



## IT250 - BAZE PODATAKA

### Arhitektura organizacije podataka-normalizacija podataka

#### Lekcija 07

PRIRUČNIK ZA STUDENTE

# IT250 - BAZE PODATAKA

## Lekcija 07

### *ARHITEKTURA ORGANIZACIJE PODATAKA-NORMALIZACIJA PODATAKA*

- ✓ Arhitektura organizacije podataka-normalizacija podataka
- ✓ Poglavlje 1: Funkcionalna zavisnost
- ✓ Poglavlje 2: Normalizacija
- ✓ Poglavlje 3: Prva normalna forma (1NF)
- ✓ Poglavlje 4: Druga normalna forma (2NF)
- ✓ Poglavlje 5: Treća normalna forma (3NF)
- ✓ Poglavlje 6: Boyce-Codd normalna forma (BCNF)
- ✓ Poglavlje 7: Četvrta normalna forma-4NF
- ✓ Poglavlje 8: Veze između normalnih formi
- ✓ Poglavlje 9: Primena normalizacije u projektovanju BP
- ✓ Poglavlje 10: Pokazna vežba
- ✓ Poglavlje 11: Domaći zadatak
- ✓ Zaključak

Copyright © 2017 – UNIVERZITET METROPOLITAN, Beograd. Sva prava zadržana. Bez prethodne pismene dozvole od strane Univerziteta METROPOLITAN zabranjena je reprodukcija, transfer, distribucija ili memorisanje nekog dela ili čitavih sadržaja ovog dokumenta., kopiranjem, snimanjem, elektronskim putem, skeniranjem ili na bilo koji drugi način.

Copyright © 2017 BELGRADE METROPOLITAN UNIVERSITY. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, without the prior written permission of Belgrade Metropolitan University.

# ▼ Uvod

## UVOD

### Šta ćemo naučiti u ovoj lekciji?

U ovoj lekciji ćemo naučiti:

*-Šta je normalizacija i koju ulogu ima u procesu dizajna baze podataka*

*-O normalnim formama 1NF, 2NF, 3NF, BCNF i 4NF*

*-Kako se normalne forme mogu transformisati iz nižih u više normalne forme*

*-Da se normalizacija i ER modeliranje koriste istovremeno kako bi se postigao dobar dizajn baze podataka*

*-Da neke situacije zahtevaju denormalizaciju kako bi se efikasno generisale informacije.*

Dobar dizajn baze podataka mora biti usklađen s dobrim strukturama tabela. Naučićemo kako da procenimo i dizajniramo dobre strukture tabela kako bismo kontrolisali redundancije podataka i izbegli anomalije podataka. Proces koji dovodi do takvih poželjnih rezultata poznat je kao normalizacija.

1.

Saznaćete kako da dizajnirate dobru strukturu tabela i kako da popravite postojeću lošu.

2.

Otkrićete ne samo da se anomalije podataka mogu eliminisati kroz normalizaciju, već i da je pravilno normalizovan skup struktura tabela zapravo manje komplikovan za korišćenje od nenormalizovanog skupa.

3.

Pored toga, naučićete da normalizovani skup struktura tabela vernije odražava stvarne operacije organizacije.

U lekciji se takođe govori i o denormalizaciji podataka što je postupak suprotan normalizaciji, a primenjuje se u slučaju baza podataka koje se samo čitaju.

## ▼ Poglavlje 1

# Funkcionalna zavisnost

## ŠTA JE FUNKCIONALNA ZAVISNOST?

*Kada vrednost jednog ili više atributa relacije odredjuje vrednosti drugih atributa relacije*

Funkcionalna zavisnost (FD) je relacija između dva atributa, obično između primarnog ključa (PK) i drugih ne-ključnih atributa unutar relacije (tabele). **Za bilo koju relaciju R, atribut B je potpuno funkcionalno zavisan od atributa A ako svaka vrednost atributa A određuje samo jednu vrednost atributa B.** Odnosno, **atribut A određuje atribut B (tj. B je funkcionalno zavisan od A) ako svi redovi u tabeli koji se slažu u vrednosti za atribut A takođe se slažu u vrednosti za atribut B.**

Ova relacija se označava na sledeći način:  $A \twoheadrightarrow B$ .

Leva strana se naziva determinanta a desna strana je zavisnost.

Funkcionalne zavisnosti su važne u projektovanju baze podataka jer nam omogućuju uklanjanje redundantnih podataka. Koristeći funkcionalne zavisnosti, možete izvršiti normalizaciju baza podataka i na taj način stvoriti efikasniju bazu podataka.

Potpuna funkcionalna zavisnost je stanje normalizovane baze podataka koje podrazumeva zadovoljenje Druge normalne forme (2NF). Ako je atribut B funkcionalno zavisan od složenog ključa A, ali ne i od bilo kog podskupa tog složenog ključa, atribut B je **potpuno funkcionalno zavisan** od A.

Primeri funkcionalne zavisnosti:

**Primer 1:**  $SIN \twoheadrightarrow Name, Address, Birthdate$

*U ovom primeru, SIN je determinanta i određuje attribute Name, Address i Birthdate. Za dati SIN, možemo odrediti bilo koji drugi atribut u tabeli.*

**Primer 2:**  $SIN, Course \twoheadrightarrow DateCompleted$

*U ovom primeru SIN i Course određuju datum završetka DateCompleted. U ovom slučaju bi imali složeni primarni ključ*

**Primer 3:**  $ISBN \twoheadrightarrow Title$

*Treći primer označava da ISBN određuje Title*

**Primer 4:** *Pretpostavimo da posedujemo torbe koje mogu biti crvene, plave ili žute boje. Takođe, možemo pretpostaviti da su crvene torbe teške 5 kg, plave 3 kg, a žute 7 kg. Ako neko pogleda u torbu i kaže vam njenu boju, vi možete reći njenu težinu. To možemo napisati kao:*

*Boja  $\rightarrow$  Težina*

*Možemo takođe definisati da od boje torbe zavisi i njen oblik:*

Boja → Oblik, što se može zajedno definisati kao:

Boja → Težina, Oblik

## PRIMERI FUNKCIONALNE ZAVISNOSTI

*Ovde je dato još nekoliko primera funkcionalne zavisnosti*

**Primer 5:** Pretpostavimo da smo kupili nekoliko kutija keksa i da svaka od njih košta po 50 dinara. Imajući ovu činjenicu u vidu, cenu nekoliko kutija keksa možemo izračunati kao:

$CenaKeksa = BrojKutija * 50 \text{ dinara}$

Uopšteniji način da se izrazi veza između CeneKeksa i BrojKutija je reći da CeneKeksa zavisi od broja kutija. Još formalnije se može reći da je CeneKeksa funkcionalno zavisna od BrojKutija što se može napisati kao:

$BrojKutija \rightarrow CenaKeksa$

Ovaj izraz se može čitati kao: BrojKutija određuje CenaKeksa.

Korišćenjem druge formule, CenaKeksa se može sračunati množenjem Količine i JediničneCene.

$Cena = Količina * JediničnaCena$

U ovom slučaju se kaže da je Cena funkcionalno zavisna od Količine i JediničneCene.

$Količina, JediničnaCena \rightarrow Cena$

U ovom slučaju je determinanta složena ( Količina, JediničnaCena ).

**Primer 6:** Ako posmatramo relaciju:

MOVIES (title, year, length, genre, studioName, starName)

možemo primetiti da u njoj možemo otkriti sledeću funkcionalnu zavisnost:

$title, year \rightarrow length, genre, studioName$

Ova funkcionalna zavisnost definiše da ukoliko dve torke imaju istu vrednost za atribut title, year, tada te dve torke moraju da imaju istu vrednost i za atribut length, genre i studioName. Ovako definisana funkcionalna zavisnost ima smisla jer verujemo da nije moguće da postoje 2 filma sa istim nazivom koji su objavljeni iste godine, a koji imaju iste vrednosti za atribut length, genre i studioName (jedinствене vrednosti title, year određuju јединствене vrednosti za length genre studioName ).

Sa druge strane, ukoliko posmatramo:

$title, year \rightarrow starName$

Vidimo da je greška i da ovo ne predstavlja funkcionalnu zavisnost. Sa sigurnošću možemo da znamo da u bazi postoji više vrednosti atributa starName za određenu vrednost atributa title, year. Razlog koji govori u prilog navedenom, je taj što se funkcionalna zavisnost odnosi na sve slučajeve u relaciji, ne samo na jedan slučaj. Činjenica da smo mogli imati više elemenata za starName za jedan film, isključuje mogućnost da title, year funkcionalno odrede starName.

# PRAVILA FUNKCIONALNE ZAVISNOSTI

*Normalizacija počinje identifikacijom zavisnosti u datoj relaciji i postepenim razdvajanjem relacije (tabele) na skup novih relacija (tabela) na osnovu identifikovanih zavisnosti.*

Normalizacija počinje identifikacijom zavisnosti u datoj relaciji i postepenim razdvajanjem relacije (tabele) na skup novih relacija (tabela) na osnovu identifikovanih zavisnosti.

Dve vrste funkcionalnih zavisnosti koje su od posebnog interesa u normalizaciji su **parcijalne zavisnosti** i **tranzitivne zavisnosti**.

Parcijalna zavisnost postoji kada postoji funkcionalna zavisnost u kojoj je determinanta samo deo primarnog ključa (pretpostavljamo da postoji samo jedan kandidatski ključ). Na primer, ako  $(A, B) \rightarrow (C, D)$ ,  $B \rightarrow C$ , i  $(A, B)$  je primarni ključ, onda je funkcionalna zavisnost  $B \rightarrow C$  parcijalna zavisnost, jer je potreban samo deo primarnog ključa (B) da bi se odredila vrednost C. Parcijalne zavisnosti su obično prilično jednostavne i lako se identifikuju.

Tranzitivna zavisnost postoji kada postoje funkcionalne zavisnosti kao što su  $X \rightarrow Y$ ,  $Y \rightarrow Z$ , pri čemu je X primarni ključ. U tom slučaju, zavisnost  $X \rightarrow Z$  je tranzitivna zavisnost jer X određuje vrednost Z preko Y. Za razliku od parcijalnih zavisnosti, tranzitivne zavisnosti je teže prepoznati među skupom podataka. Srećom, postoji lakši način za identifikaciju tranzitivnih zavisnosti.

Tranzitivna zavisnost će se javiti samo kada postoji funkcionalna zavisnost među neosnovnim atributima. U prethodnom primeru, stvarna tranzitivna zavisnost je  $X \rightarrow Z$ . Međutim, zavisnost  $Y \rightarrow Z$  ukazuje na postojanje tranzitivne zavisnosti. Stoga, tokom diskusije o procesu normalizacije, postojanje funkcionalne zavisnosti među neosnovnim atributima će se smatrati znakom tranzitivne zavisnosti. Da bi se rešili problemi vezani za tranzitivne zavisnosti, vrše se promene na strukturi tabele na osnovu funkcionalne zavisnosti koja signalizira postojanje tranzitivne zavisnosti. Da bismo pojednostavili opis normalizacije, razmatraćemo signalizirajuću zavisnost kao tranzitivnu zavisnost.

## ▼ Poglavlje 2

# Normalizacija

## ZAŠTO SE VRŠI NORMALIZACIJA?

*Da bi se eliminisala redundantnost podataka i izbegla moguća korupcija baza podataka*

Jedna od važnijih tema u oblasti upravljanja bazama podataka je proces normalizacije tabela u relacionoj bazi podataka.

Ideja normalizacije je relativno jednostavna. Normalizacijom želimo da projektujemo skup relacija-tabela za našu bazu podataka koje:

1. **sadrže sve podatke neophodne za primenu date baze podataka**
2. **imaju što manje ponavljanja**
3. **dozvoljavaju efikasno ažuriranje podataka u bazi podataka i**
4. **izbegavaju opasnosti gubitka podataka bez znanja.**

Primarni razlog za normalizaciju baza podataka leži u činjenici da je normalizacija moćno oružje protiv moguće korupcije baza podataka koja potiče od onog što nazivamo "anomalije prilikom ubacivanja", "anomalije prilikom brisanja" i "anomalije prilikom ažuriranja". Ovi tipovi grešaka mogu da postoje u nedovoljno normalizovanim bazama podataka.

## KADA NASTAJE KORUPCIJA BAZE PODATAKA ?

*Prilikom pojave anomalije brisanja, unosa i ažuriranja*

Korupcija baze podataka može nastati prilikom brisanja, unosa i ažuriranja podataka. Korupcija baze podataka se može pokazati na primeru sledeće tabele (slika 2.1) :

*Primer 1: Pod pretpostavkom da se iz relacije prikazane na slici 1 izbriše slog čiji je OправкаBroj 2100, izgubili bi se podaci ne samo o obavljenoj opravci već i o mašini nad kojom je rađena popravka. Brisanjem ovog reda, izgubili bismo informacije o dve različite stvari: mašini i njenoj opravci. To je anomalija brisanja.*

*Primer 2: Pretpostavimo sada da želimo da unesemo nove podatke o nekoj izvršenoj opravci. Da bismo to uradili potrebno je da znamo ne samo broj opravke, datum i cenu opravke, već i podatke o samoj mašini – tip i cenu koštanja, što može biti problem jer oni koji rade na opravci obično te podatke ne znaju. Struktura ove tabele nas tera da unesemo činjenice o dva entiteta što se naziva anomalija unosa.*

StavkaBroj	Tip	CenaKoštanja	OpravkaBroj	OpravkaDatum	OpravkaCena
100	Presa	3500	2000	05.05.2007	375
200	Bušilica	4750	2100	05.07.2007	255
100	Presa	3500	2200	01.02.2007	178
300	Strug	27300	2300	10.02.2007	1785
100	Presa	3500	2400	12.03.2007	0
100	Presa	3500	2600	15.04.2007	275

Slika 2.1 Primer tabele nad kojom je moguća korupcija [Izvor: NM IT350-2020/2021.]

*Primer 3: Na kraju, pretpostavimo da želimo da ažuriramo postojeće podatke. Ako menjamo vrednosti o broju opravke, datumu i ceni opravke, verovatno neće biti problema. Ali ako menjamo vrednosti koje se odnose na broj, tip i cenu koštanja mašine, može nastati problem. Pretpostavimo da vrednosti zadnjeg reda tabele menjamo na*

*(100, 'Presa', 5500, 2500, 20.05.2007, 275) čime dobijamo sledeći sadržaj tabele (slika 2.2):*

*U ovom slučaju dolazi do anomalije jer Presa ima dve različite cene koštanja što se naziva anomalijom ažuriranja.*

Normalizacija može da se posmatra kao serija koraka (nivoa) projektovanih jedan za drugim kojima se tabele mogu pojednostaviti tj. izbeći postojanje "previše komplikovanih" tabela i umanjiti mogućnost pojave anomalija u bazi podataka.

StavkaBroj	Tip	CenaKoštanja	OpravkaBroj	OpravkaDatum	OpravkaCena
100	Presa	3500	2000	05.05.2007	375
200	Bušilica	4750	2100	05.07.2007	255
100	Presa	3500	2200	01.02.2007	178
300	Strug	27300	2300	10.02.2007	1785
100	Presa	3500	2400	12.03.2007	0
100	Presa	5500	2500	20.05.2007	275

Slika 2.2 Sadržaj tabele nakon ažuriranja [Izvor: NM IT350-2020/2021.]

## PRIMER KORUPCIJE BAZA PODATAKA NA PRIMERU RELACIJE MOVIES

*Primer anomalija kod operacija unosa, brisanja, ažuriranja nad relacijom MOVIES*

1. Anomalija unosa - pojava kada se neke informacije bez potrebe pojavljuju više puta, odnosno u više torki. Primer za to su atributi *length* i *genre* ili *studioName* u primeru prikazanom na slici 2.3.

<i>title</i>	<i>year</i>	<i>length</i>	<i>genre</i>	<i>studioName</i>	<i>starName</i>
Star Wars	1977	124	SciFi	Fox	Carrie Fisher
Star Wars	1977	124	SciFi	Fox	Mark Hamill
Star Wars	1977	124	SciFi	Fox	Harrison Ford
Gone With the Wind	1939	231	drama	MGM	Vivien Leigh
Wayne's World	1992	95	comedy	Paramount	Dana Carvey
Wayne's World	1992	95	comedy	Paramount	Mike Meyers

Slika 2.3 Primer pojave redundantnih podataka kod atribura *length* i *genre* [Izvor: NM IT350-2020/2021.]



2. **Anomalije ažuriranja** - situacija kada se vrši neka promena u okviru jedne torke, a ista informacija ostane nepromenjena u nekoj drugoj torci. *Na primer, ako je ustanovljeno da film Star Wars ima dužinu od 124 minuta, možemo lako izvršiti promenu na prvoj torci sa slike 3, ali ne i u drugoj, odnosno trećoj torci. Bez obzira što se to može podvesti pod nemarnost pri ažuriranju, moguće je ovakve situacije ograničiti redizajniranjem relacije tako da se ukloni opasnost od takvih grešaka.*

3. **Anomalije brisanja** - situacija kada se skup vrednosti obriše i na taj način prouzrokuje gubljenje ostalih informacija kao sporedni efekat. *Na primer, ukoliko je, iz primera sa slike 3, potrebno obrisati Vivien Leigh iz skupa glumačkih zvezda filma Gone With the Wind, u tom slučaju kako za pomenuti film nemamo više filmskih zvezda, kao sporedni efekat nestaju i ostale informacije o filmu.*

## DEKOMPOZICIJA RELACIJE MOVIES ZBOG KORUPCIJE

### Potreba za dekompozicijom relacije MOVIES zbog uočenih anomalija

Zbog navedenih anomalija koje se javljaju, relaciju MOVIES (slika 2.4), je potrebno dekomponovati

<i>title</i>	<i>year</i>	<i>length</i>	<i>genre</i>	<i>studioName</i>	<i>starName</i>
Star Wars	1977	124	SciFi	Fox	Carrie Fisher
Star Wars	1977	124	SciFi	Fox	Mark Hamill
Star Wars	1977	124	SciFi	Fox	Harrison Ford
Gone With the Wind	1939	231	drama	MGM	Vivien Leigh
Wayne's World	1992	95	comedy	Paramount	Dana Carvey
Wayne's World	1992	95	comedy	Paramount	Mike Meyers

Slika 2.4 Relacija MOVIES koju treba dekomponovati [Izvor: NM IT350-2020/2021.]

Da bismo otklonili anomalije, relaciju MOVIES dekomponujemo na sledeće relacije:

1. Relaciju MOVIES2 koja sadrži sve atribute osim *starName* (Slika 2.5)
2. Relaciju MOVIES3 koja sadrži atribute *title*, *year* i *starName* (Slika 2.6)

<i>title</i>	<i>year</i>	<i>length</i>	<i>genre</i>	<i>studioName</i>
Star Wars	1977	124	sciFi	Fox
Gone With the Wind	1939	231	drama	MGM
Wayne's World	1992	95	comedy	Paramount

Slika 2.5 Relacija MOVIES2 [Izvor: NM IT350-2020/2021.]

<i>title</i>	<i>year</i>	<i>starName</i>
Star Wars	1977	Carrie Fisher
Star Wars	1977	Mark Hamill
Star Wars	1977	Harrison Ford
Gone With the Wind	1939	Vivien Leigh
Wayne's World	1992	Dana Carvey
Wayne's World	1992	Mike Meyers

Slika 2.6 Relacija MOVIES3 [Izvor: NM IT250-2020/2021.]

Ono što je bitno je da smo ovakvom dekompozicijom eliminisali prethodno pomenute anomalije.

1. Redundantnost je eliminisana npr. kod atributa *length* u relaciji MOVIES2, jer se svaki film pojavljuje samo jednom. Rizik od anomalija koje nastaju ažuriranjem je eliminisan jer sada ukoliko dođe do izmene dužine trajanja nekog filma, izmena je samo na jednom mestu i nema mogućnosti da se nakon ažuriranja dužina istog filma pojavi kao različita vrednost u istoj bazi podataka.
2. Takođe su uklonjene anomalije brisanja, jer uklanjanjem atributa *starName* za neki film iz relacije MOVIES3, ne utiče na relaciju MOVIES2, pa ostale informacije o filmu neće biti obrisane.

Može se desiti da se u relaciji MOVIES3 pojavi redundancija, ako se dogodi da se *title* i *year* pojave više puta. Međutim ako su ova dva atributa definisana kao ključ ove relacije i uz to relacija MOVIES3 ne pruža mogućnost za anomalije ažuriranja, to se ne može desiti.

## NORMALNE FORME

### *Dele se u tri glavne kategorije*

Postoje tri grupe normalnih formi (NF):

1. **Normalne forme koje služe da se eliminišu anomalije prouzrokovane postojanjem funkcionalne zavisnosti:** Tu spadaju: prva normalna forma (1NF) , druga normalna forma (2NF) , treća normalna forma (3NF) i Boyce-Codd normalna forma (BCNF).
2. **Normalne forme kojima se mogu eliminisati anomalije prouzrokovane postojanjem tzv. zavisnosti od više vrednosti.** Tu spada četvrta normalna forma 4NF.
3. O normalnim formama **kojima se mogu eliminisati anomalije koje se odnose na dosta retka i specifična ograničenja podataka**, gde spada 5NF, u ovoj lekciji neće biti reči.

## ▼ Poglavlje 3

### Prva normalna forma (1NF)

#### KADA JE RELACIJA U PRVOJ NORMALNOJ FORMI (1NF)?

*Ako svi njeni atributi imaju samo atomske (nedeljive) vrednosti*

Relacija je u prvoj normalnoj formi (1NF) ako svi njeni atributi imaju samo atomske (nedeljive) vrednosti. Atomske ili nedeljive vrednosti su vrlo često u zavisnosti od konteksta drugačije definisane.

Table name: RPT_FORMAT					Database name: Ch06_ConstructCo	
PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CLASS	CHG_HOUR	HOURS
15	Evergreen	103	June E. Arbough	Elect. Engineer	84.50	23.8
		101	John G. News	Database Designer	105.00	19.4
		105	Alice K. Johnson *	Database Designer	105.00	35.7
		106	William Smithfield	Programmer	35.75	12.6
		102	David H. Senior	Systems Analyst	96.75	23.8
18	Amber Wave	114	Annelise Jones	Applications Designer	48.10	24.6
		118	James J. Frommer	General Support	18.36	45.3
		104	Anne K. Ramoras *	Systems Analyst	96.75	32.4
22	Rolling Tide	112	Darlene M. Smithson	DSS Analyst	45.95	44.0
		105	Alice K. Johnson	Database Designer	105.00	64.7
		104	Anne K. Ramoras	Systems Analyst	96.75	48.4
		113	Delbert K. Joenbrood *	Applications Designer	48.10	23.6
		111	Geoff B. Wabash	Clerical Support	26.67	22.0
25	Starflight	106	William Smithfield	Programmer	35.75	12.8
		107	Maria D. Alonzo	Programmer	35.75	24.6
		115	Travis B. Bawangi	Systems Analyst	96.75	45.8
		101	John G. News *	Database Designer	105.00	56.3
		114	Annelise Jones	Applications Designer	48.10	33.1
		108	Ralph B. Washington	Systems Analyst	96.75	23.6
		118	James J. Frommer	General Support	18.36	30.5
		112	Darlene M. Smithson	DSS Analyst	45.95	41.4

Slika 3.1 Tabularna reprezentacija formata izveštaja. Izvor: [1]

Zbog toga što relacijski model posmatra podatke kao deo tabele ili skupa tabela u kojima se moraju identifikovati sve vrednosti ključeva, podaci prikazani na slici 3.1 možda neće biti sačuvani baš kao što je prikazano. Primetimo da slika 3.1 sadrži ono što je poznato kao *ponavljajuće grupe*. Ponavljajuća grupa dobija svoje ime zbog činjenice da grupa višestrukih unosa istog tipa može postojati za svako pojedinačno pojavljivanje ključnog atributa. Obratimo pažnju da se svako pojedinačno pojavljivanje broja projekta (PROJ\_NUM) može odnositi na grupu povezanih unosa podataka. Na primer, projekat Evergreen (PROJ\_NUM = 15) prikazuje pet unosa na ovom mestu - a ti unosi su povezani jer svaki deli karakteristiku PROJ\_NUM = 15. Svaki put kada se unese novi zapis za projekat Evergreen, broj unosa u grupi se povećava za jedan.

Relaciona tabela ne sme sadržati ponavljajuće grupe. Postojanje ponavljajućih grupa ukazuje da tabela RPT\_FORMAT prikazana na slici 3.1 ne zadovoljava čak ni najniže zahteve normalnih formi, što odražava redundanciju podataka. Normalizacija strukture tabele će smanjiti redundanciju podataka. Ako postoje ponavljajuće grupe, moraju biti uklonjene tako što će se osigurati da svaki red definiše jedan entitet. Pored toga, moraju se identifikovati zavisnosti

kako bi se dijagnostifikovala normalna forma. Identifikacija normalne forme će omogućiti da znamo u kojoj fazi se nalazimo u procesu normalizacije. Proces normalizacije počinje jednostavnim procedurom u tri koraka: **ukloniti ponavljajuće grupe, identifikovati primarni ključ, identifikovati sve zavisnosti.**

## UKLANJANJE PONAVLJAJUĆIH GRUPA I IDENTIFIKOVANJE PRIMARNOG KLJUČA

*Objašnjavaju se prvi i drugi korak u prevođenju relacije u 1NF*

Da biste eliminisali ponavljajuće grupe, uklonite NULL vrednosti tako što ćete se pobrinuti da svaki atribut ponavljajuće grupe sadrži odgovarajuću vrednost podataka. Ta promena pretvara tabelu sa slike 1 u 1NF (prvu normalnu formu) na slici 3.2.

Table name: DATA\_ORG\_1NF Database name: Ch06\_ConstructCo

PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CLASS	CHG_HOUR	HOURS
15	Evergreen	103	June E. Arbough	Elect. Engineer	84.50	23.8
15	Evergreen	101	John G. News	Database Designer	105.00	19.4
15	Evergreen	105	Alice K. Johnson *	Database Designer	105.00	36.7
15	Evergreen	106	William Smithfield	Programmer	35.75	12.6
15	Evergreen	102	David H. Senior	Systems Analyst	96.75	23.8
18	Amber Wave	114	Annelise Jones	Applications Designer	48.10	24.6
18	Amber Wave	118	James J. Frommer	General Support	18.36	45.3
18	Amber Wave	104	Anne K. Ramoras *	Systems Analyst	96.75	32.4
18	Amber Wave	112	Darlene M. Smithson	DSS Analyst	45.95	44.0
22	Rolling Tide	105	Alice K. Johnson	Database Designer	105.00	64.7
22	Rolling Tide	104	Anne K. Ramoras	Systems Analyst	96.75	48.4
22	Rolling Tide	113	Delbert K. Joenbrood *	Applications Designer	48.10	23.6
22	Rolling Tide	111	Geoff B. Wabash	Clenical Support	26.87	22.0
22	Rolling Tide	106	William Smithfield	Programmer	35.75	12.8
25	Starflight	107	Maria D. Alonzo	Programmer	35.75	24.6
25	Starflight	115	Travis B. Bawangi	Systems Analyst	96.75	45.8
25	Starflight	101	John G. News *	Database Designer	105.00	56.3
25	Starflight	114	Annelise Jones	Applications Designer	48.10	33.1
25	Starflight	108	Ralph B. Washington	Systems Analyst	96.75	23.6
25	Starflight	118	James J. Frommer	General Support	18.36	30.5
25	Starflight	112	Darlene M. Smithson	DSS Analyst	45.95	41.4

Slika 3.2 Tabela u prvij normalnoj formi (1NF). Izvor: [1]

Raspored na slici 3.2 predstavlja više od obične "kozmetičke" promene. Čak i neobavezni posmatrač primetiće da PROJ\_NUM nije adekvatan primarni ključ, jer broj projekta ne identifikuje jedinstveno sve preostale attribute (redove).

Na primer, vrednost PROJ\_NUM 15 može identifikovati bilo koga od pet zaposlenih. Da bi se održao odgovarajući primarni ključ koji će jedinstveno identifikovati svaku vrednost atributa, novi ključ mora biti sastavljen od kombinacije PROJ\_NUM i EMP\_NUM.

Na primer, koristeći podatke prikazane na slici 3.2, ako znate da je PROJ\_NUM = 15 i EMP\_NUM = 103, tada odgovarajući unosi za attribute PROJ\_NAME, EMP\_NAME, JOB\_CLASS, CHG\_HOUR i HOURS moraju biti Evergreen, June E. Arbough, Elect. Engineer, \$84.50 i 23.8, redom.

## IDENTIFIKOVANJE SVIH ZAVISNOSTI

*Objašnjava se treći korak u pretvaranju relacije u 1NF.*

Identifikacija primarnog ključa u Koraku 2 znači da ste već identifikovali sledeću zavisnost: PROJ\_NUM, EMP\_NUM → PROJ\_NAME, EMP\_NAME, JOB\_CLASS, CHG\_HOUR, HOURS

To znači da vrednosti PROJ\_NAME, EMP\_NAME, JOB\_CLASS, CHG\_HOUR i HOURS zavise od - odnosno, određene su kombinacijom PROJ\_NUM i EMP\_NUM.

Postoje dodatne zavisnosti. Na primer, broj projekta identifikuje (određuje) naziv projekta. Odnosno, naziv projekta zavisi od broja projekta. Ovu zavisnost možete zapisati kao:

PROJ\_NUM → PROJ\_NAME

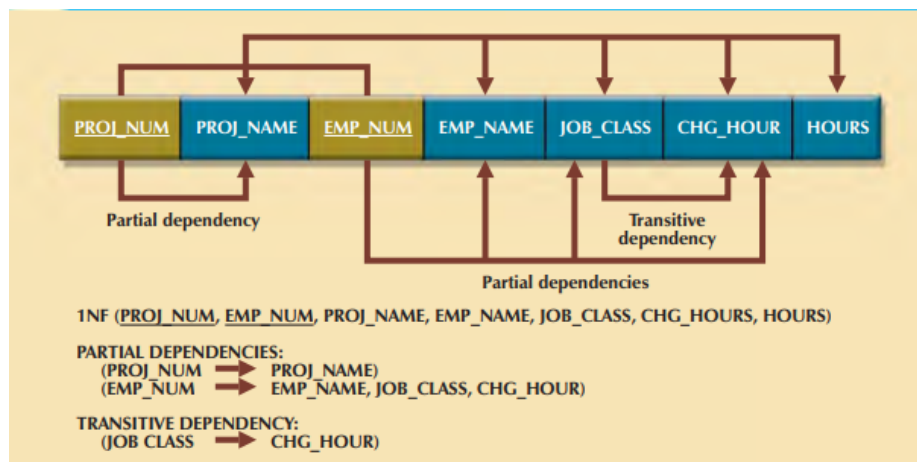
Takođe, ako znate broj zaposlenog, znate i ime tog zaposlenog, klasifikaciju posla i naknadu po satu za taj posao. Stoga možete identifikovati sledeću zavisnost:

EMP\_NUM → EMP\_NAME, JOB\_CLASS, CHG\_HOUR

Međutim, imajući u vidu prethodne delove zavisnosti, možete primetiti da poznavanje klasifikacije posla znači poznavanje naknade po satu za tu klasifikaciju posla. Odnosno, možete identifikovati poslednju zavisnost:

JOB\_CLASS → CHG\_HOUR

Ova zavisnost postoji između dva neosnovna atributa; stoga je to signal da postoji tranzitivna zavisnost, i mi ćemo je nazivati tranzitivnom zavisnošću. Zavisnosti koje ste upravo ispitali takođe se mogu prikazati pomoću dijagrama prikazanog na slici 3.3. Budući da ovakav dijagram prikazuje sve zavisnosti koje su pronađene unutar date strukture tabele, poznat je kao **dijagram zavisnosti**. Dijagrami zavisnosti su vrlo korisni kako biste dobili pregled svih veza među atributima tabele i njihova upotreba smanjuje verovatnoću da će vam promaći važna zavisnost.



Slika 3.3 Dijagram zavisnosti za prvu normalnu formu (1NF). Izvor: [1]

## ▼ Poglavlje 4

# Druga normalna forma (2NF)

## KADA JE RELACIJA U DRUGOJ NORMALNOJ FORMI (2NF)?

*Ako je svaki atribut koji nije primarni ključ funkcionalno zavisen od celog primarnog ključa.*

Relacija je u drugoј normalnoj formi (2NF) ako je svaki atribut koji nije primarni ključ funkcionalno zavisen od celog primarnog ključa.

Druga normalna forma je zadovoljena ako se može primeniti jedan od sledećih uslova:

1. **Primarni ključ se sastoji od samo jednog atributa**
2. **U relaciji ne postoje atributi koji nisu primarni ključevi**
3. **Svaki atribut koji nije primarni ključ je funkcionalno zavisen od potpunog skupa atributa primarnih ključeva.**

*Primer: ZAPOSLENI je jedan primer relacije koja nije u drugoj normalnoj formi. Skraćena notacija za tu relaciju je:*

*ZAPOSLENI (Zaposleni\_ID, Ime, Odeljenje, Plata, Kurs, Datum\_Zavrsetka)*

*Funkcionalne zavisnosti u ovoj relaciji su:*

*Zaposleni\_ID → Ime, Odeljenje, Plata*

*Zaposleni\_ID, Kurs → Datum\_Zavrsetka*

*Primarni ključ za ovu relaciju je sastavljen od Zaposleni\_ID i Kurs. Međutim, atributi koji nisu primarni ključevi, Ime, Odeljenje i Plata su funkcionalno zavisni samo od Zaposleni\_ID a ne i od Kurs. ZAPOSLENI ima redundansu što stvara teškoće kad se tabela ažurira.*

*Da bi se relacija prebacila u drugu normalnu formu, relaciju treba dekomponovati u nove relacije korišćenjem atributa determinanti koje će postati primarni ključevi tih relacija. ZAPOSLENI je dekomponovan u sledeće dve relacije:*

*ZAPOSLENI (Zaposleni\_ID, Ime, Odeljenje, Plata): ova relacija zadovoljava prvi uslov druge normalne forme*

*ZAPOSLENI\_KURS (Zaposleni\_ID, Kurs, Datum\_Zavrsetka): ova relacija zadovoljava treći uslov druge normalne forme*

## KADA RELACIJA NE ZADOVOLJAVA 2NF?

*Relacija neće zadovoljiti 2NF ako su njom predstavljena dva tipa entiteta iz relanog sveta*

Konverzija u 2NF se vrši samo kada prva normalna forma (1NF) ima složen primarni ključ. Ako 1NF ima primarni ključ sa samo jednim atributom, tada je tabela automatski u 2NF. Konverzija iz 1NF u 2NF je jednostavna. Počinjemo od formata 1NF prikazanog na slici 3.3 i sledeće korake: napraviti nove tabele kako bi se eliminisale parcijalne zavisnosti, dodeliti odgovarajuće zavisne attribute.

Koristimo sliku 3.3 da bismo odredili one attribute koji zavise od parcijalnih zavisnosti. Zavisnosti za originalne komponente ključa pronalaze se ispitivanjem strelica ispod dijagrama zavisnosti prikazanog na slici 3.3. Atributi koji zavise od parcijalnih zavisnosti se uklanjaju iz originalne tabele i smeštaju u novu tabelu zajedno sa njenom determinantom. Svi atributi koji ne zavise od parcijalnih zavisnosti će ostati u originalnoj tabeli.

Dakle, tri tabele koje rezultiraju konverzijom u 2NF dobijaju odgovarajuća imena (PROJECT, EMPLOYEE i ASSIGNMENT) i opisane su sledećim relacijskim šemama:

PROJECT ( **PROJ\_NUM**, PROJ\_NAME)

EMPLOYEE ( **EMP\_NUM**, EMP\_NAME, JOB\_CLASS, CHG\_HOUR)

ASSIGNMENT ( **PROJ\_NUM**, **EMP\_NUM**, ASSIGN\_HOURS)

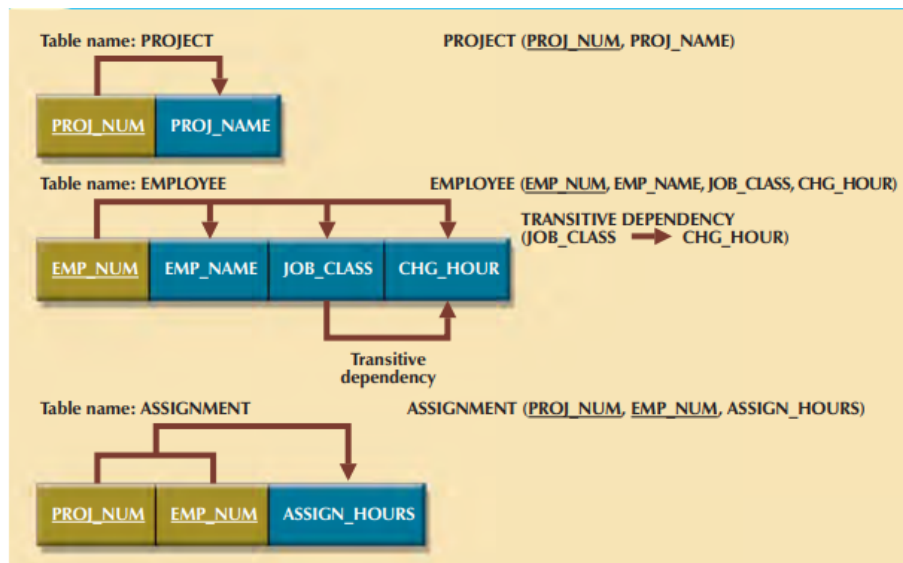
Zbog toga što je broj sati proveden na svakom projektu od strane svakog zaposlenog zavisan kako od PROJ\_NUM tako i od EMP\_NUM u tabeli ASSIGNMENT, te sate ostavljamo u tabeli ASSIGNMENT pod nazivom ASSIGN\_HOURS.

Primetimo da tabela ASSIGNMENT sadrži složeni primarni ključ sastavljen od atributa PROJ\_NUM i EMP\_NUM. Takođe primetimo da je ostavljanjem determinanata u originalnoj tabeli, kao i postavljanjem istih kao primarnih ključeva u novim tabelama, stvoreni su odnosi primarnog ključa/stranog ključa. Na primer, u tabeli EMPLOYEE, EMP\_NUM je primarni ključ. U tabeli ASSIGNMENT, EMP\_NUM je deo složenog primarnog ključa (PROJ\_NUM, EMP\_NUM) i služi kao strani ključ koji povezuje tabelu EMPLOYEE sa tabelom ASSIGNMENT.

Rezultati Koraka 1 i 2 prikazani su na slici 4.1. U ovom trenutku, većina ranije pomenutih anomalija je eliminisana. Na primer, ako sada želite da dodate, promenite ili obrišete zapis o PROJEKTU, potrebno je samo da pristupite tabeli PROJECT i izvršite promenu samo u jednom redu.

## DOVOĐENJE RELACIJE NA 2NF

*Dva tipa entiteta identifikovana u domenu problema nikada ne treba prikazati jednom, već sa dve relacije*



Slika 4.1 Rezultati konverzije u drugu normalnu formu (2NF). Izvor: [1]

Zbog toga što parcijalna zavisnost može postojati samo kada primarni ključ tabele sadrži više atributa, tabela čiji primarni ključ se sastoji samo od jednog atributa je automatski u 2NF nakon što je u 1NF.

Slika 4.1 i dalje pokazuje tranzitivnu zavisnost, koja može generisati anomalije. Na primer, ako se promeni naknada po satu za klasifikaciju posla koju ima mnogo zaposlenih, ta promena mora biti primenjena za svakog od tih zaposlenih. Ako zaboravite da ažurirate neke od zapisa o zaposlenima koji su pogođeni promenom naknade po satu, različiti zaposleni sa istim opisom posla će generisati različite naknade po satu.

Tabela je u drugoj normalnoj formi (2NF) kada:

- je u prvoj normalnoj formi (1NF).

i

- ne sadrži parcijalne zavisnosti; to znači da nijedan atribut nije zavisn samo od dela primarnog ključa.

Važno je napomenuti da je i dalje moguće da tabela u 2NF ima tranzitivnu zavisnost.



## ▼ Poglavlje 5

# Treća normalna forma (3NF)

## KADA JE RELACIJA U TREĆOJ NORMALNOJ FORMI (3NF)?

*Ako je u drugoj normalnoj formi i ako su svi atributi koji nisu deo primarnog ključa međusobno nezavisni*

Relacija je u trećoj normalnoj formi (3NF) ako je u drugoj normalnoj formi i ako su svi atributi koji nisu deo ni jednog ključa međusobno nezavisni, što se drugačije kaže da ne postoji tranzitivna zavisnost.

Jedan od načina da se relacija dovede u 3NF je na osnovu semantike modela. Znači, ne postoji univerzalno pravilo.

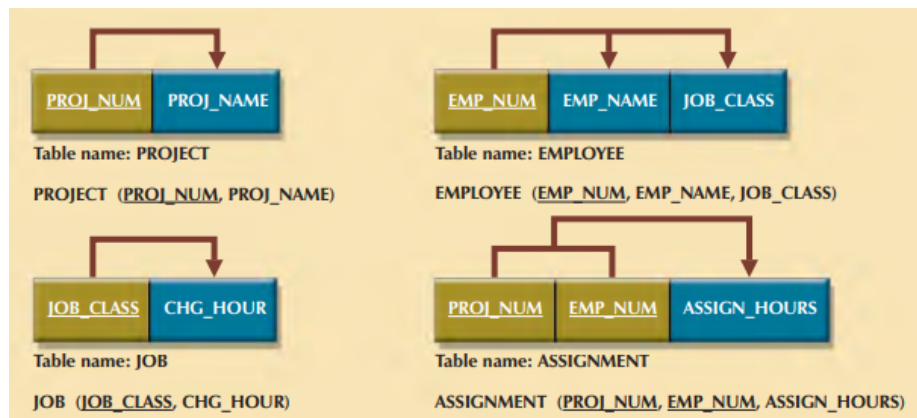
Podaci o anomalijama koje je stvorila organizacija baze podataka prikazana na slici 4.1, lako se eliminišu izvršavanjem sledeća dva koraka: napraviti nove tabele kako bi se eliminisale tranzitivne zavisnosti, premeštanjem odgovarajućih zavisnih atributa.

Za svaku tranzitivnu zavisnost, zapišite kopiju njene determinante kao primarni ključ za novu tabelu. Determinanta je svaki atribut čija vrednost određuje druge vrednosti unutar reda. Ako imate tri različite tranzitivne zavisnosti, imaćete tri različite determinante. Kao i kod konverzije u 2NF, važno je da determinanta ostane u originalnoj tabeli kako bi služila kao strani ključ.

Slika 4.1 prikazuje samo jednu tabelu koja sadrži tranzitivnu zavisnost. Stoga, determinanta za ovu tranzitivnu zavisnost je: JOB\_CLASS

Smestimo zavisne attribute u nove tabele sa njihovim determinantama i uklonimo ih iz originalnih tabela. U ovom primeru, eliminišimo CHG\_HOUR iz tabele EMPLOYEE, kako bismo ostavili definiciju zavisnosti tabele EMPLOYEE kao:

EMP\_NUM → EMP\_NAME, JOB\_CLASS



Slika 5.1 Rezultati konverzije u treću normalnu formu (3NF). Izvor: [1]

## DOVOĐENJE RELACIJE U 3NF

*Svaka tabela ima svoju determinantu i nijedna tabela ne sadrži "neprikladne" zavisnosti.*

Nacrtali smo novi dijagram zavisnosti kako bismo prikazali sve tabele koje ste definisali u Koraku 1 i 2. Imenovali smo tabele tako da odražavaju njihov sadržaj i funkciju. U ovom slučaju, "JOB" deluje adekvatno. Treba proveriti sve tabele kako bismo se uverili da svaka tabela ima determinantu i da nijedna tabela ne sadrži "neprikladne" zavisnosti. Rezultate smo videli na slici 5.1.

Nakon što je završena konverzija u treću normalnu formu (3NF), naša baza podataka će sadržati četiri tabele:

PROJECT ( **PROJ\_NUM**, PROJ\_NAME)

EMPLOYEE ( **EMP\_NUM**, EMP\_NAME, JOB\_CLASS)

JOB ( **JOB\_CLASS**, CHG\_HOUR)

ASSIGNMENT ( **PROJ\_NUM**, **EMP\_NUM**, ASSIGN\_HOURS)

Primetimo da je ovom konverzijom eliminisana originalna tranzitivna zavisnost iz tabele EMPLOYEE; tabele su sada u trećoj normalnoj formi (3NF).

*Tabela je u trećoj normalnoj formi (3NF) kada:*

- je u drugoj normalnoj formi (2NF).

i

- ne sadrži tranzitivne zavisnosti.

Bez obzira da li je "problematična" zavisnost parcijalna zavisnost ili tranzitivna zavisnost, rešenje je isto. Kreirajte novu tabelu za svaku problematičnu zavisnost. Determinanta problematične zavisnosti ostaje u originalnoj tabeli i postavlja se kao primarni ključ nove tabele. Zavisni atributi problematične zavisnosti se uklanjaju iz originalne tabele i postavljaju kao neosnovni atributi u novu tabelu. Postojanje više kandidata za ključ takođe može uticati na identifikaciju tranzitivnih zavisnosti. Prethodno je definisano da tranzitivna zavisnost postoji kada jedan neosnovni atribut određuje drugi neosnovni atribut. U prisustvu više kandidata za ključ, definicija neosnovnog atributa kao atributa koji nije deo bilo kog kandidata za ključ je ključna. Ako determinanta funkcionalne zavisnosti nije primarni ključ, već je deo nekog drugog kandidata za ključ, tada nije neosnovni atribut i ne signalizira prisustvo tranzitivne zavisnosti.

## ✓ Poglavlje 6

### Boyce-Codd normalna forma (BCNF)

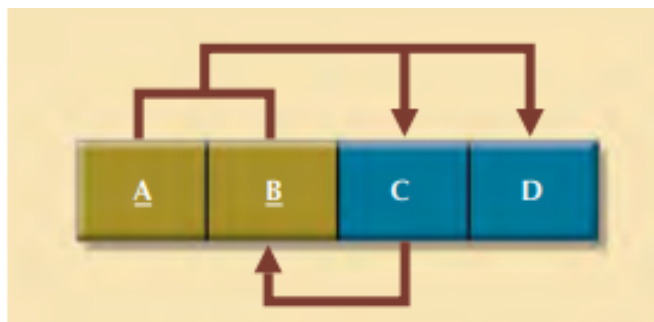
#### KADA JE RELACIJA U BOYCE-CODD NORMALNOJ FORMI (BCNF)?

*Kada svaka determinanta postane kandidat za ključ. Dovođenjem relacije na BCNF se u potpunosti eliminiše funkcionalna zavisnost*

Često je dovođenje relacije na 3NF dovoljno da se eliminišu sve funkcionalne zavisnosti, ali postoje slučajevi gde problemi ostaju. Ti problemi se mogu eliminisati ukoliko se relacije dizajniraju ili redizajniraju svođenjem na Boyce-Codd-ovu normalnu formu (BCNF).

Tabela je u Boyce-Codd normalnoj formi (BCNF) kada je svaka determinanta u tabeli kandidat za ključ. Dovođenje relacije na BCNF se vrši njenom dekompozicijom na nekoliko drugih relacija.

Tabela je u 3NF kada je 2NF i nema tranzitivnih zavisnosti. Međutim, šta ako neki atribut koji nije ključ određuje ključni atribut? Ta situacija ne krši 3NF, ali ne ispunjava zahteve BCNF-a, jer BCNF zahteva da svaka determinanta u tabeli bude kandidat za ključ. Situacija koja je opisana (tabela u 3NF koja ne ispunjava zahteve BCNF-a) prikazana je na slici 6.1.



Slika 6.1 Tabela koja je u trećoj normalnoj formi (3NF) ali nije u Boyce-Codd normalnoj formi (BCNF).  
Izvor: [1]

Primetimo ove funkcionalne zavisnosti na slici 6.1:

$A + B \rightarrow C, D$

$A + C \rightarrow B, D$

$C \rightarrow B$

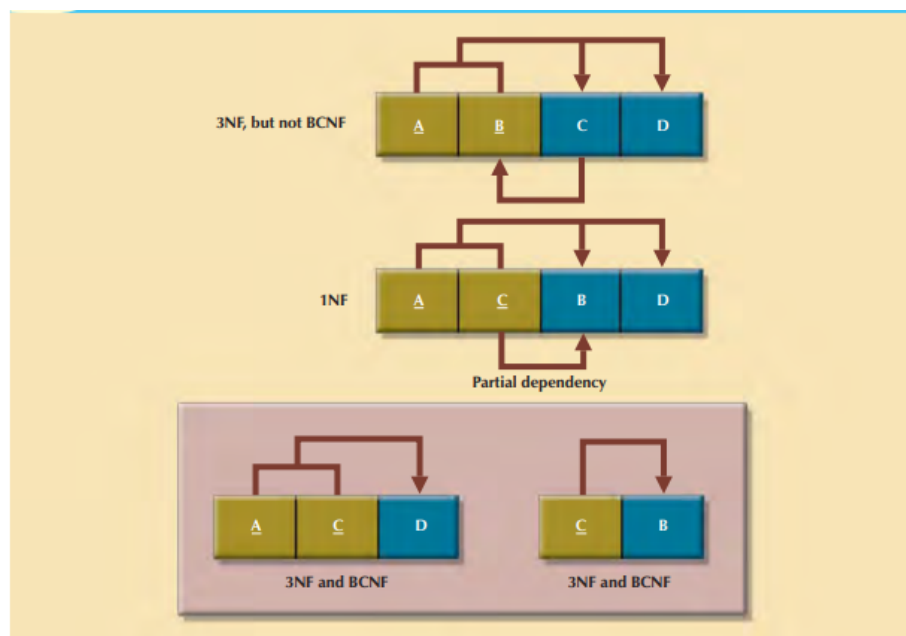
Napomenimo da ova struktura ima dva kandidata za ključeve:  $(A + B)$  i  $(A + C)$ . Struktura tabele prikazana na slici 6.1 nema parcijalne zavisnosti, niti tranzitivne zavisnosti. (Uslov  $C \rightarrow B$  ukazuje na to da neosnovni atribut određuje deo primarnog ključa - i ta zavisnost nije tranzitivna ili parcijalna jer je zavisni atribut primarni atribut!) Dakle, struktura tabele na slici

6.1 ispunjava zahteve za treću normalnu formu (3NF). Ipak, uslov  $C \rightarrow B$  uzrokuje da tabela ne ispunjava zahteve za Boyce-Codd normalnu formu (BCNF).

Da biste konvertovali strukturu tabele sa slike 6.1 u strukture tabele koje su u trećoj normalnoj formi (3NF) i u Boyce-Codd normalnoj formi (BCNF), prvo promenite primarni ključ u  $A + C$ . To je odgovarajuća akcija jer zavisnost  $C \rightarrow B$  znači da je  $C$ , u stvari, nadskup  $B$ . U ovom trenutku, tabela je u 1NF jer sadrži parcijalnu zavisnost  $C \rightarrow B$ . Zatim, sledite standardne postupke dekompozicije kako biste dobili rezultate prikazane na slici 6.2.

## STRATEGIJA SVOĐENJA RELACIJE NA BCNF

*Navedena je generalna strategija za svođenje relacije na BCNF*



Slika 6.2 Dekompozicija u Boyce-Codd normalnu formu (BCNF). Izvor: [1]

Generalna strategija je:

1. Identifikovati svaku funkcionalnu zavisnost
2. Identifikovati svaki kandidat ključ
3. Ako postoji funkcionalna zavisnost koja ima determinantu koja nije kandidat ključ:
  - kolone te funkcionalne zavisnosti treba prebaciti u novu relaciju
  - determinantu te funkcionalne zavisnosti treba proglasiti primarnim ključem nove relacije
  - kopiju determinante treba ostaviti kao sekundarni ključ originalne relacije
  - kreirati ograničenje referencijalnog integriteta između nove i originalne relacije
4. Ponavljati korak 3. sve dok je svaka determinanta svake relacije kandidat ključ

## PRIMER SVOĐENJA IZMENJENE RELACIJE NA BC NORMALNU FORMU

*Primer svođenja izmenjene relacije na BCNF*

STU_ID	STAFF_ID	CLASS_CODE	ENROLL_GRADE
125	25	21334	A
125	20	32456	C
135	20	28458	B
144	25	27563	C
144	20	32456	B

Slika 6.3 Uzorak podataka za konverziju u Boyce-Codd normalnu formu (BCNF). Izvor: [1]

Tabela na slici 6.3 prikazuje sledeće uslove:

Svaki CLASS\_CODE jedinstveno identifikuje jedan čas. Ovo ilustruje situaciju u kojoj jedan kurs može generisati mnogo časova. Na primer, kurs označen kao INFS 420 može se predavati u dva različita časa (sekcije), svaka sa jedinstvenim kodom radi lakše registracije. Tako će CLASS\_CODE 32456 identifikovati INFS 420, čas sekciju 1, dok će CLASS\_CODE 32457 identifikovati INFS 420, čas sekciju 2. Ili, CLASS\_CODE 28458 će identifikovati QM 362, čas sekciju 5.

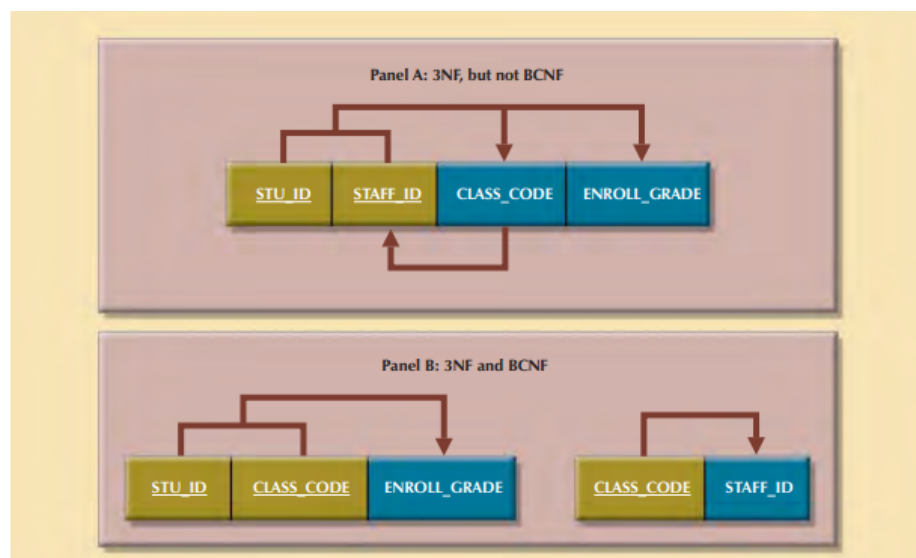
Jedan student može pohađati mnogo časova. Na primer, student 125 je pohađao i časove sa identifikatorima 21334 i 32456, pri čemu je postigao ocene A i C, redom.

Jedan član osoblja može predavati mnogo časova, ali svaki čas se predaje samo od strane jednog člana osoblja. Na primer, član osoblja sa identifikatorom 20 predaje časove sa identifikatorima 32456 i 28458.

Struktura prikazana u Tabeli sa slike 6.3, odražava se u Panelu A slike 6.4:

STU\_ID + STAFF\_ID → CLASS\_CODE, ENROLL\_GRADE

CLASS\_CODE → STAFF\_ID



Slika 6.4 Još jedna dekompozicija u Boyce-Codd normalnu formu (BCNF). Izvor: [1]

Panel A slike 6.4 prikazuje strukturu koja je jasno u 3NF, ali tabela koju ova struktura predstavlja ima veliki problem jer pokušava opisati dve stvari: dodeljivanje osoblja časovima i informacije o upisu studenata. Takva dvostruka svrha tabele izazvaće anomalije. Na primer, ako je drugi član osoblja dodeljen da predaje čas 32456, dve vrste će zahtevati ažuriranja, što dovodi do anomalije ažuriranja. Ako student 135 odustane od časa 28458, informacije o tome ko je predavao taj čas će biti izgubljene, što dovodi do anomalije brisanja. Rešenje ovog problema je dekompozicija tabele (Panel B na slici 6.4 - strukture koje odgovaraju zahtevima i 3NF i BCNF) .

## PRIMER: DOKAZ DA JE SVAKA RELACIJA SA DVA ATRIBUTA U BCNF

### *Dokaz da je svaka relacija od dva atributa u BCNF*

**PRIMER 3:** Možemo da tvrdimo da je svaka relacija od dva atributa u BCNF. Moramo ispitati sve moguće netrivialne funkcionalne zavisnosti sa jednim atributom na desnoj strani. Nema mnogo slučajeva koje treba razmotriti. Ako pretpostavimo da su  $A$  i  $B$  atributi:

1. Ne postoji ne trivijalna funkcionalna zavisnost. U ovom slučaju je ispunjen uslov za BCNF jer njega mogu da naruše samo ne trivijalne funkcionalne zavisnosti. U tom slučaju je jedini mogući ključ  $\{A, B\}$
2. Postoji funkcionalna zavisnost  $A \rightarrow B$  ali ne i zavisnost  $B \rightarrow A$ . U ovom slučaju  $A$  je jedini ključ i sve netrivialne zavisnosti sadrže  $A$  sa leve strane. U tom slučaju važi BCNF
3. Postoji funkcionalna zavisnost  $B \rightarrow A$ , obrnuto  $A \rightarrow B$  ne. Ovaj slučaj je simetričan slučaju 2.
4. Postoji funkcionalna zavisnost  $A \rightarrow B$  i  $B \rightarrow A$ . Tada su i  $A$  i  $B$  ključevi. Tada su i  $A$  i  $B$  ključevi i bar jedna od ovih funkcionalnih zavisnosti ima bar jedan od atributa  $A$  ili  $B$  s leve strane, tako da BCNF nije narušena.

Važno je pomenuti da iz slučaja 4 može nastati više od jednog ključa za relaciju, kao i to da BCNF uslov zahteva da se samo jedan ključ pojavljuje sa leve strane za svaku netrivialnu funkcionalnu zavisnost. Takođe situacija gde imamo relacije između dva atributa, gde svaki funkcionalno određuje drugi, nije nemoguća. Na primer, neka kompanije svojim zaposlenima može dodeliti jedinstvene lične karte i zasebno čuvati broj osiguranja. U tom slučaju svaki od atributa je ključan jer ne očekujemo da će se pojaviti sve torke koje će se slagati po oba atributa.

## ▼ Poglavlje 7

# Četvrta normalna forma-4NF

## ZAVISNOST OD VIŠE VREDNOSTI

*Javlja se kada jedna determinanta određuje skup vrednosti a ne samo jednu vrednost. Determinanta zavisnosti od više vrednosti ne može nikada biti primarni ključ*

Zavisnost od više vrednosti se javlja kada jedna determinanta određuje skup vrednosti a ne samo jednu vrednost.

Možete se susresti sa loše dizajniranim bazama podataka, ili vam može biti zatraženo da konvertujete tabele u formatu tabelarnih proračuna u bazu podataka u kojoj postoje višestruki više vrednosni atributi. Na primer, razmislite o mogućnosti da zaposleni ima više zaduženja i učestvuje u više organizacija za pružanje usluga. Pretpostavimo da zaposleni sa identifikatorom 10123 volontira u Crvenom krstu i Udruženju za pomoć. Takođe, isti zaposleni može biti dodeljen za rad na tri projekta: 1, 3 i 4. Slika 7.1 ilustruje kako taj skup činjenica može biti zabeležen na veoma različite načine.

Database name: Ch06\_Service

Table name: VOLUNTEER_V1		
EMP_NUM	ORG_CODE	ASSIGN_NUM
10123	RC	1
10123	UW	3
10123		4

Table name: VOLUNTEER_V2		
EMP_NUM	ORG_CODE	ASSIGN_NUM
10123	RC	
10123	UW	
10123		1
10123		3
10123		4

Table name: VOLUNTEER_V3		
EMP_NUM	ORG_CODE	ASSIGN_NUM
10123	RC	1
10123	RC	3
10123	UW	4

Slika 7.1 Tabele sa višestrukim zavisnostima (multivalued dependencies). Izvor: [1]

**Suprotno funkcionalnoj zavisnosti, determinanta zavisnosti od više vrednosti ne može nikada biti primarni ključ.**

Atributi ORG\_CODE i ASSIGN\_NUM mogu imati mnogo različitih vrednosti. U terminologiji normalizacije, ova situacija se naziva višestruka zavisnost (eng. **multivalued dependency**). Višestruka zavisnost se javlja kada jedan ključ određuje više vrednosti dva druga atributa, a ti atributi su nezavisni jedni od drugih. (Jedan zaposleni može imati mnogo unosa o učešću u organizacijama za pružanje usluga i mnogo unosa o zaduženjima. Stoga, jedan EMP\_NUM može odrediti više vrednosti za ORG\_CODE i više vrednosti za ASSIGN\_NUM; međutim, ORG\_CODE i ASSIGN\_NUM su nezavisni jedni od drugih.) Prisustvo višestruke zavisnosti znači da ako se primenjuju verzije 1 i 2, tabele će verovatno sadržati dosta null vrednosti; zapravo, tabele nemaju prikladan kandidat za ključ. (Vrednosti EMP\_NUM nisu jedinstvene, tako da ne mogu biti primarni ključevi. Nijedna kombinacija atributa u verzijama tabela 1 i 2 ne može se

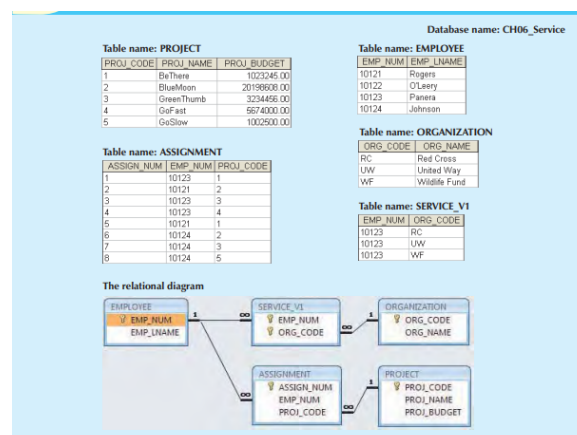
koristiti za kreiranje primarnog ključa jer neki od njih sadrže null vrednosti.) Ovakvo stanje nije poželjno, pogotovo kada postoji hiljade zaposlenih, mnogi od njih mogu imati višestruka zaduženja i učešće u mnogim aktivnostima. Verzija 3 ima bar jedan primarni ključ, ali sastoji se od svih atributa u tabeli. Međutim, verzija 3 ispunjava zahteve za treću normalnu formu (3NF), ali sadrži mnogo redundancija koje su očigledno neželjene.

## PRIMER KOMBINOVANJA DVE ZAVISNOSTI OD VIŠE VREDNOSTI U JEDNU TABELU

*Kombinovanje dve zavisnosti od više vrednosti u jednu tabelu, dovodi do anomalije ažuriranja*

Rešenje je da eliminišemo probleme koji su izazvani višestrukom zavisnošću. To postizemo tako što kreiramo nove tabele za komponente višestruke zavisnosti. U ovom primeru, višestruka zavisnost se rešava stvaranjem tabela ASSIGNMENT i SERVICE\_V1 prikazanih na slici 7.2. Primetite da ni tabela ASSIGNMENT ni tabela SERVICE\_V1 ne sadrže višestruku zavisnost. Te tabele su u 4NF.

Svi atributi moraju zavisiti od primarnog ključa, ali moraju biti nezavisni jedni od drugih. Nijedan red ne sme sadržavati dve ili više višestrukih činjenica o entitetu.



Slika 7.2 Skup tabela u četvrtoj normalnoj formi (4NF). Izvor: [1]

Zavisnost od više vrednosti se može eliminisati ako se ona izoluje u posebnu tabelu. Rezultat je uvek tabela sa dve kolone koje zajedno čine primarni ključ. Kada se zavisnost od više vrednosti izoluje na taj način, za tabelu se može reći da je četvrtoj normalnoj formi (4NF).

Relacijaje u četvrtoj normalnoj formi (4NF) ako je u BCNF i ako nema zavisnosti od više vrednosti. Zavisnost od više vrednosti se javlja kada jedna determinanta određuje **skup vrednosti, a ne samo jednu vrednost**.



## ▼ Poglavlje 8

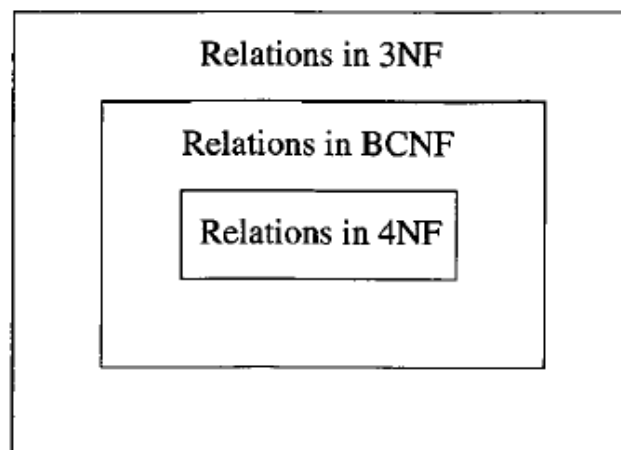
# Veze između normalnih formi

## KAKVA JE VEZA IZMEĐU NORMALNIH FORMI?

*4NF podrazumeva da je zadovoljena BCNF, koja podrazumeva da je zadovoljena 3NF.*

4NF podrazumeva da je zadovoljena BCNF, koja podrazumeva da je zadovoljena 3NF. Skup relacionih šema koje zadovoljavaju 3 normalnu formu su prikazane na slici 8.1.

Sa slike se vidi da ako je relacija u 4NF, ona je takođe u BCNF i 3NF. Takođe ako je relacija u BCNF ona je i u 3NF.



Slika 8.1 Veze između normalnih formi [Izvor: NM IT350-2020/2021.]

Drugi način da se uporede normalne forme jeste da se garantuje da se njima kreira skup relacija koje bi bile dobijene na osnovu dekompozicije u tu normalnu formu, što je prikazano na slici 8.2.

Tako, BCNF (a samim tim i 4NF) eliminišu redundantnost i druge anomalije koje nastaju usled postojanja funkcionalne zavisnosti, dok 4NF eliminiše dodatnu redundantnost koja je prouzrokovana postojanjem zavisnosti od više vrednosti, koja nije funkcionalna zavisnost. BCNF garantuje da je sprečeno postojanje funkcionalnih zavisnosti a sve normalne forme garantuju da su sprečene zavisnosti od više vrednosti.

Svojstvo	3NF	BCNF	4NF
Eliminiše redundantnost na osnovu postojanja funkcionalne zavisnosti	NE	DA	DA
Eliminiše redundantnost na osnovu postojanja zavisnosti od više vrednosti	NE	NE	DA
Ostavlja funkcionalne zavisnosti	DA	NE	NE
Ostavlja zavisnosti od više vrednosti	NE	NE	NE

Slika 8.2 Svojstva normalnih formi i njihova dekompozicija [Izvor: NM IT350-2020/2021.]

## VIDEO

*Database Normalization - Explained with Examples; Normalization - 3NF, BCNF, Multivalued Dependency, 4NF, 5NF*

**Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.**

**Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.**

## ▼ Poglavlje 9

# Primena normalizacije u projektovanju BP

## KADA PRIMENITI NORMALIZACIJU PRI PROJEKTOVANJU BP?

*Samo kod baza podataka podložnih ažuriranju, a retko kod baza podataka koje se samo čitaju*

Normalizacija se u dizajniranju baza podataka primenjuje kada se iz nekih izvora dobiju podaci koje treba smestiti u bazu podataka. Postavlja se pitanje: da li te podatke, treba na neki način transformisati ili zapamtiti takve kakvi jesu? U donošenju takve odluke, veliku ulogu ima normalizacija.

Pri donošenju konačne odluke treba takođe pažljivo proučiti principe normalizacije tj. znati šta su njene prednosti i mane što se može sumirati na sledeći način:

### 1. **Prednosti normalizacije**

- Eliminisanje anomalija
- Smanjenje redundantnih (duplih) podataka čime se eliminiše problem integriteta podataka i čuva prostor na diskovima

### 2. **Nedostaci normalizacije**

- Komplikovaniji SQL upiti sa join-om više tabela i više pod upita
- Sporiji rad aplikacija

Takođe, treba znati da li će se nova baza podataka koristiti samo za potrebe pretraživanja tj. čitanja ili će ona biti podložna i ažuriranjima.

**Normalizaciju treba primeniti samo kod baza podataka podložnih ažuriranju. Normalizacija se retko primenjuje kod baza podataka koje se samo čitaju (baze podataka koje se koriste za postavljanje upita, izveštavanje i data mining aplikacije).**

## DA LI NORMALIZACIJU UVEK TREBA PRIMENITI?

*Postoje i izuzeci kada je ne treba primeniti*

Kada se dizajniraju baze podataka podložne ažuriranju, treba se usredsrediti na anomalije modifikacije i nekonzistentnost podataka. Kao što je poznato, anomalije koje nastaju zbog postojanja funkcionalne zavisnosti se mogu eliminisati stavljanjem tabela u BCNF.

Međutim postoje i izuzeci kada ne treba vršiti normalizaciju.

*Primer: Razmotrimo tabelu:*

*KUPAC (Broj, Ime, Ulica, Mesto, Država, Poštanski\_broj)*

*Funkcionalne zavisnosti u ovoj tabeli su :*

1.

*Broj → Ime, Ulica, Mesto, Država, Poštanski\_broj*

2.

*Poštanski\_broj → Mesto, Država*

*Ova tabela nije u BCNF jer je Poštanski\_broj determinanta koja nije kandidat ključ. Tabela se može normalizovati kreiranjem dve tabele na sledeći način:*

1.

*KUPAC\_2 (Broj, Ime, Ulica, Poštanski\_broj)*

2.

*POŠTANSKI\_BROJ (Poštanski\_broj, Grad, Država)*

*Sa ograničenjem: KUPAC\_2. Poštanski\_broj mora da postoji u POŠTANSKI\_BROJ. Poštanski\_broj*  
Ove dve tabele su u BCNF ali njih treba proanalizirati u svetlu prednosti i nedostataka normalizacije koje su date u prethodnoj tabeli. Normalizacijom se eliminišu anomalije modifikacije, ali koliko se često ažuriraju podaci o Poštanskom broju? Koliko se često podaci o poštanskom broju menjaju za datu državu i grad? Skoro nikada. Tako, iako dizajn omogućava pojavu anomalija, one se neće nikada pojaviti jer se podaci neće nikada menjati.

Pored toga, dve zasebne tabele umesto jedne zahtevaju da se programima aplikacije pišu složeniji SQL upiti. Obrada dve umesto jedne tabele, takođe može usporiti samu aplikaciju. Mereći tako prednosti i nedostatke, mnogi praktičari će reći da je u ovom slučaju normalizacija loša. Podaci o poštanskom broju treba da ostanu u originalnoj tabeli.

Nasuprot anomalijama koje postoje zbog funkcionalne zavisnosti, anomalije zbog zavisnosti od više vrednosti su toliko ozbiljne da ih treba uvek eliminisati. **Kolone između kojih postoji zavisnost od više vrednosti uvek treba izdvojiti u posebnu tabelu.**

## DA LI TREBA PRIMENJIVATI NORMALIZACIJU KOD BP KOJE SE SAMO ČITAJU?

*Normalizacija je u slučaju ovakvih baza podataka retko prednost zbog nekoliko razloga*

Baze podataka koje se samo čitaju (eng. **read-only database**) se koriste za postavljanje upita, izveštavanje i **data mining** aplikacije. Pošto se takve baze podataka nikad ne ažuriraju, preporuke za njihovo dizajniranje se razlikuju od onih koje se odnose na baze podataka koje su podložne ažuriranju.

**Normalizacija je u slučaju ovakvih baza podataka retko prednost zbog nekoliko razloga:**

1.

Ako se baza podataka nikada ne ažurira, ne mogu se pojaviti anomalije modifikacije. Tako proizilazi da je jedini razlog za normalizaciju read-only baza podataka smanjiti ponavljanje podataka.

2.

Ako nema aktivnosti ažuriranja, tada ne postoji rizik od problema integriteta podataka, pa je jedini razlog za sprečavanje ponavljanja podataka štednja prostora na diskovima.

Međutim, prostor na diskovima je danas veoma jeftin pa su troškovi skladištenja podataka minimalni. **Ako su podaci normalizovani, potrebno je čitati ih iz dve ili više tabela i vreme potrebno za spajanje – join tih tabela je mnogo veće od pretraživanja u okviru jedne tabele.** U gotovo svim slučajevima normalizacija tabela u bazama podataka koje se samo čitaju je loša ideja.

Često su podaci za read-only baze podataka dobijeni iz operativnih baza podataka. Pošto su takve baze podložne ažuriranjima, oni su verovatno normalizovani.

Na osnovu iznetih razloga, verovatno je da podaci prilikom prebacivanja u read-only baze podataka ne treba da ostanu u normalizovanoj formi. U tom slučaju potrebno je izvršiti denormalizaciju ili join prethodno zapamćenih podataka.

## ▼ 9.1 Denormalizacija baza podataka

### ŠTA JE DENORMALIZACIJA?

#### *Karakteristike, razlozi, nedostaci i prednosti denormalizacije*

Važno je zapamtiti da optimalna implementacija relacione baze podataka zahteva da sve tabele budu bar u 3NF. Dobar relacioni sistem za upravljanje bazama podataka (DBMS) odlično se snalazi u upravljanju normalizovanim relacijama, tj. relacijama koje su oslobođene nepotrebnih redundancija koje mogu uzrokovati anomalije u podacima. Iako je kreiranje normalizovanih relacija važan cilj dizajna baze podataka, to je samo jedan od mnogih takvih ciljeva. Dobar dizajn baze podataka takođe uzima u obzir zahteve za obradom (ili izveštavanjem) i brzinu obrade. Problem sa normalizacijom je što se kako se tabele dekomponuju da bi se pridržavale zahteva normalizacije, broj tabela u bazi povećava. Stoga, kako bi se generisale informacije, podaci moraju biti izvučeni iz različitih tabela. Spajanje velikog broja tabela zahteva dodatne operacije ulaz/izlaz (I/O) i obradnu logiku, čime se smanjuje brzina sistema. Većina relacionih sistema za upravljanje bazama podataka može veoma efikasno rukovati spajanjem tabela. Međutim, retke i povremene okolnosti mogu dozvoliti neki stepen denormalizacije kako bi se povećala brzina obrade.

Sveobuhvatan primer potrebe za denormalizacijom zbog zahteva za izveštavanjem je slučaj izveštaja o evaluaciji fakultetskog osoblja u kojem svaki red navodi ocene dobijene tokom poslednja četiri semestra koji su predavani, slika 10.1. Iako ovaj izveštaj deluje dovoljno jednostavno, problem nastaje zbog činjenice da su podaci sačuvani u normalizovanoj tabeli u kojoj svaki red predstavlja

Instructor	Department	I		II		III		IV		Last Two Sem. Avg.
		Semester	Mean	Semester	Mean	Semester	Mean	Semester	Mean	
Alton	INFS	2003S	2.91	2004F	2.84	2004S	2.55	2003F	2.51	2.875
Arns	INFS	2003S	3.24	2004F	3.36	2004S	3.31	2003F	3.19	3.250
Crandon	INFS	2003S	3.93	2004F	3.95	2004S	3.91	2003F	3.88	3.940
Dumas	MGMT	2004F	3.66	2004S	3.69	2003F	3.56	2003S	3.72	3.675
London	BKOM	2003S	3.57	2004F	3.64	2004S	3.39	2003F	3.57	3.605
Lothar	ECON	1999F	3.53	1998F	3.53					3.530
Robins	INFS	1998S	3.50							3.500

Slika 9.1.1 Izveštaj o evaluaciji fakultetskog osoblja. Izvor: [1]

različitu ocenu za određenog člana fakulteta u određenom semestru, slika 10.2.

Table name: EVALDATA

ID	INSTRUKTOR	DEPARTMENT	MEAN	SEMESTER
370	Arns	INFS	3.2	2003F
371	Arns	INFS	2.95	2003S
372	Arns	INFS	3.15	2003S
373	Arns	INFS	3.15	2003S
374	Arns	INFS	2.94	2004F
375	Arns	INFS	2.95	2004S
376	Arns	INFS	3.91	2004F
377	Arns	INFS	2.55	2004S
378	Arns	INFS	2.84	2004F
379	Arns	INFS	3.95	2004S
380	Arns	INFS	3.36	2004F
381	Arns	INFS	3.31	2004S
382	Arns	INFS	3.57	2004F
383	Arns	INFS	3.64	2004S
384	Arns	INFS	3.39	2004F
385	Arns	INFS	3.53	2004S
386	Arns	INFS	3.53	2004F
387	Arns	INFS	3.50	2004S
388	Arns	INFS	3.50	2004F
389	Arns	INFS	3.50	2004S
390	Arns	INFS	3.50	2004F
391	Arns	INFS	3.50	2004S
392	Arns	INFS	3.50	2004F
393	Arns	INFS	3.50	2004S
394	Arns	INFS	3.50	2004F
395	Arns	INFS	3.50	2004S
396	Arns	INFS	3.50	2004F
397	Arns	INFS	3.50	2004S
398	Arns	INFS	3.50	2004F
399	Arns	INFS	3.50	2004S
400	Arns	INFS	3.50	2004F
401	Arns	INFS	3.50	2004S
402	Arns	INFS	3.50	2004F
403	Arns	INFS	3.50	2004S
404	Arns	INFS	3.50	2004F
405	Arns	INFS	3.50	2004S
406	Arns	INFS	3.50	2004F
407	Arns	INFS	3.50	2004S
408	Arns	INFS	3.50	2004F
409	Arns	INFS	3.50	2004S
410	Arns	INFS	3.50	2004F
411	Arns	INFS	3.50	2004S
412	Arns	INFS	3.50	2004F
413	Arns	INFS	3.50	2004S
414	Arns	INFS	3.50	2004F
415	Arns	INFS	3.50	2004S
416	Arns	INFS	3.50	2004F
417	Arns	INFS	3.50	2004S
418	Arns	INFS	3.50	2004F
419	Arns	INFS	3.50	2004S
420	Arns	INFS	3.50	2004F
421	Arns	INFS	3.50	2004S
422	Arns	INFS	3.50	2004F
423	Arns	INFS	3.50	2004S
424	Arns	INFS	3.50	2004F
425	Arns	INFS	3.50	2004S
426	Arns	INFS	3.50	2004F
427	Arns	INFS	3.50	2004S
428	Arns	INFS	3.50	2004F
429	Arns	INFS	3.50	2004S
430	Arns	INFS	3.50	2004F
431	Arns	INFS	3.50	2004S
432	Arns	INFS	3.50	2004F
433	Arns	INFS	3.50	2004S
434	Arns	INFS	3.50	2004F
435	Arns	INFS	3.50	2004S
436	Arns	INFS	3.50	2004F
437	Arns	INFS	3.50	2004S
438	Arns	INFS	3.50	2004F
439	Arns	INFS	3.50	2004S
440	Arns	INFS	3.50	2004F
441	Arns	INFS	3.50	2004S
442	Arns	INFS	3.50	2004F
443	Arns	INFS	3.50	2004S
444	Arns	INFS	3.50	2004F
445	Arns	INFS	3.50	2004S
446	Arns	INFS	3.50	2004F
447	Arns	INFS	3.50	2004S
448	Arns	INFS	3.50	2004F
449	Arns	INFS	3.50	2004S
450	Arns	INFS	3.50	2004F
451	Arns	INFS	3.50	2004S
452	Arns	INFS	3.50	2004F
453	Arns	INFS	3.50	2004S
454	Arns	INFS	3.50	2004F
455	Arns	INFS	3.50	2004S
456	Arns	INFS	3.50	2004F
457	Arns	INFS	3.50	2004S
458	Arns	INFS	3.50	2004F
459	Arns	INFS	3.50	2004S
460	Arns	INFS	3.50	2004F
461	Arns	INFS	3.50	2004S
462	Arns	INFS	3.50	2004F
463	Arns	INFS	3.50	2004S
464	Arns	INFS	3.50	2004F
465	Arns	INFS	3.50	2004S
466	Arns	INFS	3.50	2004F
467	Arns	INFS	3.50	2004S
468	Arns	INFS	3.50	2004F
469	Arns	INFS	3.50	2004S
470	Arns	INFS	3.50	2004F
471	Arns	INFS	3.50	2004S
472	Arns	INFS	3.50	2004F
473	Arns	INFS	3.50	2004S
474	Arns	INFS	3.50	2004F
475	Arns	INFS	3.50	2004S
476	Arns	INFS	3.50	2004F
477	Arns	INFS	3.50	2004S
478	Arns	INFS	3.50	2004F
479	Arns	INFS	3.50	2004S
480	Arns	INFS	3.50	2004F
481	Arns	INFS	3.50	2004S
482	Arns	INFS	3.50	2004F
483	Arns	INFS	3.50	2004S
484	Arns	INFS	3.50	2004F
485	Arns	INFS	3.50	2004S
486	Arns	INFS	3.50	2004F
487	Arns	INFS	3.50	2004S
488	Arns	INFS	3.50	2004F
489	Arns	INFS	3.50	2004S
490	Arns	INFS	3.50	2004F
491	Arns	INFS	3.50	2004S
492	Arns	INFS	3.50	2004F
493	Arns	INFS	3.50	2004S
494	Arns	INFS	3.50	2004F
495	Arns	INFS	3.50	2004S
496	Arns	INFS	3.50	2004F
497	Arns	INFS	3.50	2004S
498	Arns	INFS	3.50	2004F
499	Arns	INFS	3.50	2004S
500	Arns	INFS	3.50	2004F

Table name: FACHIST

ID	INSTRUKTOR	DEPARTMENT	LASTSEM	LASTMEAN	LASTSEM	LASTMEAN	LASTSEM	LASTMEAN	LASTSEM	LASTMEAN
370	Arns	INFS	2003F	3.2	2003F	3.2	2003F	3.2	2003F	3.2
371	Arns	INFS	2003S	2.95	2003S	2.95	2003S	2.95	2003S	2.95
372	Arns	INFS	2003S	3.15	2003S	3.15	2003S	3.15	2003S	3.15
373	Arns	INFS	2003S	3.15	2003S	3.15	2003S	3.15	2003S	3.15
374	Arns	INFS	2004F	2.94	2004F	2.94	2004F	2.94	2004F	2.94
375	Arns	INFS	2004S	2.95	2004S	2.95	2004S	2.95	2004S	2.95
376	Arns	INFS	2004F	3.91	2004F	3.91	2004F	3.91	2004F	3.91
377	Arns	INFS	2004S	2.55	2004S	2.55	2004S	2.55	2004S	2.55
378	Arns	INFS	2004F	2.84	2004F	2.84	2004F	2.84	2004F	2.84
379	Arns	INFS	2004S	3.95	2004S	3.95	2004S	3.95	2004S	3.95
380	Arns	INFS	2004F	3.36	2004F	3.36	2004F	3.36	2004F	3.36
381	Arns	INFS	2004S	3.31	2004S	3.31	2004S	3.31	2004S	3.31
382	Arns	INFS	2004F	3.57	2004F	3.57	2004F	3.57	2004F	3.57
383	Arns	INFS	2004S	3.64	2004S	3.64	2004S	3.64	2004S	3.64
384	Arns	INFS	2004F	3.39	2004F	3.39	2004F	3.39	2004F	3.39
385	Arns	INFS	2004S	3.53	2004S	3.53	2004S	3.53	2004S	3.53
386	Arns	INFS	2004F	3.53	2004F	3.53	2004F	3.53	2004F	3.53
387	Arns	INFS	2004S	3.50	2004S	3.50	2004S	3.50	2004S	3.50
388	Arns	INFS	2004F	3.50	2004F	3.50	2004F	3.50	2004F	3.50
389	Arns	INFS	2004S	3.50	2004S	3.50	2004S	3.50	2004S	3.50
390	Arns	INFS	2004F	3.50	2004F	3.50	2004F	3.50	2004F	3.50
391	Arns	INFS	2004S	3.50	2004S	3.50	2004S	3.50	2004S	3.50
392	Arns	INFS	2004F	3.50	2004F	3.50	2004F	3.50	2004F	3.50
393	Arns	INFS	2004S	3.50	2004S	3.50	2004S	3.50	2004S	3.50
394	Arns	INFS	2004F	3.50	2004F	3.50	2004F	3.50	2004F	3.50
395	Arns	INFS	2004S	3.50	2004S	3.50	2004S	3.50	2004S	3.50
396	Arns	INFS	2004F	3.50	2004F	3.50	2004F	3.50	2004F	3.50
397	Arns	INFS	2004S	3.50	2004S	3.50	2004S	3.50	2004S	3.50
398	Arns	INFS	2004F	3.50	2004F	3.50	2004F	3.50	2004F	3.50
399	Arns	INFS	2004S	3.50	2004S	3.50	2004S	3.50	2004S	3.50
400	Arns	INFS	2004F	3.50	2004F	3.50	2004F	3.50	2004F	3.50
401	Arns	INFS	2004S	3.50	2004S	3.50	2004S	3.50	2004S	3.50
402	Arns	INFS	2004F	3.50	2004F	3.50	2004F	3.50	2004F	3.50
403	Arns	INFS	2004S	3.50	2004S	3.50	2004S	3.50	2004S	3.50
404	Arns	INFS	2004F	3.50	2004F	3.50	2004F	3.50	2004F	3.50
405	Arns	INFS	2004S	3.50	2004S	3.50	2004S	3.50	2004S	3.50
406	Arns	INFS	2004F	3.50	2004F	3.50	2004F	3.50	2004F	3.50
407	Arns	INFS	2004S	3.50	2004S	3.50	2004S	3.50	2004S	3.50
408	Arns	INFS	2004F	3.50	2004F	3.50	2004F	3.50	2004F	3.50
409	Arns	INFS	2004S	3.50	2004S	3.50	2004S	3.50	2004S	3.50
410	Arns	INFS	2004F	3.50	2004F	3.50	2004F	3.50	2004F	3.50
411	Arns	INFS	2004S	3.50	2004S	3.50	2004S	3.50	2004S	3.50
412	Arns	INFS	2004F	3.50	2004F	3.50	2004F	3.50	2004F	3.50
413	Arns	INFS	2004S	3.50	2004S	3.50	2004S	3.50	2004S	3.50
414	Arns	INFS	2004F	3.50	2004F	3.50	2004F	3.50	2004F	3.50
415	Arns	INFS	2004S	3.50	2004S	3.50	2004S	3.50	2004S	3.50
416	Arns	INFS	2004F	3.50	2004F	3.50	2004F	3.50	2004F	3.50
417	Arns	INFS	2004S	3.50	2004S	3.50	2004S	3.50	2004S	3.50
418	Arns	INFS	2004F	3.50	2004F	3.50	2004F	3.50	2004F	3.50
419	Arns	INFS	2004S	3.50	2004S	3.50	2004S	3.50	2004S	3.50
420	Arns	INFS	2004F	3.50	2004F	3.50	2004F	3.50	2004F	3.50
421	Arns	INFS	2004S	3.50	2004S	3.50	2004S	3.50	2004S	3.50
422	Arns	INFS	2004F	3.50	2004F	3.50	2004F	3.50	2004F	3.50
423	Arns	INFS	2004S	3.50	2004S	3.50	2004S	3.50	2004S	3.50
424	Arns	INFS	2004F	3.50	2004F	3.50	2004F	3.50	2004F	3.50
425	Arns	INFS	2004S	3.50	2004S	3.50	2004S	3.50	2004S	3.50
426	Arns	INFS	2004F	3.50	2004F	3.50	2004F	3.50	2004F	3.50
427	Arns	INFS	2004S	3.50	2004S	3.50	2004S	3.50	2004S	3.50
428	Arns	INFS	2004F	3.50	2004F	3.50	2004F	3.50	2004F	3.50
429	Arns	INFS	2004S	3.50	2004S	3.50	2004S			

Negativni aspekti spojenih tabela su, kao i kod mnogih oblika denormalizacije, teško održavanje ažurnosti podataka. Spojena tabela može vrlo brzo da izgubi sinhronizaciju sa tabelama od kojih je kreirana. Zbog toga, prethodno spojene tabele su mnogo korisnije kod relativno statičkih nego kod dinamičkih podataka.

## DENORMALIZACIJA DUPLICIRANJEM/DELJENJEM TABELA

*Dupliciranje tabela se koristi kada treba napraviti: (1) tabelu za izveštavanje (2) izvršiti dupliciranje podataka; Deljenje tabele se vrši kada različiti korisnici pristupaju delovima tabele*

Dupliciranje tabela se koristi kada treba napraviti: (1) tabelu za izveštavanje (2) izvršiti dupliciranje podataka;

1. **Kreiranje tabela za izveštavanje** :Ovaj tip denormalizacije se koristi u slučajevima kad nije moguće razviti izveštaj za korisnika uz pomoć SQL-a. Ovakvi izveštaji zahtevaju dodatnu manipulaciju podacima i specijalne oblike formatiranja. Ako neki kritični ili često korišćeni izveštaji ovakve prirode treba da se urade u realnom vremenu, može se koristiti tabela koja predstavlja taj izveštaj. Podaci za ovakav izveštaj se kreiraju najčešće aplikativnim programom ili SQL-om u batch okruženju, a onda se učitaju u tabelu kao sekvencijalni podaci. Ove tabele su idealne za smeštanje rezultata višestrukih spajanja tabela, međuzavisnih upita ili ostalih složenih SQL iskaza. Ako se napravi složen upit i izvrši, a njegov rezultat smesti u tabelu, da bi se rezultati učitali umesto složenih i najčešće sporijih upita koji su bili korišćeni za popunjavanje tabele izveštaja može se koristiti jednostavna SELECT naredba.

2. **Dupliciranje podataka**: Ako je grupi korisnika redovno potreban samo podskup podataka iz neke tabele onda se kritična tabela može ponoviti u podskup za tu grupu. Za grupe korisnika koji vide samo iskopirane podatke se može postaviti pitanje latentnosti.

**Deljenje tabela** se vrši ako se različitim aplikacijama ili od strane različitih korisnika pristupa samo pojedinim delovima tabele. Tada je moguće podeliti tabelu na dva denormalizovana dela – po jedan za svaku grupu korisnika ili aplikacija. **Tabele mogu biti podeljene na dva načina: horizontalno i vertikalno.**

1.

**Vertikalno podeljena tabela** odvaja samo neke kolone u nove tabele, odnosno jedan skup kolona je u jednoj a drugi u drugoj tabeli. Primarni ključ se nalazi u obe nove tabele.

2.

Kod **horizontalne podele** tabele slogovi su klasifikovani u grupe u zavisnosti od načina grupisanja ključa tabele. Kolone u novim tabelama su iste kao i kolone originalne tabele. Treba voditi računa o dupliranju slogova u novim tabelama a. To se obezbeđuje korišćenjem primarnog ključa za deljenje tabele i to tako da se vrednost ključa dodeljuje samo jednoj novoj tabeli.

## DENORMALIZACIJA DELJENJEM KOLONA

### *Vrši se u slučaju postojanja dugačkih tekstualnih kolona*

Pretpostavimo da postoji tabela ko ja sadrži opise artikala i robe.

ARTIKAL (IdArtikla, ArtikalVelicina, ArtikalBoja, ArtikalOpis)

Opis artikla može biti dugačak i nekoliko stotina karaktera, ali mnogi artikli zahtevaju samo 10 do 20.

U nastavku je dat primer podele tabele ARTIKAL na dve tabele

1. ARTIKAL (IdArtikla, ArtikalVecina, ArtikalBoja)
2. ARTIKAL\_OPIS (IdArtikla, ArtikalOpis)

Značaj ove podele se ogleda u smanjenim zahtevima za U/I prema disku, a što je slog glavne tabele manji, više slogova se može staviti na istu stranicu diska. Postoje različite varijacije ovog tipa denormalizacije

Napomenimo da iako se tabele u 2NF-u ne mogu uvek izbeći, problem rada sa tabelama koje sadrže parcijalne i/ili tranzitivne zavisnosti u proizvodnom okruženju baze podataka ne treba umanjivati. Osim mogućnosti stvaranja problematičnih anomalija u podacima, nenormalizovane tabele u proizvodnoj bazi podataka često pate od sledećih nedostataka:

*Ažuriranje podataka je manje efikasno jer programi koji čitaju i ažuriraju tabele moraju raditi sa većim tabelama.*

*Indeksiranje je komplikovanije. Jednostavno nije praktično izgraditi sve indekse potrebne za mnoge attribute koji mogu biti smešteni u jednoj ne-normalizovanoj tabeli.*

*Nenormalizovane tabele ne pružaju jednostavne strategije za kreiranje virtuelnih tabela poznatih kao pogledi (eng. **views**).*

## DENORMALIZACIJA DODAVANJEM KOLONA

### *Primenjuje se kada se jednoj ili više kolona jedne tabele pristupa svaki put kad se traže podaci iz drugih tabela*

Ponekad se jednoj ili više kolona jedne tabele pristupa svaki put kad se traže podaci iz drugih tabela. U takvim slučajevima potrebno je razmotriti mogućnost **dodavanja redundantnih kolona tako da podaci iz tabela kojima se pristupa budu dostupni u drugim tabelama čime se eliminiše potreba za spajanje tabela što poboljšava performanse izvršenja upita.**

*Posmatrajmo primer tabela ST UDENT i SMER. Mnogi od upita nad podacima studenta zahtevaju i podatak o nazivu usmerenja koji student pohađa. S obzirom na to, naziv smer se može dodati kao redundantna kolona u tabelu STUDENT (slika 10.3 ).*

**Kolone koje se mogu dodati kao redundantne treba da imaju sledeće karakteristike:**



1. Za podršku redundantnosti je potreban samo mali broj kolona
2. Kolone treba da budu stabilne i da se retko ažuriraju
3. Treba da im pristupa veliki broj korisnika

Dodavanjem redundantnih kolona se za mnoge upite eliminiše potreba za spajanjem.

**Problemi koji nastaju ovakvim rešenjem su:**

1. zahteva se održavanje novih kolona. Sve promene se moraju uneti u dve tabele i verovatno u mnogo redova u jednoj od tabela.
2. zahteva se više prostora na disku, jer se *Naziv Smera* duplira.

STUDENT

Student	Ime	S	Prezime	Smer
1/05	Petar	P	Petrović	01
10/05	Jovan	M	Jovanović	02
134/06	Dejan	R	Stanković	01
...	...	...	...	...

SMER

Smer	Naziv Smera
01	Informatika
02	Matematika
03	Telekomunikacije
...	...

STUDENT (dodavanje kolone Naziv Smera)

Student	Ime	S	Prezime	Smer	Naziv Smera
1/05	Petar	P	Petrović	01	Informatika
10/05	Jovan	M	Jovanović	02	Matematika
134/06	Dejan	R	Stanković	01	Informatika
...	...	...	...	...	...

Slika 9.1.3 Dodavanje kolona na primeru tabela STUDENT i SMER [Izvor: NM IT350-2020/2021.]

## VIDEO

*SQL Server training :- When do we apply de-normalization ?(Interview question)*

**Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.**

## ✓ Poglavlje 10

### Pokazna vežba

#### NAČIN ORGANIZACIJE POKAZNIH VEŽBI

*Vežba je organizovana kroz uvod deo i deo za samostalni rad studenata*

Vežba je organizovana kroz uvod deo i deo za samostalni rad studenata.

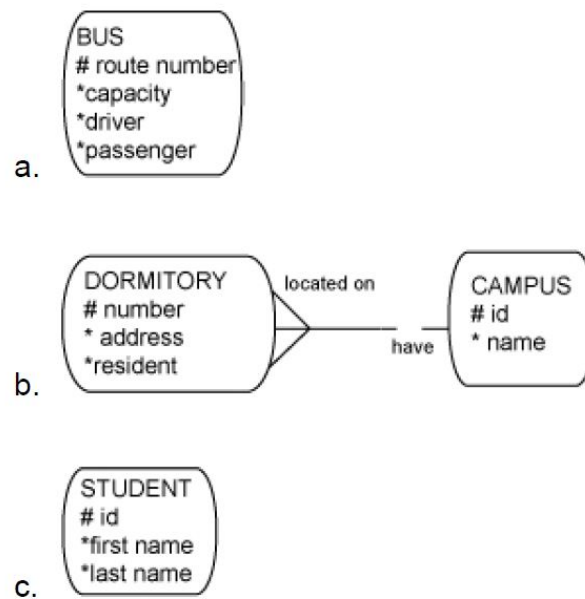
1. U uvodnom delu pokaznih vežbi se daje pokazni primer koji studentima treba da pomogne u samostalnom rešavanju zadataka.
2. Zadatke koji su zadati za samostalni rad student samostalno rešava uz pomoć asistenta.

## ✓ 10.1 Pokazni primer

### SVOĐENJE RELACIJA NA 1NF - (3 MIN.)

*Primer svođenja relacija na 1NF*

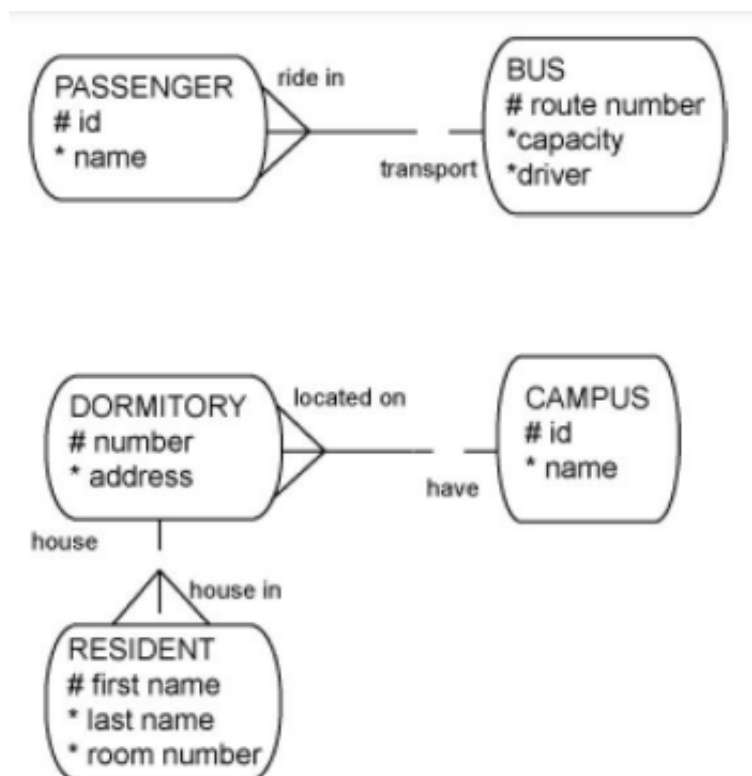
Proveriti da li su relacije prikazane na slici 11.1 u PRVOJ NORMALNOJ FORMI i ako nisu izvršiti potrebne korekcije kako bi bile. Rešenja predstaviti preko ER dijagrama



Slika 10.1.1 Zadatak 1 [Izvor: NM IT350-2020/2021.]

REŠENJE: Rešenje zadatka je prikazano na slici 11.2.

Relacije a) i c) je bilo potrebno dovesti u 1NF, dok relacija b) već ispunjava 1NF



Slika 10.1.2 Zadatak 1 - rešenje [Izvor: NM IT350-2020/2021.]

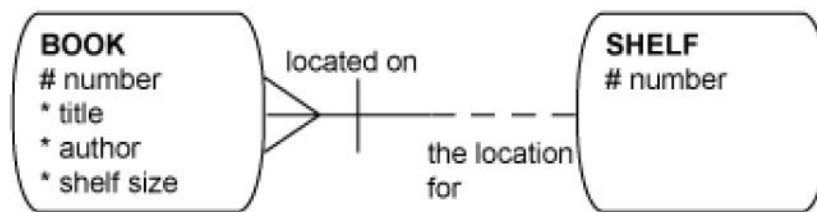
## SVOĐENJE RELACIJA NA 2NF - PRIMER 1 (3 MIN.)

### *Primer svođenja relacija na 2NF*

Proveriti da li relacija na slici 11.3 ispunjava uslove 2NF. Ukoliko ne, izvršiti izmene kako bi relacija zadovoljila drugu normalnu formu.

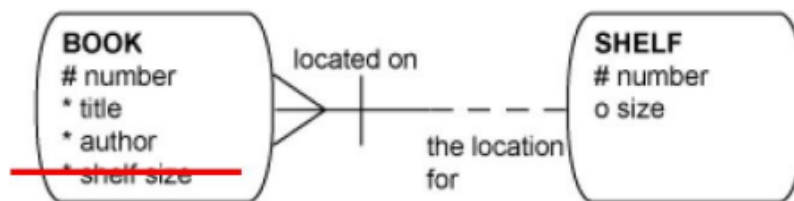
Napomena: UID za knjigu u biblioteci sadrži BOOK number i SHELF number.

Rešenja predstaviti preko ER dijagrama



Slika 10.1.3 Zadatak 2 [Izvor: NM IT350-2020/2021.]

**REŠENJE:** predstavljeno je na slici 11.4.



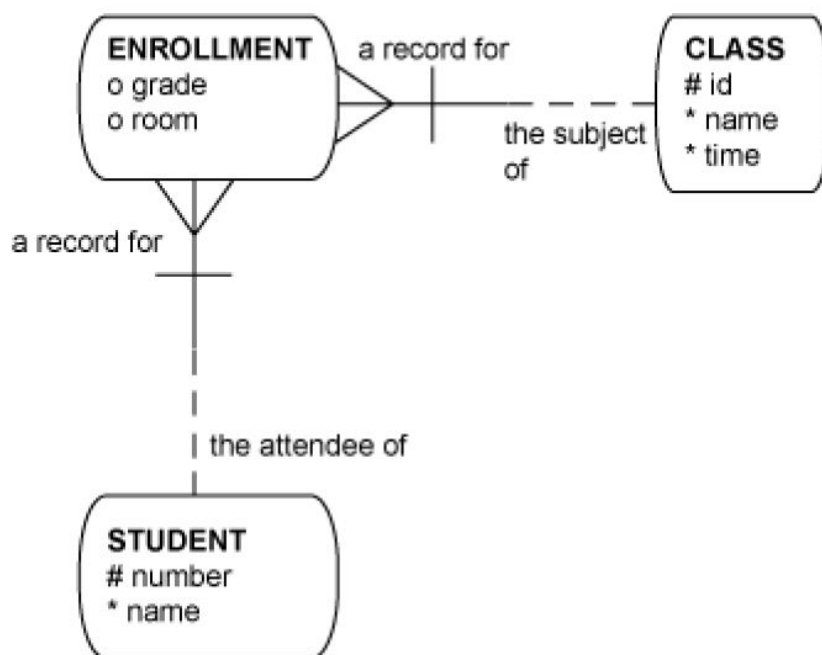
Slika 10.1.4 Rešenje zadatka 2 [Izvor: NM IT350-2020/2021.]

## SVOĐENJE RELACIJA NA 2NF - PRIMER 2 (3 MIN.)

### *Primer svođenja relacije na 2NF*

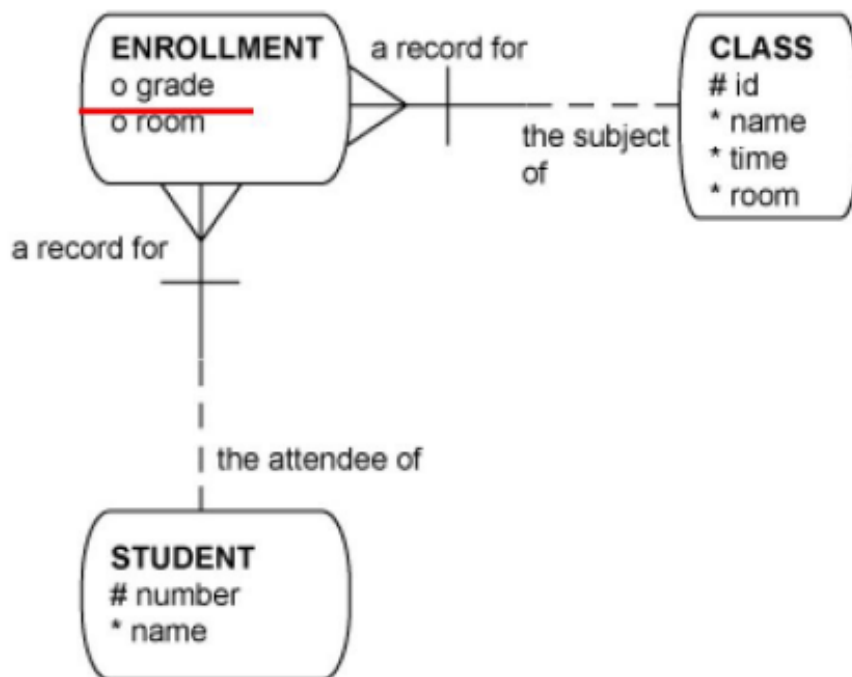
Proveriti da li relacije na slici 11.5 ispunjavaju uslove za 2NF. Ukoliko ne, izvršiti izmene kako bi relacija zadovoljila drugu normalnu formu.

Rešenja predstaviti preko ER dijagrama



Slika 10.1.5 zadatak 3 [Izvor: NM IT350-2020/2021.]

**Rešenje:** predstavljeno na slici 11.6. Atribut ROOM se odnosi na CLASS tip entiteta i zavisi od class id. Nema potrebe da u među entitetu ENROLLMENT, gde ćemo čuvati samo ocene studenata za pojedine predmete, čuvamo i class id i room.



Slika 10.1.6 Rešenje zadatka 3 [Izvor: NM IT350-2020/2021.]

## SVOĐENJE RELACIJE NA 3NF (3 MIN.)

### Primer svođenja relacije na 3NF

Proveriti da li relacija predstavljena na slici 11.7 ispunjava uslove 3NF. Ukoliko ne, izvršiti izmene kako bi relacija zadovoljila treću normalnu formu.

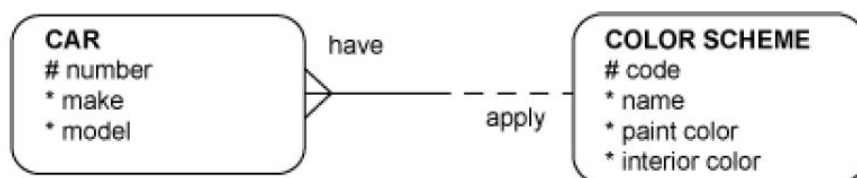
Rešenja predstaviti preko ER dijagrama.



Slika 10.1.7 Zadatak 4 [Izvor: NM IT350-2020/2021.]

**Rešenje:** predstavljeno je na slici 11.8. Paint color i interior color su atributi koji zavise od COLOR SCHEME.

Kreiraćemo novi tip entiteta COLOR SCHEME i dodati relaciju između novog tipa entiteta i tipa entiteta CAR.



Slika 10.1.8 Rešenje zadatka 4 [Izvor: NM IT350-2020/2021.]

## SVOĐENJE RELACIJE NA BCNF - PRIMER 1 (5 MIN.)

### Dovođenje tabele SKU\_PROIZVOD na BCNF

Razmotrimo tabelu:

SKU\_PROIZVOD (SKU, SKU\_Opis, Odeljenje, Kupac)

Ova tabela ima tri funkcionalne zavisnosti:

1. SKU  $\rightarrow$  (SKU\_Opis, Odeljenje, Kupac)
2. SKU\_Opis  $\rightarrow$  (SKU, Odeljenje, Kupac)
3. Kupac  $\rightarrow$  Odeljenje

SKU i SKU\_Opis kao determinante određuju sve kolone u tabeli pa su oni kandidati za ključ.

Kupac je determinanta, ali on ne određuje sve kolone pa stoga nije kandidat za ključ.

Relacija SKU\_PROIZVOD ima determinante koje nisu kandidati za ključ pa stoga nije u BCNF.

Da bi ovu anomaliju otklonili:

1. kolone funkcionalne zavisnosti čija determinanta nije kandidat ključ stavljamo u novu tabelu: KUPAC (Kupac, Odeljenje)
2. kao primarni ključ te nove tabele proglašavamo determinantu: KUPAC (Kupac, Odeljenje)
3. zatim kopiju determinante stavljamo u originalnu tabelu kao strani ključ SKU\_PROIZVOD\_2 (SKU, SKU\_Opis, Kupac)

Na taj način dobijamo dve rezultujuće tabele:

1. SKU\_PROIZVOD\_2 (SKU, SKU\_Opis, Kupac)
2. KUPAC (Kupac, Odeljenje)

Takodje moramo definisati referencijalni integritet:

SKU\_PROIZVOD\_2.Kupac mora postajati u KUPAC.kupac

## SVOĐENJE RELACIJE NA BCNF - PRIMER 2 (3 MIN.)

### *Svođenje tabele PRODAJA na BCNF*

U relaciji PRODAJA (Kupac\_ID, Kupac\_Ime, Prodavac, Region) koja je prikazana na slici 11.9 postoje sledeće funkcionalne zavisnosti:

1. Kupac\_ID  $\rightarrow$  Kupac\_ime, Prodavac, Region (Kupac\_ID je primarni ključ)
2. Prodavac  $\rightarrow$  Region (Svaki prodavac je dodeljen jednom regionu)

Region je funkcionalno zavistan od Prodavca a Prodavac je funkcionalno zavistan od Kupac\_ID. Kao rezultat toga u PRODAJI imamo problem sa održavanjem podataka:

1. Novi prodavac (Robinson) koji je dodeljen severnom regionu ne može biti unet dok se tom prodavcu ne dodeli kupac (jer mora da postoji vrednost za Kupac\_ID prilikom unošenja sloga u tabelu);
2. Ako se kupac broj 6837 izbriše iz tabele, gubimo informaciju da je prodavac Hernandez bio dodeljen istočnom regionu;

Kupac_ID	Kupac_Ime	Prodavac	Region
8023	Anderson	Smith	Južni
9167	Bancroft	Hicks	Zapadni
7924	Hobbs	Smith	Južni
6837	Tucker	Hernandez	Istočni
8596	Eckersley	Hicks	Zapadni
7018	Arnold	Faulb	Severni

Slika 10.1.9 Relacija PRODAJA [Izvor: NM IT350-2020/2021.]

3. Ako prodavac Smith pređe u istočni region, mora se promeniti nekoliko redova tabele.

Ovi problemi se mogu sprečiti dekompozicijom relacije PRODAJE u dve relacije, PRODAVAC i PRODAJA1, na osnovu dve determinante, što je prikazano na slici 11.10. Prodavac je primarni ključ u relaciji PRODAVAC. Prodavac je takodje strani ključ u relaciji PRODAJA1.

Prodavca u tabeli PRODAJA1 je ograničena na postojeće vrednosti Prodavca u tabeli PRODAVAC

Kupac_ID	Kupac_Ime	Prodavac
8023	Anderson	Smith
9167	Bancroft	Hicks
7924	Hobbs	Smith
6837	Tucker	Hernandez
8596	Eckersley	Hicks
7018	Arnold	Faulb

Prodavac	Region
Smith	Južni
Hicks	Zapadni
Hernandez	Istočni
Faulb	Severni

Slika 10.1.10 Relacije PRODAVAC i PRODAJA1 dobijene dovođenjem relacije PRODAJA na BCNF [Izvor: NM IT350-2020/2021.]

## SVOĐENJE RELACIJE NA BCNF - PRIMER 3 (3 MIN.)

### *Svođenje tabele OPRAVKA\_UREĐAJA na BCNF*

Struktura relacije OPRAVKA\_UREĐAJA ( (StavkaBroj, Tip, CenaKoštanja, OpravkaBroj, OpravkaDatum, OpravkaCena) je prikazana na slici 11.11. Kod nje postoje dve funkcionalne zavisnosti:

1. StavkaBroj → ( Tip, CenaKoštanja)
2. OpravkaBroj → (StavkaBroj, Tip, CenaKoštanja, OpravkaDatum, OpravkaCena)



StavkaBroj	Tip	CenaKoštanja	OpravkaBroj	OpravkaDatum	OpravkaCena
100	Presa	3500	2000	05.05.2007	375
200	Bušilica	4750	2100	05.07.2007	255
100	Presa	3500	2200	01.02.2007	178
300	Strug	27300	2300	10.02.2007	1785
100	Presa	3500	2400	12.03.2007	0
100	Presa	5500	2500	20.05.2007	275

Slika 10.1.11 Relacija OPRAVKA\_UREĐAJA [Izvor: NM IT350-2020/2021.]

StavkaBroj i OpravkaBroj su determinante ali je samo StavkaBroj kandidat za ključ. Saglasno tome, OPRAVKA\_UREĐAJA nije u BCNF i može se pojaviti anomalija ažuriranja.

Funkcionalno zavisne kolone treba staviti u posebnu tabelu na sledeći način:

1. STAVKA (StavkaBroj, Tip, CenaKoštanja)
2. OPRAVKA (StavkaBroj, OpravkaBroj, OpravkaDatum, OpravkaCena)

OPRAVKA.StavkaBroj mora postojati u STAVKA. StavkaBroj

## SVOĐENJE RELACIJE NA BCNF - PRIMER 4 (3 MIN.)

### *Svođenje tabele AKTIVNOSTI\_STUDENATA na BCNF*

Proanalizirajmo tabelu AKTIVNOSTI\_STUDENATA (SID, Ime, Klub, Cena, PlaćenaTaksa) prikazanu na slici 11.12 u kojoj se čuvaju podaci o aktivnostima studenata, gde je SID-identifikator studenta, Ime-ime studenta, Klub-ime kluba, Cena-članarina kluba i PlaćenaTaksa-iznos koji je student platio za ulazak u klub. Na osnovu sadržaja ove tabele mogu se otkriti sledeće funkcionalne zavisnosti

1. SID → Ime
2. SID → Klub.

Ove dve funkcionalne zavisnosti treba detaljno proanalizirati:

SID	Ime	Klub	Cena	PlaćenaTaksa
100	Jones	Scuba	400	0
200	Chau	Scuba	400	400
200	Chau	Skilling	550	550
300	Garett	Climbing	150	150
400	Jones	Skilling	550	550

Slika 10.1.12 Tabela AKTIVNOSTI\_STUDENATA [Izvor: NM IT350-2020/2021.]

1. *Funkcionalna zavisnost SID → Ime*: sigurno postoji ali postavlja se pitanje da li važi i obrnuto tj. da li ime determiniše SID? Da li je na primer vrednost „Jones” uvek uparena sa istom vrednošću SID? Ne, postoje dva studenta sa imenom „Jones” i oni imaju različite vrednosti za SID. Ime ne determiniše ni jednu drugu kolonu u tabeli.
2. *Funkcionalna zavisnost SID → Klub*: postoji ako student pripada samo jednom klubu, ali ne postoji ako student pripada u više od jednog kluba. Na osnovu sadržaja tabele se vidi da student 200 pripada u dva različita kluba, tako da SID ne determiniše Klub. SID ne determiniše ni Cenu i PlaćenaTaksa.

## SVOĐENJE RELACIJE NA BCNF - PRIMER 4, NASTAVAK I (3 MIN.)

*Imamo tri determinante: SID, Klub i (SID, Klub); Relacija AKTIVNOSTI\_STUDENATA nije u BCNF jer su kolone SID i Klub determinante ali ni jedna od njih nije kandidat ključ.*

Razmotrimo sledeću kolonu Klub. Mi znamo da jednom klubu može pripadati više studenata. Zbog toga, Klub ne determiniše SID ili Ime. Da li Klub determiniše Cenu. Da li je vrednost „Scuba” na primer uvek uparena sa istom vrednošću za Cenu. Iz sadržaja tabele to je očigledno i koristeći upravo ovaj primer podataka može se zaključiti da Klub determiniše Cenu: Klub → Cena

Tako postoje dve funkcionalne zavisnosti:

1.

SID → Ime

2.

Klub → Cena

Da li postoje neke funkcionalne zavisnosti sa složenim determinantama? Vrednost kolone PlaćenaTaksa ne određuje samo jedna kolona pa za nju treba razmotriti mogući složenu determinantu. PlaćenaTaksa zavisi i od Studenta i od Kluba kojem se student pridružio. Zbog toga, on je determinisan kombinacijom determinanti Student i Klub. Tako se može reći: (SID, Klub) → PlaćenaTaksa

Sada imamo tri determinante: SID, Klub i (SID, Klub).

Da li je neka od njih kandidat ključ. Da li neka od ovih determinanti jedinstveno identifikuje red tabele.

Na osnovu podataka, čini se da (SID, Klub) jedinstveno identifikuje red i da je ona kandidat ključ. U realnoj situaciji, učinjena pretpostavka se mora proveriti sa korisnikom.

Relacija AKTIVNOSTI\_STUDENATA sada nije u BCNF jer su kolone SID i Klub determinante ali ni jedna od njih nije kandidat ključ. SID i Klub su samo delovi kandidat ključa (SID, Klub).

Da bi se ova tabela normalizovala, potrebno je konstruisati tabele tako da je svaka determinanta kandidat ključ. To je moguće uraditi kreiranjem posebnih tabela za svaku funkcionalnu zavisnost.

## SVOĐENJE RELACIJE NA BCNF - PRIMER 4, NASTAVAK II (3 MIN.)

*Kreiranjem posebnih tabela za svaku funkcionalnu zavisnost se početna relacija AKTIVNOSTI\_STUDENATA dovodi na BCNF*

Kreiranjem posebnih tabela za svaku funkcionalnu zavisnost se dobija sledeće:

1. STUDENT (SID, Ime)
2. KLUB (Klub, Cena)
3. PLAĆANJE (SID, Klub, PlacenaTaksa)

Sa ograničenjem referencijalnog integriteta

1. PLAĆANJE.SID mora postajati u STUDENT.SID i
2. PLAĆANJE..Klub mora postajati u KLUB.Klub

Ove tabele su u BCNF i nemaju anomalije prouzrokovane funkcionalnom zavisnošću. Primeri podataka za normalizovane tabele su prikazani na slici 11.13 .

Relacija STUDENT	
SID	Ime
100	Jones
200	CHau
300	Garrett
400	Jones

Relacija KLUB	
Klub	Cena
Climbing	150
Scuba	400
Skilling	550

Relacija PLAĆANJE		
SID	Klub	PlaceniZnos
100	Scuba	0
100	Skilling	550
200	Scuba	400
300	Climbing	150
400	Skilling	550

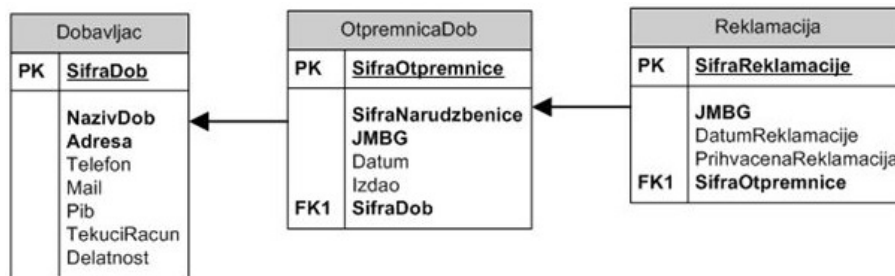
Slika 10.1.13 Normalizovana tabela AKTIVNOSTI\_STUDENATA [Izvor: NM IT350-2020/2021.]

## DENORMALIZACIJA -PRIMER 1 (4 MIN.)

### *Short-Circuit Keys tehnika denormalizacije*

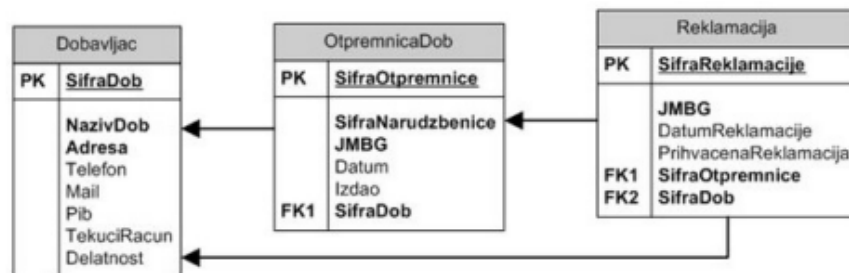
Na slici 11.14 je dat fizički model normalizovanih tabela DOBAVLJAC, OTPREMNICADOB i REKLAMACIJA. Postupak denormalizacije tabele REKLAMACIJA tako što se tabela po vezuje sa tabelom Dobavljac i dodaje joj se atribut SifraDob koji u njoj postaje strani ključ naziva se Short-Circuit Keys tehnika denormalizacije.

Tako podatke o reklamacijama i dobavljačima možemo dobiti spajanjem samo te dve tabele i izbegavamo spajanje sa tabelom OTPREMNICADO.



Slika 10.1.14 Normalizovane tabele [Izvor: NM IT350-2020/2021.]

Fizički model denormalizovanih tabela je prikazan na slici 11.15.



Slika 10.1.15 Denormalizovane tabele [Izvor: NM IT350-2020/2021.]

## DENORMALIZACIJA - PRIMER 2. (3 MIN.)

### *Storing Derivable Values tehnika denormalizacije*

Na slici 11.16 je prikazan fizički model podataka koji se sastoji od dve normalizovane tabele: PRIJEMNICA i STAVKAPRIJEMNICE. Denormalizacija tabele PRIJEMNICA tako što se tabeli doda atribut Ukupno, k oji sadrži sumu svih proizvoda  $\text{PrimljenaKolicina} * \text{Cena}$  iz stavki prijemnice se naziva Storing Derivable Values tehnika denormalizacije.

Kada se menjaju stavke prijemnice, potrebno je svaki put ponovo izračunavati vrednost polja Ukupno, a takođe i zabraniti da se denormalizovana kolona (Ukupno) menja direktno. Dodavanjem ovog redundantnog polja se izbegavaju kalkulacije tokom izveštaja i upita.

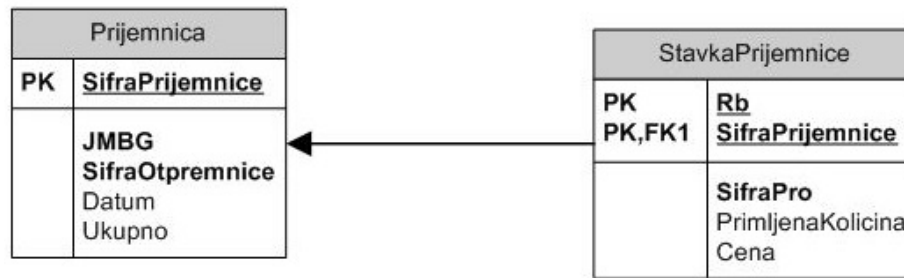


Slika 10.1.16 Normalizovane tabele [Izvor: NM IT350-2020/2021.]

U relacionom modelu se menja relacija Prijemnica:

Prijemnica (SifraPrijemnice#, Datum, Ukupno, SifraOtpremnice#, JMBG#)

Izgled fizičkog modela podšeme baze podataka posle denormalizacije prikazan je na slici 11.17.



Slika 10.1.17 Denormalizovane tabele [Izvor: NM IT350-2020/2021.]

## DENORMALIZACIJA - PRIMER 3. (3 MIN.)

### Denormalizacija tabele Proizvod

Na slici 11.18 je predstavljen fizički model baze podataka koju treba denormalizovati a koji se sastoji od relacija PROIZVOD, PRIJEMNICA i STAVKAPRIJEMNICE. Denormalizacija relacije PROIZVOD je izvršena tako što je tabeli do data kolona KolicinaUSkladistu, koja je suma PrimljenihKolicina tog proizvoda na stavkama prijemnica.



Slika 10.1.18 Baza pre denormalizacije [Izvor: NM IT350-2020/2021.]

Svaki put kada se unese, ažurira ili izbaci StavkaPrijemnice sa određenom SifromPro i PrimljenomKolicinom taj iznos se dodaje ili oduzima konkretnom proizvodu i čuva kao vrednost atributa KolicinaUSkladistu ( $KolicinaUSkladistu = KolicinaUSkladistu + / - PrimljenaKolicina$ ). Na slici 11.19 je predstavljen fizički model denormalizovane baze podataka sa izmenjenom relacijom PROIZVOD:

PROIZVOD (SifraProizvoda#, Naziv, JedMere, Opis, KolicinaUSkladistu)



Slika 10.1.19 Baza posle denormalizacije [Izvor: NM IT350-2020/2021.]

## DENORMALIZACIJA - PRIMER 4. (3 MIN.)

### Denormalizacija vertikalnom podelom tabele na dve i smeštanjem primarnog ključa u obe tabele

Denormalizaciju tabele DOBAVLJAC (slika 11.20) ćemo uraditi vertikalnom podelom tabele (odvajaju se samo neke kolone u novu tabelu). Primarni ključ se smešta u obe tabele. Tabela sa podacima o dobavljaču sadrži pored najčešće korišćenog podatka (naziv dobavljača) i mnogo drugih podataka kao što su adresa, telefon, mail, pib, tekući račun itd. Predlaže se formiranje dve tabele:

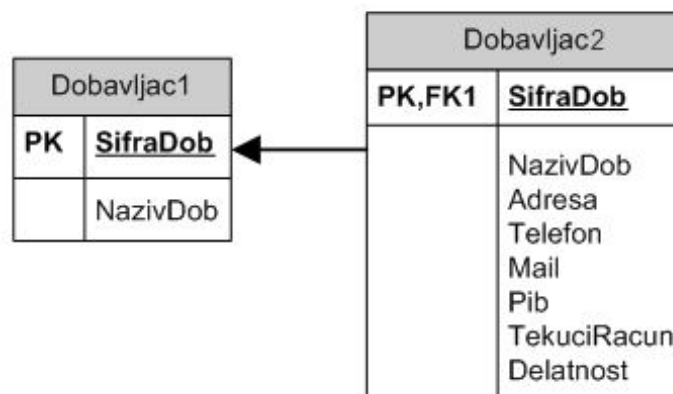
1. DOBAVLJAC1(SifraDob#, NazivDob)
2. DOBAVLJAC2(SifraDob#, NazivDob, Adresa, Telefon, Mail, Pib, TekuciRacun, Delatnost)

Tabele DOBAVLJAC1 i DOBAVLJAC2 su u vezi 1:1. Tabela DOBAVLJAC1 se vezuje za sve ostale tabele modela gde je to potrebno.

Dobavljac	
PK	<u>SifraDob</u>
	NazivDob Adresa Telefon Mail Pib TekuciRacun Delatnost

Slika 10.1.20 Baza posle denormalizacije [Izvor: NM IT350-2020/2021.]

Nakon denormalizacije se dobija fizički model predstavljen na slici 11.21.



Slika 10.1.21 Baza pre denormalizacije [Izvor: NM IT350-2020/2021.]

## ✓ 10.2 Pokazna vežba - samostalni rad

ZADATAK ZA SAMOSTALNI RAD 1 I 2 - (20 MIN.)

*Dovođenje relacija PREDMET i STUDENT na normalizovan oblik*

Relaciju PREDMET sa slike 11.1. dovesti na BCNF formu.

### Nenormalizovana

Predmet	Odjel	Nastavnik	Lekcije
P1	O1	N1	L1, L2
P2	O1	N1	L1, L3
P3	O1	N2	L4
P4	O2	N3	L1, L5
P5	O2	N4	L6

Slika 10.2.1 Relacija PREDMET [Izvor: NM IT350-2020/2021.]

### Potrebno vreme za rešavanje zadatka 10 minuta

Relaciju STUDENT sa slike 11.2. prevesti na BCNF formu.

STUDENTI						
brindeksa	SifraPredmeta	Ime	Prezime	Semestar	Ocena	Predmet
123	MA101	Marija	Dragić	1	6	Matematika 1
123	IT350				8	Baze podataka
452	CS101	Nikola	Marić	2	8	Programiranje 1
215	ENG101	Ana	Nikolić	1	9	Engleski 1
215	MA101				10	Matematika 1
215	IT350				9	Baze podataka

Slika 10.2.2 Relacija STUDENT [Izvor: NM IT350-2020/2021.]

### Potrebno vreme za rešavanje zadatka 10 minuta

## ZADATAK ZA SAMOSTALNI RAD 3 I 4 - (25 MIN.)

### *Dovođenje relacija IZDAVACI i ARANZMANI na normalnu formu*

Relacija na slici 11.3 pokazuje da autor za svaku prodatu knjigu od svojih izdavača dobija određeni honorar. Iznos honorara zavisi od autora, knjige i izdanja knjige. Primarni ključ relacije čine ISBN i AutorKnjigID.

Normalizujte ovu relaciju tako da dobijete model baze podataka u 3NF i predstavite je ER dijagramom. Na dijagramu označite identifikatore tipova entiteta.

ISBN	NazivKnjige	AutorKnjigID	AutorKnjigIme	Izdavač	Honorar	Izdanje
1	Koreni	1	Dobrica Ćosić	Prosveta	15.000	1
1	Koreni	1	Dobrica Ćosić	Nolit	12.000	2
2	Pesme za decu	2	Dobrica Erić	Prosveta	10.000	1
2	Pesme za decu	2	Dobrica Erić	Službeni glasnik	11.000	2
2	Pesme za decu	2	Dobrica Erić	Nolit	9.000	3
3	Na Drini ćuprija	3	Ivo Andrić	Prosveta	10.000	1
3	Na Drini ćuprija	3	Ivo Andrić	Službeni glasnik	8.000	2

Slika 10.2.3 Ne-normalizovan oblik relacije o knjigama i njihovim autorima

### Potrebno vreme za rešavanje zadatka 10 minuta [Izvor: NM IT350-2020/2021.]



Relacija na slici 11.4 pokazuje agente, njihove matične agencije i aranžmane iz ponude koje su oni prodali.

Normalizujte ovu relaciju tako da dobijete model baze podataka u BCNF i predstavite je ER dijagramom. Na dijagramu označite identifikatore tipova entiteta.

Tura	AgentID	Ag. Ime	Agencija	PonudaID	Noćenja	Cena	Br. Rezer.	Polazak	Kod aero.	Naziv aerodroma	Grad
1	76	Petar Kostić	NiTravel	B563	10	363	10	26.06.2018. 9:00	1	Konstantin Veliki	Niš
				B248	11	248	20	24.06.2018. 10:00	12	Aleksandar Veliki	Skoplje
				B428	10	322	18	25.06.2018. 9:00	11	Sofia Airport	Sofija
2	142	Mika Lazić	MetTravel	B563	10	363	15	07.07.2018. 11:00	1	Konstantin Veliki	Niš
				C930	12	568	2	06.07.2018. 9:00	14	Nikola Tesla	Beograd
				A270	15	972	1	09.07.2018. 12:00	14	Nikola Tesla	Beograd
				B728	9	248	5	08.07.2018. 9:00	12	Aleksandar Veliki	Skoplje
3	76	Petar Kostić	NiTravel	C930	12	568	11	15.07.2018. 9:00	1	Konstantin Veliki	Niš
				A430	9	279	15	16.07.2018. 10:00	11	Sofia Airport	Sofija

Slika 10.2.4 Prodaja aranžmana agenata

**Potrebno vreme za rešavanje zadatka 15 minuta** [Izvor: NM IT350-2020/2021.]



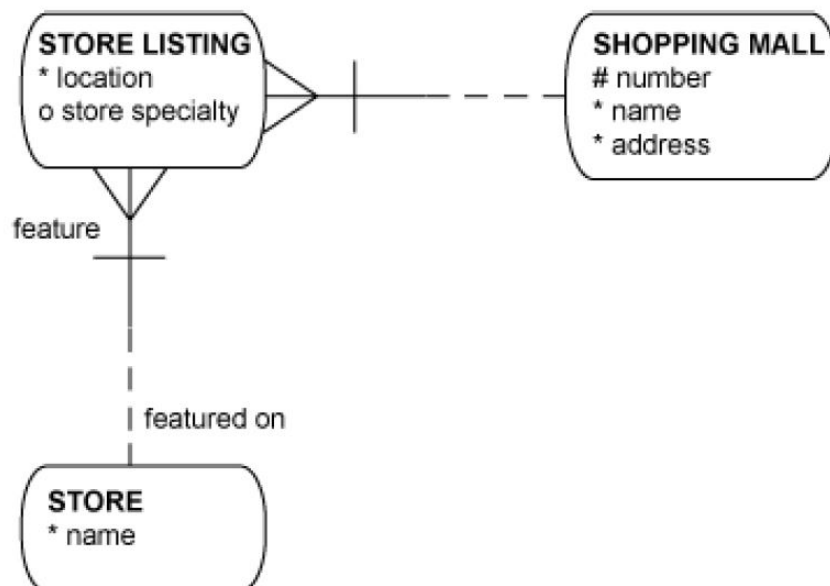
## ✓ Poglavlje 11

### Domaći zadatak

#### DOMAĆI ZADATAK 7 - ZADACI 1 I 2

*Tekstovi zadataka 1 i 2 - potrebno vreme izrade svakog zadatka 45 min.*

ZADATAK 1. Proveriti da li sledeća relacija ispunjava uslove 2NF. Ukoliko ne, izvršiti izmene kako bi relacija zadovoljila drugu normalnu formu i nacrtajte odgovarajući ER model.



Slika 11.1.1 Zadatak 1 za domaći [Izvor: NM IT350-2020/2021.]

ZADATAK 2. Ako imamo scenario da:

1. Jedan atleta zapošljava jednog agenta
2. Jedan agent može raditi za jednog ili više atleta
3. Atleta može igrati za jedan tim
4. Tim može imati jednog ili više igrača

Proveriti da li je sledeći primer u 3NF i ako nije izvršiti korekcije, kako bi 3NF bila zadovoljena. Za relaciju svedenu na 3NF nacrtajte odgovarajući E/R model.



Slika 11.1.2 Zadatak 2 za domaći [Izvor: NM IT350-2020/2021.]

## DOMAĆI ZADATAK - ZADACI 3 I 4

*Tekstovi zadataka 3 i 4 - potrebno vreme izrade svakog zadatka 45 min.*

ZADATAK 3. Relaciju IZNAJMLJIVANJE\_STANA sa slike 11.3 prevesti u BCNF. Za relaciju svedenu na BCNF nacrtajte odgovarajući ER model.

Klijent_ID	Klijent_Ime	Stan_ID	Stan_Adresa	Stan_Cena	Stan_Vlasnik_ID	Stan_Vlasnik_Ime
1	Milan	1	Prvomajska 4b, Niš	500	10	Jovan
		2	Milentijeva 3, Beograd	300	20	Jasna
4	Dušan	3	Skerlićeva 5, Beograd	300	30	Srdan
		4	Nušićeve 5, Novi Sad	400	40	Novica

Slika 11.1.3 Relacija IZNAJMLJIVANJE\_STANA [Izvor: NM IT350-2020/2021.]

ZADATAK 4: Relaciju IZNAJMLJENI\_FILMOVI prevesti u BCNF. Za relaciju svedenu na BCNF nacrtajte odgovarajući ER model.

Ime	Adresa	Rentirani filmovi	Zvanje
Jovan	Prvomajska 4, Niš	Pirati sa Kariba, Titanik	Mr
Ana	Jovana Jovanovića 10, Beograd	Tatina mala kći, Neretva	Ms
Milan	Njegoševa 3, Novi Sad	Titanik, Svi znaju	Mr

Slika 11.1.4 Relaciju IZNAJMLJENI\_FILMOVI [Izvor: NM IT350-2020/2021.]

## DOMAĆI ZADATAK - ZADATAK 5

*Tekst zadataka 5 - potrebno vreme izrade svakog zadatka 45 min.*

ZADATAK 5. Relaciju ZAPOSLENI sa slike 11.5 prevesti u BCNF. Za relaciju svedenu na BCNF nacrtajte odgovarajući ER model.

emp_id	emp_name	emp_zip	emp_state	emp_city	emp_district
1001	John	282005	UP	Agra	Dayal Bagh
1002	Ajeet	222008	TN	Chennai	M-City
1006	Lora	282007	TN	Chennai	Urrapakkam
1101	Lilly	292008	UK	Pauri	Bhagwan
1201	Steve	222999	MP	Gwalior	Ratan

Slika 11.1.5 Relacija ZAPOSLENI [Izvor: NM IT350-2020/2021.]

## UPUTSTVO ZA SLANJE DOMAĆEG ZADATKA

*Domaći zadatak treba izabrati na sledeći način i poslati prema priloženom uputstvu*

Raspodelu zadataka vrši asistent tako što:

1. zadatak br. 1 i 2 dodeljuju se studentima čiji se broj indeksa završava na 0 i 5.
2. zadatak br. 2 i 3 dodeljuju se studentima čiji se broj indeksa završava na 1 i 6.
3. zadatak br. 3 i 4 dodeljuju se studentima čiji se broj indeksa završava na 2 i 7.
4. zadatak br. 4 i 5 dodeljuju se studentima čiji se broj indeksa završava na 3 i 8.
5. zadatak br. 5 i 2 dodeljuju se studentima čiji se broj indeksa završava na 4 i 9.

Asistent ili profesor su u obavezi da studentima daju sve potrebne dodatne informacije vezano zabaze podatakakoji analiziraju pri rešavanju domaćih zadataka.

Prilikom slanja domaćih zadatka, neophodno je da ispunite sledeće:

- Subject mail-a mora biti IT250-DZbr (u slučaju kada šaljete domaći za ovu nedelju to je IT250-DZ07)
- U prilogu mail-a treba da se nalazi arhiviran projekat koji se ocenjuje imenovan na sledeći način: IT250-DZbr-BrojIndeksa-Ime Prezime. Na primer, IT250-DZ07-1234-VeljkoGrkovic
- Telo mail-a treba da ima pozdravnu poruku
- Arhivu sa zadatkom poslati na adresu predmetnog asistenta:  
milica.vlajkovic@metropolitan.ac.rs (studenti u Beogradu i online studenti) ili  
tamara.vukadinovic@metropolitan.ac.rs (studenti u Nišu).

Svi poslati mail-ovi koji ne ispunjavaju navedene uslove NEĆE biti pregledavani. Za sva pitanja ili nedoumice u vezi zadatka, možete se obratiti asistentu

## ✓ 11.1 Zadatak za samostalni rad

### ZADATAK ZA SAMOSTALNI RAD STUDENATA - (15 MIN.)

#### *Dovođenje relacije o iznajmljenim filmovima na normalni oblik*

Normalizujte relaciju sa slike 12.1 tako da dobijete model baze podataka u 3NF i predstavite je E/R dijagramom. Na dijagramu označite identifikatore tipova entiteta.

Ime i prezime	Fizička adresa	Grad	Rentirani filmovi	Titula
Ana Jović	Prvomajska 4b	Niš	Pirati sa Kraiba	Ms.
Ana Jović	Prvomajska 4b	Niš	Titanik	Ms.
Milan Canić	Dobrice Erić 5	Beograd	Titanik	Mr.
Milan Canić	Dobrice Erić 5	Beograd	Pirati sa Kariba	Mr.
Milan Canić	Dobrice Erić 5	Beograd	Maratonci trče počasni krug	Mr.
Jovana Milić	Ive Andrića 10	Kragujevac	Maratonci trče počasni krug	Ms.
Jovana Milić	Ive Andrića 10	Kragujevac	Poslednji tango u Parizu	Ms.

Slika 11.2.1 Ne-normalizovan oblik relacije o iznajmljenim filmovima [Izvor: NM IT350-2020/2021.]

**Potrebno vreme za rešavanje zadatka 15 minuta**

## ▼ Zaključak

### ZAKLJUČAK

#### *Šta smo naučili u ovoj lekciji?*

Normalizacija je tehnika koja se koristi za dizajniranje tabela u kojima su minimizirane redundancije podataka. Prve tri normalne forme (1NF, 2NF i 3NF) su najčešće korišćene. Sa strukturnog gledišta, više normalnih formi su bolje od nižih, jer više normalnih formi rezultira relativno manje redundancije podataka u bazi podataka. Skoro svi poslovni dizajni koriste 3NF kao idealnu normalnu formu. Takođe se koristi posebna, strožija 3NF poznata kao Boyce-Codd normalna forma, ili BCNF.

Normalizacija je važan deo, ali samo deo procesa dizajna. Kako se entiteti i atributi definišu tokom procesa modeliranja ER dijagrama, svaki entitet (skup) podvrgava se proveru normalizacije i formira se novi entitet (skup) ako je potrebno. Uključite normalizovane entitete u ERD i nastavite iterativni proces modeliranja ER dijagrama dok se ne definišu svi entiteti i njihovi atributi i dok svi ekvivalentni skupovi podataka ne budu u 3NF-u.

Što je veći broj tabela, to je potrebno više dodatnih operacija ulaza/izlaza i logike obrade za njihovo spajanje. Zbog toga se ponekad tabele denormalizuju kako bi se smanjio broj operacija ulaza/izlaza i povećala brzina obrade. Nažalost, sa većim tabelama, plaćate veću brzinu obrade po cenu manje efikasnih izmena podataka, komplikovanijeg indeksiranja i uvođenja redundancija podataka koje mogu dovesti do problema sa podacima.

Denormalizacija je postupak suprotan normalizaciji (znači dodavanje redundantnih podataka tabelama i vrši se pod specijalnim okolnostima, uglavnom nad bazama podataka koje se samo čitaju (služe za izveštavanje).

### LITERATURA:

#### *Za pisanje ove lekcije korišćena ja sledeća literatura:*

1. [http://corpgov.crew.ee/Materjalid/Database%20Systems%20-%20Design,%20Implementation,%20and%20Management%20\(9th%20](http://corpgov.crew.ee/Materjalid/Database%20Systems%20-%20Design,%20Implementation,%20and%20Management%20(9th%20edition).pdf)
2. Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom, DATABASE SYSTEMS –The Department of Computer Science, Stanford University, 2009 by Pearson Education Inc.
3. C. J. Date, An introduction to Database Systems, Addison-Wesley Publishing Company, 1990
4. <https://www.guru99.com/database-normalization.html>
5. <https://www.studytonight.com/dbms/database-normalization.php>