

Normalizacija relacij (tabel)

- Relacijski podatkovni model
- Relacija, atribut, relacijska shema
- Odvisnosti med atributi relacije:
 - Funkcionalne
 - Večvrednostne
 - Stične
- Ažurirne anomalije
- Normalne oblike relacij (1, 2, 3, BCNO, 4, 5) in postopki za normalizacijo
- Kako določiti ključ relacije na podlagi funkcionalnih odvisnosti?

Odvisnosti med atributi relacije

- Relacija se podreja integritetnim omejitvam iz realnega sveta, ki omogočajo le določene kombinacije vrednosti atributov.
- Integritetne omejitve v modelu določimo s pomočjo funkcionalnih in drugih odvisnosti.
- Odvisnosti so sredstvo, s katerim lahko v relacijskem modelu povemo, katere vrstice relacij (kombinacije vrednosti atributov) so oziroma bi lahko bile veljavne in katere sploh ne morejo obstajati.

Funkcionalne odvisnosti

- Funkcionalne odvisnosti veljajo na nivoju relacijske sheme, torej za vse relacije, ki pripadajo isti shemi.
- Imejmo relacijsko shemo R s podmnožicama atributov X in Y .
- V relacijski shemi R velja $X \rightarrow Y$ (X funkcionalno določa Y oziroma Y je funkcionalno odvisen od X), če v nobeni relaciji, ki pripada shemi R , ne obstajata dve n -terici, ki bi se ujemali v vrednostih atributov X in se ne bi ujemali v vrednostih atributov Y .
- Preprosto povedano, obstaja neka funkcija, s pomočjo katere lahko iz vrednosti X izračunamo vrednosti Y .

Upoštevanje odvisnosti

- Pri ažuriranju relacij (tabel) je treba odvisnosti upoštevati, sicer pride do ažurirnih anomalij.
- Več možnosti za upoštevanje:
 - Uporabnik se zaveda vseh odvisnosti in jih upošteva
 - SUPB preverja vse odvisnosti (časovno zahtevno)
 - Preoblikovanje relacij na način, da do ažurirnih anomalij sploh ne more priti (normalizacija)

Ažurirne anomalije

- Relacije, ki vsebujejo odvečne podatke lahko povzročajo ažurirne anomalije pri operacijah nad podatki.
- Poznamo več vrst anomalij:
 - Anomalije pri dodajanju n-teric v relacijo
 - Anomalije pri brisanju n-teric iz relacije
 - Anomalije pri spreminjanju n-teric

Anomalije pri dodajanju vrstic

- Dodajanje novih članov oddelka: ponovno moramo (pravilno) vpisati naslov oddelka
- Dodajanje novega oddelka: za podatke o članu vpišemo NULL

Ime	Priimek	Oddelek	Naslov
Janez	Novak	1A	Tržaška 25
Peter	Klepec	1A	Tržaška 25
Marija	Kovač	2A	Dunajska 6
Janko	Jankovič	1A	Tržaška 52
NULL	NULL	3A	Celovška 12

Anomalije pri brisanju vrstic

- Brisanje edinega člana oddelka: izgubimo tudi vse informacije o tem oddelku (šifra oddelka, naslov)

Ime	Priimek	Oddelek	Naslov
Janez	Novak	1A	Tržaška 25
Peter	Klepec	1A	Tržaška 25
Marija	Kovač	2A	Dunajska 6

Anomalije pri spreminjanju vrstic

- Oddelek 1A se preseli na Jadransko 21. Naslov je treba pravilno popraviti pri vseh članih oddelka!

Ime	Priimek	Oddelek	Naslov
Janez	Novak	1A	Tržaška 25
Peter	Klepec	1A	Tržaška 25
Marija	Kovač	2A	Dunajska 6

Primarni ključ

- Imejmo relacijsko shemo R s podmnožico atributov X .
- X je ključ relacijske sheme R , če velja
 1. $X \rightarrow R$
 2. Za noben atribut A iz X ne velja $(X-A) \rightarrow R$ (minimalnost)
- Shema ima lahko več ključev, izberemo enega najprimernejšega, ki mu pravimo primarni ključ. Ostalim pravimo alternativni ključi.
- Specifikacija primarnih in alternativnih ključev omogoča dosledno spoštovanje nekaterih omejitev!

Pomožni koncepti za iskanje ključev na podlagi funkcionalnih odvisnosti

- Imejmo shemo R v kateri velja množica funkcionalnih odvisnosti F
- Osnovni atribut: del nekega (ne nujno primarnega) ključa
- Kanonična oblika funkcionalne odvisnosti: na desni strani je največ en atribut
- Logična izpeljava odvisnosti: $F \Rightarrow X \rightarrow Y$
- Zaprtje množice **odvisnosti**: $F^+ = \{X \rightarrow Y: F \Rightarrow X \rightarrow Y\}$
(vse možne izpeljane odvisnosti)

Zaprte (closure) množice atributov

- Zaprtje množice **atributov** X glede na F
 $X^+ = \{A: X \rightarrow A \in F^+\}$
- Postopek za izračun X^+

Vhod: X, F

Izhod: X^+

$X^+ = X$

ponavljaj:

$\text{stari}X^+ = X^+$

 za vsako odvisnost $Y \rightarrow Z \in F$ naredi:

 če $Y \subseteq X^+$ potem:

$X^+ = X^+ \cup Z$

dokler ni $\text{stari}X^+ = X^+$

Primer izračuna zaprtja

$R=ABCDEFGG$

$F=\{A \rightarrow B, BE \rightarrow G, EF \rightarrow A, D \rightarrow AC\}$

Iščemo $\{EF\}^+$:

1. $\{EF\}^+ = EF$
2. $\{EF\}^+ = EFA$
3. $\{EF\}^+ = EFAB$
4. $\{EF\}^+ = EFABG$
5. $\{EF\}^+ = EFABG$

Minimalno pokritje množice odvisnosti (olajša iskanje ključev)

- F pokriva E: $\forall f \in E: F \Rightarrow f$ oziroma $E^+ \subseteq F^+$
- Minimalno pokritje F_{\min} : zahtevamo $F_{\min}^+ = F^+$, vendar ohranimo samo neredundantne odvisnosti
 - Kanonična oblika (en atribut na desni)
 - Minimalnost (ne moremo odstraniti nobene odvisnosti, da bi še vedno veljalo $F_{\min}^+ = F^+$)
 - $\forall F_{\min}$ ne moremo zamenjati nobene $X \rightarrow A$ z $Y \rightarrow A$, $Y \subset X$, da bi še vedno veljalo $F_{\min}^+ = F^+$)
- Postopek v praksi: narišemo graf odvisnosti, kjer odstranimimo tranzitivne povezave. Rezultat ni nujno enoličen!
- Primer: $R = ABC$, $F = \{A \rightarrow B, B \rightarrow A, B \rightarrow C, A \rightarrow C, C \rightarrow A\}$

Izpeljevanje funkcionalnih odvisnosti in določanje ključev

- Armstrongovi aksiomi in izpeljana pravila sklepanja (kogar zanima, v literaturi)
- Trivialne odvisnosti: $X \rightarrow Y$ kadar $Y \subseteq X$ (vedno veljajo, lahko jih izpustimo)
- Postopki za določanje kandidatov za ključe na osnovi funkcionalnih odvisnosti:
 - Elmasri-Navathe
 - Saiedian-Spencer

Elmasri-Navathe algoritem za določanje enega ključa

- Vhod: relacijska shema R , množica funkcionalnih odvisnosti F
 1. Postavi $K =$ začetni kandidat, npr. R (vsi atributi)
 2. Za vsak atribut $X \in K$
 - Izračunaj $\{K-X\}^+$ glede na F
 - Če $\{K-X\}^+$ vsebuje vse attribute R postavi $K = K - \{X\}$
 3. Kar ostane v K je ključ.
- Problem: vrne samo en ključ, ki je odvisen od vrstnega reda pregledovanja atributov

Primer (Elmasari-Navathe):

$R=ABCDEFGG$

$F=\{A \rightarrow D, AG \rightarrow B, B \rightarrow G, B \rightarrow E, E \rightarrow B, E \rightarrow F\}$

- $K=ABCDEFGG, X=A$
 $K-X=BCDEFG, \{K-X\}^+=BCDEFG$
manjka A
- $K=ABCDEFGG, X=B$
 $K-X=ACDEFG, \{K-X\}^+=ABCDEFG$ $\{AG \rightarrow B\}$
- $K=ACDEFG, X=C$
 $K-X=ADEFG, \{K-X\}^+=ABDEFG$ $\{AG \rightarrow B\}$
manjka C
- $K=ACDEFG, X=D$
 $K-X=ACEFG, \{K-X\}^+=ABCDEFG$ $\{AG \rightarrow B, A \rightarrow D\}$

Primer (Elmasari-Navathe):

$R=ABCDEFGG$

$F=\{A \rightarrow D, AG \rightarrow B, B \rightarrow G, B \rightarrow E, E \rightarrow B, E \rightarrow F\}$

- $K=ACEFG, X=E$
 $K-X=ACFG, \{K-X\}^+=ABCDEF$ $\{AG \rightarrow B, A \rightarrow D, B \rightarrow E\}$
- $K=ACFG, X=F$
 $K-X=ACG, \{K-X\}^+=ABCDEF$ $\{AG \rightarrow B, A \rightarrow D, B \rightarrow E, E \rightarrow F\}$
- $K=ACG, X=G$
 $K-X=AC, \{K-X\}^+=ACD$ $\{A \rightarrow D\}$
manjkajo BEFG

Ključ je
 $ABCDEF - BDEF$
torej ACG

Iskanje ključev relacije na podlagi funkcionalnih odvisnosti

- Splošni veljavne resnice
 - Atribut, ki **ne nastopa** na desni strani **nobene** funkcionalne odvisnosti, **mora** biti vsebovan v **vsakem** ključu
 - Atribut, ki **nastopa** na desni strani neke funkcionalne odvisnosti in **ne nastopa** na levi strani nobene funkcionalne odvisnosti, **ne more** biti vsebovan v **nobenem** ključu
 - Dobri kandidati za ključe so leve strani funkcionalnih odvisnosti in njihove unije

Saiedian-Spencer algoritem za določanje vseh ključev

- Vhod: relacijska shema R ,
min. pokritje funkcionalnih odvisnosti F_{\min}
 1. Poišči množice \mathcal{L} (atributi samo na levi strani odvisnosti in atributi ki ne nastopajo v nobeni odvisnosti), \mathcal{R} (atributi samo na desni strani odvisnosti) in \mathcal{B} (atributi na levi in desni strani odvisnosti)
 2. Preveri množico \mathcal{L} . Če $\mathcal{L}^+ = R$, je edini ključ in lahko končaš, sicer nadaljuj na koraku 3.
 3. Preveri množico \mathcal{B} tako da v \mathcal{L} vstavljaš po vrsti vse možne kombinacije atributov X iz \mathcal{B} , začenši s posameznimi atributi. Kadar dobimo $\{\mathcal{L} \cup X\}^+ = R$, smo našli ključ. N-teric, ki vsebujejo X , dalje ne obravnavamo več.

Primer (Saiedian-Spencer):

$R=ABCDEF G$

$F=\{A \rightarrow D, AG \rightarrow B, B \rightarrow G, B \rightarrow E, E \rightarrow B, E \rightarrow F\}$

1. $\mathcal{L}=CAGBE=CA$
 $\mathcal{R}=DBGEF=DF$
 $\mathcal{B}=BGE$
2. $\mathcal{L}^+=CAD \subseteq R$
3. $X=B \quad \mathcal{L}=CAB$
 $\mathcal{L}^+=CABDGEF = R$
4. $X=G \quad \mathcal{L}=CAG$
 $\mathcal{L}^+=CAGDBEF = R$
5. $X=E \quad \mathcal{L}=CAE$
 $\mathcal{L}^+=CAEDBFG = R$

Ključni: ABC, ACG, ACE

Normalizacija

- Normalizacija je postopek, s katerem pridemo do množice primerno strukturiranih relacij, ki ustrezajo kriteriju normalne oblike.
- Lastnosti primernih relacij:
 - Relacije imajo minimalen nabor atributov
 - Atributi, ki so logično povezani, so zajeti v isti relaciji
 - Med atributi relacij je minimalna redundanca, vsak atribut (razen tujih ključev) je predstavljen samo enkrat.

Prva normalna oblika

- Relacija je v prvi normalni obliki, če:
 - **Nima večvrednostnih atributov**, kar pomeni, da ima vsak atribut lahko le eno vrednost (torej vrednost ne more biti množica). Primer: vzdevek
 - **Nima sestavljenih atributov** (torej vrednost ne more biti relacija). Primer: naslov
 - **Ima definiran ključ** in določene **funkcionalne odvisnosti**
- Koraki:
 - Eliminiranje ponavljajočih skupin (večvrednostnih sestavljenih atributov)
 - Določitev funkcionalnih odvisnosti
 - Določitev ključa

Primer normalizacije v 1. NO

Voznik (ime, priimek, stdov, (datum, znesek, davčna))
Prekršek

- Odpravimo ponavljajočo skupino:
Voznik (ime, priimek, stdov)
Prekršek(datum, znesek, davčna)
- Določimo ključe:
Voznik (ime, priimek, stdov)
Prekršek(#stdov, datum, znesek, davčna)
Relacijska shema Prekršek vključuje ključ originalne sheme.
- Določimo funkcionalne odvisnosti:
stdov → ime, stdov → priimek,
stdov, datum → znesek, stdov → davčna, davčna → stdov

Druga normalna oblika

Shema: ABCDE

$ABC \rightarrow D$

$ABC \rightarrow E$

$B \rightarrow E$ parcialna

- Relacija je v drugi normalni obliki:
 - Če je v prvi normalni obliki
 - **Ne vsebuje parcialnih odvisnosti**: noben atribut ni funkcionalno odvisen le od dela primarnega ključa, temveč od celotnega ključa
- Nekaj pogostih primerov:
 - Relacija, katere primarni ključ je sestavljen le iz enega atributa, je v drugi normalni obliki
 - Relacija, katere primarni ključ je sestavljen iz vseh atributov, je v drugi normalni obliki
- 2. NO je definirana kot pomožna NO za definicijo 3. NO

Primer normalizacije v 2. NO

- Voznik (ime, priimek, stdov)
stdov → ime, stdov → priimek ✓
- Prekršek(stdov, datum, znesek, davčna) ✗
stdov, datum → znesek, stdov → davčna, davčna → stdov
- Postopek normalizacije: problematične neosnovne attribute (tiste, ki niso del ključa) in so delno odvisni od njega prenesemo v novo tabelo in dodamo še dele ključa, od katerih so odvisni
- Prekršek(stdov, datum, znesek)
stdov, datum → znesek, ✓
- Davek(stdov, davčna)
stdov → davčna, davčna → stdov ✓

Tretja normalna oblika

- Relacija je v tretji normalni obliki (tradicionalna definicija):
 - Če je v drugi normalni obliki
 - Če **ne vsebuje tranzitivnih funkcionalnih odvisnosti**: ni funkcionalnih odvisnosti med atributi, ki niso del primarnega ključa oz. ne obstaja atribut, ki ni del primarnega ključa, ki bi bil funkcionalno odvisen od drugega atributa, ki ravno tako ni del primarnega ključa
- Nekaj pogostih primerov:
 - Relacija, katere primarni ključ je sestavljen iz vseh atributov, je v tretji normalni obliki
 - Relacija, kjer le en atribut izmed vseh ni del primarnega ključa, je v tretji normalni obliki

PK:

ABC

D

tranzitivna

E

Tretja normalna oblika

- Relacija je v 3. NO (formalna definicija), če za vsako odvisnost $X \rightarrow A \in F$ velja vsaj eden izmed pogojev:
 1. $X \rightarrow A$ je trivialna odvisnost ($A \subseteq X$)
 2. X je nadključ sheme R
 3. A je osnovni atribut (del nekega ključa)
- Normalizacija v 3. NO je neizgubna; s stikom dobljenih relacij lahko dobimo nazaj originalno relacijo (obstoj neizgubnega stika).

Postopek normalizacije v 3. NO

- Dekompozicija relacijske sheme R v ρ
 - Določimo F (še bolje: izračunamo F_{\min})
 - Vsaki problematični odvisnosti $X \rightarrow A \in F$ priredimo novo relacijsko shemo XA v ρ , razen v primeru, če že obstaja kakšna shema, ki XA vključuje kot podmnožico. Desno stran odvisnosti (A) izločimo iz originalne sheme.
 - Kar ostane od originalne relacijske sheme dodamo v ρ , razen če v ρ že obstaja kakšna shema, ki jo vsebuje.

Primer normalizacije v 3. NO

R=ABCDEF~~G~~

F={A →D, AG →B, B →G, B →E, E →B, E →F}

Ključ: ACG

1. $\rho = \{\}$
2. A →D: $\rho = \rho \cup \{AD\}$, paricalna
3. AG →B: $\rho = \{AD\} \cup \{AGB\}$, parcialna
4. B → G: ni problematična
5. B → E: $\rho = \{AD, AGB\} \cup \{BE\}$, tranzitivna
6. E → B: $\{EB\} \subseteq \{BE\}$, je že vsebovana
7. E → F: $\rho = \{AD, AGB, BE\} \cup \{EF\}$, tranzitivna
8. Končamo: $\rho = \{AD, AGB, BE, EF\} \cup \{ACG\}$

$\rho = \{AD, AGB, BE, EF, ACG\}$

Ali so spodnje relacije v 3. NO?

- Voznik (ime, priimek, stdov)
stdov → ime, stdov → priimek ✓
- Prekršek(stdov, datum, znesek)
stdov, datum → znesek, ✓
- Davek(stdov, davčna)
stdov → davčna, davčna → stdov ✓
- PrekršekDavek (stdov, datum, znesek, davčna)
stdov, datum → znesek, stdov → davčna, davčna → stdov ✗

Primer normalizacije

Predavanja(Šifra predmeta, Ime predmeta, Predavatelj, Katedra)

$F = \{ \text{Šifra predmeta} \rightarrow \text{Ime predmeta},$
 $\text{Šifra predmeta} \rightarrow \text{Predavatelj},$
 $\text{Šifra predmeta} \rightarrow \text{Katedra},$
 $\text{Predavatelj} \rightarrow \text{Katedra} \}$

- a) Poiščite vse ključe.
- b) V kateri normalni obliki je relacija Predavanja?
- c) Normalizirajte relacijo Predavanja v 3. normalno obliko, če je to potrebno.

Primer normalizacije – a) ključi

Predavanja(Šifra predmeta, Ime predmeta, Predavatelj, Katedra)

$F = \{ \text{Šifra predmeta} \rightarrow \text{Ime predmeta},$
 $\text{Šifra predmeta} \rightarrow \text{Predavatelj},$
 $\text{Šifra predmeta} \rightarrow \text{Katedra},$
 $\text{Predavatelj} \rightarrow \text{Katedra} \}$

Saiedian-Spencer:

- $\mathcal{L} = \{ \text{Šifra predmeta}, \text{Predavatelj} \} = \{ \text{Šifra predmeta} \}$
 $\mathcal{R} = \{ \text{Ime predmeta}, \text{Predavatelj}, \text{Katedra} \}$
 $= \{ \text{Ime predmeta}, \text{Katedra} \}$
 $\mathcal{B} = \{ \text{Predavatelj} \}$
- $\mathcal{L}^+ = \{ \text{Šifra predmeta}, \text{Ime predmeta}, \text{Predavatelj}, \text{Katedra} \}$
- Ključ = Šifra predmeta

Primer normalizacije – b) najvišja NO

Predavanja(Šifra predmeta, Ime predmeta, Predavatelj, Katedra)

$F = \{$ Šifra predmeta \rightarrow Ime predmeta,
Šifra predmeta \rightarrow Predavatelj,
Šifra predmeta \rightarrow Katedra,
Predavatelj \rightarrow Katedra $\}$

- 1. NO je!
- 2. NO: ni delnih odvisnosti, torej je!
- 3. NO: tranzitivna odvisnost Predavatelj \rightarrow Katedra, torej ni!
- Normalizacija v 3. NO je torej potrebna!

Primer normalizacije – c) 3. NO

Predavanja(Šifra predmeta, Ime predmeta, Predavatelj, Katedra)
 $F = \{ \text{Šifra predmeta} \rightarrow \text{Ime predmeta}, \text{Šifra predmeta} \rightarrow \text{Predavatelj}, \text{Šifra predmeta} \rightarrow \text{Katedra}, \text{Predavatelj} \rightarrow \text{Katedra} \}$

- $\rho = \{ \}$
 $R = \{ \text{Šifra predmeta}, \text{Ime predmeta}, \text{Predavatelj}, \text{Katedra} \}$
- Problematična tranzitivna odvisnost Predavatelj \rightarrow Katedra:
 $\rho = \rho \cup \{ \{ \text{Predavatelj}, \text{Katedra} \} \}$
 $R = \{ \text{Šifra predmeta}, \text{Ime predmeta}, \text{Predavatelj}, \text{Katedra} \}$
- Končamo (dodamo, kar je ostalo):
 $\rho = \rho \cup \{ \{ \text{Šifra predmeta}, \text{Ime predmeta}, \text{Predavatelj} \} \}$
- Končni rezultat – dekompozicija v dve relaciji:
P1 (Šifra predmeta, Ime predmeta, Predavatelj)
P2 (Predavatelj, Katedra)

Vaja: primer stare izpitne naloge

- Za relacijsko shemo R s funkcionalnimi odvisnostmi F poiščite oba ključa, določite minimalno pokritje F_{\min} in ugotovite, v kateri najvišji normalni obliki se nahaja! Odgovore utemeljite!

$R = ABCD$

$F = \{ACD \rightarrow B, BCD \rightarrow A, ACD \rightarrow AB, BD \rightarrow D, AC \rightarrow C, AB \rightarrow B\}$

Vaje: odvisnosti in ključi (1/2)

1. Imamo relacijo (tabelo)
Ocenelzpitov(VpisnaSt, Predmet, Semester, Ocena)
Določite funkcionalne odvisnosti in
(a) en ključ (Elmasri-Navathe)
(b) vse ključe (Saiedian-Spencer)
2. Imamo relacijo (tabelo)
PostavkaNarocila(SifraNarocila, CrtnaKodalzdelka, Sifralzdelka, Opislzdelka, Cenalzdelka, Kolicina)
Določite funkcionalne odvisnosti in
(a) en ključ (Elmasri-Navathe)
(b) vse ključe (Saiedian-Spencer)
3. Imamo relacijo (tabelo) s podanimi funkcionalnimi odvisnostmi. Določite
(a) en ključ (Elmasri-Navathe)
(b) vse ključe (Saiedian-Spencer)
Zaloga (Sifralzdelka, SifraAkcije, Proizvajalec, Opislzdelka, Cenalzdelka)
Sifralzdelka, SifraAkcije → Proizvajalec, Opislzdelka, Cenalzdelka
Sifralzdelka → Proizvajalec, Opislzdelka
4. Imamo relacijo (tabelo) s podanimi funkcionalnimo odvisnostjo. Ob predpostavki, da imajo vsi produkti istega založnika enako garancijo določite še ostale funkcionalne odvisnosti in
(a) en ključ (Elmasri-Navathe)
(b) vse ključe (Saiedian-Spencer)
ProgramskaOprema(Založnik, Produkt, Verzija, SystemskeZahteve, Cena, Garancija)
Založnik, Produkt, Verzija → SystemskeZahteve, Cena, Garancija

Vaje: odvisnosti in ključi (2/2)

5. Imamo relacijo (tabelo) s podanimi funkcionalnimi odvisnostmi. Določite
(a) en ključ (Elmasri-Navathe)
(b) vse ključe (Saiedian-Spencer)
R1(H, I, J, K, L, M, N, O)
H, I → J, K, L
J → M K → N L → O
6. Imamo relacijo (tabelo) s podanimi funkcionalnimi odvisnostmi. Določite
(a) en ključ (Elmasri-Navathe)
(b) vse ključe (Saiedian-Spencer)
R2(D, O, N, T, C, R, Y)
D, O → N, T, C, R, Y
C, R → D
D → N
7. Imamo relacijo (tabelo) s podanimi funkcionalnimi odvisnostmi. Določite
(a) en ključ (Elmasri-Navathe)
(b) vse ključe (Saiedian-Spencer)
Shipping (ShipName, ShipType, VoyageID, Cargo, Port, Date)
ShipName → ShipType
VoyageID → ShipName, Cargo
ShipName, Date → VoyageID, Port
Date je datum prihoda ladje v pristanišče (Port).