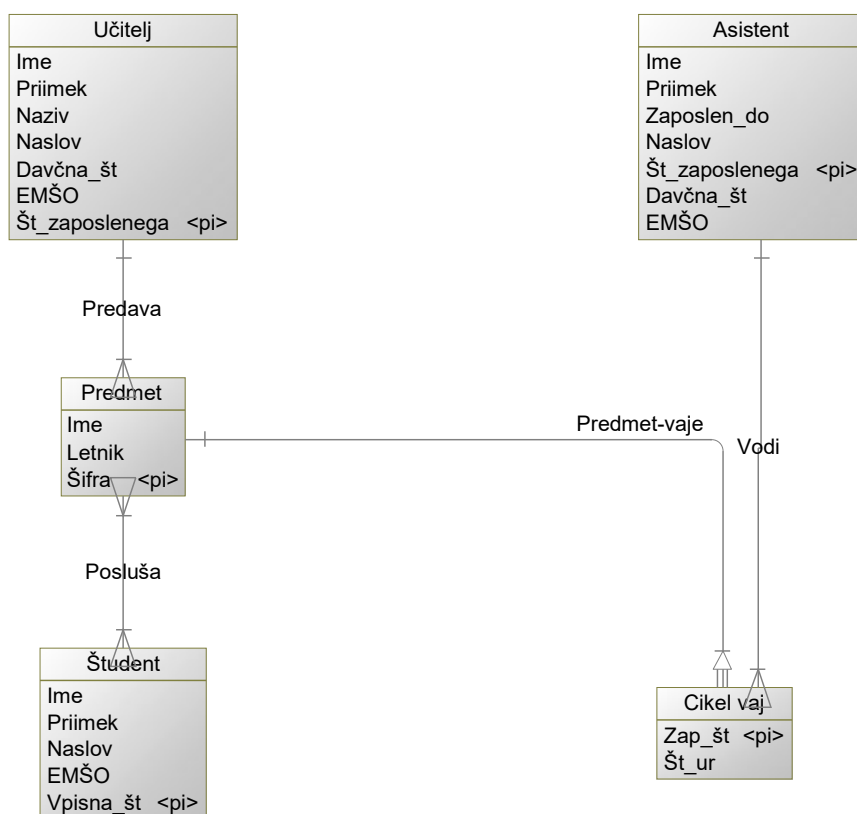


## Tehnologija upravljanja podatkov

Čas pisanja: 75 minut. Literatura je dovoljena.

Liste z nalogami podpišite in obvezno oddajte skupaj z rešitvami!

1. Podano imamo naslednjo konceptualno shemo, kjer so s <pi> označeni identifikatorji.
  - a) V originalni konceptualni shemi kategorizirajte gradnike na: razmerja, močne in šibke entitetne tipe, ter vsakemu entitetnemu tipu določite **celoten** identifikator.
  - b) Konceptualno shemo dopolnite s povezavo med entitetnima tipoma študent in cikel vaj, ki govori o tem, da študent obiskuje dani cikel vaj.
  - c) Dopolnjeno konceptualno shemo pretvorite v relacijske sheme in atributom določite domene. Pri tem lahko izbirate med VARCHAR(x), NUMBER(x) in DATE, kjer je x dolžina (število mest) atributa.



(30) \_\_\_\_ točk

2. Prestavljajte si, da originalno konceptualno shemo iz prve naloge poenostavimo tako, da ohranimo samo entitetne tipe **Učitelj**, **Predmet** in **Cikel\_vaj** ter razmerji **Predava** in **Predmet-vaje**.

- Implementirajte opisano konceptualno shemo v MongoDB.
- Ali bi na učinkovit in neredundanten način lahko dodali tudi entitetni tip **Študent** in razmerje **Poslušā**? Kratko in jedrnato utemeljite (implementacija ni potrebna).

(20) \_\_\_\_ točk

3. V podatkovni bazi spletne učilnici FRI imamo v SUPB MariaDB tabelo Ucilnica z naslednjo shemo:

```
Ucilnica (vpst:VARCHAR(10), predmet:VARCHAR(100), tocke:INTEGER)
```

SUPB uporablja rigorozno dvofazno zaklepanje z možnostjo nadgrajevanja. Sočasno se lahko izvaja poljubno mnogo kopij spodnjih dveh programov.

- Nastavite stopnjo izolacije na »READ COMMITTED«.
- Dodajte manjkajoče ukaze za začetek/zaključek transakcij.
- Do katerih ažurirnih anomalij lahko še vedno pride? Navedite konkretne primere in predlagajte rešitev!

```
def posodobi(c1, predmet, vpst, dodatne_tocke):  
    tocke = c1.execute("SELECT tocke FROM ucilnica WHERE predmet=? and vpst=?", predmet, vpst)  
    nove_tocke = tocke + dodatne_tocke  
    c1.execute("UPDATE ucilnica set tocke=? WHERE predmet=? and vpst=?", nove_tocke, predmet, vpst)  
    if nove_tocke > 100:  
        c1.rollback()  
    else:  
        c1.commit()  
  
def povp(c2, predmet):  
    tocke = c2.execute("SELECT tocke FROM ucilnica WHERE predmet=?", predmet)  
    N = 0  
    sum = 0.0  
    for t in tocke:  
        sum += t  
        N += 1  
    return sum/N
```

(5+5+10) \_\_\_\_ točk

4. V podatkovni bazi uporabljamo rigorozni protokol dvofaznega zaklepanja (2PL), ki omogoča nadgrajevanje zaklepanj. Pri naslednjem razporedu ukazov:
- Uporabite opisan 2PL protokol in narišite dopolnjen razpored ukazov (dodajte ukaze za zaklepanje in vpišite morebitno čakanje na odobritev). Dopolnite delno rešitev v tabeli na naslednji strani!
  - Pri pravilni uporabi 2PL pride do mrtve zanke. Ob nastopu mrtve zanke narišite čakalni graf in predlagajte, katero transakcijo bi bilo najboljše prekiniti in razveljaviti. Odgovor utemeljite!

Čas	Transakcija T1	Transakcija T2	Transakcija T3	Transakcija T4
1	BEGIN TRANSACTION			
2	READ(X)			
3		BEGIN TRANSACTION		
4		READ(Y)		
5			BEGIN TRANSACTION	
6			READ(Z)	
7				BEGIN TRANSACTION
8				S=0
9	X=X+10			
10	WRITE(X)			
11		Y=Y+20		
12		WRITE(Y)		
13			Z=Z+30	
14			WRITE(Z)	
15				READ(X)
16				S=S+X
17	ROLLBACK			
18		READ(X)		
19		X=X+10		
20			READ(Y)	
21			Y=Y+20	READ(Y)
22				S=S+Y
23		WRITE(X)		
24		COMMIT		
25			WRITE(X)	
26			ROLLBACK	
27				READ(Z)
28				S=S+Z
29				WRITE(S)
30				COMMIT

(30) \_\_\_\_ točk

Skupaj: \_\_\_\_ točk

Čas	Transakcija T1	Transakcija T2	Transakcija T3	Transakcija T4
1	BEGIN TRANSACTION			
2				
3	READ(X)			
4		BEGIN TRANSACTION		
5				
6		READ(Y)		
7			BEGIN TRANSACTION	
8				
9			READ(Z)	
10				BEGIN TRANSACTION
11				S=0
12	X=X+10			
13				
14	WRITE(X)			
15		Y=Y+20		
16				
17		WRITE(Y)		
18			Z=Z+30	
19				
20			WRITE(Z)	
21				
22	ROLLBACK			
23				
24		READ(X)		
25		X=X+10		
26				
27				
28				READ(X)
29				S=S+X
30		WRITE-LOCK(X)		
31				READ-LOCK(Y)
32				
33				
34				
35				
36				
37				
38				
39				
40				