VEZE IZMEĐU TABELA

Kako poslovno pravilo postaje kardinalnost, ključ i ograničenje u bazi

1 : 1

jedan ↔ jedan

1 : N

jedan → više

M : N

više ↔ više

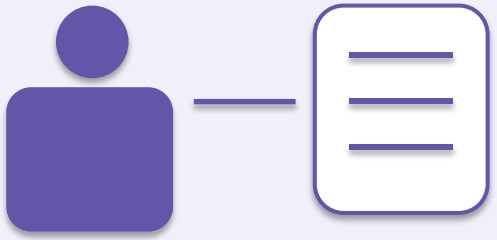
Student prvo razume odnos u realnom svetu — tek onda vidi tabele i PK/FK.

KARDINALNOST | KLJUČEVI | INTEGRITET | PROJEKTOVANJE | PROVERA

TRI TIPA VEZE — TRI REALNE SITUACIJE

Kardinalnost opisuje koliko zapisa može biti povezano.

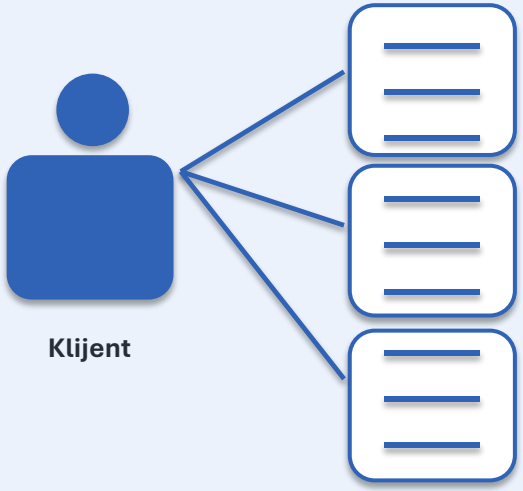
1 : 1



Zaposleni — Kartica

Jedan zaposleni ↔ jedna službena kartica

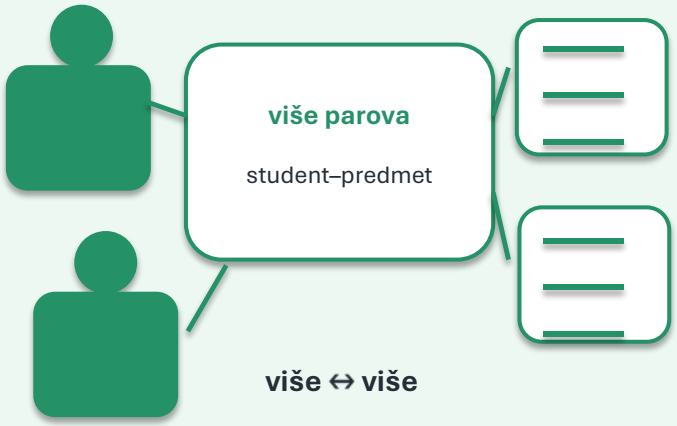
1 : N



Klijent

Jedan klijent → više porudžbina

M : N



više ↔ više

Kardinalnost je prvo poslovno pravilo — ne tehnička odluka.

KARDINALNOST | KLJUČEVI | INTEGRITET | PROJEKTOVANJE | PROVERA

KARDINALNOST + OBAVEZNOST

Broj „N“ kaže maksimum; poslovno pravilo može da kaže i da li je veza obavezna.

A **1 : N**

Klijent može imati više porudžbina.

B **0 : N**

Novi klijent može postojati i bez porudžbine.

C **1 : 1**

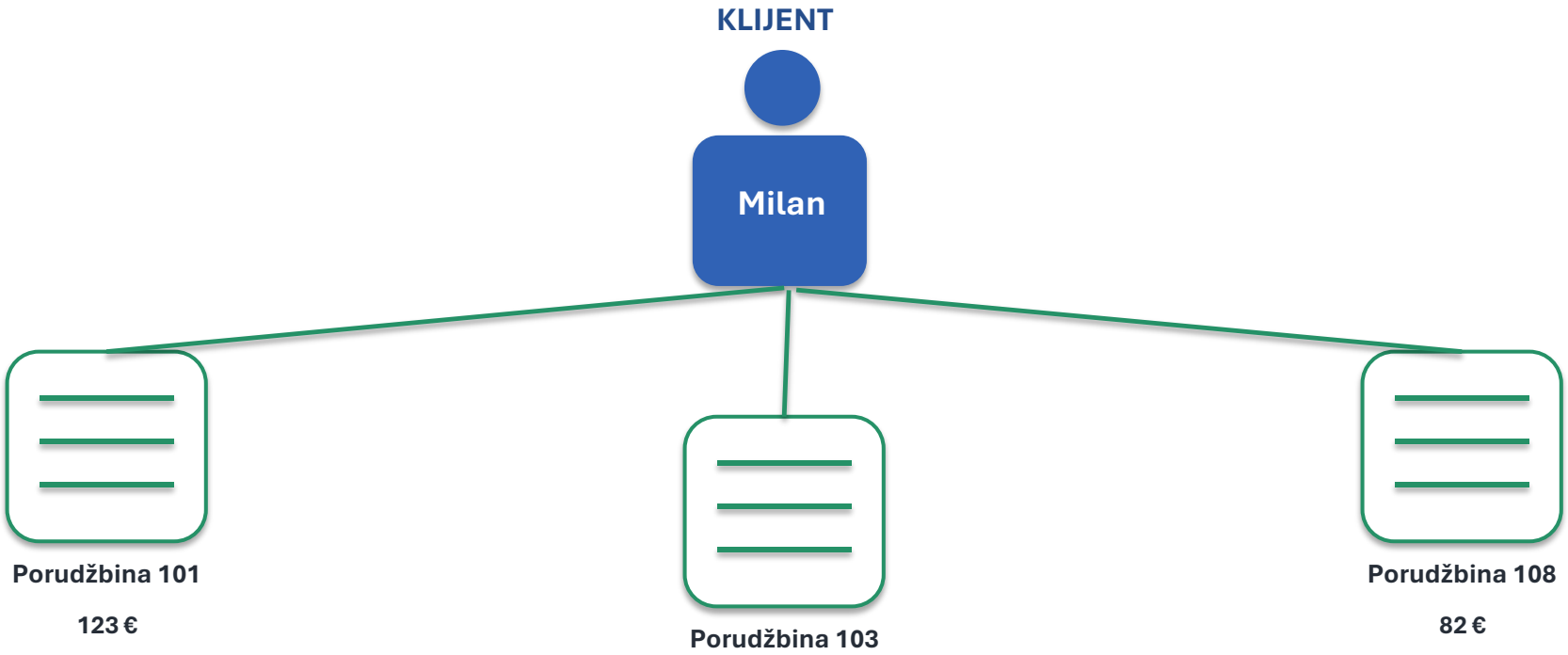
Porudžbina mora pripadati jednom klijentu.

U praksi modelujemo i „koliko najviše“ i „da li mora da postoji“.

KARDINALNOST | **KLJUČEVI** | **INTEGRITET** | **PROJEKTOVANJE** | **PROVERA**

1 : N

Jedan klijent pravi više porudžbina tokom vremena.



„Jedan klijent može imati više porudžbina.“

Ova rečenica određuje da je odnos KLIJENT → PORUDŽBINA tipa 1:N.

KARDINALNOST | KLJUČEVI | INTEGRITET | PROJEKTOVANJE | PROVERA

1 : N — GDE IDE STRANI KLJUČ?

Sada realnu vezu prevodimo u strukturu baze.

KLIJENT

PK **KlijentID**

Ime

Email

1 ——— N

KlijentID se ponavlja u više porudžbina.
Zato FK ide na stranu N.

PORUDŽBINA

PK **PorudzbinaID**

Datum

Ukupno

FK KlijentID

Kod 1:N, strani ključ se nalazi u tabeli na strani „više“.

1 : N — ISTA VEZA U REDOVIMA

Sada se vidi zašto se strani ključ ponavlja.

KlijentID (PK)	Ime
233	Milan
235	Marina

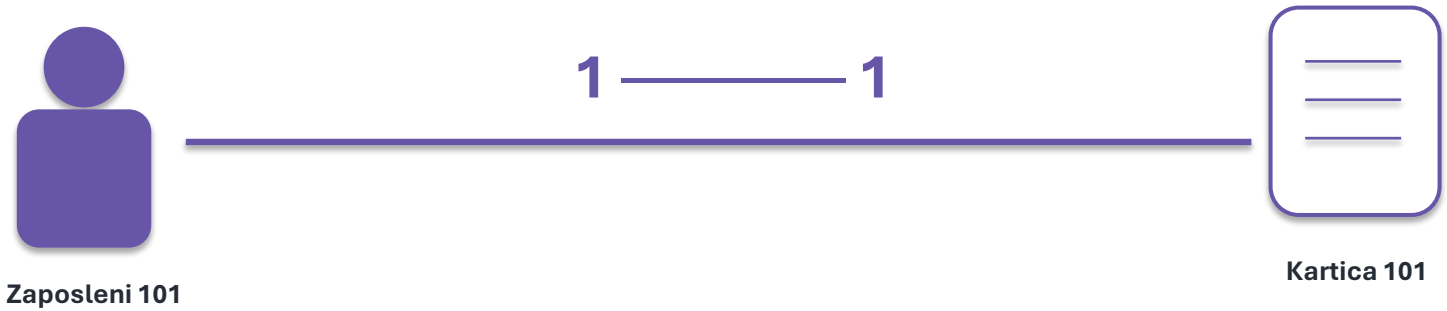
PorudžbinaID	Datum	Ukupno	KlijentID (FK)
101	01.05.	123	233
103	02.05.	45	233
108	07.05.	82	233
102	01.06.	345	235

Klijent 233 → porudžbine 101, 103 i 108

Ponavljanje FK vrednosti nije dupliranje podataka — to je način predstavljanja veze 1:N.

1 : 1 — JEDAN SA JEDNIM

Primer: zaposleni i njegova službena kartica.



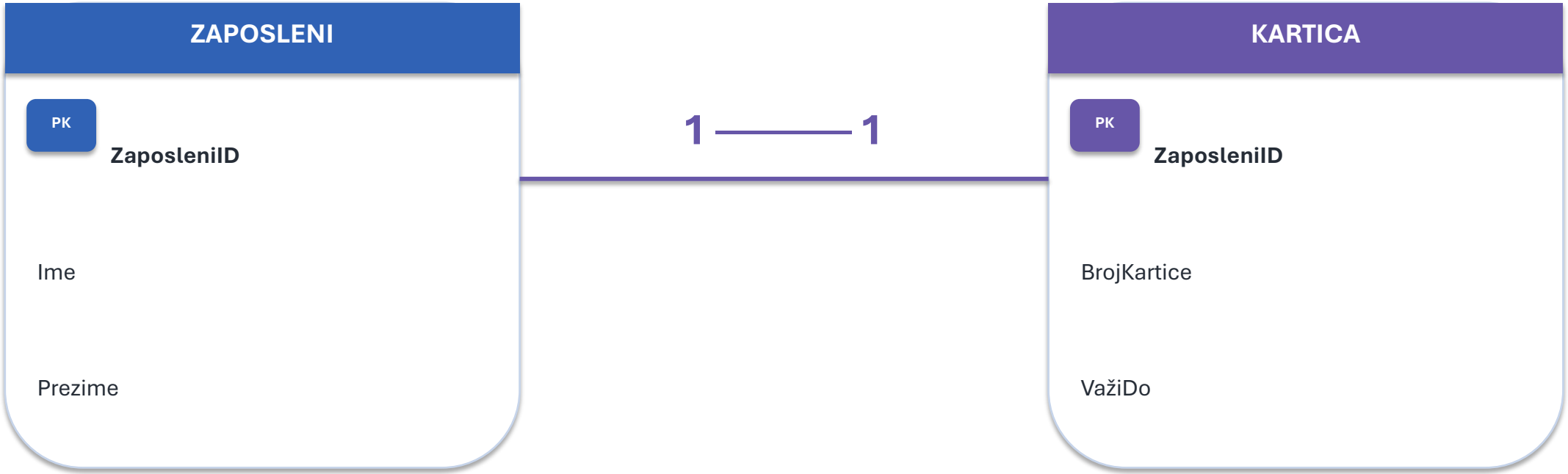
**Svaki zaposleni ima najviše jednu karticu,
a svaka kartica pripada najviše jednom zaposlenom.**

1:1 je ređi; uvek proverimo da li dve tabele zaista treba da budu odvojene.

KARDINALNOST | KLJUČEVI | INTEGRITET | PROJEKTOVANJE | PROVERA

1 : 1 — IMPLEMENTACIJA

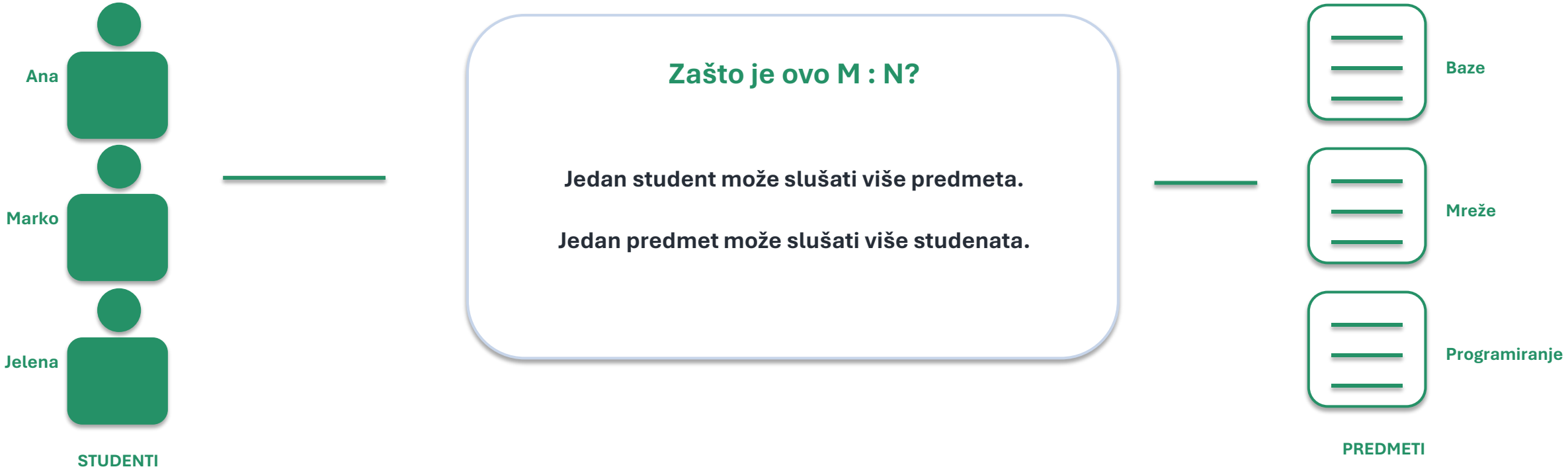
Jedan ključ može istovremeno biti PK i FK u zavisnoj tabeli.



Jedinstveno ograničenje ili PK/FK kombinacija sprečava da jedan zaposleni dobije više kartica.

M : N — KAKO IZGLEDA U REALNOM SVETU?

Više studenata sluša više predmeta.



M:N je samo mnogo pojedinačnih veza.

KARDINALNOST | KLJUČEVI | INTEGRITET | PROJEKTOVANJE | PROVERA

9. ZAŠTO JEDAN FK NE MOŽE DA REŠI M : N?

Pokušaj direktnog smeštanja FK-a odmah stvara problem.

A **FK u STUDENT**

Ako student ima samo jedno polje PredmetID, može sačuvati samo jedan predmet.

B **FK u PREDMET**

Ako predmet ima samo jedno polje StudentID, može sačuvati samo jednog studenta.

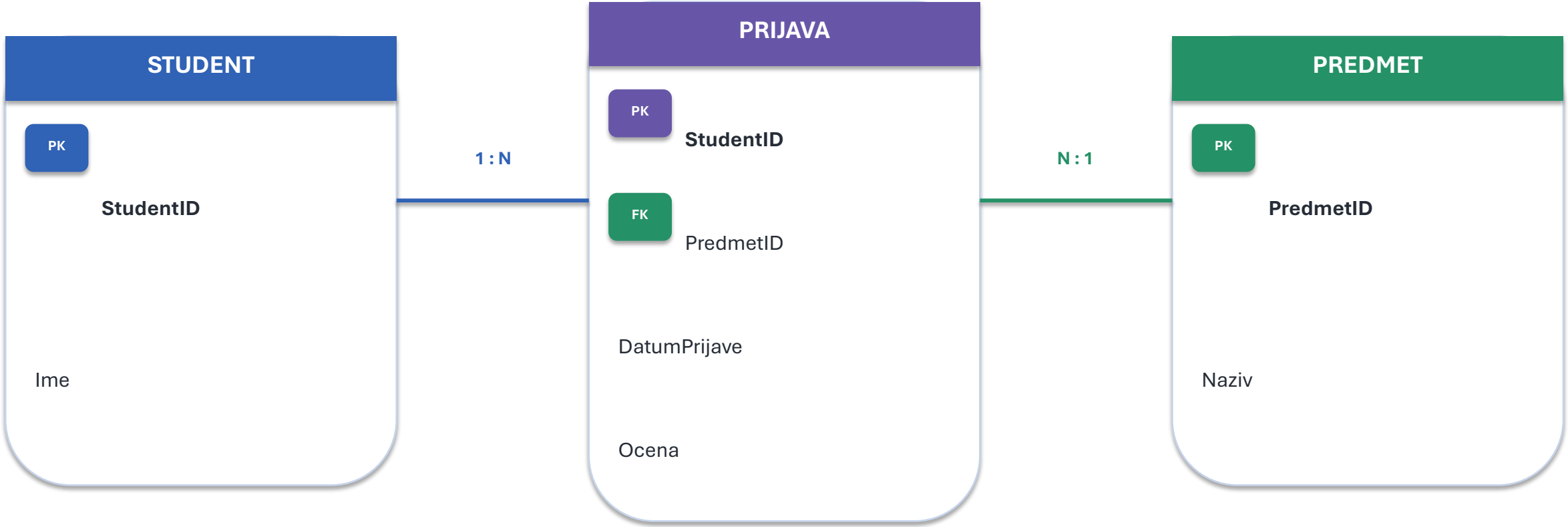
C **REŠENJE**

Vezu pretvaramo u novu tabelu koja čuva po jedan red za svaku kombinaciju.

M:N zahteva spojnu tabelu jer jedna kolona ne može uredno čuvati listu više ključeva.

10. M : N → DVE VEZE 1 : N

Spojna tabela predstavlja samu vezu.



PRIJAVA nije samo pomoćna tabela — ona može čuvati datum, ocenu i druge podatke same veze.

KARDINALNOST | KLJUČEVI | INTEGRITET | PROJEKTOVANJE | PROVERA

SPOJNA TABELA — VEZE POSTAJU REDOVI

Svaka kombinacija student–predmet je jedan red.

StudentID	Ime
50	Ana
51	Marko
52	Jelena

StudentID	PredmetID	Ocena
50	111	9
50	113	8
51	111	10
51	112	9
52	112	8
52	113	10

PredmetID	Naziv
111	Baze
112	Mreže
113	Programiranje

Pitanje: koje predmete sluša Ana (50)? → 111 i 113

12. PK I FK — PK identifikuje, FK povezuje

PK „KO JE OVAJ RED?“

- jedinstven
- ne ponavlja se
- stabilan identitet

Primer: KlijentID

FK „NA KOGA SE OSLANJAM?“

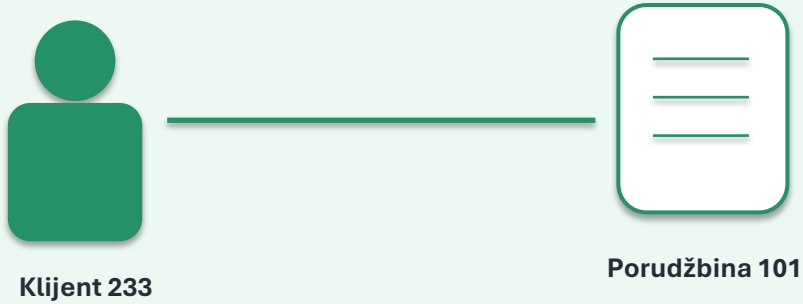
- može da se ponavlja
- upućuje na drugi red
- ostvaruje vezu

Primer: Porudžbina.KlijentID

13. REFERENCIJALNI INTEGRITET — „DA LI VEZA POSTOJI?“

DBMS proverava da li FK pokazuje na postojeći roditeljski red.

✓ DOZVOLJENO



FK = 233 → klijent 233 postoji

✕ BLOKIRANO



FK = 400 → klijent 400 ne postoji

Referencijalni integritet sprečava „siročad“ — redove koji pokazuju na nepostojeći roditeljski zapis.

14. ŠTA KADA BRIŠEMO RODITELJA?

DBMS mora unapred da zna sudbinu povezanih redova.

RESTRICT

Ne dozvoli brisanje.

CASCADE

Obriši i zavisne redove.

SET NULL

Zadrži red, ukloni vezu.

SET DEFAULT

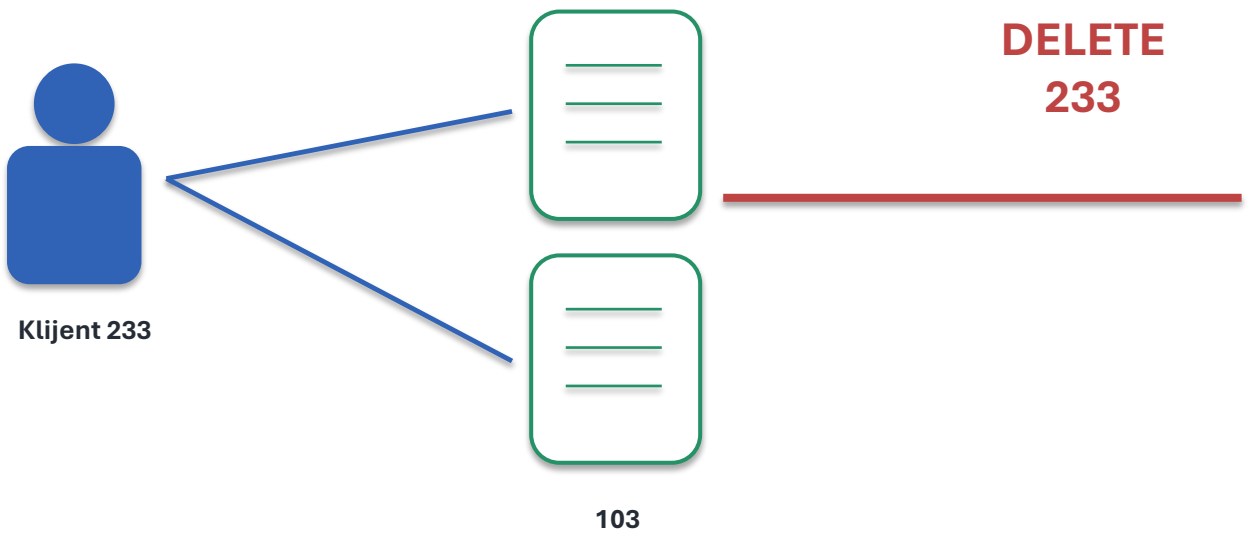
Postavi podrazumevanu vezu.

Referencijalna akcija je poslovna odluka: šta treba da ostane validno posle brisanja?

KARDINALNOST | KLJUČEVI | **INTEGRITET** | PROJEKTOVANJE | PROVERA

15. RESTRICT / NO ACTION — „NE MOŽE JOŠ“

Roditelj ostaje dok postoje zavisni redovi.



BRISANJE ODBIJENO

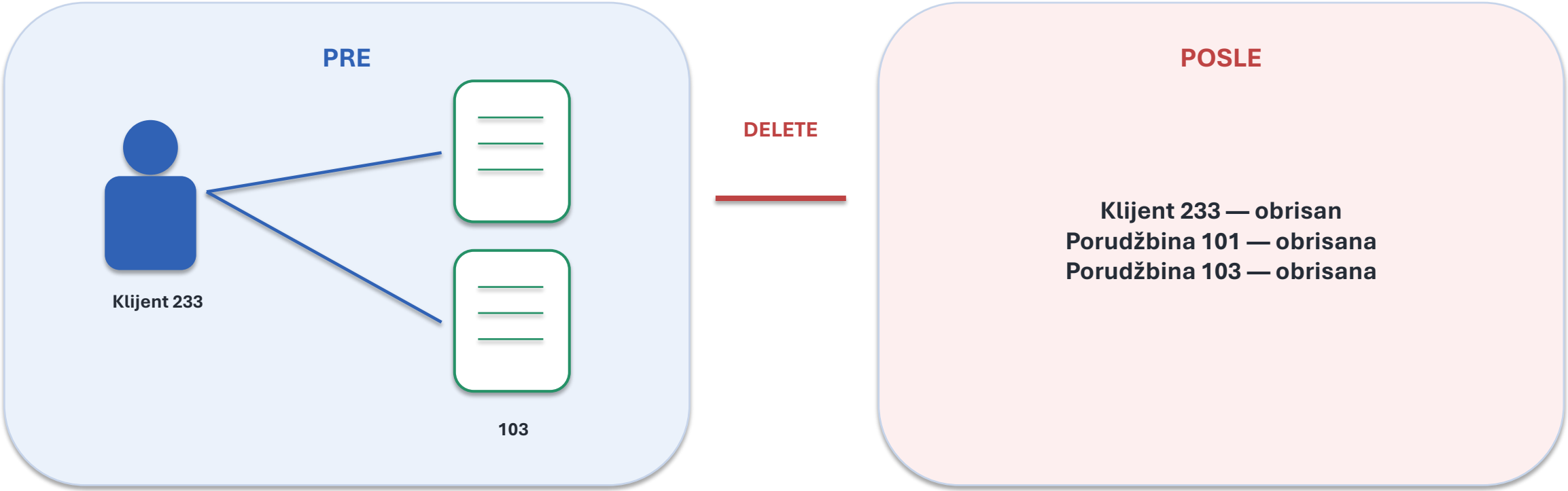
Porudžbine 101 i 103 još zavise od klijenta.

RESTRICT je dobar kada zavisni podatak ne sme da izgubi roditelja.

KARDINALNOST | KLJUČEVI | **INTEGRITET** | PROJEKTOVANJE | PROVERA

16. CASCADE — „OBRIŠI I ONO ŠTO ZAVISI“

Brisanje roditelja automatski uklanja zavisne redove.



CASCADE je prirodan kada dete nema smisla bez roditelja — npr. stavka porudžbine bez porudžbine.

17. SET NULL — „RED OSTAJE, VEZA NESTAJE“

FK se postavlja na NULL, a zavisni red ostaje.

PRE

Porudžbina 101 → KlijentID 233

Porudžbina 103 → KlijentID 233

POSLE

Porudžbina 101 → KlijentID NULL

Porudžbina 103 → KlijentID NULL

SET NULL ima smisla samo ako FK kolona dozvoljava NULL i red može da postoji bez roditelja.

18. ON UPDATE CASCADE

Promena roditeljskog ključa može se preneti na sve povezane FK vrednosti.

1 PRE

KlijentID = 233
Porudžbine: 101 → 233, 103 → 233

2 PROMENA

KlijentID: 233 → 300

3 POSLE

Porudžbine: 101 → 300, 103 → 300

U praksi stabilni interni ID ključevi obično se retko menjaju.

19. KAKO IZABRATI DELETE AKCIJU?

Ne biramo naredbu — biramo poslovno ponašanje.

Da li dete sme da postoji bez roditelja?

NE

RESTRICT ili CASCADE?

Da li dete treba blokirati ili automatski obrisati?

DA

SET NULL može imati smisla

Veza je opciona, a red može ostati samostalan.

Primer: STAVKA bez PORUDŽBINE → CASCADE. PORUDŽBINA bez KLIJENTA → često RESTRICT.

20. TRI FAZE PROJEKTOVANJA

Od realnog sveta do SQL-a — bez preskakanja koraka.

1 KONCEPTUALNO

Šta postoji?
Ko je sa kim povezan?

Izlaz: ER/EER

2 LOGIČKO

Koje tabele?
Gde su PK/FK?

Izlaz: relaciona šema

3 IMPLEMENTACIONO

Koji tipovi?
Koji SQL i indeksi?

Izlaz: DDL

Ako preskočimo razumevanje domena, lako završimo sa tehnički ispravnom, ali loše modelovanom bazom.

KARDINALNOST | KLJUČEVI | INTEGRITET | PROJEKTOVANJE | PROVERA

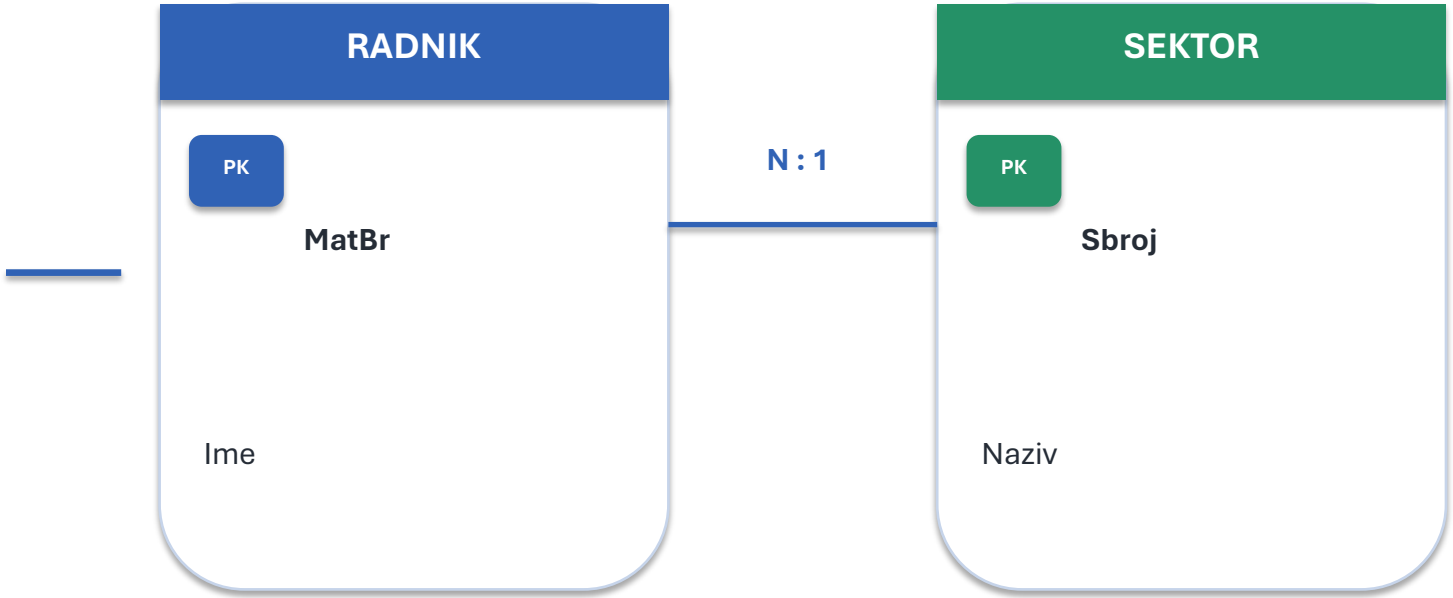
21. KONCEPTUALNO — OD REČENICE DO ER MODELA

Primer: „Radnik radi u jednom sektoru; sektor ima više radnika.“

POSLOVNA REČENICA

Radnik radi u jednom sektoru.

Sektor ima više radnika.

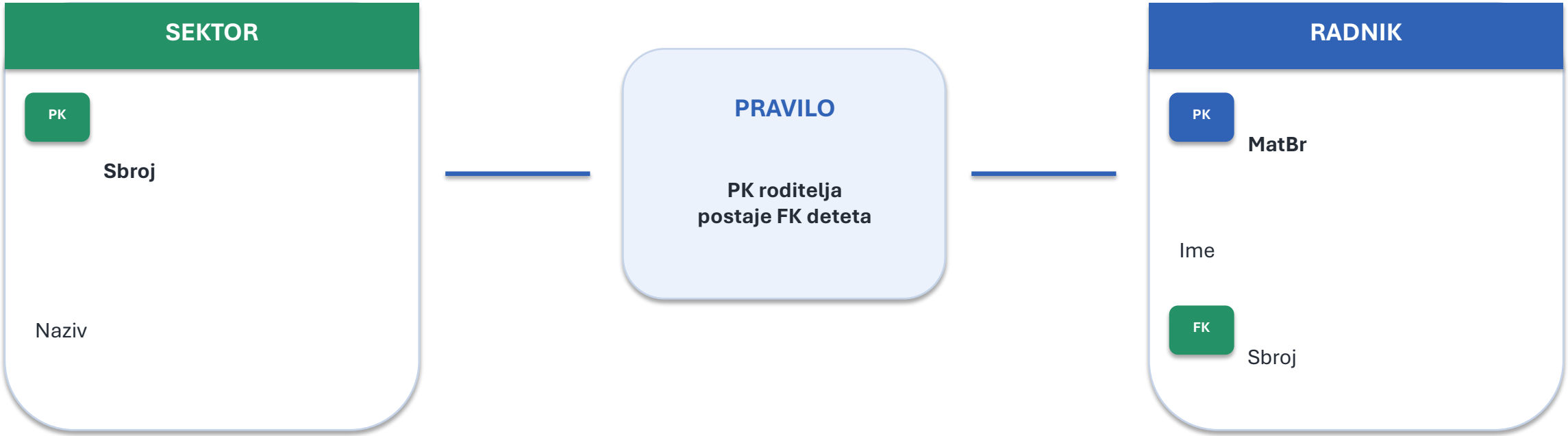


Konceptualni model opisuje značenje, ne SQL sintaksu.

KARDINALNOST | KLJUČEVI | INTEGRITET | PROJEKTOVANJE | PROVERA

22. LOGIČKO — ER VEZE POSTAJU RELACIJE

Kod 1:N, ključ roditelja prelazi u tabelu na strani N.



Logičko projektovanje pretvara značenje veze u konkretne ključeve i relacije.

23. IMPLEMENTACIONO — RELACIJA POSTAJE SQL

Na kraju biramo konkretne tipove podataka i FK akcije.

```
CREATE TABLE Radnik (
  MatBr INT PRIMARY KEY,
  Ime VARCHAR(50) NOT NULL,
  Sbroj INT NOT NULL,

  FOREIGN KEY (Sbroj)
    REFERENCES Sektor(Sbroj)
    ON DELETE RESTRICT
);
```

PRIMARY KEY

MatBr identifikuje radnika

FOREIGN KEY

Sbroj povezuje radnika sa sektorom

ON DELETE

RESTRICT čuva poslovno pravilo

PITALICA 1/5

Gde ide FK u vezi KLIJENT 1:N PORUDŽBINA?

A) Klijent.PorudzbinaID

B) Porudzbina.KlijentID

C) U obe tabele isti FK

D) FK nije potreban

PITALICA 2/5

Zašto M:N veza STUDENT–PREDMET zahteva tabelu PRIJAVA?

A) Da ukloni PK

B) Da student ima jedan predmet

C) Da svaka kombinacija postane red

D) Zato što SQL nema FK

PITALICA 3/5

Šta se dešava kod ON DELETE CASCADE?

A) FK postaje NULL

B) Brisanje je blokirano

C) Brišu se zavisni redovi

D) Menja se PK deteta

PITALICA 4/5

Kada je SET NULL dobar izbor?

A) Kada dete može postojati bez roditelja

B) Kada dete mora biti obrisano

C) Kada je FK NOT NULL

D) Uvek kod 1:N

PITALICA 5/5

Koji je pravilan redosled projektovanja?

A) SQL → ER → relacije

B) ER/EER → relacije → SQL

C) Indeksi → entiteti → veze

D) Tabele → zahtevi → ER