

# *Projektovanje relacionih baza podataka*

## **NORMALIZACIJA**

# *Dizajn baze podataka*

- Dizajn baze podataka je proces kreiranja detaljnog modela podataka baze podataka.
- Ovaj logički model sadrži sve što je potrebno za izbor fizičkog i logičkog dizajna i parametara fizičkog skladišta, koji su potrebni da bi se dobio skript u **DDL**-u (Data Definition Language), koji se dalje koristi za kreiranje baze podataka.
- Model podataka sadrži i detaljne atributе za svaki entitet.

# *Model podataka*

- Model podataka je intelektualno sredstvo za opis statičkih karakteristika sistema, opis karakteristika sistema u nekom stacionarnom stanju.
- Stacionarno stanje nekog sistema karakteriše se skupom zavisnosti koje postoji između objekata sistema.
- Ove zavisnosti se, u modelu podataka, mogu predstaviti bilo strukturom podataka, bilo skupom ograničenja na vrednosti podataka.

# *Normalizacija baze podataka*

- Normalizacija je postupak projektovanja logičke strukture baze podataka
- Uobičajeno je da se koristi za projektovanje logičke strukture relacionog modela.
- Postupak normalizacije ima opštiji značaj i treba ga primenjivati i na druge modele

# *Normalizacija baze podataka*

- To je proces efikasnog organizovanja podataka u bazi podataka
- Dva osnovna cilja procesa normalizacije su:
  - Eliminisanje redundantnih podataka (skladištenje istih podataka na više mesta u bazi podataka)
  - Osiguravanje da su zavisnosti podataka logične (u jednoj tabeli se nalaze samo povezani podaci)
- Cilj je da se smanji veličina prostora koju baza zauzima na disku i da su podaci logično uskladišteni

# *Normalizacija baze podataka*

- Primena normalnih formi na analizu relacija
- Primena normalnih formi na sintezu relacija
- Transformacija drugih modela u relacioni model

# *Normalizacija baze podataka*

- Normalizacija je proces primene skupa pravila na postojeći dizajn baze, uglavnom da bi se postigla **minimalna redudansa podataka**.
- Većina udžbenika predstavljaju kao proces od tri koraka, koji se vezuje za normalne forme, koje se mogu obaviti u skoro algoritamskom poretku.

# *Normalizacija baze podataka*

- Jedan od ključnih ciljeva relacione baze podataka je da se spriči nepotrebno dupliranje podataka.
- U stvari, ovo je jedan od glavnih razloga zašto se koriste relacione baze podataka, a ne fajlovi, koji podatke skladište u jednoj tabeli.
- Ponekad se klasa ili objekat dizajnira korektno (zavisi koji model podataka se koristi), a u relacionoj šemi se tek uoči problem.

# *Normalizacija baze podataka*

- U teoriji, trebalo bi krenuti od jedinstvene relacione šeme (nekad se naziva i univerzalna šema, ili U) koja sadrži sve atributе baze podataka
- zatim se rekurzivno primenjuju pravila kako bi se dobio skup sve više i više normalizovanih šema tj. pod-relacija.
- Kada su sve šeme u trećoj normalnoj formi, onda je cela baza normalizovana.

# *Normalizacija baze podataka*

- U praksi, češće se pravila postepeno primenjuju, dok se šema svake relacije ne razvije na način na koji se dobija iz UML dijagrama klasa ili dijagrama E-R modela.
- Konačna struktura tabela treba da bude ista bez obzira koji se metod (ili kombinacija metoda) primenjuje.

# *Nenormalizovana relacija*

- Anomalije u ažuriranju (dodavanje, brisanje, izmena)
- Anomalije u izveštavanju

Bl	Ime	Sem	ŠSmer	ImeRuk	ŠPred	Nazpred	Ocena
21	Goran	5	01	Bata	121 323 056	MAT BP OET	7 8 8
77	Ana	7	01	Bata	056 121	OET MAT	10 8

# *Prva normalna forma (1NF)*

- *postavlja* najosnovnija pravila za organizovanu bazu podataka:
- Eliminisanje duplih kolona iz iste tabele
- Kreiranje posebne tabele za istu grupu povezanih podataka i identifikovanje svakog reda jedinstvenom kolonom (primarni ključ)

# *Prva normalna forma (1NF)*

- Prvo pravilo znači da ne možemo imati duple podatke u istom redu jedne tabele. Ovo se označava kao atomarnost tabele, tj. tabele koje zadovoljavaju ovo pravilo se nazivaju atomske.
- Primer: Jedan rukovodilac može imati više zaposlenih kojima rukovodi.

# *Prva normalna forma (1NF)*

- Intuitivno možemo kreirati tabelu sa sledećim poljima:
- Manager
- Subordinate1
- Subordinate2
- Subordinate3
- Subordinate4

# *Prva normalna forma (1NF)*

- Međutim, ako se setimo prvog pravila 1NF: eliminacija duplih kolona iz iste tabele, onda možemo zaključiti da su kolone Subordinate1-Subordinate4 duplikati.
- Ako menadžer ima samo jednog zaposlenog kojim rukovodi, onda su kolone Subordinate2-Subordinate4 prazne, i uzalud trošimo prostor
- A ako menadžer već ima 4 zaposlena kojima rukovodi, onda ukoliko se pojavi još jedan, mora da se menja struktura baze podataka

# *Prva normalna forma (1NF)*

- Rešenje za početnike u dizajnu baze podataka bi bilo da imamo jednu kolonu menadžer i jednu zaposleni, u kojoj bi skladištili imena zaposlenih.
- Međutim, ta nova kolona može sadržati duplike i nije atomska.
- Šta se dešava ako treba da dodamo ili uklonimo zaposlenog, morali bi celo polje da menjamo, ukoliko imamo nekoliko zaposlenih to nije veliki problem, ali šta se dešava ako menadžer ima nekoliko stotina zaposlenih.
- Ovo nam komplikuje i upite za prikaz podataka.

# *Prva normalna forma (1NF)*

- Evo tabele koja zadovoljava 1NF:
- Manager
- Subordinate
- U ovom slučaju svaki zaposleni se nalazi u jednom redu, ali menadžeri se ponavljaju.

# *Prva normalna forma (1NF)*

- Što se tiče drugog pravila: identifikovanje svakog reda jedinstvenom kolonom ili skupom kolona (primarni ključ)
- Možemo koristiti kolonu zaposleni kao primarni ključ.
- Međutim, šta se dešava ako zaposlimo novog zaposlenog koji ima isto ime kao zaposleni koji je već u tabeli. Najbolje rešenje je da koristimo stvarno jedinstveno polje, kao ID zaposlenog, za primarni ključ-

# *Prva normalna forma (1NF)*

- Konačna tabela ima sledeći izgled.
- Manager ID
- Subordinate ID

# *Prva normalna forma*

- Relacija R je u Prvoj normalnoj formi (1NF) ako su sve vrednosti njenih atributa atomske
- Relacioni model zahteva da su sve relacije u 1NF
- I dalje postoje anomalije ažuriranju

Bl	Ime	Sem	ŠSmer	ImeRuk	ŠPred	Nazpred	Ocena
21	Goran	5	01	Bata	121	MAT	7
21	Goran	5	01	Bata	323	BP	8
21	Goran	5	01	Bata	056	OET	8
77	Ana	7	01	Bata	056	OET	10
77	Ana	7	01	Bata	121	MAT	8

# *Baza podataka bez anomalija*

- Relacija Student predstavlja spoj više različitih objekata sistema.
- Ona istovremeno predstavlja objekte Student, Predmet, Smer, Prijava i njihove međusobne veze
- Dekompozicijom na relacije koje predstavljaju navedene objekte dobili bi sledeće relacije, koje su bez anomalija

# *Baza podataka bez anomalija*

Normalizovane relacije dobijene iz početne  
relacije Student

Student(BI,Ime,Sem,ŠSmer)

Smer(ŠSmer,ImeRuk)

Predmet(ŠPredmet,NazPRed)

Prijava(BI,ŠPred,Ocena)

# Funkcionalne zavisnosti

- Definicije druge, treće i Boyce-Codd-ove normalne forme se zasnivaju na konceptu **funkcionalne zavisnosti** atributa relacije

# *Funkcionalne zavisnosti atributa relacije*

- Data je relacija R sa atributima X i Y, moguće složenim. Atribut Y je funkcionalno zavistan od atributa X (ili funkcionalno određuje Y)
- $R.X \rightarrow R.Y$ ,
- ako i samo ako svakoj vrednosti X odgovara jedna i samo jedna vrednost Y.

# *Funkcionalne zavisnosti atributa relacije*

- BI, ŠPred->Ime, sem, ŠSmer, ImeRuk, NazPred,Ocena
- BI->Ime, Sem, ŠSmer, ImeRuk
- ŠSmer->ImeRuk
- ŠPred->NazPred

# *Definicije ključa i nadključa*

- Definicije ključa i nadključa relacije na osnovu definicije funkcionalne zavisnosti
- Atribut X moguće složeni je nadključ neke relacije R akko funkcionalno određuje sve ostale atribute relacije R
- Atribut X, moguće složeni, je ključ relacije R ako je nadključ, a nijedan njegov pravi podskup nema tu osobinu.
- U relaciji student složeni atribut BI, ŠPred je ključ

# Funkcionalna zavisnost

- Ako su X i Y atributi, onda se notacija  $X \rightarrow Y$  čita kao „X funkcionalno određuje Y“ ili „Y je funkcionalno zavisno od X“.
- Ovo znači da ako dobijemo tabelu sa podacima i vrednost X , da možemo da odredimo Y.
- Nadključ uvek funkcionalno određuje sve attribute u relaciji (kao i sebe).
- Ovo je „dobra“ FD.

# *Funkcionalna zavisnost*

- „Loša“ FD se dešava kada postoji atribut ili skup atributa koji predstavljaju nadključ za neke attribute, ali ne i za celu relaciju.
- Ovakvi atributi se nazivaju podključevi (subkey) relacije.

# *Podključevi i normalizacija*

- **Normalizacija** znači praćenje procedure ili skupa pravila kako bi se osigurali da je baza podataka dobro dizajnirana.
- Većina pravila normalizacije znači da se eliminišu redundantni podaci u bazi podataka.

# *Podključevi i normalizacija*

- Podključevi uvek znače redundantne podatke.
- Ako u tabelama ne postoje podključevi, onda se ona nalazi u trećoj normalnoj formi.
- Međutim, ova normalna forma dozvoljava podključeve u nekim izuzetnim slučajevim, a normalna forma koja striktno ne dozvoljava podključeve je BCNF (Boyce-Coddova normalna forma).

# *Funkcionalne zavisnosti atributa relacije*

- Atribut ocena **potpuno funkcionalno zavisi** od složenog atributa BI, ŠPred

BI,ŠPred->Ocena

BI-/->Ocena

ŠPred-/-> Ocena

- NazPred je **nepotpuno funkcionalno zavistan** od složenog atributa BI, ŠPred, jer je funkcionalno zavistan i od njegovog dela ŠPred

BI,ŠPred->NazPred

BI-/->NazPred

ŠPred-> NazPred

# *Funkcionalna zavisnost*

- Postoji jednostavan način od tri koraka koji rešava ovaj problem:
  1. Sve atribute koji zavise od podključa treba otkloniti i staviti ih u novu šemu.
  2. Duplicirani podključ u novoj tabeli postaje primarni ključ.
  3. Ostaviti kopiju ovog atributa u originalnoj šemi, gde postaje spoljni ključ.

# *Tranzitivna funkcionalna zavisnost*

- Data je relacija R sa atributima A,B i C, moguće složenim. Ako u relaciji R važi:

A->B

B->C

A->C

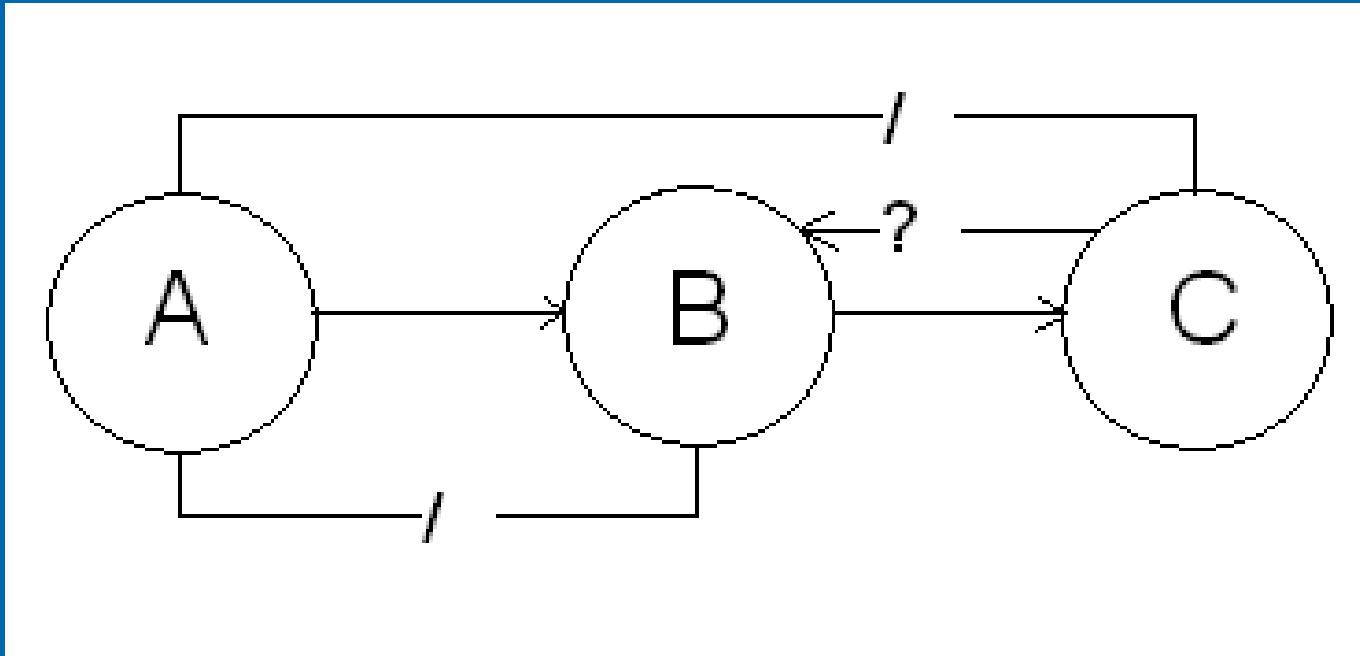
B-/->A

C-/->A

Atribut C je **tranzitivno funkcionalno zavistan** od atributa A

Jednostavnije rečeno, atribut C je tranzitivno funkcionalno zavistan od atributa A, ako je funkcionalno zavistan od A i ako je funkcionalno zavistan od nekog atributa B, koji je sam funkcionalno zavistan od A.

# *Tranzitivna funkcionalna zavisnost*



# **DRUGA NORMALNA FORMA**

- Relacija je u Drugoj normalnoj formi (2NF) ako je u 1NF i svi njeni neključni atributi funkcionalno zavise od primarnog ključa (neklučni atributi su atributi koji nisu kandidati za ključ, niti deo kandidata za ključ)

# Druga normalna forma

- Osnovni zahtevi 2NF su
- Uklanjanje podskupova podataka koji se nalaze u više redova i njihovo smeštanje u posebne tabele.
- Kreiranje veza između novih tabela i tabela sa kojima su spojene korišćenjem spoljnih ključeva

# Druga normalna forma

- Ova pravila se mogu sumirati u jednostavnu rečenicu:
- 2NF ima za cilj smanjenje količine redundantnih podataka, tako što se oni “izvlače” iz tabele, stavlju u nove tabele i kreiraju veze između ovih tabela.

# Druga normalna forma

- Primer: Tabela Kupci jedne online prodavnice sadrži sledeće podatke:
- CustNum
- FirstName
- LastName
- Address
- City
- State
- ZIP

# Druga normalna forma

- Može se desiti da za različite kupce skladištimos iste podatke o gradu – City, State, Zip.
- Ovo prouzrokuje anomalije u ažuriranju (dodavanje, izmena i brisanje).
- Rešenje je da se podaci o gradu prebace u novu tabelu, Grad, a da se u početnoj tabeli ostavi polje Zip, koje postaje spoljašnji ključ.

# Druga normalna forma

- Takođe, tabela Grad se može popuniti i unapred podacima dobijenim iz pošte
- Tabela Kupac dobija sledeću strukturu.
- CustName
- FirstName
- LastName
- Address
- ZIP
- Na ovaj nači minimizirana je količina redundantnih podataka.

# **TREĆA NORMALNA FORMA**

- Relacija R je u trećoj normalnoj formi akko je u 2NF i ako svi njeni neključni atributi netranzitivno zavise od primarnog ključa

# Treća normalna forma

- Osnovni zahtevi 3NF
- Tabela je u 1NF i 2NF
- Uklonjene su kolone koje nisu potpuno zavisne od primarnog ključa.

# Treća normalna forma

- Primer: Tabela sa narudžbinama koja sadrži sledeće attribute:
- Order Number
- Customer Number
- Unit Price
- Quantity
- Total

# Treća normalna forma

- Prvo treba proveriti da li je tabela u 1NF i 2NF. Da li postoji duplirane kolone, ne, da li postoji primarni ključ, da, Order Number. Dakle, tabela je u 1NF.
- Da li postoji podskupovi podataka koji se ponavljaju u više redova? Ne, dakle, tabela je u 2NF.

# Treća normalna forma

- Da li postoje kolone koje ne zavise u potpunosti od primarnog ključa?
- Customer Number se razlikuje u zavisnosti od OrderNumber, i ne zavisi od drugih polja.
- UnitPrice zavisi od CustomerNumber, u slučaju kada svakom kupcu naplaćujemo različitu cenu, što se i dešava, tako da cena zavisi od OrderNumber. Isto važi i za Quantity.

# Treća normalna forma

- Total se dobija kada se UnitPrice pomnoži sa Quantity, i on zavisi od ta dva polja a ne od OrderNumber. Zato se ova polja uklanjaju iz tabele i dobijamo novu tabelu koja je u 3NF.
- Order Number
- Customer Number
- Unit Price
- Quantity

# NORMALIZACIJA

- STUDENT(BI,ŠPRED,IME,SEM,ŠSMER,IMERUK ,NAZPRED,OCENA)
- ZBOG NEPOTPUNIH ZAVISNOSTI

BI,ŠPRED->NAZPRED  
ŠPRED->NAZPRED

BI,ŠPRED->IME,SEM,  
ŠSMER,IMERUK  
BI->IME,SEM,ŠSMER,IMERUK

- NIJE U 2NF I DEKOMPONUJE SE NA SLEDEĆE PROJEKCIJE

# **NORMALIZACIJA**

STUDENT1 (BI, IME, SEM, ŠSMER, IMERUK)  
PREDMET(ŠPRED, NAZPRED)  
PRIJAVA(BI, ŠPRED, OCENA)

U relaciji STUDENT1 postoji tranzitivna zavisnost

BI->ŠSMER  
ŠSMER->IMERUK  
BI->IMERUK

Pa ona nije u 3NF i dekomponuje se na sledeće projekcije

STUDENT2 (BI, IME, SEM, ŠSMER)  
SMER(ŠSMER, IMERUK)

# *PRIMENA NORMALNIH FORMI NA PROJEKTOVANJE BAZE PODATAKA ANALIZOM RELACIJA*

- Dat je skup nenormalizovanih relacija
- Primenom definicije normalnih formi relacije se normalizuju, primenjuju se redom definicije 2NF,3NF,...
- Relacije sa istim ključem se mogu spojiti ako se time ne ugrožava semantika sistema
- Dekompozicija bez gubitka informacija

- Za tabele prikazane na slici :
- a) odrediti kojoj normalnoj formi pripada,
- b) dati definiciju te normalne forme,
- c) navesti i objasniti anomalije ako postoje,
- d) opisati kako se anomalije uklanjaju,
- e) dati definiciju normalne forme u koju je prevedena.

# 1. zadatak

IDBR	IME	POSAO	PLATA	BROD	IMEOD	MESTO
5497	Aco	radnik	1000	10	Komercijala	Novi Beograd
5519	Vaso	prodavac	1200	10	Komercijala	Novi Beograd
5367	Petar	vozač	1300	20	Plan	Dorćol
5780	Bozo	upravnik	2200	20	Plan	Dorćol
5953	Pero	nabavljač	1100	30	Prodaja	Stari Grad
5842	Savo	direktor	3000	40	Direkcija	Banovo Brdo
5867	Simo	savetnik	2750	40	Direkcija	Banovo Brdo
6789	Janko	rukovodilac	3900	40	Direkcija	Banovo Brdo

## 2. zadatak

- ProjektniTim (ŠifraProjekta, JMBG, NazivProjekta, ImePrezime, PeriodAngažovanja, ŠifraSektora, NazivSektora)
- ŠifraProjekta, JMBG -> PeriodAngažovanja, ŠifraSektora, NazivSektora, NazivProjekta, ImePrezime
- ŠifraProjekta -> NazivProjekta
- JMBG -> ImePrezime, ŠifraSektora, NazivSektora
- ŠifraSektora -> NazivSektora

ŠifraProjekta	JMBG	NazivProjekta	ImePrezime	PeriodAngažovanja	ŠifraSektora	NazivSektora
123	1203971710001	IS	Zoran	2 godine	12	Projektovanje
123	0906975715001	IS	Maja	2 godine	13	Programiranje
131	1111961710001	ELearn	Petar	1 godina	12	Projektovanje

# 3. zadatak

ra_ucenika	prezime	ime	dat_rodjenja	sifra_odeljenja	razred	broj	generacija	smer
123	Antić	Dragan	23.11.1990	12	1	2	2005/2006	prirodno-matemat
1224	Sarić	Dobrivoje	2.8.1990	12	1	2	2005/2006	prirodno-matemat
1621	Stanković	Maja	21.9.1990	16	1	6	2005/2006	prirodno-matemat
4817	Milošević	Petar	12.12.1987	48	4	8	2005/2006	drustveno-jezicki
4824	Stanković	Marija	1.8.1987	48	4	8	2005/2006	drustveno-jezicki
273	Babić	Goran	5.10.1989	27	2	7	2005/2006	drustveno-jezicki