NORMALIZACIJA BAZE PODATAKA

Od problema u jednoj tabeli do jasnog relacionog modela

JEDNA VELIKA TABELA

1NF

2NF

3NF

Normalizacija nije „pravljenje što više tabela“, već uklanjanje pogrešnih zavisnosti.

Šta je normalizacija i zašto je koristimo?

Skup pravila za organizovanje podataka tako da baza ima manje redundanse i manje anomalija.

MANJE PONAVLJANJA

Ista činjenica se čuva na jednom mestu.

MANJE GREŠAKA

Unos, izmena i brisanje ne prave neočekivane posledice.

JASNIJE VEZE

Svaka tabela predstavlja jednu vrstu poslovne činjenice.

Dobra normalizacija olakšava čitanje, ažuriranje i brisanje podataka

Kako prepoznati da tabela traži normalizaciju?

Tri tipična simptoma.

ANOMALIJA UNOSA

Ne možemo sačuvati činjenicu dok ne postoji neka druga.

Primer: novi kurs bez zakazanog termina.

ANOMALIJA IZMENE

Isti podatak se ponavlja u više redova.

Promena mora da se uradi na više mesta.

ANOMALIJA BRISANJA

Brisanjem jednog reda slučajno gubimo i drugu činjenicu.

Normalizacija razdvaja činjenice koje imaju različit životni ciklus.

Tri normalne forme kao tri pitanja

1NF

Da li jedna ćelija sadrži jednu vrednost?

2NF

Da li atribut zavisi od celog ključa?

3NF

Da li neključni atribut zavisi od drugog neključnog?

Svaka sledeća forma polazi od toga da je prethodna već ispunjena.

1NF: jedna ćelija — jedna vrednost

Radnik može da koristi više računara.

IdRadnika	Ime	Prezime	Email	IdRacunara
123	Mika	Lazić	m.l@itc.rs	WIN81234323, XP7856234
124	Ana	Pešić	m.p@vts.rs	WA623543452
125	Jela	Porić	j.p@itc.rs	WIN818874323, XP7859934

Problem: IdRacunara nije jedna vrednost, već lista.

1NF traži atomsku vrednost u svakoj ćeliji.

Viševrednosni atribut je signal da jedna kolona pokušava da čuva skup vrednosti.

Kako se rešava viševrednosni atribut?

Tri mogućnosti — samo jedna dobro skalira.

1. LISTA U ĆELIJI

IdRacunara =
„PC1, PC2, PC3“

✗ teško filtriranje
✗ teško povezivanje
✗ krši 1NF

2. VIŠE KOLONA

IdRacunara1
IdRacunara2
IdRacunara3

✗ ograničen broj
✗ prazna polja
✗ loše skalira

3. POSEBNA TABELA

RAČUNARI
(IdRacunara, IdRadnika, Opis)

✓ neograničen broj
✓ jedan računar = jedan red
✓ FK povezuje radnika

Preferirano rešenje: viševrednosni atribut pretvoriti u više redova povezane tabele.

Preferirano rešenje 1NF

Radnik ostaje u jednoj tabeli, računari prelaze u povezanu tabelu.

IdRadnika (PK)	Ime	Email
123	Mika	m.l@itc.rs
124	Ana	m.p@vts.rs
125	Jela	j.p@itc.rs

1 : N

IdRacunara (PK)	IdRadnika (FK)	Opis
WIN81234323	123	Dell
XP7856234	123	Lenovo
WA623543452	124	HP
WIN818874323	125	HP
XP7859934	125	Dell

Broj računara više ne utiče na strukturu tabele — samo dodajemo novi red.

Instagram objava sa više hashtagova

LOŠ MODEL

POST
PostID = 501
Tekst = „Jutarnja kafa“
Hashtagovi = „#coffee, #nis, #morning“

Problem: Hashtagovi je lista u jednoj ćeliji.

DOBAR MODEL

POST(PostID, Tekst)
HASHTAG(HashtagID, Naziv)
POST_HASHTAG(PostID, HashtagID)

Jedan post može imati više hashtagova, a jedan hashtag više postova.

Kada viševrednosna vrednost predstavlja drugi entitet, dobijamo spojnu tabelu M:N veze.

2NF: kada imamo kompozitni ključ

Svaki neključni atribut mora da zavisi od celog ključa.

Ako atribut zavisi samo od jednog dela kompozitnog ključa → parcijalna zavisnost.

KADA JE 2NF VAŽNA?

Kada primarni ključ ima dve ili više kolona.

ŠTA TRAŽIMO?

Atribut koji „pripada“ samo jednom delu ključa.

Ako tabela ima jednostavan ključ i već je u 1NF, problem parcijalne zavisnosti ne postoji.

Originalni primer 2NF: OBUKE

Isti kurs može da se održava više puta, ali ne dva puta istog dana.

KursId	Datum	NazivKursa	Učionica	BrojMesta	SlobodnaMesta
MAT1	01.05.2017	Matematika	A3	30	15
BP12	10.02.2017	Baze podataka	A2	50	10
FIZ	24.03.2017	Fizika	A1	25	9
BP12	10.04.2017	Baze podataka	A2	50	30

Kompozitni ključ: (KursId, Datum)

NazivKursa zavisi samo od KursId

NazivKursa je parcijalno zavisan → tabela nije u 2NF.

Kako razmišljamo u 2NF?

Razdvojimo činjenice o kursu od činjenica o konkretnom terminu.

(KursId, Datum) — **Učionica, SlobodnaMesta**

KursId — parcijalna zavisnost — **NazivKursa**

NazivKursa prebacujemo u tabelu KURS, a podatke o održavanju u TERMIN.

Rešenje 2NF

Informacija o kursu se više ne ponavlja u svakom terminu.

KursId (PK)	NazivKursa
MAT1	Matematika
BP12	Baze podataka
FIZ	Fizika

1:N

KursId (FK)	Datum	Učionica	BrojMesta	SlobodnaMesta
MAT1	01.05.	A3	30	15
BP12	10.02.	A2	50	10
FIZ	24.03.	A1	25	9
BP12	10.04.	A2	50	30

Ako promenimo naziv BP12, menjamo jedan red u KURS — ne sve termine.

Novi primer 2NF: student–predmet

Kompozitni ključ (StudentID, PredmetID).

LOŠA TABELA

PRIJAVA
StudentID
PredmetID
ImeStudenta
NazivPredmeta
ESPB
Ocena

Ključ = (StudentID, PredmetID)

ŠTA ZAVISI OD ČEGA?

StudentID → ImeStudenta
PredmetID → NazivPredmeta, ESPB
(StudentID, PredmetID) → Ocena

Student i predmet moraju u svoje tabele.

Ocena pripada vezi student–predmet; ime studenta i naziv predmeta ne.

3NF: neključni atribut ne treba da zavisi od drugog neključnog

Tražimo tranzitivnu zavisnost.



PK → A → B

Ako B zapravo pripada entitetu A, izdvajamo novu tabelu.

Originalni primer 3NF: učionica i broj mesta

Kapacitet pripada učionici, ne konkretnom terminu kursa.

KursId	Datum	Učionica	BrojMesta	SlobodnaMesta
MAT1	01.05.	A3	30	15
BP12	10.02.	A2	50	10
FIZ	24.03.	A1	25	9
BP12	10.04.	A2	50	30

A2 → 50 mesta, bez obzira na termin.

BrojMesta zavisi od Učionice.

Učionica → BrojMesta je tranzitivna zavisnost u tabeli TERMIN.

Rešenje 3NF

Uvodimo tabelu PROSTORIJE.

KursId	Datum	Učionica (FK)	SlobodnaMesta
MAT1	01.05.	A3	15
BP12	10.02.	A2	10
FIZ	24.03.	A1	9
BP12	10.04.	A2	30

N:1

Učionica (PK)	BrojMesta
A3	30
A2	50
A1	25

Promena kapaciteta učionice sada se radi u jednom redu tabele PROSTORIJE.

Izvedene vrednosti: da li sve što prikazujemo moramo i da čuvamo?

Primer: Ukupno = Količina × JediničnaCena.

ID	Proizvod	Količina	JediničnaCena	Ukupno
SR112	A100	5	1000 RSD	5000 RSD
SR103	C23	10	2340 RSD	23400 RSD
SR104	B10	1	500 RSD	500 RSD

Ukupno = Količina × JediničnaCena

Ako izvedenu vrednost ručno čuvamo, možemo dobiti nekonzistentne podatke.

Normalizacija i self-join: hijerarhija zaposlenih

Isti entitet može da se povezuje sam sa sobom.

LOŠ MODEL

RADNIK
RadnikID
Ime
SefIme
SefEmail

Podaci o istom šefu ponavljaju se kod svih podređenih.

NORMALIZOVAN MODEL

RADNIK
RadnikID (PK)
Ime
SefID (FK → RADNIK.RadnikID)

Podaci o šefu postoje jednom, kao običan radnik.

Normalizacija ne uklanja self-join — uklanja dupliranje; self-join ostaje način čitanja hijerarhije.

Denormalizacija: svesno odstupanje od normalizacije

Radimo je tek kada postoji konkretna korist.

NORMALIZOVAN MODEL

✓ manje redundanse
✓ lakše izmene
✓ jasna pravila

Cena: više JOIN-ova i ponekad skuplji upiti.

DENORMALIZOVAN MODEL

✓ brže čitanje u specifičnom scenariju
✓ manje JOIN-ova / preagregirani podaci

Cena: redundansa i složenija sinhronizacija.

Denormalizacija je optimizacija nakon merenja — ne početna prečica.

Kada je denormalizacija korisna? Dashboard

Isti skupi JOIN i agregacija izvršavaju se hiljadama puta.

IZVOR ISTINE

PORUDŽBINA
STAVKA
PROIZVOD
KUPAC

Dashboard za svaki zahtev računa JOIN + SUM + GROUP BY.

→ preagregacija →

DENORMALIZOVANI READ MODEL

DNEVNA_PRODAJA
Datum
BrojPorudžbina
Prihod
TopProizvod

Osvežava se periodično; dashboard čita direktno.

Korisno kada je čitanje mnogo češće od izmene, a upit je skup i predvidiv.

Kada NE treba denormalizovati?

Denormalizacija bez razloga pravi više problema nego koristi.

MALI SISTEM

Upiti su već brzi; dodatna redundansa samo komplikuje održavanje.

ČESTE IZMENE

Podaci se stalno menjaju; sinhronizacija kopija postaje skupa.

NEMA MERENJA

Ne znamo gde je usko grlo, pa optimizujemo naslepo.

Pravilo: normalizuj prvo, izmeri drugo, denormalizuj ciljano.

Praktičan algoritam za normalizaciju

Kako rešavati zadatak korak po korak.

1

Šta predstavlja jedan red?

Definiši poslovnu činjenicu.

2

Koji je ključ?

Nađi kolonu ili kombinaciju koja jednoznačno određuje red.

3

Ima li lista u ćeliji?

Ako da → reši 1NF.

4

Od čega zavise atributi?

Deo ključa → 2NF; drugi neključni atribut → 3NF.

Tek kada je model ispravan, razmatramo denormalizaciju zbog konkretnih performansi.

Vežba — polazimo od tabele koja je već u 1NF

Jedan red predstavlja: jedan student + jedan predmet.

StudentID	PredmetID	ImeStudenta	SmerID	NazivSmera	NazivPredmeta	ESPB	Ocena
1001	BP101	Ana Ilić	IT	Informacione tehnologije	Baze podataka	6	9
1001	RM201	Ana Ilić	IT	Informacione tehnologije	Računarske mreže	6	8
1002	BP101	Marko Jović	IT	Informacione tehnologije	Baze podataka	6	8
1002	PR101	Marko Jović	IT	Informacione tehnologije	Programiranje	7	10
1003	RM201	Jelena Nikolić	SA	Saobraćaj	Računarske mreže	6	9

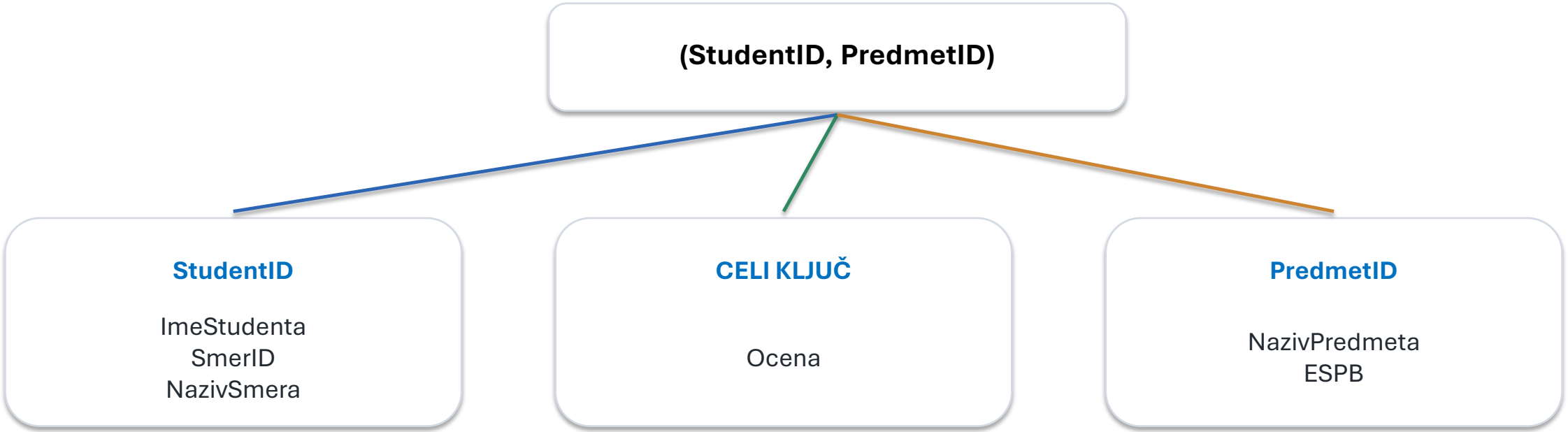
Kompozitni ključ = (StudentID, PredmetID)

Ocena opisuje baš tu kombinaciju.

Prvo proveravamo 2NF: da li svaki neključni atribut zavisi od celog kompozitnog ključa?

2NF — otkrivamo parcijalne zavisnosti

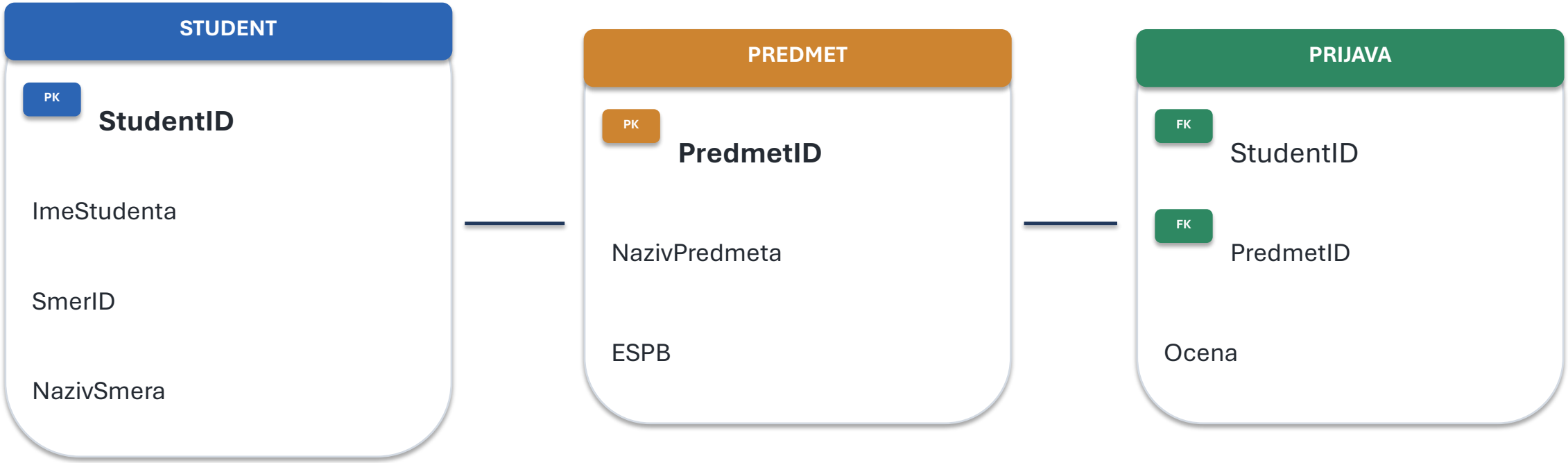
Nisu svi atributi zavisni od celog kompozitnog ključa.



ImeStudenta i podaci o smeru zavise samo od StudentID; podaci o predmetu samo od PredmetID.

Rešenje 2NF — razdvajamo student, predmet i prijavu

Sada parcijalnih zavisnosti više nema.



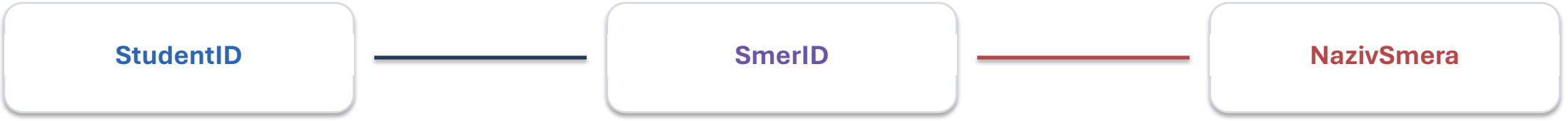
Ali pogledajte tabelu STUDENT...

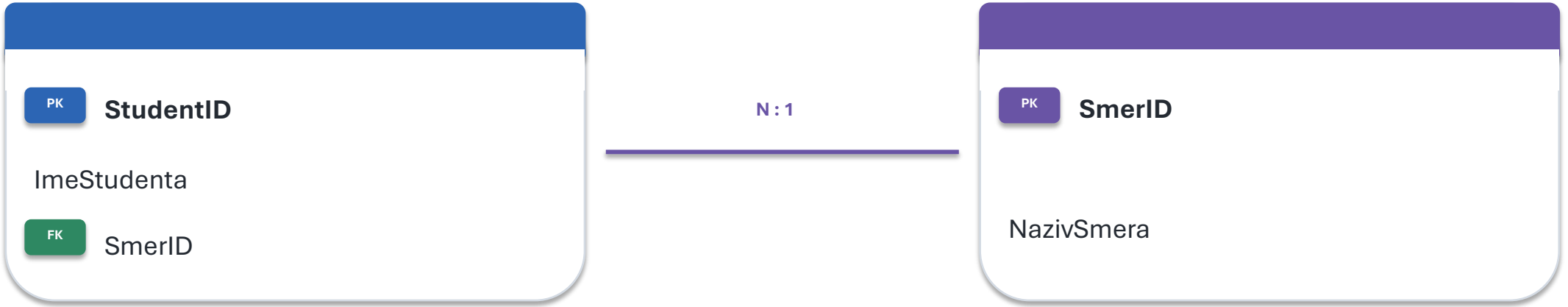
StudentID → SmerID → NazivSmera

2NF je rešena, ali STUDENT još ima tranzitivnu zavisnost — prelazimo na 3NF.

3NF — uklanjamo tranzitivnu zavisnost

StudentID → SmerID → NazivSmera.





Konačno: STUDENT čuva samo SmerID; naziv smera se čuva jednom u tabeli SMER.

Pitanje 1/5

Koji primer najjasnije krši 1NF?

A) Email = NULL

B) IdRacunara = „PC1, PC2“

C) Dva radnika rade u istoj firmi

D) Jedan radnik nema računar

Obrazložite odgovor jednom rečenicom — navedite zavisnost ili pravilo.

Pitanje 2/5

Ključ je (KursId, Datum), a NazivKursa zavisi samo od KursId. Šta imamo?

A) Viševrednosni atribut

B) Parcijalnu zavisnost

C) Tranzitivnu zavisnost

D) Denormalizaciju

Obrazložite odgovor jednom rečenicom — navedite zavisnost ili pravilo.

Pitanje 3/5

U TERMIN-u važi Učionica → BrojMesta. Šta je najbolje uraditi?

A) Duplirati BrojMesta

B) Izdvojiti PROSTORIJE

C) Ukloniti Učionicu

D) Spojiti Datum i Učionicu

Obrazložite odgovor jednom rečenicom — navedite zavisnost ili pravilo.

Pitanje 4/5

Zašto je SefID dobar model hijerarhije zaposlenih?

A) Uklanja potrebu za JOIN-om

B) Čuva podatke šefa jednom i omogućava self-join

C) Pretvara hijerarhiju u M:N

D) Eliminiše strani ključ

Obrazložite odgovor jednom rečenicom — navedite zavisnost ili pravilo.

Pitanje 5/5

Kada je denormalizacija opravdana?

A) Uvek kada ima JOIN

B) Kada je dokazano usko grlo i postoji kontrola sinhronizacije

C) Kada ne znamo ključ

D) Kada želimo manje tabela

Obrazložite odgovor jednom rečenicom — navedite zavisnost ili pravilo.