K2.2

Normalizacija in podatkovne baze

- Spada na širše področje načrtovanja PB
- Samostojna uporaba primerna za manj kompleksne PB (do okoli 10 tabel)

- Imamo problem iz realnega sveta, opisan z atributi združenimi v relacije in omejitvami vrednosti atributov.
- Normalizacija odgovarja na vprašanja:
  - ali je **obstoječa** struktura relacij oz. tabel v PB primerna za obravnavo danega problema (pasivna vloga normalizacije)
  - kako **združiti atribute v tabele**, da dobimo najprimernejšo strukturo PB (aktivna vloga normalizacije)

Modeliranje omejitev s funkcionalnimi odvisnostmi

- Relacija je model nekega stanja v svetu, torej njena vsebina ne more biti poljubna.
- Realne omejitve ne dovoljujejo, da bi bili odnosi v svetu kakršnikoli; možna so le določena stanja (tudi: poslovna pravila).
- Omejitve v relacijskem modelu formalno opišemo s pomočjo odvisnosti
- Odvisnosti so sredstvo, s katerim v relacijskem podatkovnem modelu povemo, katere vrednosti relacij (komb. vrednosti atributov v vrsticah) so veljavne in katere sploh ne morejo obstajati

Funkcionalne odvisnosti..

- Imejmo relacijsko shemo R z množico atributov, katere podmnožici sta X in Y.

- V relacijski shemi R velja funkcionalna odvisnost X → 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 (vrstici), ki bi se ujemali v vrednostih atributov X in se ne bi ujemali v vrednostih atributov Y.

- Preprosto povedano, **obstaja** neka funkcija f, s pomočjo katere lahko iz vrednosti X izračunamo vrednosti Y = f(X).

Primeri funkcionalnih odvisnosti

- Imamo relacijo s shemo

  Stevila(X, Y, Z)

  z naslednjim pomenom:

  X in Y sta celi števili, Z pa njuna vsota

- Funkcionalne odvisnosti relacijske sheme so:

  $F \equiv \{ XY \rightarrow Z, XZ \rightarrow Y, YZ \rightarrow X\}$

- S pomočjo funkcionalnih odvisnosti definiramo tudi pojem ključa, ki implementira določene integritetne omejitve.

Primeri funkcionalnih odvisnosti

- Imamo relacijo s shemo

  Izpit(VpŠt, Priimek, Ime, ŠifraPredmeta, Datum izpita, OcenaPisno, OcenaUstno)

  z naslednjim pomenom:

  Študent z vpisno številko VpŠt ter priimkom Priimek in imenom Ime je na DatumIzpita opravljal izpit iz predmeta s šifro ŠifraPredmeta. Dobil je oceni OcenaPisno in OcenaUstno.

- Funkcionalne odvisnosti relacijske sheme Izpit so:

  F ≡ { VpŠt → (Priimek, Ime), (VpŠt, ŠifraPredmeta, DatumIzpita) → (OcenaPisno, OcenaUstno) }

Ključ relacije r / relacijske sheme R

- Ker je relacija množica n-teric, se v njej vse n-terice med seboj razlikujejo.
- Če v shemi nastopajo funkcionalne odvisnosti za sklicevanje na posamezno n-terico ni potrebno poznati vseh vrednosti atributov n-terice.
- Vsaki množici atributov, ki enolično določa vsako n-terico relacije r, pravimo ključ relacije r oziroma pravilneje ključ relacijske sheme R.
- Lastnosti ključa $K \subseteq R$:
  a) $K \rightarrow R$ oziroma $K^+ = R$ (zaprtje množice atributov)
  b) K je minimalna: ne moremo odstraniti nobenega atributa, da še vedno velja (a).

Iskanje **naravnih** ključev relacije na podlagi specificiranih funkcionalnih odvisnosti

- Izhajamo iz relacijske sheme R (množica atributov) in nad njo definirane množice funkcionalnih odvisnosti F
- Pomožni pojmi (zaprtje množice atributov)
- Različni postopki (algoritmi) za določanje enega ali vseh naravnih ključev
- Za kaj potrebujemo naravne ključe
  - integritetne omejitve
  - normalizacija
- Kompaktna notacija zapisa: $X \rightarrow Y$
  X in Y sta množici atributov

Pomožen pojem: zaprtje (closure) množice atributov

- Zaprtje X^+ množice atributov X glede na množico funkcionalnih odvisnosti F
- X^+ = {A: A lahko "izračunamo" iz X s pomočjo odvisnosti iz F}

- Postopek za izračun X^+ (algoritem fiksne točke)

```
Vhod: X, F
Izhod: X+

X+ = X
ponavljaj
    stari X+ = X+
    za vsako odvisnost Y → Z ∈ F naredi
          če Y ⊆ X+ potem
                X+ = X+ ∪ Z
dokler ni stari X+ = X+
```

Primer izračuna zaprtja

R=ABCDEFG

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

Iščemo $\{EF\}^+$:

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

5. Končni rezultat: $\{EF\}^+ = EF^+ = EFABG$

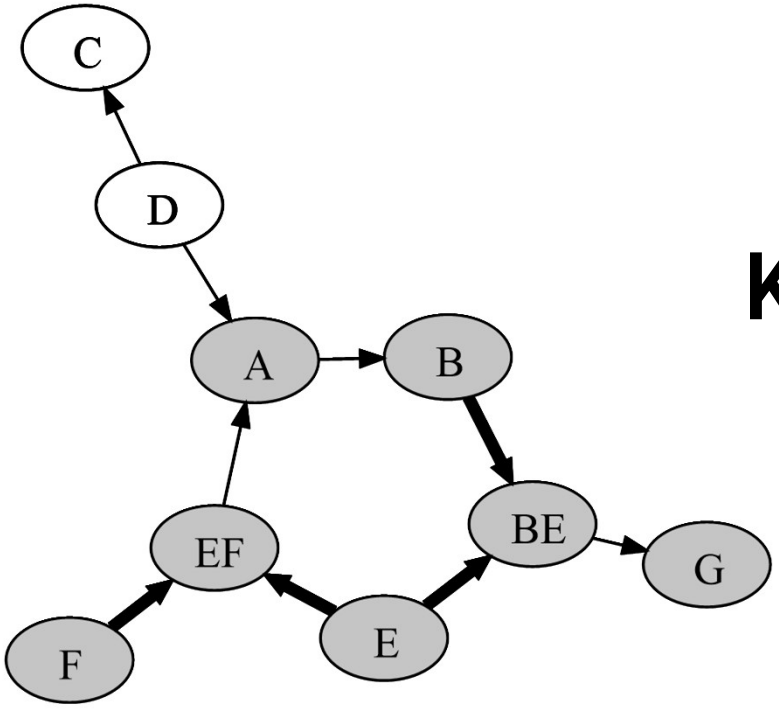
Ali je EF ključ ali vsak kandidat za ključ? Ne, ker $\{EF\}^+ \subset R$.

Za vsakega kandidata za ključ K namreč velja: $K^+=R$.

R=ABCDEFG
F={A→B, BE→G, EF→A, D→AC}

Primer izračuna zaprtja

$\{EF\}^+$



KONEC!

Iskanje naravnih ključev relacije na podlagi podanih 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

Elmasari-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 $\{K-X\}^+ = R$ (vsebuje vse atribute R)
   postavi K= K- {X}
3. Kar ostane v K je ključ.

- Problem: vrne samo en ključ, odvisen od vrstnega reda pregledovanja atributov

Primer (Elmasari-Navathe):

R=ABCDEFG

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

- K=ABCDEFG, X=A
  K-X= BCDEFG, $(K-X)^+$= BCDEFG
  manjka A
- K=ABCDEFG, 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 ($AG\rightarrow B$, $A\rightarrow D$)
- K=ACEFG, X=E
  K-X= ACFG, $(K-X)^+$= ABCDEFG ($AG\rightarrow B$, $A\rightarrow D$,$B\rightarrow E$)
- K=ACFG, X=F
  K-X= ACG, $(K-X)^+$= ABCDEFG ($AG\rightarrow B$, $A\rightarrow D$,$B\rightarrow E$,$E\rightarrow F$)
- K=ACG, X=G ($A\rightarrow D$)
  K-X= AC, $(K-X)^+$= ACD manjkajo BEFG

Ključ je ABCDEFG - BDEF torej ACG

Saiedian-Spencer algoritem za določanje **vseh** ključev

- Vhod: relacijska shema R,
  množica funkcionalnih odvisnosti F

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

1. $\mathcal{L}$= CAGBE – BGE = CA
   $\mathcal{R}$= DBGEF – BGE = DF
   $\mathcal{B}$ = BGE

2. $\mathcal{L}^+$=CAD $\subseteq$ R

3. X=B $\mathcal{L}$=CAB
   $\mathcal{L}^+$=CABDGEF = R

4. X=G $\mathcal{L}$=CAG
   $\mathcal{L}^+$=CAGDBEF = R

5. X=E $\mathcal{L}$=CAE
   $\mathcal{L}^+$=CAEDBFG = R

Ključi: ABC, ACG, ACE

Integritetne omejitve in ključi

- Integritetna omejitev v splošnem zagotavlja smiselno vsebino podatkov – njihovo celovitost
- Primarni ključ izberemo kot enega izmed naravnih ali umetnih ključev.
- Primarni ključ (in tudi vsi ostali - alternativni – ključi) zagotavljajo tudi integritetne omejitve enoličnosti vključenih atributov, kar zagotavlja možnost enolične identifikacije tudi v prihodnje.
- To še posebej velja za naravne ključe.
- Fizična implementacija v SQL:
  - PRIMARY KEY (PK)
  - UNIQUE (AK), NOT NULL (za vsak atribut iz AK)

Normalizacija

- Problemi zaradi redundance podatkov v osnovnih relacijah.
- Namen normalizacije.
- Uporabnost normalizacije pri načrtovanju relacijske podatkovne baze.
- Postopek normalizacije.
- Normalne oblike glede na restriktivnost in stopnjo neredundantnosti:
  - I. normalna oblika
  - II. normalna oblika
  - III. normalna oblika
  - Boyce-Coddova normalna oblika
  - IV. normalna oblika (večvrednostne odvisnosti)
  - V. normalna oblika (stične odvisnosti)
  - VI. Normalna oblika (ni netrivialnih stičnih odvisnosti, upošteva časovni okvir)

Kaj je normalizacija

- Normalizacija je postopek, s katerem pridemo do množice primernih (primerno strukturiranih) relacij, ki ustrezajo potrebam uporabe
- Nekaj lastnosti primernih relacij:
  - Relacije imajo minimalen nabor atributov → zgolj tiste, ki so potrebni za pokritje potreb poslovnega sistema;
  - 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.

Načrtovanje PB in normalizacija

- Osnovni cilj načrtovanja relacijske podatkovne baze je smiselno grupirati atribute v relacije na način, da bo med podatki **čim manj redundance**.
- Potencialne koristi pravilnega načrtovanja so:
  - Spremembe podatkov v podatkovni bazi dosežemo z minimalnim številom operacij → večja učinkovitost; manj možnosti za podatkovne nekonsistentnosti.
  - Manjše potrebe po diskovnih kapacitetah za shranjevanje osnovnih relacij → manjši stroški.
- Normalizacija (oziroma upoštevanje njenih principov) je pomemben korak v postopku načrtovanja.

Primer nenormalizirane relacije

- Relacija Osebje ima odvečne podatke.

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

Odvečni (ponavljajoči se) naslovi.

Ažurirne anomalije

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

Anomalije pri dodajanju vrstic

- Dodajanje novih članov oddelka: ponovno moramo (pravilno) vpisati naslov oddelka
- Dodajanje novega oddelka: za podatke o članu vpišemo NULL;
  - Problem: kaj pa (primarni) ključ?

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 Večno pot 113. 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

Postopek normalizacije

- Postopku preoblikovanja relacij v obliko, pri kateri do ažurirnih anomalij ne more priti, pravimo normalizacija.
- Obstaja več stopenj normalnih oblik (form):
  - I. normalna oblika,
  - II. normalna oblika,
  - III. normalna oblika,
  - Boyce-Coddova normalna oblika
  - IV. normalna oblika
  - V. normalna oblika
  - VI. normalna oblika.

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: telefonska številka (fiksna, mobilna, službena)
  - Nima sestavljenih atributov (torej vrednost ne more biti relacija). Primer: naslov (ulica, hišna številka, pošta, kraj, država)
  - Ima definiran primarni ključ in določene funkcionalne odvisnosti
- Koraki normalizacije:
  - Elimininiranje ponavljajočih skupin (večvrednostne in sestavljene atribute skupno obravnavamo kot ponavljajoče skupine)
  - Določitev funkcionalnih odvisnosti
  - Določitev primarnega ključa

Prva normalna oblika..

- Primer relacijske sheme:
  - Študent(VŠ, priimek, ime, pst, kraj, {(šifra predmeta, naziv, ocena)})
- Določimo primarni ključ:
  - VŠ
- Vmesni korak: odpravimo ponavljajoče skupine:
  - Študent(VŠ, priimek, ime, pst, kraj)
  - PredmetOcena(šifra predmeta, naziv, ocena)
- Določimo primarna ključa obeh relacijskih shem:
  - Študent(VŠ, priimek, ime, pst, kraj)
  - PredmetOcena(#VŠ, šifra predmeta, naziv, ocena)

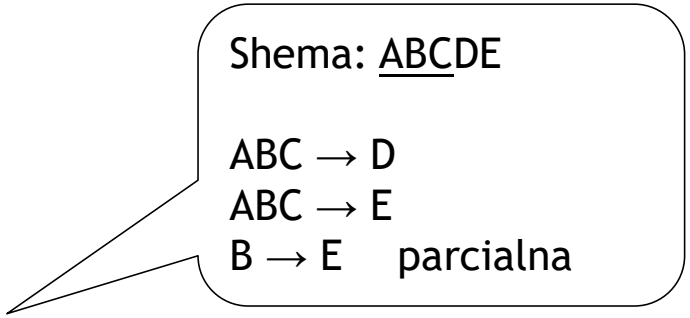
Primarni ključ glavne relacije se kot **tuji ključ** prenese k ponavljajočim skupinam in postane del njihovega primarnega ključa.

Prva normalna oblika

- Gledamo obe relacijski shemi …
  - Študent(VŠ, priimek, ime, pst, kraj)
  - PredmetOcena(#VŠ, šifra predmeta, naziv, ocena)
- … in določimo funkcionalne odvisnosti:
  - VŠ → Ime
  - VŠ → Priimek
  - VŠ → pst
  - VŠ → kraj
  - Pst → kraj

  - Šifra predmeta → naziv
  - VŠ, šifra predmeta → naziv
  - VŠ, šifra predmeta → ocena

Druga normalna oblika..



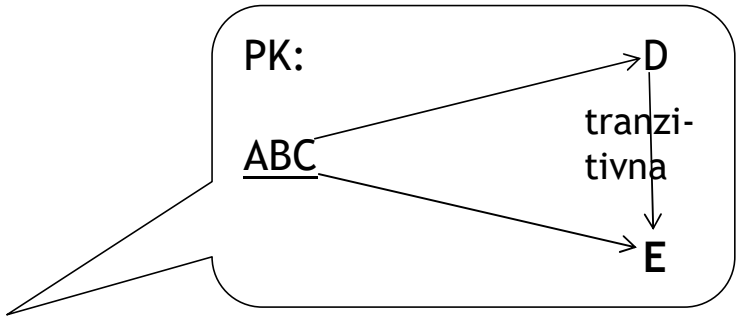
- Relacija je v drugi normalni obliki:
  - Če je v prvi normalni obliki
  - Ne vsebuje parcialnih (delnih) odvisnosti: noben atribut ni funkcionalno odvisen le od dela primarnega ključa, temveč le od celotnega ključa
  - Normalizacija: problematičnim (parcialnim) odvisnostim dodelimo nove relacijske sheme (vsaki svojo). Desne strani izločimo iz originalne sheme.
- Ugotovitev:
  - 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

Druga normalna oblika

- Nadaljevanje predhodnega primera:
  - Študent(VŠ, priimek, ime, pst, kraj)
  - PredmetOcena(#VŠ, šifra predmeta, naziv, ocena)
- Funkcionalne odvisnosti:
  - VŠ → Ime
  - VŠ → Priimek
  - VŠ → pst
  - VŠ → kraj
  - Pst → kraj
  - **Šifra predmeta → naziv**
  - VŠ, šifra predmeta → naziv
  - VŠ, šifra predmeta → ocena

- Relacijska shema Študent je v drugi normalni obliki
- Relacijsko shemo PredmetOcena normaliziramo :
  - Predmet_študent (#VŠ, #šifra predmeta, ocena)
  - Predmet (šifra predmeta, naziv)

Tretja normalna oblika..



- Relacija je v tretji normalni obliki:
  - 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
- Ugotovitev:
  - 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

Tretja normalna oblika

- Nadaljevanje predhodnega primera:
  - Študent(VŠ, priimek, ime, pst, kraj)
  - Predmet (šifra predmeta, naziv)
  - Predmet_študent (#VŠ, #šifra predmeta, ocena)
- Funkcionalne odvisnosti:
  - VŠ → Ime
  - VŠ → Priimek
  - **VŠ → pst**
  - **VŠ → kraj**
  - **Pst → kraj**
  - Šifra predmeta → naziv
  - VŠ, šifra predmeta → naziv
  - VŠ, šifra predmeta → ocena

- Relaciji
  Predmet
  in
  Predmet_Študent
  sta v tretji normalni obliki

- Relacija Študent:
  - Študent (VŠ, priimek, ime, #pst)
  - Pošta (pst, kraj)

Postopek normalizacije v 3. NO

- Normalizacija relacijske sheme R v ρ
  - $\rho = \{\}$
  - Vsaki problematični odvisnosti $X \rightarrow A \in F$ priredimo novo relacijsko shemo XA v ρ, razen v primeru, če že obstaja kakšna shem, 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

    Odvisnost je problematična, kadar je tranzitivna (3. NO) ali parcialna (2. NO)

Primer normalizacije v 3. NO

R=ABCDEFG
F={A $\rightarrow$D, AG $\rightarrow$B, B $\rightarrow$G, B $\rightarrow$E, E $\rightarrow$B, E $\rightarrow$F}
Primarni ključ: ACG Alternativni ključi: ABC, ACE

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

ρ= {AD, AGB, BE, EF, ACG}

Boyce-Coddova normalna oblika (BCNO)

- V primerjavi s 3.NO vpelje BCNO dodatne omejitve.
- Osnovne normalne oblike preverjajo parcialne (2.NO) in tranzitivne (3.NO) odvisnosti **glede na primarni ključ**. Ne preverjajo pa se odvisnosti glede na ostale kandidate za ključ (alternativne ključe).
- BCNO razširja omejitve na vse kandidate za ključ.
- Ugotovitev:
  - Če ima shema samo enega kandidata za ključ, potem je njena BCNO enaka 3.NO!

Boyce-Coddova normalna oblika (BCNO)

- Relacija R je v BCNO, če za vsako odvisnost $X \rightarrow A \in F_R$ velja vsaj eden izmed pogojev:

    1. $X \rightarrow A$ je trivialna odvisnost ($A \subseteq X$)
    2. X je (nad)ključ sheme R

- Normalizacija v BCNO je lahko izgubna; s stikom dobljenih relacij ne dobimo nazaj originalne relacije (neizgubni stik ne obstaja).

Postopek normalizacije v BCNO

- Dekompozicija relacijske sheme R v ρ
  - Izračunamo F_{min} (izločimo trivialne in redundantne odvisnosti)
  - Postavimo $\rho=\{\}$, S=R (množica preostalih atributov) in F = F_{min} (množica še veljavnih odvisnosti v S)
  - Vsaki problematični odvisnosti $X \rightarrow A \in F$ priredimo novo relacijsko shemo XA v ρ, razen v primeru, če že obstaja kakšna shem, ki XA vključuje kot podmnožico. Desno stran odvisnosti (A) izločimo iz originalne sheme in sproti prilagajamo množico odvisnosti!
    Napotek: v množicah S in F si sproti vodimo trenutno množico atributov in še veljavnih odvisnosti v njej!
  - Kar ostane od originalne relacijske sheme dodamo v ρ, razen če v ρ že obstaja kakšna shema, ki jo vsebuje

Primer normalizacije v BCNO

R=ABCDEFG

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

Ključ: ACG, ABC, ACE

1. Začetek ρ={} **S=ABCDEFG**
   F={A $\rightarrow$D, AG $\rightarrow$B, B $\rightarrow$G, B $\rightarrow$E, E $\rightarrow$B, E $\rightarrow$F}
2. A $\rightarrow$D: $\rho = \rho \cup$ {AD} **S=ABCEFG**
   F={ AG $\rightarrow$B, B $\rightarrow$G, B $\rightarrow$E, E $\rightarrow$B, E $\rightarrow$F}
3. AG $\rightarrow$B: ρ= {AD} $\cup$ {AGB} **S=ACEFG**
   F={E $\rightarrow$F}
4. E $\rightarrow$ F: ρ= {AD, AGB} $\cup$ {EF} **S=ACEG**
   F={}
5. Končamo: ρ= {AD, AGB, EF} $\cup$ {ACEG}

ρ= {AD, AGB, EF, ACEG}

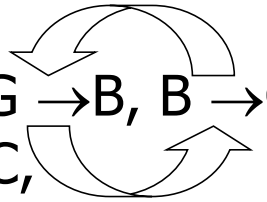
Izgubili smo: B $\rightarrow$E, E $\rightarrow$B.

Primer normalizacije v BCNO

R=ABCDEFG

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

Ključ: ACG, ABC, ACE



1. Začetek ρ={}
   S=ABCDEFG
   F={A $\rightarrow$D, AG $\rightarrow$B, B $\rightarrow$G, B $\rightarrow$E, E $\rightarrow$B, E $\rightarrow$F}
2. A $\rightarrow$D: $\rho = \rho \cup$ {AD}
   S=ABCEFG
   F={ AG $\rightarrow$B, B $\rightarrow$G, B $\rightarrow$E, E $\rightarrow$B, E $\rightarrow$F}
3. B $\rightarrow$ G: ρ= {AD} $\cup$ {BG}
   S=ABCEF
   F={ B $\rightarrow$E, E $\rightarrow$B, E $\rightarrow$F}
4. B $\rightarrow$ E: ρ= {AD, BG} $\cup$ {BE}
   S=ABCF
   F={}
5. Končamo: ρ= {AD, BG, BE} $\cup$ {ABCF}

ρ= {AD, BG, BE, ABCF}

Izgubili smo: AG $\rightarrow$B, E $\rightarrow$F.

Ali so spodnje relacije v BCNO?

- Voznik (ime, priimek, <u>stdov</u>)
  stdov $\rightarrow$ ime, stdov $\rightarrow$ priimek
- Prekršek(<u>stdov, datum</u>, znesek)
  stdov,datum $\rightarrow$ znesek,
- Davek(<u>stdov</u>, davčna)
  stdov $\rightarrow$ davčna, davčna $\rightarrow$ stdov
- PrekrškekDavek (<u>stdov, datum</u>, znesek, davčna)
  stdov,datum $\rightarrow$ znesek, stdov $\rightarrow$ davčna, davčna $\rightarrow$ stdov

Ali je spodnja relacija v BCNO?

- Izpit(<u>student, predavatelj</u>, predmet, ocena)
  student,predavatelj $\rightarrow$ predmet
  student,predavatelj $\rightarrow$ ocena
  predmet $\rightarrow$ predavatelj

- Primarni ključ: student, predavatelj
- Sekundarni (alternativni) ključ: student, predmet
  student,predmet $\rightarrow$ predavatelj
  student,predmet $\rightarrow$ ocena

- Ali je relacija v 3.NO?
- Ali je relacija v BCNO?

Normalizacija v BCNO

- Izpit(student, predmet, ocena)
  Predavanje(predavatelj, predmet)
- Kako je s ključi in funkcionalnimi odvisnostmi?
- Primarni ključ: student, predavatelj
  student,predavatelj $\rightarrow$ predmet
  student,predavatelj $\rightarrow$ ocena
  predmet $\rightarrow$ predavatelj
- Sekundarni (alternativni) ključ: student, predmet
  student,predmet $\rightarrow$ predavatelj
  student,predmet $\rightarrow$ ocena
  predmet $\rightarrow$ predavatelj

Normalizacija v BCNO

- Izpit(<u>student, predmet</u>, ocena)
  student,predmet $\rightarrow$ ocena (izpeljana odvisnost v začetni shemi)

- Predavanje(predavatelj, <u>predmet</u>)
  predmet $\rightarrow$ predavatelj

Večvrednostne odvisnosti

- $A \twoheadrightarrow B$ (A večvrednostno določa B)
- Nastanejo tipično, ko nadomestimo večvrednostne atribute s kartezičnim produktom enovrednostnih
- $A \twoheadrightarrow B$ je trivialna, če:
  - $B \subseteq A$
  - $A \cup B = R$
- Obravnavamo samo netrivialne odvisnosti
- Primer: Oseba(<u>Ime</u>, Jezik, Šport)
  Ime $\twoheadrightarrow$ Jezik, Ime $\twoheadrightarrow$ Šport

Večvrednostne odvisnosti

Ime	Jezik	Šport
Janez	Ang	Tek
Janez	Nem	Plavanje
Janez	Nem	Tek
Janez	Ang	Plavanje

(Janez, {Ang,Nem}, {Tek, Plavanje})

Ime ↠ Jezik
Ime ↠ Šport

4NO – četrta (poslovna) normalna oblika

- Relacija je v četrti normalni obliki, če je v Boyce-Coddovi normalni obliki in ne vsebuje netrivialnih večvrednostnih odvisnosti
- Postopek normalizacije: netrivialne večvrednostne odvisnosti zamenjamo z novimi relacijskimi shemami (kot v BCNO)

Primer 1: normalizacija v 4. NO

- Oseba(<u>Ime</u>, Jezik, Šport)
  Ime ↠ Jezik, Ime ↠ Šport

- Oseba(<u>Ime</u>)
  ImeJezik(<u>#Ime, Jezik</u>)
  ImeŠport(<u>#Ime, Šport</u>)

5. Normalna oblika

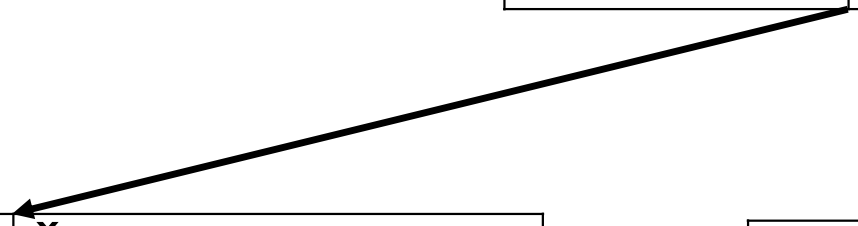
- Stične odvisnosti
- Nastanejo, ko je relacija R rezultat stika dveh ali več relacij
- Anomalije: po ažuriranju relacije R mora ta še vedno imeti obliko stika več relacij

- Relacija je v 5. NO, če je v četrti NO in ne vsebuje stičnih odvisnosti.
- Normalizacija na originalne stične tabele

5. Normalna oblika

Stične odvisnosti:
⋈(Univerza-Študij, Študij-Stopnja)

Univerza	Študij	Stopnja
Ljubljana	Računalništvo	1
Ljubljana	Računalništvo	2
Ljubljana	Bioinformatika	3
Maribor	Računalništvo	1
Maribor	Računalništvo	2



Univerza	Študij
Ljubljana	Računalništvo
Ljubljana	Bioinformatika
Maribor	Računalništvo

Študij	Stopnja
Računalništvo	1
Bioinformatika	3
Računalništvo	2

5. Normalna oblika

Študij računalništva se v **Ljubljani** začne izvajati tudi na **3. stopnji**.

Univerza	**Študij**	**Stopnja**
Ljubljana	Računalništvo	1
Ljubljana	Računalništvo	2
Ljubljana	Bioinformatika	3
Maribor	Računalništvo	1
Maribor	Računalništvo	2

Univerza	**Študij**
Ljubljana	Računalništvo
Ljubljana	Bioinformatika
Maribor	Računalništvo

Študij	**Stopnja**
Računalništvo	1
Bioinformatika	3
Računalništvo	2

5. Normalna oblika

Študij računalništva se v **Ljubljani** začne izvajati tudi na **3. stopnji**.

Univerza	**Študij**	**Stopnja**
Ljubljana	Računalništvo	1
Ljubljana	Računalništvo	2
Ljubljana	Bioinformatika	3
Maribor	Računalništvo	1
Maribor	Računalništvo	2
Ljubljana	Računalništvo	3

Univerza	**Študij**
Ljubljana	Računalništvo
Ljubljana	Bioinformatika
Maribor	Računalništvo

Študij	**Stopnja**
Računalništvo	1
Bioinformatika	3
Računalništvo	2
Računalništvo	3

5. Normalna oblika

Študij računalništva se v **Ljubljani** začne izvajati tudi na **3. stopnji**.

Univerza	Študij	Stopnja
Ljubljana	Računalništvo	1
Ljubljana	Računalništvo	2
Ljubljana	Bioinformatika	3
Maribor	Računalništvo	1
Maribor	Računalništvo	2
Ljubljana	Računalništvo	3
Maribor	Računalništvo	3

Univerza	Študij
Ljubljana	Računalništvo
Ljubljana	Bioinformatika
Maribor	Računalništvo

Študij	Stopnja
Računalništvo	1
Bioinformatika	3
Računalništvo	2
Računalništvo	3

6. Normalna oblika

- Definicija: stična odvisnost $\bowtie(X_1,X_2, \ldots , X_n)$ je trivialna, kadar vsaj en X_i vsebuje vse atribute
- 6. normalna oblika:
  - ni netrivialnih stičnih odvisnosti (torej je v 5. NO)
  - upošteva časovni okvir veljavnosti vrednosti atributov (del ključa)
  - posledica: vse relacije v obliki $R_i(\underline{ključ_i}, vrednost_i)$ ali $R_i(\underline{ključ_i})$, kar povzroči dodatno izgubo kompleksnejših odvisnosti (poslovnih pravil)
- Neformalno:
  - Relacijo poleg primarnega ključa sestavlja še največ en atribut.
  - Vsi atributi so obvezni
- Teoretična osnova dimenzijskih tabel v podatkovnih skladiščih (ki pa se je v praksi pogosto ne držijo – denormalizacija)