtegljaj, če nakazilo ni uspelo.

8. Ali sqlite3 potrebuje ločen strežnik?

Ne. SQLite je vgrajena podatkovna knjižnica, Pythonov sqlite3 pa je del običajne standardne namestitve; podatkovna baza je lahko v datoteki ali v pomnilniku.

7 Dodatni odgovori za demonstracijo v Notebooku

Vprašanja ob vektorski celici

Če q spremenimo na (0.95, 0.05, 0.10), zmaga A. Dimenzije v učnem primeru označujejo tri izmišljene značilnosti/teme; v pravem embeddingu posamezna koordinata običajno nima enostavne razlage. Pri ničelnem vektorju je kosinus nedefiniran (deljenje z 0).

Vprašanja ob retrieval celici

Za »Kaj zagotavlja transakcija?« vrne D3. Iskanje po besedah ne prepozna nujno sinonimov, parafraz ali konteksta. Če ni uporabnega odlomka, mora sistem povedati, da nima ustreznega vira, razširiti iskanje ali predati zahtevo v preverjanje - ne izmišljati si vira.

Naloge ob transakcijski celici v Notebooku

(1) Po uspešnem prenosu 30 je stanje (70,90). Prenos 15 nazaj z računa 2 na 1 da (85,75) in vsoto 160. (2) Prenos 1000 ob preverjanju stanja vrne ROLLBACK, salda ostaneta nespremenjena. (3) Vsota saldov je invariant poslovnega pravila, ker denarja pri prenosu ne ustvarjamo in ne izgubljamo. (4) Če potrdimo odtegljaj pred uspešnim nakazilom, poznejši ROLLBACK že potrjenega odtegljaja ne more več popraviti - atomarnost je porušena.

Priporočeni vrstni red za hitro preverjanje pri vajah

Del	Minimalni preverljivi izid
Neo4j	N1 vrne 2 čolna; N2 2 rezervaciji; N3 prazno; N4 3 pare; N5 6 povezav.
Vektorji	Novo vprašanje → A (kosinus ≈ 0.997). Prvotni q → C pred B.
Retrieval	Vprašanje o transakciji → D3, ujemanje 2.
SQLite	COMMIT prenosa 30 → (70,90); ROLLBACK → izhodiščno stanje.

Povezave in gradiva

Neo4j – Cypher Manual

Neo4j – GraphAcademy

SQLite – Transaction

Neo4j – Python Driver

Vir učnih primerov: priloženi dokumenti predmeta TUP in spremljevalni PDF za študente (november 2026). Primeri in pričakovani izpisi so vezani na točno navedene učne podatke; ob drugačnem ali že spremenjenem grafu se rezultati razlikujejo.