

## Osnove relacionog modela podataka

Model podataka sadrži:

- strukturalnu komponentu
- operacijsku komponentu
- integritetnu komponentu

Nivoi apstrakcije - određeni modelom podataka:

- nivo intenzije , konteksta, tipa, opisuje LSO, šeme
- nivo ekstenzije, konkretizacije, pojave tipa, opisuje LSP

## Strukturalna komponenta

Modelira statičku strukturu sistema, LSO, šeme BP.

Sastoji se od:

- primitivnih i složenih koncepata - gradivnih elemenata modela podataka
- pravila za kreiranje složenih koncepata

Primitivni koncepti u relacionom modelu podataka:

- Nivo intenzije
  - obeležje (atribut) - predstavlja osobinu (svojstvo) klase entiteta ili poveznika u realnom sistemu
  - domen - specifikacija skupa mogućih vrednosti koja neka obeležja mogu da dobiju, svako obeležje ima pridružen tačno jedan domen
- Nivo ekstenzije
  - vrednost

Polazne pretpostavke na kojoj se zasnivaju neke tehnike projektovanja relacione šeme baze podataka (metoda dekompozicije, metoda sinteze):

- postojanje šeme univerzalne relacije
  - poznat je univerzalni skup obeležja
  - poznat je skup svih ograničenja realnog sistema
  - uvek egzistira u imaginarnom svetu
  - posledica :
    - jedinstvena uloga svakog obeležja
      - ne postoje sinonimi - dva obeležja sa istom ulogom
      - ne postoje homonimi - obeležja sa više od jedne uloge
    - svako obeležje u budućoj šemi baze podataka identifikuje se isključivo putem svog naziva
- univerzalna relacija

- pojava nad šemom univerzalne relacije
- reprezentuje stanje realnog sistema
- apstraktni pojam - preuzet iz imaginarnog sveta
- praktično je nemoguće implementirati je pod nekim SUBP
  - prepreke logičkog karaktera
  - prepreke vezane za moguću fizičku organizaciju podataka
- teoretski zahtev - stanje relacione baze podataka nad šemom baze podataka treba u informativnom smislu da odgovara sadržaju univerzalne relacije
- šema baze podataka treba da zadovolji projektantske kriterijume

Nivo intenzije	Nivo ekstenzije
Domen	Vrednost
Obeležje	Podataka
Skup obeležja	Torka
Šema relacije	Relacija
Šema baze podataka	Baza podataka

Složeni koncepti:

- torka
  - predstavlja pojavu entiteta ili poveznika, svakom obeležju iz nekog skupa obeležja se dodeljuje konkretna vrednost iz skupa mogućih vrednosti definisanog domenom
- restrikcija ("skraćenje") torke na skup obeležja
  - svakom obeležju iz skupa nad kojim se vrši restrikcija pridružuje se ona vrednost koju ima polazna torka
- relacija nad skupom obeležja
  - konačan skup torki, predstavlja skup realnih entiteta ili poveznika, podskup svih mogućih torki nad skupom obeležja
  - ne može da sadrži dve identične torke
  - uobičajena reprezentacije relacija je pomoću tabele, relaciju predstavlja kompletan njen sadržaj
  - poredak obeležja i torki ne utiče na informacije koje sa sobom nosi relacija
- šema relacije
  - definisana nazivom (opciono), skupom obeležja i skupom ograničenja
- pojava nad šemom relacije
  - bilo koja relacija nad skupom obeležja šeme relacija takva da zadovoljava skup ograničenja šeme relacije
- relaciona šema baze podataka
  - definisana skupom šema relacija i skupom međurelacionih ograničenja
  - statička, sporo promenljiva kategorija sistema baze podataka
- relaciona baza podataka
  - pojava nad zadatom relacionom šemom baze podataka
  - svakoj šemi relacije u relacionoj šemi baze podataka odgovara jedna njena pojava
  - reprezentuje jedno stanje realnog sistema

- ažurira se, jer promene stanja realnog sistema treba da prate odgovorajuće promene podataka u bazi podataka
- dinamička, stalno promenljiva kategorija sistema baze podataka
- u formalno konzistentnom stanju je ako svaka relacija zadovoljava sva ograničenja odgovarajuće šeme relacije i ako sve relacije zadovoljavaju međurelaciona ograničenja - SUBP može da kontroliše
- u suštinski konzistentnom stanju je ako je u formalno konzistentnom stanju i ako predstavlja vernu sliku stanja realnog sistema

## Operacijska komponenta

Modelira dinamiku izmene stanja.

Sadrži:

- upitni jezik (Query Language)
  - operacije za izražavanje upita nad jednom ili skupom relacija, pružanje podataka na uvid korisniku
  - sadrži operatore za izražavanje upita, pravila za njihovu primenu i pravila za formiranje operanada upita - izraza
  - vrste teoretskih upitnih jezika u relacionom modelu podataka:
    - relaciona algebra - zasnovana na teoriji skupova i skupovnih operacija: unija, presek i razlika
    - relacioni račun - operacije nad torkama i domenima, zasnovano na predikatskom računu prvog reda
- jezik za manipulisanje podacima (Data Manipulation Language)
  - operacije za ažuriranje relacija: dodavanje nove, brisanje postojeće ili modifikacija podataka postojeće torke
- jezik za definiciju podataka (Data Definition Language)
  - operacije za upravljanje šemom baze podataka: kreiranje, brisanje i modifikacija njenih delova

Operacije relacionog računa:

- selekcija torki iz relacije po nekom kriterijumu, logičkoj formuli
- projekcija (restrikcija) relacije na podskup skupa obeležja
  - izdvajanje vrednosti pojedinih obeležja iz relacije
- prirodni spoj relacija
  - spajanje torki različitih relacija po osnovu istih vrednosti zajedničkih obeležja
  - ukoliko nema zajedničkih obeležja, nastaje dekartov proizvod relacija
- dekartov proizvod relacija
  - spajanje torki formiranje svih mogućih kombinacija torki iz dve relacije koje ne sadrže zajednička obeležja
- theta spajanje relacija
  - selektovanje torki iz dekartovog proizvoda relacija po nekom kriterijumu

## Integritetna komponenta

Sadrži skup tipova ograničenja, uslova integriteta.

Karakteristike tipa ograničenja:

- definicija
  - formalizam za zapisivanje
- pravilo za interpretaciju, validaciju
- oblast definisanosti
  - tip logičke strukture obeležja nad kojom se ograničenje definiše
  - podela na vanrelaciono, jednorelaciono i višerelaciono ograničenje, u zavisnosti od skupa šema relacija
- oblast interpretacije
  - tip logičke strukture podataka nad kojom se ograničenje interpretira
  - podela na ograničenje vrednosti, torke, relaciono i međurelaciono ograničenje
  - relacija, u ovom slučaju, se može odnositi i na relaciju koje su nastale primenom izraza relacione algebre i/ili računa nad jednom ili više drugih relacija
- skup operacija nad bazom podataka koje mogu dovesti do narušavanja ograničenja datog tipa
- skup mogućih akcija kojima se obezbeđuje očuvanje validnosti baze podataka, pri pokušaju narušavanja ograničenja datog tipa, definisan za svaku operaciju koja može dovesti do narušavanja ograničenja

Tipovi ograničenja u relacionom modelu podataka:

- ograničenje domena
  - specifikacija domena sadrži ograničenje domena i opciono predefinisanu vrednost domena koje zadovoljava ograničenje
  - definiše se u okviru specifikacije domena pomoću:
    - tipa podatka
      - primitivni domen ili oznaka prethodno definisanog domena
      - jedina obavezna komponenta specifikacije ograničenja domena
    - dužine tipa podatka
      - navodi se samo za tipove koji su primitivni domen koji to zahtevaju
    - logičkog uslova
      - mora da ga zadovoljava svaka vrednost iz skupa mogućih vrednosti domena
  - nula, nedostajuća, izostavljena vrednost je specijalna vrednost koja označava nepoznatu postojeću, nepostojeću ili neinformativnu vrednost obeležja, koja se dodaje u skup mogućih vrednosti domena i a priori zadovoljava svako ograničenje domena
  - interpretacija ograničenja - moguća za bilo koju vrednost
- ograničenje vrednosti obeležja
  - specifikacija obeležja šeme relacija se zadaje za svako obeležje šeme relacije i sadrži ograničenje vrednosti obeležja i opciono predefinisanu vrednost obeležja
    - ukoliko se predefinisana vrednost ne navede, važeća je predefinisana vrednost odgovarajućeg domena ili prvog roditeljskog domena koji je ima
  - definiše se u okviru specifikacije obeležja obavezno pomoću:

- oznake, naziva domena obeležja
  - ograničenja nula vrednosti obeležja, dozvole/zabrane dodele nula vrednosti obeležju
  - interpretacija ograničenja - moguća za bilo koju vrednost obeležja
- ograničenje torke
  - izražava ograničenja na moguće vrednosti unutar jedne torke
  - predstavlja skup ograničenja vrednosti obeležja kojem je pridodat logički uslov koji svaka torka mora da zadovolji i može da sadrži bilo koje obeležje date šeme relacije
  - interpretacija ograničenja - moguća za bilo koju torku nad skupom obeležja
- ograničenje ključa, integritet entiteta
  - definiše se pomoću ključa šeme relacija
    - ključ je minimalni podskup skupa obeležja šeme relacije, na osnovu kojeg se jedinstveno može identifikovati svaka torka relacije nad datom šemom
    - skup obeležja predstavlja ključ ako važi:
      - svojstvo jedinstvenosti
        - za svake dve torke iz relacije važi da ako su jednake projekcije torki nad tim skupom onda su to dve iste torke
        - postoji funkcionalna zavisnosti takva da skup obeležja određuje sva obeležja šeme relacije
      - svojstvo minimalnosti - ukoliko ne postoji podskup tog skupa obeležja za koje važi svojstvo jedinstvenosti
    - generisanje jednog ključa šeme relacije
      - polazi se od skupa obeležja šeme relacije i vrši se redukcija izbacivanjem obeležja i izračunavanjem zatvarača ostatka skupa
      - postupak se nastavlja nad ostatkom skupa sve dok se ne nađe skup obeležja za koje važi svojstvo jedinstvenosti i minimalnosti ključa
    - generisanje alternativnih ključeva
      - polazi se od prvog generisanog ključa
      - ukoliko se podskup ključa nalazi sa desne strane neke funkcionalne zavisnosti, menja se njenom levom stranom, vrši se redukcija i dobija se alternativni ključ
  - za sva obeležja ključa (primarna, ključna obeležja, obeležja koja pripadaju barem jednom ključu) nula vrednosti su zabranjene
  - svaka šema relacije mora posedovati najmanje jedan ključ (proizilazi iz definicije pojma relacije)
  - ekvivalentni ključevi - svi ključevi skupa ključeva
  - primarni ključ - jedan izabrani ključ od ekvivalentnih ključeva, svaka šema relacije mora da poseduje tačno jedan, koristi se u ulozi asocijativne (simboličke) adrese za povezivanje podataka u relacijama
  - oblast interpretacije - skup torki nad datom šemom relacije
  -
- ograničenje jedinstvenosti vrednosti obeležja
  - zahteva da ne-nula kombinacija vrednosti obeležja bude jedinstvena u relaciji nad odgovarajućom šemom relacije
  - oblast interpretacije - skup torki nad datom šemom relacije

- zavisnost sadržavanja
  - važi za niz od dve šeme relacije ukoliko:
    - šeme sadrže domenski kompatibilne nizove obeležja
    - za svaku torku iz prve relacije važi jedno od sledećeg za svako obeležje njene šeme relacije:
      - vrednost obeležja u toj torci je nula vrednost
      - postoji torca u drugoj relaciji čija vrednost odgovarajućeg obeležja iz niza obeležja njene šeme relacije je jednaka vrednosti obeležja prve torke
  - oblast definisanosti - niz od dve šeme relacije
  - oblast interpretacije - relacije nad šemama relacije u nizu
- ograničenje referencijalnog integriteta
  - predstavlja zavisnost sadržavanja u kojoj je obeležje iz druge šeme relacije ključ
  - prva šema relacije je referencirajuća, a druga je referencirana
- funkcionalna zavisnost
  - definiše se pomoću niza od 2 skupa obeležja (levog i desnog)
  - ako su vrednosti obeležja iz prvog skupa poznate, poznate su i vrednosti obeležja drugog skupa obeležja
  - relacija zadovoljava funkcionalnu zavisnost ako za svake dve torke relacije važi da ako su njihove restrikcije nad prvim skupom obeležja jednake, onda su jednake i projekcije nad drugim skupom obeležja
  - oblast interpretacije - relacija
  - funkcionalna zavisnost je trivijalna ako je zadovoljena u bilo kojoj relaciji, odnosno kada je drugi skup obeležja podskup prvog
  - funkcionalna zavisnost je logička posledica skupa funkcionalnih zavisnosti ako svaka relacija koja zadovoljava taj skup zadovoljava i logičku posledicu
  - skup funkcionalnih zavisnosti je logička posledica originalnog skupa funkcionalnih zavisnosti ako za svaku funkcionalnu zavisnost iz skupa koji predstavlja logičku posledicu važi da je logička posledica originalnog skupa funkcionalnih zavisnosti
  - zatvarač skupa funkcionalnih zavisnosti - skup svih logičkih posledica, nadskup originalnog skupa funkcionalnih zavisnosti
  - skupovi funkcionalnih zavisnosti su ekvivalentni ako su oba logičke posledice drugog skupa, odnosno ako su im zatvarači jednaki
  - implikacioni problem - da li za neku funkcionalnu zavisnost važi da je logička posledica?
    - rešava se pomoću Armstrongovih pravila izvođenja:
      - refleksivnost
      - proširenje
      - pseudotranzitivnost
    - izvedena pravila izvođenja
      - uniranje desnih strana
      - dekompozicija desnih strana
      - tranzitivnost
    - sistem Armstrongovih pravila izvođenja je
      - korektan

- svaka funkcionalna zavisnost koja se izvede primenom pravila iz nekog skupa funkcionalnih zavisnosti predstavlja logičku posledicu tog skupa
- kompletan
  - svaka logička posledica nekog skupa funkcionalnih zavisnosti može se izvesti primenom pravila iz tog skupa
- neredundantan (minimalan)
  - ne može se eliminisati kao suvišno ni jedno od tri pravila izvođenja, a da su prethodna dva svojstva očuvana
- funkcionalna zavisnost je nepotpuna ako leva strana sadrži logički suvišno obeležje na levoj strani
- funkcionalna zavisnost A u C je tranzitivna ako važi da A određuje B, B C, a B ne određuje A
- zatvarač skupa obeležja - skup svih obeležja koja funkcionalno zavise od tog skupa
  - algoritam za izračunavanje zatvarača:
    - naći sve funkcionalne zavisnosti čija leva strana je podskup skupa obeležja čiji se zatvarač traži
    - naći zatvarač unije obeležja sa desne strane pronađenih funkcionalnih zavisnosti i originalnog skupa obeležja
- projekcija skupa funkcionalnih zavisnosti na skup obeležja
  - predstavlja skup svih funkcionalnih zavisnosti koje logički slede iz originalnog skupa, a definisane su u skupu obeležja nad kojim se vrši projekcija

## Formalne matematičke definicije

Univerzalni skup obeležja	$U = \{A_1, \dots, A_n\}$
Univerzalni skup domena	$D = \{D_1, \dots, D_k\}$
Domen obeležja	$Dom: U \rightarrow D,$ $(\forall A_i \in U)(Dom(A_i) \in D)$
Skup svih mogućih vrednosti	$DOM = \cup_{i=1}^n (dom(A_i))$
Torka	$t: U \rightarrow DOM,$ $(\forall A_i \in U)(t(A_i) \in dom(A_i))$
Restrikcija (projekcija) torke	$X \subseteq U, t: U \rightarrow DOM$ $t[X]: X \rightarrow DOM$ $(\forall A \in X)(t[X](A) = t(A))$
Relacija	$r(U) \subseteq \{t \mid t: U \rightarrow DOM\},  r  \in \mathbb{N}_0$
Unija	$r(R) \cup s(R) = \{t \mid t \in r \wedge t \in s\}$
Presek	$r(R) \cap s(R) = \{t \mid t \in r \vee t \in s\}$
Razlika	$r(R) - s(R) = \{t \mid t \in r \wedge t \notin s\}$
Selekcija	$\sigma_F(r(R)) = \{t \in r \mid F(t)\}$
Projekcija	$X \subseteq R$ $\pi_X(r(R)) = \{t[X] \mid t \in r(R)\}$
Prirodni spoj	$r(R) \bowtie s(S) = \{t \in Tuple(RS) \mid t[R] \in r \wedge t[S] \in s\}$
Dekartov proizvod	$R \cap S = \emptyset$ $r(R) \times s(S) = \{t \in Tuple(RS) \mid t[R] \in r \wedge t[S] \in s\}$
Theta spajanje	$r(R) \bowtie_\theta F s(S) = \sigma_F(r \times s)$
Šema relacije	$N(R, O)$
Pojava nad šemom relacije	$r(R)$

Relaciona šema baze podataka	$(S, I)$
Relaciona baza podataka	$s: S \rightarrow \{r_i \mid i \in \{1, \dots, n\}\}, (\forall i) s(R_i, O_i) = r_i$
Specifikacija domena	$D(id(D), Predef)$
Ograničenje domena	$id(D) = (Tip, Dužina, Uslov)$
Specifikacija obeležja šeme relacije	$(id(N, A), Predef)$
Ograničenje vrednosti obeležja	$id(N, A) = (Domen, Null)$
Ograničenje torke	$id(N) = id(R) = (\{id(N, A) \mid A \in R\}, Uslov)$
Svojstvo jedinstvenosti	$(\forall u, v \in r(R))(u[X] = v[X] \Rightarrow u = v)$ $X \rightarrow R(X \rightarrow R \in F^+)$
Svojstvo minimalnosti	$(\forall Y \subset X)(\neg 1^0)$ $\neg(\exists X' \subset X)(X' \rightarrow R \in F^+)$
Ključ šeme relacije	$N(R, K)$
Ograničenje ključa (integritet entiteta)	$Key(N, X)$ $(\forall K_i \in K)(\forall A \in K_i)(Null(N, A) = \perp)$ $K \neq \emptyset$
Ograničenje jedinstvenosti	$Unique(N, X)$ $(\forall u, v \in r(R))((\forall A \in X)(u[A] \neq \omega \wedge v[A] \neq \omega) \Rightarrow (u[X] = v[X] \Rightarrow u = v))$ $Uniq = \{Unique(N, X) \mid X \subseteq R\}$
Zavisnost sadržavanja	$X = (A_1, \dots, A_n), (\forall l \in \{1, \dots, n\})(A_l \in R_i),$ $Y = (B_1, \dots, B_n), (\forall l \in \{1, \dots, n\})(B_l \in R_j),$ $(\forall l \in \{1, \dots, n\})(dom(A_l) \subseteq dom(B_l))$ $N_i[X] \subseteq N_j[Y]$ $(\forall u \in r)(\exists v \in s)(\forall l \in \{1, \dots, n\})(u[A_l] = \omega \vee u[A_l] = v[B_l])$
Ograničenje referencijalnog integriteta	$N_i[X] \subseteq N_j[Y]$ $Key(N_j, Y)$
Funkcionalna zavisnost	$f: X \rightarrow Y$ $(\forall u, v \in r(R))(u[X] = v[X] \Rightarrow u[Y] = v[Y])$
Trivijalna funkcionalna zavisnost	$X \rightarrow Y, Y \subseteq X$
Logička posledica	$F  = f$ $(\forall r \in SAT(U))(r  = F \Rightarrow r  = f)$ $F_1  = F_2$ $(\forall f \in F_2)(F_1  = f)$
Ekvivalentnost skupova funkcionalnih zavisnosti	$(F_1 \equiv F_2) \Leftrightarrow (F_1  = F_2 \wedge F_2  = F_1) \Leftrightarrow F_1^+ = F_2^+$
Zatvarač skupa funkcionalnih zavisnosti	$F^+ = \{f \mid F  = f\}$ $F \subseteq F^+$
Refleksivnost	$Y \subseteq X \mid - X \rightarrow Y$
Proširenje	$X \rightarrow Y, W \subseteq V \mid - XV \rightarrow YW$
Pseudotranzitivnost	$X \rightarrow Y, YV \rightarrow Z \mid - XV \rightarrow Z$
Uniranje desnih strana	$X \rightarrow Y, X \rightarrow Z \mid - X \rightarrow YZ$
Dekompozicija desnih strana	$X \rightarrow Y, V \subseteq Y \mid - X \rightarrow V$
Tranzitivnost	$X \rightarrow Y, Y \rightarrow Z \mid - X \rightarrow Z$



Nepotpuna funkcionalna zavisnost	$X \rightarrow Y \in F$ $(\exists X' \subset X)(X' \rightarrow Y \in F^+)$
Tranzitivna funkcionalna zavisnost	$X \rightarrow Z$ $X \rightarrow Y, Y \rightarrow Z, \neg(Y \rightarrow X)$
Zatvarač skupa obeležja	$X_F^+ = \{A \in U \mid X \rightarrow A \in F^+\}$
Algoritam za izračunavanje zatvarača $X_F^+$	$X_0 \leftarrow X$ $(i \geq 0)(X_{i+1} \leftarrow (X_i \cup \{A \in U \mid (\exists V \rightarrow W \in F)(V \subseteq X_i \wedge A \in W)\}))$ $(n \geq 0)(X_{n+1} = X_n \Rightarrow X_F^+ = X_n)$
Generisanje jednog ključa šeme relacije	$X \leftarrow R$ $Red(X): (\forall A \in X)(A \in (X \setminus \{A\})_F^+ \Rightarrow X \leftarrow X \setminus \{A\})$
Generisanje svih alternativnih ključeva	$K \leftarrow \{X\}$ $(\forall X \in K)(\forall V \rightarrow W \in F)(X \cap W \neq \emptyset \Rightarrow X_{newk} \leftarrow (X \setminus W)V)$ $Red(X_{newk}): K \leftarrow K \cup \{Red(X_{newk})\}$
Projekcija skupa funkcionalnih zavisnosti na skup obeležja	$X \subseteq U$ $F _X = \{V \rightarrow W \mid F _V = V \rightarrow W \wedge VW \subseteq X\}$

## Projektovanje šeme relacione baze podataka

Motivacija - šema univerzalne relacije se praktično ne može implementirati, jer:

- univerzalni skup obeležja, skup ograničenja i univerzalna relacija su preglomazni
- nemoguće je pomenute skupove sagledati u celini, niti ima logičkog smisla
- javljaju se anomalije ažuriranja
- javlja se nepotrebna redundansa podataka

## Anomalije ažuriranja

### Anomalije upisa

pri pokušaju upisa podataka o jednom entitetu se moraju znati vrednosti obeležja svih povezanih entiteta.

Potrebno je zadati sve vrednosti obeležja ključa, a neke od njih nisu poznate u trenutku upisa.

### Anomalije brisanja

Brisanjem jedne torke gube se na neželjan način informacije o različitim realnim entitetima, koji su povezani sa entitetom kojeg reprezentuje brisana torka.

### Anomalije modifikacije (redundanse)

Modifikacija vrednosti obeležja istog realnog entiteta obavlja se na više mesta u relaciji, jer se nepotrebno ponavlja na više mesta u relaciji.

Često se za takvu operaciju zahteva prolaz kroz celu relaciju.

## Osnovni projektantski kriterijumi

U teoriji relacionog modela podataka polazi se od pretpostavke da je univerzalna šema relacija inicijalni model realnog sistema.

Projektovanje šeme baze podataka se može vršiti dekomponovanjem (rastavljanjem) šeme univerzalne relacije na više drugih šema relacija baze podataka.

Šema baze podataka treba da zadovolji sledeće kriterijume u odnosu na šemu univerzalne relacije:

1. da predstavlja dekompoziciju šeme univerzalne relacije
  - a. da unija svih obeležja svih šema relacija šeme baze podataka bude jednaka univerzalnom skupu obeležja
2. da se garantuje spojivost bez gubitaka informacija
3. da skup svih ograničenja bude ekvivalentan polaznom skupu ograničenja
4. da se otklone anomalije ažuriranja

Kriterijumi se mogu "oslabiti" po potrebi, jer je često praktično nemoguće ili nepotrebno strogo ih ispoštovati.

## Spojivost bez gubitaka

Prirodni spoj služi za dobijanje odgovora na upite koji traže spajanje baznih relacija.

Bazne relacije su dobijene od hipotetičke univerzalne relacije, dekomponovanjem, primenom operatora projekcije.

Prirodnim spajanjem relacija bi se morala dobiti hipotetička univerzalna relacija da bi važila spojivost bez gubitaka informacije (viška podataka, lažnih torki).

## Pravilo za dekomponovanje i spajanje bez gubitaka za dve šeme relacije:

Pri projektovanju šeme baze podataka, polaznu šemu relacije treba dekomponovati na dve šeme relacije, tako da:

- unija obeležja obe šeme relacija predstavlja skup obeležja polazne šeme
- jedna šema relacije sadrži ključ druge šeme relacije
  - samo tad se relacije nad dekomponovanim šemama mogu spajati
- skup funkcionalnih zavisnosti dekomponovanih šema relacije predstavlja projekciju skupa funkcionalnih zavisnosti početne šeme na obeležja odgovarajuće dekomponovane šeme relacije

## Spoj sa gubicima

$$r(U), X_1 \subseteq U, X_2 \subseteq U, X_1 X_2 = U$$

$$\pi_{X_1}(r) \bowtie \pi_{X_2}(r) \neq r$$

$$r \subseteq \pi_{X_1}(r) \bowtie \pi_{X_2}(r)$$

$$(\forall t \in r)(t \in \pi_{X_1}(r) \bowtie \pi_{X_2}(r))$$

$$t \in r \Rightarrow t[X_1] \in \pi_{X_1}(r) \wedge t[X_2] \in \pi_{X_2}(r) \Rightarrow t \in \pi_{X_1}(r) \bowtie \pi_{X_2}(r)$$

## Spoj bez gubitaka

$$r = \pi_{X_1}(r) \bowtie \pi_{X_2}(r) \Leftarrow X_1 \cap X_2 \rightarrow U \in F^+$$

$$r(U), X \rightarrow Y, r \models X \rightarrow Y \Rightarrow \pi_{XY}(r) \bowtie \pi_{X(U \setminus Y)}(r) = r$$

$$\pi_{XY}(r) \bowtie \pi_{X(U \setminus Y)}(r) \subseteq r$$

### Izvođenje

$$t \in \pi_{XY}(r) \bowtie \pi_{X(U \setminus Y)}(r) \Rightarrow t[XY] \in \pi_{XY}(r) \wedge t[X(U \setminus Y)] \in \pi_{X(U \setminus Y)}(r)$$

$$(\exists t_1 \in r)(t_1[XY] = t[XY]) \wedge (\exists t_2 \in r)(t_2[X(U \setminus Y)] = t[X(U \setminus Y)])$$

$$t_1 \in r \wedge t_1[XY] = t[XY] \wedge t_2 \in r \wedge t_2[X(U \setminus Y)] = t[X(U \setminus Y)]$$

$$r \models X \rightarrow Y$$

$$\begin{array}{lcl} & t_1[Y] = t[Y] & \\ t_1[X] = t[X] & \Rightarrow & t_2[X] = t[X] \\ t_2[X] = t[X] & \Rightarrow & t_2[Y] = t[Y] \\ & t_2[U \setminus Y] = t[U \setminus Y] & \end{array} \Rightarrow t = t_2 \Rightarrow t_2 \in r \Rightarrow t \in r$$

### Definicija

$$\pi_{XY}(r) \bowtie \pi_{X(U \setminus Y)}(r) = r \Rightarrow r \models X \rightarrow Y$$

## Višeznačna zavisnost

### Definicija

$$X \twoheadrightarrow Y; X, Y \subseteq U$$

$$(\forall u, v \in r)(u[X] = v[X]) \Rightarrow (\exists t \in r)(u[XY] = t[XY] \wedge v[X(U \setminus Y)] = t[X(U \setminus Y)])$$

### Neispunjena višeznačna zavisnost

r	A	B	C
	a1	b1	c1
	a1	b2	c2

Ne postoji torka

a1	b1	c2
----	----	----

$$A \twoheadrightarrow B ? \perp$$

### Ispunjena višeznačna zavisnost

r	A	B	C
	a1	b1	c1
	a1	b1	c2
	a1	b2	c1
	a1	b2	c2

$$A \twoheadrightarrow B ? \top$$

## Višeznačna zavisnost i spojivost bez gubitaka

$$r \models X \twoheadrightarrow Y \Leftrightarrow \pi_{XY}(r) \bowtie \pi_{X(U \setminus Y)}(r) = r$$

## Pravila izvođenja za funkcionalne zavisnosti i višeznačne zavisnosti

Funkcionalne zavisnosti (Armstrongova pravila):

$$Y \subseteq X \vdash X \rightarrow Y$$

$$X \rightarrow Y, W \subseteq V \vdash XV \rightarrow YW$$

$$X \rightarrow Y, YV \rightarrow Z \vdash XV \rightarrow Z$$

Višeznačne zavisnosti:

$$X \twoheadrightarrow Y \vdash X \twoheadrightarrow U \setminus (XY)$$

$$X \twoheadrightarrow Y, W \subseteq V \vdash XV \twoheadrightarrow YW$$

$$X \twoheadrightarrow Y, Y \twoheadrightarrow Z \vdash X \twoheadrightarrow Z \setminus Y, X \twoheadrightarrow Y \setminus Z, X \twoheadrightarrow Z \cap Y$$

$$X \rightarrow Y \vdash X \twoheadrightarrow Y$$

$$X \twoheadrightarrow Y, Y \rightarrow Z \vdash X \rightarrow Z \setminus Y$$

## Dokaz odnosa višeznačne zavisnosti i spojivosti bez gubitaka

$$r \models X \twoheadrightarrow Y \Leftrightarrow \pi_{XY}(r) \bowtie \pi_{X(U \setminus Y)}(r) = r$$

### Dokaz implikacije s leva na desno

$$r \models X \twoheadrightarrow Y \Rightarrow \pi_{XY}(r) \bowtie \pi_{X(U \setminus Y)}(r) = r$$

sledi iz definicija višeznačne zavisnosti, prirodnog spoja i projekcije

$$(\forall u, v \in r)(u[X] = v[X]) \Rightarrow (\exists t \in r)(u[XY] = t[XY] \wedge v[X(U \setminus Y)] = t[X(U \setminus Y)])$$

$$t \in \pi_{XY}(r) \bowtie \pi_{X(U \setminus Y)}(r) \Rightarrow t[XY] \in \pi_{XY}(r) \wedge t[X(U \setminus Y)] \in \pi_{X(U \setminus Y)}(r)$$

$$(\exists t_1 \in r)(t_1[XY] = t[XY]) \wedge (\exists t_2 \in r)(t_2[X(U \setminus Y)] = t[X(U \setminus Y)])$$

ili

$$(\forall u, v \in r)(u[X] = v[X]) \Rightarrow u[XY] \in \pi_{XY}(r) \wedge u[X(U \setminus Y)] \in \pi_{X(U \setminus Y)}(r) \wedge v[XY] \in \pi_{XY}(r) \wedge v[X(U \setminus Y)] \in \pi_{X(U \setminus Y)}(r)$$

$$(\forall u, v \in r)(u[X] = v[X]) \Rightarrow (\exists t \in r)(\boxed{u[XY] = t[XY]} \wedge \boxed{v[X(U \setminus Y)] = t[X(U \setminus Y)]})$$

$$t \in \pi_{XY}(r) \bowtie \pi_{X(U \setminus Y)}(r) \Rightarrow \boxed{t[XY] \in \pi_{XY}(r)} \wedge \boxed{t[X(U \setminus Y)] \in \pi_{X(U \setminus Y)}(r)}$$

$$(\exists t_1 \in r)(t_1[XY] = t[XY]) \wedge (\exists t_2 \in r)(t_2[X(U \setminus Y)] = t[X(U \setminus Y)])$$

ili

$$(\forall u, v \in r)(u[X] = v[X]) \Rightarrow \boxed{u[XY] \in \pi_{XY}(r)} \wedge \boxed{u[X(U \setminus Y)] \in \pi_{X(U \setminus Y)}(r)} \wedge \boxed{v[XY] \in \pi_{XY}(r)} \wedge \boxed{v[X(U \setminus Y)] \in \pi_{X(U \setminus Y)}(r)}$$

$$\boxed{X \twoheadrightarrow Y \vdash X \twoheadrightarrow U \setminus (XY)}$$

### Dokaz implikacije s desna na levo

$$r \models X \twoheadrightarrow Y \Leftrightarrow \pi_{XY}(r) \bowtie \pi_{X(U \setminus Y)}(r) = r$$

sledi iz definicije spoja bez gubitaka i pravila izvođenja

$$\pi_{XY}(r) \bowtie \pi_{X(U \setminus Y)}(r) = r \Rightarrow r \models X \rightarrow Y$$

$$X \rightarrow Y \vdash X \twoheadrightarrow Y$$

### Zavisnost spoja

$$X \twoheadrightarrow Y | U \setminus XY$$

$$\bowtie (XY, X(U \setminus Y)) \Leftrightarrow \pi_{XY}(r) \bowtie \pi_{X(U \setminus Y)}(r) = r$$

intenzioni (viši apstraktni) nivo  $\Leftrightarrow$  ekstenzioni (niže apstraktni) nivo

Zavisnost spoja (potpuna zavisnost) je generalizacija višeznačne zavisnosti.

Važi ekvivalencija:

$$F | = \bowtie (R_1, R_2) \Leftrightarrow F | = R_1 \cap R_2 \rightarrow R_1 \setminus R_2 \vee R_1 \cap R_2 \rightarrow R_2 \setminus R_1$$

### Definicija

$$\bowtie (X_1, \dots, X_k), X_i \subseteq U, \bigcup_{i=1}^k X_i = U$$

$$(\forall t_1, \dots, t_k \in r)(\forall i, j \in \{1, \dots, k\})(t_i[X_i \cap X_j] = t_j[X_i \cap X_j]) \Rightarrow (\exists t \in r)(\forall i \in \{1, \dots, k\})(t[X_i] = t_i[X_i])$$

r	A	B	C	D
	a1	b1	c1	d1
	a2	b1	c2	d2
	a3	b2	c2	d3
	a1	b1	c2	d3
	...			

$$\bowtie (X_i, i = \{1, \dots, k\}) \Leftrightarrow \pi_{X_1}(r) \bowtie \dots \bowtie \pi_{X_i}(r) \bowtie \dots \bowtie \pi_{X_k}(r) = r$$

## Očuvanje skupa ograničenja

Očuvanje spojivosti bez gubitaka informacija ne određuje očuvanost polaznog skupa funkcionalnih zavisnosti, kao ni obrnuto, ali najbolje bi bilo imati očuvanost oba.

## Formalne matematičke definicije

Dekompozicija šeme	$(\forall N_i \in S)(R_i \neq \emptyset) \wedge \bigcup_{N_i \in S} (R_i) = U$
Ekvivalentnost skupa svih ograničenja sa polaznim skupom ograničenja	$\bigcup_{N_i \in S} (O_i) \cup I \equiv OGR$
Spojivost bez gubitaka	$r(U, OGR) = \bowtie_{N_i \in S} (r_i(R_i))$
Pravilo za dekomponovanje i spajanje bez gubitaka	$(U, F)(R_1, F_1)(R_2, F_2)$ $R_2 \cup R_1 = U$ $K_1 \subseteq R_2 \cap R_1 \vee K_2 \subseteq R_2 \cap R_1$ $F_1 = F _{R_1} \wedge F_2 = F _{R_2}$ $F  = \bowtie (R_1, R_2) \Leftrightarrow F  = R_1 \cap R_2 \rightarrow R_1 \setminus R_2 \vee R_1 \cap R_2 \rightarrow R_2 \setminus R_1$ $r = \pi_{R_1}(r) \bowtie \pi_{R_2}(r) \Leftarrow R_1 \cap R_2 \rightarrow U \in F^+$
Višeznačna zavisnost	$X \twoheadrightarrow Y; X, Y \subseteq U$ $(\forall u, v \in r)(u[X] = v[X]) \Rightarrow (\exists t \in r)(u[XY] = t[XY] \wedge v[X(U \setminus Y)] = t[X(U \setminus Y)])$
Višeznačna zavisnost i spojivost bez gubitaka	$r \models X \twoheadrightarrow Y \Leftrightarrow \pi_{XY}(r) \bowtie \pi_{X(U \setminus Y)}(r) = r$
Zavisnost spoja	$\bowtie (X_1, \dots, X_k), X_i \subseteq U, \bigcup_{i=1} X_i = U$ $(\forall t_1, \dots, t_k \in r)(\forall i, j \in \{1, \dots, k\})(t_i[X_i \cap X_j] = t_j[X_i \cap X_j])$ $\Rightarrow (\exists t \in r)(\forall i \in \{1, \dots, k\})(t[X_i] = t_i[X_i])$ $\bowtie (X_i, i = \{1, \dots, k\}) \Leftrightarrow \pi_{X_1}(r) \bowtie \dots \bowtie \pi_{X_i}(r) \bowtie \dots \bowtie \pi_{X_k}(r) = r$

## Normalne forme i normalizacije

Moguće je izbegavanje ili, u idealnom slučaju, potpuno uklanjanje anomalija ažuriranja.

Šema baze podataka treba da zadovolji kriterijum odgovarajuće normalne forme.

Postoji 7 normalnih formi.

Ukoliko su sve šeme relacije šeme baze podataka u određenoj normalnoj formi, i šema baze podataka je u toj normalnoj formi.

## 1NF

Šema relacija je u 1NF ako:

- njen skup obeležja sadrži samo elementarna obeležja
- za svaku pojavu relacije nad šemom važi da su sve vrednosti svih obeležja iz njenog skupa atomarne, odnosno da ne predstavljaju niz ili skup drugih vrednosti iz domena obeležja

## 2NF

Šema relacija je u 2NF ako:

- je u 1NF
- je svako neprimarno obeležje u potpunoj funkcionalnoj zavisnosti od svakog ključa

## 3NF

Šema relacija je u 3NF ako:

- je u 1NF
- je svako neprimarno obeležje samo u netranzitivnoj funkcionalnoj zavisnosti od svakog ključa
  - odnosno, svaka netrivialna funkcionalna zavisnost bilo kog neprimarnog obeležja mora sadržati ključ sa leve strane (Odnos uslova normalnih formi)

## BCNF

Šema relacija je u BCNF ako:

- je u 1NF
- je svaka netrivialna funkcionalna zavisnost bilo kog atributa mora sadržati ključ sa leve strane

## 4NF

Šema relacija je u 4NF ako:

- je u 1NF
- je svaka netrivialna višeznačna zavisnost logička posledica funkcionalne zavisnosti

### Kontra primer

$N(\{A, B, C\}, \{A \twoheadrightarrow B\}), K = \{ABC\}$

$$A \twoheadrightarrow B|C \Leftrightarrow (AB, AC) \Leftrightarrow \pi_{AB}(r(N)) \bowtie \pi_{AC}(r(N)) = r(N)$$

r	A	B	C
	a1	b1	c1
	a1	b2	c2
	a1	b2	c1
	a1	b1	c2

## BCNF

Anomalija upisa - mora se uneti svako (primarno) obeležje

Anomalija modifikacije - mnogo ponavljanja

Anomalija brisanja - ako se briše jedna brišu se još neke torke

Rešenje - dekomponovati na šeme  $(\{A,B\},\{\})$  i  $(\{A,C\},\{\})$

## 5NF, PJNF (Project-Join NF)

Šema relacija je u 5NF ako:

- je u 1NF
- je svaka netrivialna zavisnost spoja logička posledica funkcionalne zavisnosti

### Kontra primer

$N(\{A, B, C, D\}, \{\bowtie (AB, BC, CD)\}), K = \{ABCD\}$

$$\bowtie (X_1, \dots, X_n) \Leftrightarrow \pi_{X_1}(r) \bowtie \dots \bowtie \pi_{X_n}(r) = r \Leftrightarrow \bigwedge_{i=1}^n \pi_{X_i}(r) = r$$

r	A	B	C	D
	a1	b1	c1	d1
	a2	b1	c2	d2
	a3	b2	c2	d3
	a1	b1	c2	d3
			...	

Nema netrivialnih višeznačnih zavisnosti, stoga važi 4NF.

Anomalija upisa - mora se uneti svako (primarno) obeležje

## DKNF (Domain-Key NF)

Šema relacija je u DKNF ako:

- je u 1NF
- se u skupu ograničenja nalaze samo ograničenja ključa i domena

### Kontra primer

$Stavka(\{DOCID, RBR, MATID, JCEN, KOL, CENA\}, \{DOCID + RBR, DOCID + MATID\})$

$CENA = KOL * JCEN \Rightarrow CENA \leftarrow KOL + JCEN - \text{ograničenje torke, ne zadovoljava DKNF}$

## Odnos uslova normalnih formi

1NF je potreban uslov za sve više normalne forme.

$$3NF \Rightarrow 2NF$$



Dokaz kontrapozicijom:

$$\neg 2NF \Rightarrow (\exists A \in R \setminus Kpr)(\exists X \in K)(\exists Y \subseteq R\{A\})(F| = Y \rightarrow A) \Rightarrow X \rightarrow A \wedge Y \rightarrow A \wedge \neg(Y \rightarrow A) \Rightarrow \neg 3NF$$

$$BCNF \Rightarrow 3NF$$

$$(\forall A \in R)(\forall Y \subseteq R\{A\})(F| = Y \rightarrow A \Rightarrow (\exists X \in K)(X \subseteq Y)) \Rightarrow$$

$$\Rightarrow (\forall A \in R \setminus Kpr)(\forall Y \subseteq R\{A\})(F| = Y \rightarrow A \Rightarrow (\exists X \in K)(X \subseteq Y))$$

Dokaz, obratom po kontrapoziciji, s leva na desno:

$$(\forall A \in R \setminus Kpr)(\forall Y \subseteq R\{A\})(F| = Y \rightarrow A \wedge (\forall X \in K)(X \not\subseteq Y)) \Rightarrow$$

$$\Rightarrow (\forall X \in K)(F| = X \rightarrow A \wedge F| = Y \rightarrow A \wedge F| \neq Y \rightarrow A) \Rightarrow \neg 3NF$$

Dokaz, obratom po kontrapoziciji, sa desna na levo:

$$(\forall A \in R \setminus Kpr)(\forall Y \subseteq R\{A\})(F| = Y \rightarrow A \wedge F| \neq Y \rightarrow X) \Rightarrow$$

$$\Rightarrow (\forall A \in R \setminus Kpr)(\forall Y \subseteq R\{A\})(F| = Y \rightarrow A \wedge (\forall X \in K)(X \not\subseteq Y)) \Rightarrow \neg 3NF$$

Alternativna (ekvivalentna) formulacija uslova 3NF:

- 1NF
- svaka netrivialna funkcionalna zavisnost bilo kog neprimarnog obeležja mora sadržati ključ sa leve strane

Zabrana \ Normalna forma	1NF	2NF	3NF	BCNF
Nizovi ili skupovi vrednosti obeležja, umesto jedne vrednosti iz domena	-	-	-	-
Nepotpune funkcionalne zavisnosti neprimarnih obeležja od ključa iz F <sup>+</sup>	+	-	-	-
Netrivialne funkcionalne zavisnosti neprimarnih obeležja koje ne sadrže ključ s leve strane - tranzitivne funkcionalne zavisnosti neprimarnih obeležja iz F <sup>+</sup>	+	+	-	-
Netrivialne funkcionalne zavisnosti F <sup>+</sup> koje ne sadrže ključ sa leve strane	+	+	+	-

## Normalizacija

Postupak projektovanja takvog skupa šema relacija:

- kod kojeg su u celosti ili delimično ostvareni osnovni projektantski kriterijumi
- zadovoljena je odgovarajuća normalna forma

Dve metode normalizacije:

- metoda dekompozicije
- metoda sinteze

Kada je broj obeležja mali:

- primena normalizacije se čini nepotrebnom

- iste rezultate bi dao i intuitivni ekspertski pristup.

Kada je broj obeležja veliki:

- manuelna primena normalizacije je teška i podložna greškama
- ekspertski pristup daje teško predvidive rezultate

Rešenje je primena CASE / MDSD alata:

- projektovanje formalnih specifikacija konceptualne šeme baze podataka
- niz transformacija formalnih specifikacija za dolazak do
  - implementacione šeme baze podataka
  - interne šeme baze podataka, opisa šeme baze podataka u jeziku ciljno SUBP-a

## Formalne matematičke definicije

2NF	$Kpr = \bigcup_{X \in K} (K)$ $(\forall A \in R \setminus Kpr)(\forall X \in K)(\forall Y \subset K)(F  \neq Y \rightarrow A)$
3NF	$Kpr = \bigcup_{X \in K} (K)$ $(\forall A \in R \setminus Kpr)(\forall X \in K)(\forall Y \subseteq R \setminus \{A\})(F  = Y \rightarrow A \Rightarrow F  = Y \rightarrow X)$ <p style="text-align: center;">ili</p> $(\forall A \in R \setminus Kpr)(\forall Y \subseteq R \setminus \{A\})(F  = Y \rightarrow A \Rightarrow (\exists X \in K)(X \subseteq Y))$
BCNF	$(\forall A \in R)(\forall Y \subseteq R \setminus \{A\})(F  = Y \rightarrow A \Rightarrow (\exists X \in K)(X \subseteq Y))$
4NF	$(\forall X \twoheadrightarrow Y \in F)(\exists X \rightarrow Y \in F^+)$
5NF	$(\forall \bowtie (X, Y) \in F)(\exists X \rightarrow Y \in F^+)$

## Metoda dekompozicije

Da bi se šema relacije dekompovala na dve podšeme, potrebno je odabrati funkcionalnu zavisnost iz skupa funkcionalnih zavisnosti te šeme po kojoj treba dekomponovati originalnu šemu.

Po odabranoj funkcionalnoj zavisnosti nastaju dve podšeme, pri čemu prva sadrži sva obeležja opisana u funkcionalnoj zavisnosti, a druga uniju obeležja sa leve strane funkcionalne zavisnosti i sva ostala obeležja osim obeležja na desnoj strani.

$$(U, F), X \rightarrow Y \in F, (R_1, F_1) = (XY, F|_{XY}), (R_2, F_2) = (X(U \setminus Y), F|_{X(U \setminus Y)})$$

Postupak obezbeđuje spoj bez gubitaka, jer:

- $X$  je sigurno superključ u  $(R_1, F_1)$  i prenet je u  $R_2$
- ključ od  $(R_1, F_1)$  sigurno je prenet u  $R_2$

## Kriterijumi izbora funkcionalne zavisnosti

Prioritet	Kriterijum	Posledica	Matematički zapisan uslov
1	Netrivijalna funkcionalna zavisnost, takva da ne obuhvata ceo skup obeležja originalne šeme relacije	Obezbeđenje progressa algoritma.	$(Y \not\subseteq X) \wedge (XY \subset U)$
2	Funkcionalna zavisnost na kraju mogućih "lanaca izvođenja"	Obezbeđenje uslova za očuvanje polaznog skupa funkcionalnih zavisnosti	$(\forall W \rightarrow V \in F)((X)_F^+ \neq (W)_F^+ \Rightarrow X \rightarrow W \notin F^+)$
3	Funkcionalna zavisnost takva da je unija skupova funkcionalnih zavisnosti podšema ekvivalentna polaznom skupu	Očuvanje ekvivalntnosti s polaznim skupo funkcionalnih zavisnosti	$F \equiv (F _{XY} \cup F _{X(U \setminus Y)})$

## Strategije izbora funkcionalne zavisnosti:

1. idealna strategija - ispunjeni svi kriterijumi
2. Strategija očuvanja polaznog skupa funkcionalnih zavisnosti - 1. i 3. kriterijumi ispunjeni
3. Strategija obezbeđenja progressa algoritma - 1. i 2. kriterijumi ispunjeni

Ekvivalentni ključevi su ključevi čiji zatvarači po originalnom skupu funkcionalnih zavisnosti su jednaki.

Nakon dekomponovanja sledi objedinjavanje šema relacija sa ekvivalentnim ključevima.

- može dovesti do degradacije postignute normalne forme
- može dovesti do nadoknade prethodno izgubljene funkcionalne zavisnosti

Na kraju se definišu međurelaciona ograničenja.

Anomalije ažuriranja su uglavnom izbegnute.

## Metoda sinteze

Motivacija:

- automatsko generisanje skupa šema relacija i skupa međurelacionih ograničenja
  - polazeći od univerzalnog skupa obeležja i funkcionalnih zavisnosti
  - uklanjanjem suvišnih funkcionalnih zavisnosti i suvišnih obeležja iz levih strana funkcionalnih zavisnosti
- zadovoljenje uslova 3NF
- očuvanje polaznog skupa funkcionalnih zavisnosti
- očuvanje spojivosti bez gubitaka
  - na nivou celokupne šeme baze podataka ili na nivou podšema

## Koraci

### Formiranje kanoničkog pokrivača

Kanonički pokrivač  $kp(F)$  datog skupa funkcionalnih zavisnosti  $F$ :

- je ekvivalentan sa polaznim skupom  $F$
- sve desne strane funkcionalnih zavisnosti iz kanoničkog pokrivača sadrže tačno jedno obeležje
- sve funkcionalne zavisnosti iz kanoničkog pokrivača su potpune (levo redukovane)
- ne sadrži redundantne funkcionalne zavisnosti

### Dekompozicija desnih strana skupa funkcionalnih zavisnosti

Inicijalni skup transformiše u ekvivalentni oblik, tako da svaka funkcionalna zavisnost sa desne strane sadrži samo jedno obeležje.

Motivacija - može doći do kasnijeg različitog tretmana levih strana funkcionalnih zavisnosti.

### Redukcija levih strana funkcionalnih zavisnosti

Inicijalni skup funkcionalnih zavisnosti se transformiše u ekvivalentni oblik, u kom su uklonjena logički suvišna obeležja iz levih strana svake funkcionalne zavisnosti, odnosno u kom su sve funkcionalne zavisnosti potpune.

Motivacija - eliminacija faktora narušavanja 2NF.

### Eliminacija redundantnih funkcionalnih zavisnosti

Redundantne (suvišne) funkcionalne zavisnosti su one koje su logičke posledica ostatka skupa funkcionalnih zavisnosti - tranzitivne, pseudotranzitivne ili trivijalne.

Motivacija - eliminacija faktora narušavanja 3NF.

### Transformacija kanoničkog pokrivača

Motivacija:

- pronalaženje budućih ekvivalentnih ključeva
- stvaranje osnova da se, umesto više šema relacija, formira samo jedna šema relacije s ekvivalentnim ključevima
- u opštem slučaju, za isti ulazni skup funkcionalnih zavisnosti, može se generisati više različitih kanoničkih pokrivača
  - nije svaki jednako "pogodan" za sintezu skupa šema relacija

### Particioniranje kanoničkog pokrivača

Podela kanoničkog pokrivača skupa funkcionalnih zavisnosti na podskupove sa istim levim stranama.

### Određivanje ekvivalentnih levih strana

Za sve podskupove funkcionalnih zavisnosti čije leve strane su iste se računa zatvarač leve strane.

Uniraju se podskupovi čiji zatvarači levih strana po originalnom skupu funkcionalnih zavisnosti su jednaki, odnosno čije leve strane su ekvivalentne.

Moguća je reukrzivna primena postupka uniranja grupa, dokle god postoje parovi sa ekvivalentnim levim stranama.

### ***Uklanjanje tranzitivnih zavisnosti***

Moguća je modifikacija kanoničkog pokrivača skupa funkcionalnih zavisnosti, pomoću formiranja skupa funkcionalnih zavisnosti ekvivalentnih levih strana  $J$ .

Skup  $J$  je inicijalno prazan, pa se puni funkcionalnim zavisnostima čije strane predstavljaju različita obeležja koja su ekvivalentna. Ove funkcionalne zavisnosti se ne moraju nalaziti u originalno dobijenom kanoničkom pokrivaču.

Nakon pravljenja skupa  $J$ , iz podskupova u  $G$  se izbacuju funkcionalne zavisnosti koje se nalaze u skupu  $J$ .

Formira se novi skup funkcionalnih zavisnosti  $M$  koji je unija podskupova iz  $G$  i skupa  $J$ .

Iz novog skupa se uklanjaju logički suvišne funkcionalne zavisnosti koje pripadaju podskupovima iz  $G$ . Zbog funkcionalnih zavisnosti u  $J$ , neke druge mogu postati suvišne.

### ***Rekonstrukcija particije kanoničkog pokrivača***

Svaka funkcionalna zavisnost iz  $J$  se vraća u odgovarajuću grupu.

### ***Formiranje relacije šeme baze podataka***

#### ***Formiranje skupa šema relacija***

Svaka grupa daje jednu šemu relacije u finalnom skupu šema relacija.

Obeležja iz skupa obeležja jedne šeme relacije se pojavljuju u odgovarajućem skupu funkcionalnih zavisnosti.

Skup funkcionalnih zavisnosti jedne šeme relacije predstavlja zapravo skup od kog se napravila šema relacije.

Skup ključeva predstavlja skup levih strana svih funkcionalnih zavisnosti iz skupa od kog se napravila šema relacije.

Nazive šema relacije ne može generisati algoritam, već ih zadaje projektant baze podataka.

#### ***Formiranje ograničenja stranog ključa***

Na osnovu formiranog skupa šema relacija, ukoliko za 2 šeme relacija važi:

- da je skup obeležja prve šeme pravi podskup zatvarača obeležja druge po originalnom skupu funkcionalnih zavisnosti i
- da je ključ iz prve šeme podskup skupa obeležja druge šeme

formira se ograničenje stranog ključa, gde je prva šema referencirajuća, a druga referencirana.

### ***Očuvanje spoja bez gubitaka***

Spojivost bez gubitaka je očuvana ako postoji šema relaciju u skupu šema relacija sa ključem šeme univerzalne relacije.

U tom slučaju skup šema relacija predstavlja dekompoziciju šeme univerzalne relacije sa spojem bez gubitaka informacija.

Ukoliko ne postoji, treba dodati u skup šema relacija šemu relacije sa skupom obeležja koji odgovara skupu obeležja jednog, izabranog ključa šeme univerzalne relacije, sa ključem koji odgovara izabranom ključu šeme univerzalne relacije.

Šema baze podataka može zadovoljavati spoj bez gubitaka kada je mala i kada je celu koristi jedan program, što je praktično nerealan slučaj.

Šema baze podataka, po pravilu, ne zadovoljava spoj bez gubitaka kada je velika, što je praktično uvek slučaj, jer:

- ne postoji program koji koristi celu šemu baze podataka za rad nad samom bazom podataka
- ključ šeme univerzalne relacije sastojao bi se iz veoma velikog broja obeležja
- ne postoji način ili opravdanje da se obezbedi održavanje podataka nad šemom relacije koja sadrži ključ šeme univerzalne relacije

### Podšema

Logička struktura obeležja, formirana na osnovu šeme baze podataka.

Služi za realizaciju barem jednog, ili grupe sličnih transakcionih programa.

Podšema praktično mora zadovoljavati uslov spoja bez gubitaka, jer:

- služi za korišćenje baze podataka od strane barem jednog programa
- reprezentuje jedan pogled korisnika na bazu podataka.

### Funkcionalna zavisnost kao posledica ključa

Cilj algoritma sinteze:

- u skup šema relacija, ugraditi sve i samo bitne, redukovane i neredundantne funkcionalne zavisnosti polaznog skupa, putem ključeva
- obezbeđeno očuvanje polaznog skupa funkcionalnih zavisnosti

SUBP može da proverava važenje samo onih funkcionalnih zavisnosti u bazi podataka koje su posledica ključeva šema relacija.

Provera funkcionalnih zavisnosti se svodi na proveru jedinstvenosti vrednosti ključa, najčešće uz pomoć fizičkih struktura "unique" indeksa, kreiranih nad ključevima.

### Formalne matematičke definicije

Kanonički pokrivač	$F \equiv kp(F)$ $(\forall X \rightarrow A \in kp(F))(A \in U)$ $(\forall X \rightarrow A \in kp(F))(\forall X' \subset X)(X' \rightarrow A \notin F^+)$ $\neg(\exists X \rightarrow A \in kp(F))(kp(F)\{X \rightarrow A\} \equiv kp(F))$
Redukcija levih strana funkcionalnih zavisnosti	$(\exists X \rightarrow A \in F)(\exists B \in X)(X\{B\} \rightarrow A \in F^+) \Rightarrow F \leftarrow (F \setminus \{X \rightarrow A\} \cup \{X \setminus \{B\} \rightarrow A\})$
Eliminacija redundantnih funkcionalnih zavisnosti	$(\exists X \rightarrow A \in F)(F \setminus \{X \rightarrow A\} \models X \rightarrow A) \Rightarrow F \leftarrow F \setminus \{X \rightarrow A\}$
Particioniranje kanoničkog pokrivača	$G = \{G(X_i)   i \in \{1, \dots, n\}\}$ $G(X_i) = \{Y \rightarrow A \in kp(F)   Y = X_i\}$ $(\forall i, j \in \{1, \dots, n\})(X_i \neq X_j)$ $(\forall Y \rightarrow A \in kp(F))(\exists G(X_i) \in G)(Y = X_i)$

Određivanje ekvivalentnih levih strana	$(X_i)^+_F = (X_j)^+_F \Leftrightarrow \{X_i \rightarrow X_j, X_j \rightarrow X_i\} \subseteq F^+$ $G(X_i, X_j) = G(X_i) \cup G(X_j)$ $G \leftarrow (G \setminus \{G(X_i), G(X_j)\}) \cup \{G(X_i, X_j)\}$
Uklanjanje tranzitivnih zavisnosti	$J \leftarrow \emptyset$ $J \leftarrow J \cup \{X_i \rightarrow X_j, X_j \rightarrow X_i\}$ $G(\dots, X_i, X_j) \leftarrow G(\dots, X_i, X_j) \setminus (\{X_i \rightarrow A   A \in X_j\} \cup \{X_j \rightarrow A   A \in X_i\})$ $M = \bigcup_{G_X \in G} (G_X) \cup J$ $(\forall X \rightarrow A \in (M \setminus \{X \rightarrow A\})^+) (G_X \leftarrow G_X \setminus \{X \rightarrow A\})$
Rekonstrukcija particije kanoničkog pokrivača	$G(\dots, X_i, X_j) \leftarrow G(\dots, X_i, X_j) \cup \{X_i \rightarrow X_j \in J   X_i \in \{X_{i_1}, \dots, X_{i_n}\}\}$
Formiranje skupa šema relacija	$S = \{N_i(R_i, K_i)   i \in \{1, \dots, n\}\}$
Formiranje ograničenja stranog ključa	$N_i(R_i, K_i)$ $N_j(R_j, K_j)$ $R_i \subset (R_j)^+_F$ $(\exists X_i \in K_i)(X_i \subseteq R_j)$ $N_j[X_i] \subseteq N_i[X_i]$

## Prevođenje ER modela u relacioni model podataka

Mogući postupak projektovanja šeme baze podataka:

- konceptualna šema u ER modelu podataka
- implementaciona šema u relacionom modelu podataka
- relaciona šema baze podataka se može dobiti transformacijom (prevođenje) ER šeme baze podataka

## Regularni tip entiteta

Prevodi se u šemu relacije koja sadrži:

- skup obeležja entiteta kao skup obeležja šeme relacije
- skup ograničenja entiteta kao skup ograničenja šeme relacije, prvenstveno ključevi

## Binarni tip poveznika i binarni gerund

### Maksimalni kardinaliteti M:N

Tip poveznika postaje šema relacije koja sadrži ključeve povezanih entiteta i obeležja poveznika.

Ključ te šeme je uglavnom podskup unije ključeva povezanih entiteta.

Važe ograničenja referencijalnih integriteta - obeležja koja je nasledila nova šema relacija od entiteta su strani ključevi od šema relacija napravljenih od entiteta.

Ukoliko je minimalni kardinalitet = 1 jednog entiteta prema drugo, prvi entitet je egzistencijalno zavisian od drugog - važi zavisnost sadržavanja, ograničenje inverznog referencijalnog integriteta

### **Maksimalni kardinaliteti N:1**

Tip poveznika se predstavlja putem prostiranja primarnog ključa.

Prostiranje primarnog ključa - skup obeležja šeme relacije, nastale od tipa entiteta na "1" strani se proširuje obeležjima primarnog ključa tipa entiteta na "N" strani, odnosno postoji strani ključ (ograničenje referencijalnog integriteta) u šemi relacije na "1" strani.

Minimalni kardinalitet na "1" strani definiše da li je obeležje stranog ključa nullable, odnosno da li može da sadrži nula vrednost.

Ako je minimalni kardinalitet na "1" strani jednak 0, obeležja je nullable, u suprotnom nije, jer je egzistencijalno zavisian prvi entitet od drugog entiteta.

Ako je minimalni kardinalitet na "N" strani jednak 1, taj entitet je egzistencijalno zavisian od drugog, i važi zavisnost sadržavanja, odnosno ograničenje inverznog referencijalnog integriteta.

### **Maksimalni kardinaliteti 1:1**

Prevođenje se sprovodi u zavisnosti od vrednosti minimalnih kardinaliteta.

#### ***Minimalni kardinaliteti jednaki nuli***

Analogno slučaju kardinaliteta M:N ili N:1.

**M:N**

Tip poveznika sadrži samo ključeve povezanih entiteta i svoja obeležja.

Važi ograničenje referencijalnog integriteta, strani ključevi u tipu poveznika.

**N:1**

Skup obeležja jedne šeme relacije se proširuje ključem druge šeme relacije.

Važi ograničenje referencijalnog integriteta, strani ključ se nalazi u proširenoj šemi relacije.

Dodato obeležje je nullable i jedinstveno.

#### ***Minimalni kardinaliteti različiti***

Analogno slučaju kardinaliteta N:1.

Šemi relacije koja se nalazi na strani gde je minimalni kardinalitet jednak 1, dodaje ključ druge šeme relacije i sadrži strani ključ.

#### ***Minimalni kardinaliteti jednaki jedinici***

Oba tipa entiteta i tip poveznika se prevode u jednu šemu relacije.

### **Dva tipa poveznika između dva ista tipa entiteta**

Jedan entitet prve klase može biti, posredno ili neposredno, povezan sa jednim entitetom druge klase po više osnova.



Entiteti klasa mogu imati različite uloge u vezi.

Po prethodno navedenim pravilima, dobilo bi se više identičnih šema relacija, sa istim skupom obeležja i istim ograničenjima referencijalnog integriteta.

- To predstavlja koliziju sa pretpostavkom o šemi univerzalne relacije jer:
  - semantiku šeme relacije u relacionom modelu podataka nosi njen skup obeležja, a ne naziv
  - šema relacije baze podataka treba da sadrži takav skup šema relacija od kojih ne postoje dve sa istim skupom obeležja

Prevođenjem u jednu šemu relacije se gubi semantika.

Moguća rešenja:

- uvođenjem novog obeležja - uloge
  - primenjuje se samo za tipove poveznika koji imaju iste maksimalne kardinalitete
  - svi takvi tipovi poveznika između dva tipa entiteta se prevode u jednu šemu relacije
  - skup obeležja te šeme relacije se proširuje novim obeležjem
  - vrednost tog obeležja ukazuje na ulogu entiteta u povezniku
  - moguće je, kada su minimalni kardinaliteti tipova poveznika isti, realizovati već na nivou ER modela podataka
  - važe ograničenja referencijalnog integriteta u šemi relacija nastale od poveznika
  - u slučaju različitih minimalnih kardinaliteta, postoji i novo međurelaciono ograničenje, koje se može ugraditi u samu specifikaciju ER modela podataka
- uvođenje novog obeležja preimenovanjem obeležja ključa barem jednog tipa entiteta
  - mora se primeniti za tipove poveznika sa različitim maksimalnim kardinalitetima, a može i sa istim
  - obeležja stranog ključa, dobijenog propagacijom primarnog ključa, se preimenuju, i domenski su kompatibilni sa originalnim obeležjima
  - uvode se nova obeležja u relacionu šemu baze podataka
  - uobičajeno rešenje
  - podržavaju ga svi ozbiljni CASE alati namenjeni za podršku projektovanja ER šema baze podataka
  - važe ograničenja referencijalnog integriteta u šemi relacije nastaloj od poveznika, i za preimenovana obeležja
  - od minimalnih kardinaliteta zavisi ograničenje inverznog referencijalnog integriteta i nullable svojstvo
  - izbegava se kolizija s pretpostavkom o postojanju šeme univerzalne relacije
  - iskazuje se semantika dva posebna odnosa putem različitih šema relacija

### Rekurzivni tip poveznika

Jedan poveznik povezuje dva različita entiteta iste klase.

Kolizija sa činjenicom da šema relacije ne sme sadržati dva ista obeležja u različitim ulogama.

Tip poveznika se predstavlja šemom relacija koja sadrži ključ tipa entiteta i preimenovani ključ tipa entiteta, kompatibilnog domena. Važe ograničenja referencijalnog integriteta u novoj šemi relacije, nastaloj od tipa poveznika.

## Identifikaciono zavisni tip entiteta

Identifikaciono zavisni, slabi tip entiteta ne poseduje "samostalni" identifikator.

Ključ takvog tipa entiteta se formira korišćenjem ključa nadređenih tipova entiteta.

Postupak formiranja ključa identifikaciono zavisnost tipa entiteta je rekurzivan - nadređeni tip entiteta može ponovo biti identifikaciono zavisan, završava se dolaskom do regularnog tipa entiteta.

Važi ograničenje referencijalnog integriteta, strani ključ u slabom tipu entiteta referencira ista obeležja u nadređenom tipu entiteta.

## IS-A hijerarhija

Potklasa je identifikaciono zavisna od superklase i nasleđuje sva obeležja, uključujući i primarni ključ.

Postoji 4 načina prevođenja IS-A hijerarhije u relacioni model podataka.

### Šema relacije za sve tipove entiteta

Šema superklase sadrži sva svoja obeležja, uključujući i klasifikaciono obeležje.

Šeme potklase sadrže nasleđeni primarni ključ i svoja obeležja.

Sve šeme relacije nastale od potklase imaju ograničenje referencijalnog integriteta, referenciraju šemu relacije superklase.

Za totalne IS-A hijerarhije važi i međurelaciono ograničenje da projekcija roditeljske šeme relacije nad primarnim ključem bude sadržana u uniji projekcija šema potklase nad istim obeležjem.

Za nepresečne IS-A hijerarhije važi i međurelaciono ograničenje da presek nad svake dve projekcije šema potklase nad primarnim ključem bude prazan skup.

### Šema relacije za potklase

Moguće samo kod totalnih IS-A hijerarhija, minimalnih kardinalitet 1.

Sve šeme imaju nasleđena, zajednička obeležja, osim klasifikacionog.

Za nepresečne IS-A hijerarhije važi i međurelaciono ograničenje da presek nad svake dve projekcije šema potklase nad primarnim ključem bude prazan skup.

### Jedna šema relacije za celu IS-A hijerarhiju

Jedna šema sadrži sva obeležja superklase i potklase.

Važi ograničenje torke:

- za nepresečnu IS-A se određuje koja obeležja imaju nula vrednost a koje ne sme da ima, u zavisnosti od klasifikacionog obeležja
- za presečnu IS-A klasifikaciono obeležje mora biti takvo da može da predstavi bilo koju kombinaciju potklase i u zavisnosti od njegove vrednosti određena polja smeju ili ne smeju sadržati null vrednost

## **Jedna šema relacije za izabrane potklase i jedna šema relacije za sve ostale potklase i superklasu**

Kombinacija prvog (šema relacija za sve tipove entiteta) i trećeg pristupa (šema relacije za celu IS-A hijerarhiju).

## **Kategorizacija**

Od svakog tipa u kategorizaciji formira se posebna šema relacije.

Postoje dva načina prevođenja u relacioni model podataka.

### **Prenošenje primarnih ključeva svih kategorija**

Primarni ključevi svih kategorija prenose se kao strani ključevi u zavisni tip entiteta.

Uvodi se specijalno ograničenje torke u zavisnom tipu entiteta u zavisnosti od kardinaliteta i vrednosti klasifikacionog obeležja.

### **Formiranje jednog stranog ključa**

Formira se jedan strani ključ u zavisnom tipu entiteta.

Ako ne postoji, uvodi se novo obeležje koje označava vrstu kategorija.

Njegova vrednost ukazuje, za svaku pojavu zavisnog tipa entiteta, s kojom konkretnom kategorijom je povezana.

Broj obeležja u primarnim ključevima svih kategorija je isti.

Svi primarni ključevi svih kategorija su domenski kompatibilni.

Strani ključ je domenski kompatibilan sa svim primarnim ključevima svih kategorija.

Ograničenje referencijalnog integriteta zavisi od vrednosti klasifikacionog obeležja.

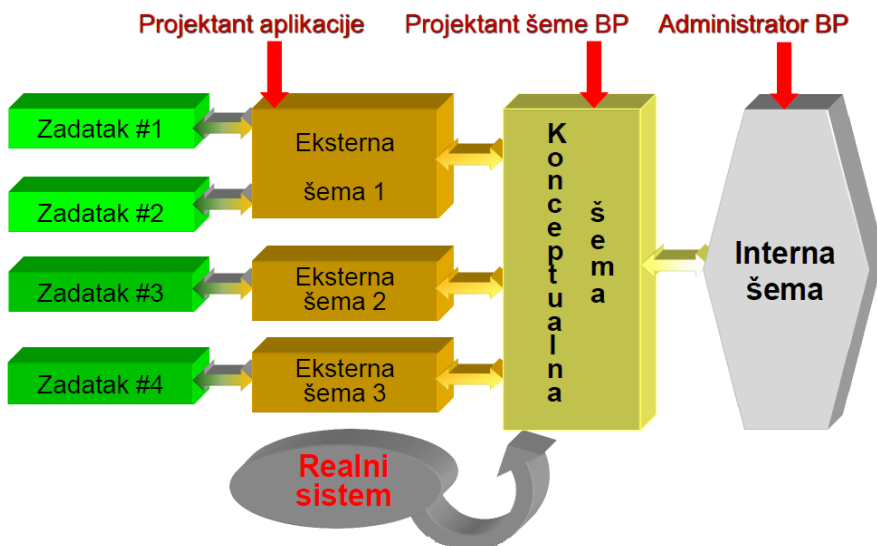
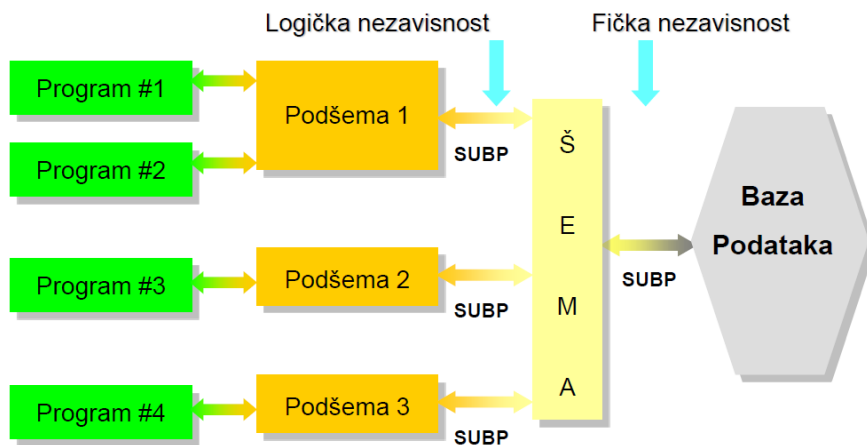
## **N-arni tip poveznika i n-arni gerund**

Ključ tipa poveznika zavisi od maksimalnih kardinaliteta povezanih tipova entiteta.

Važi ograničenje referencijalnog integriteta, tip poveznika referencija primarne ključeve entiteta koje povezuje.

## Metodologija projektovanja

### Višenivovska arhitektura baze podataka



### Konceptualna šema

Centralno mesto u opsi baze podataka.

Model statičke strukture realnog sistema.

Struktura nad skupom upotreba koncepata modela podataka.

Platformski nezavisni model.

Koncept - apstraktna predstava neke klase delova realnog sveta.

Izrada putem odabranog modela podataka.

Zahtevi:

- adaptivnost
  - mogućnost jednostavnih izmena u cilju praćenja razvoja realnog sistema
- amortizacija
  - obezbeđenje nezavisnosti promena prema eksternim šemama i internoj šemi
- usaglašenost
  - obezbeđenje veze prema svakoj eksternoj šemi
  - moguće je generisati pojavu svake eksterne šeme na osnovu sadržaja baze podataka, nad datom konceptualnom šemom
- fizička nezavisnost
  - ne sadrži informacije o fizičkoj (internoj) strukturi baze podataka
  - služi za projektovanje interne šeme (fizičke strukture baze podataka)
- kompletnost
  - obezbeđenje mogućnosti za zadovoljavanje svih predviđenih informacionih zahteva
- platformska nezavisnost
  - nezavisnost samo od izabranog SUBP
  - nezavisnost od modela podataka koji podržava izabrani SUBP i od izabranog SUBP

## Eksterna šema (podšema)

Logička struktura obeležja.

Pogled (viđenje) grupe srodnih korisnika na odgovarajući deo baze podataka i informacionog sistema, iskazana putem konceptualnog modela podataka.

Model strukture podataka nad kojom se izvršava bar jedan od radnih zadataka korisnika, koristeći usluge transakcionih programa informacionog sistema.

## Interna šema

Opis fizičke strukture baze podataka u jeziku SUBP.

U potpunosti prilagođen mogućnostima i karakteristikama odabranog SUBP.

## Platformska nezavisnost šeme baze podataka

### Nezavisnost šeme od modela podataka koji podržava izabrani SUBP i izabranog SUBP

Izbor platformski nezavisnog modela podataka - ER model podataka ili UML meta-model klasa, prevodivi u bilo koji model podataka koji podržavaju konkretni SUBP.

Obuhvata razvoj konceptualne šeme baze podataka, odnosno platformski nezavisnog modela koji je:

- semantički dovoljno bogat
- bliži percepciji i konceptima korisnika informacionog sistema
- nezavisan od bilo kojih implementacionih koncepata

## Nezavisnost šeme samo od izabranog SUBP

Izbor platformski zavisnog modela podataka - najčešće relacioni model podataka.

Obuhvata razvoj implementacione šeme baze podataka koja:

- očuvava semantiku ugrađenu u konceptualnu šemu baze podataka
- iskazana je putem koncepata koje podržava izabrani SUBP

## Implementaciona šema

Slika konceptualne šeme.

Projektuje se transformacijom prethodno isprojektovane konceptualne šeme.

Nezavisna od interne šeme.

Zavisna od modela podataka izabranog SUBP.

Dva aspekta:

- platformski zavisni model
  - s obzirom na referentni platformski nezavisni model izrađen u ER modelu podataka
- referentni platformski nezavisni model
  - s obzirom na internu šemu kao platformski zavisni model

## Pristup razvoju softvera zasnovan na modelima - Model Driven Software Development

Obuhvata izradu:

- platformski nezavisnih modela
  - na osnovu računarski nezavisnih modela.
- platformski zavisnih modela
  - sprovođenjem niza (idealno automatskih) model-model transformacija od platformski nezavisnih.
- softvera (izvršnih softverskih specifikacija)
  - sprovođenjem niza (idealno automatskih) model-kod transformacija od platformski zavisnog modela do programskog koda

## Razvoj šeme baze podataka

Obuhvatu izradu:

- konceptualne šeme baze podataka
  - platformski nezavisnog modela putem ER modela podataka
- implementacione šeme baze podataka
  - platformski nezavisnih/zavisnih modela putem relacionog modela podataka, transformacijom platformski nezavisnih modela
- interne šeme baze podataka
  - platformski zavisnog modela, transformacijom platformskih nezavisnih/zavisnih modela

- programiranje opisa šeme baze podataka u jeziku SUBP
  - softverske specifikacije za implementaciju šeme baze podataka

## Metodologija razvoja baze podataka

Sistem metoda, tehnika, postupaka i uputstava za projektovanje i realizaciju baze podataka takav da obezbeđuje zadovoljenje postavljenih ciljeva projektovanja.

Metodologije razvoja baze podataka i informacionog sistema su međusobno povezane.

Razvoj baze podataka predstavlja veliki i značajan proces u procesu razvoja informacionog sistema.

## Ciljevi projektovanja baze podataka i informacionog sistema

Ciljevi:

- zadovoljavanje informacionih zahteva korisnika
  - postizanje zahtevane funkcionalnosti softvera
- obezbeđenje zadovoljavajuće "udobnosti" (lakoće) korišćenja aplikacija informacionog sistema
- obezbeđenje prirodnog i lako razumljivog strukturiranja podataka
- obezbeđenje zadovoljavajućih performansi rada sistema
- obezbeđenje zahtevane sigurnosti i bezbednosti sistema
- obezbeđenje dugovečnosti razvijenog sistema
- stvaranje osnova za obezbeđenje kvaliteta softvera:
  - normalno održavanje sistema u eksploataciji
  - potrebne izmene funkcionalnosti i mogući reinženjering sistema
  - optimizaciju angažovanih resursa u razvoju i eksploataciji sistema (idealno: "najjeftinije" za "najkraće" vreme)

Ciljevi su međusobno često kolizionirani.

Pogodan kompromis u ostvarenju ovih ciljeva:

1. zadovoljstvo naručioca, investitora i korisnika softverskog proizvoda
2. zadovoljstvo isporučioaca i realizatora softverskog proizvoda
3. duži životni vek softverskog proizvoda

## Projektovanje informacionog sistema

Podrazumeva paralelno sprovođenje postupaka:

- projektovanje strukture i sadržaja baze podataka
- projektovanje softverske podrške informacionog sistema
  - transakcionih programa i aplikacija

Praktično je nemoguće izolovano projektovati samo jedan od ova dva aspekta sistema, pogotovo u slučaju sistema višeg stepena složenosti.

## Aktivnosti projektovanja baze podataka

Aktivnosti:

- snimanje, analiza i specificiranje korisničkih zahteva
  - izrada neformalnih, semi-formalnih i formalnih računarski nezavisnih modela zahteva
- konceptualno projektovanje baze podataka
  - izrada semi-formalnih i formalnih platformski nezavisnih modela
- implementaciono (logičko) projektovanje baze podataka
  - izrada semi-formalnih i formalnih platformski nezavisnih/zavisnih modela
    - s obzirom na model podataka koji podržava izabrani SUBP
- fizičko projektovanje baze podataka
  - izrada formalni platformski zavisnih modela
    - s obzirom na koncepte koje podržava izabrani SUBP
- projektovanje arhitekture sistema baze podataka

## Aktivnosti realizacije baze podatka

Aktivnosti:

- specifikacija šeme baze podataka putem DDL jezika i mehanizama izabranog SUBP
- instalacija arhitekture sistema baze podataka
- implementacija i testiranje upotrebe baze podataka
- punjenje baze podataka realnim podacima
- praćenje rada, otklanjanje nedostataka i podešavanje performansi baze podataka

## Razvoj aplikacija informacionih sistema

### Aktivnosti projektovanja aplikacija informacionih sistema

Aktivnosti:

- snimanje, analiza i specificiranje korisničkih zahteva
  - izrada neformalnih, semi-formalnih i formalnih računarski nezavisnih modela zahteva
- konceptualno projektovanje softverske podrške
  - izrada semi-formalnih i formalnih platformski nezavisnih modela
- implementaciono projektovanje softverske podrške
  - transformacijom prethodno razvijenih platformski nezavisnih modela u platformski zavisne modele
    - s obzirom na koncepte koje podržava razvojno okruženje i raspoloživi generatori programskog koda

### Aktivnosti realizacije aplikacija informacionih sistema

Aktivnosti:

- programiranje aplikacija i transakcionih programa
- testiranje softverskih aplikacija
- uvođenje softverskih aplikacija u upotrebu



- praćenje rada, otklanjanje nedostataka i podešavanje performansi

## Inženjerstvo korisničkih sistema

Snimanje, analiza i specificiranje korisničkih zahteva:

- Generiše ulazne podatke za sve naredne aktivnosti razvoja informacionog sistema, u celini.
- Veoma bitna, osetljiva i mukotrpa aktivnost, često sa nedovoljno preciznim rezultatima.
- Zahteva detaljno upoznavanje i što formalniju specifikaciju poslovnih procesa realnog sistema.

Zadaci:

- identifikacija ciljeva, potreba, oblasti primene i grupa korisnika budućeg informacionog sistema.
- identifikacija procesa poslovanja koje treba podržati, kao i postupaka njihovog izvođenja
- identifikacija proavila poslovanja i ograničenja u realnom sistemu, koja treba podržati
- identifikacija dokumentacije i tokova dokumenata koji se stvaraju i koriste u procesima poslovanja
- analiza postojećeg i planiranog načina obuhvata i korišćenja podataka u nutar sistema
- identifikacija problema, rizika, mogućih rešenja, ključnih faktora uspeha i indikatora ostvarenja

Tehnike:

- intervju
- analiza relevantne dokumentacije
  - poslovne, sistema kvaliteta, tehničko-tehnološke, svih relevantnih spoljnih i internih izvora dokumenata
- posmatranje poslovanja u realnim uslovima
- poređenja s poslovanjem sličnih realnih sistema
- analiza funkcionisanja postojećih informacionih sistema iste ili slične namene
  - postojećih informacionih sistema u realnom sistemu, ili drugim sličnim sistemima u postmatranoj oblasti poslovanja

Rezultati:

- specifikacije u manje ili više strukturiranom obliku, neformalne, semi-formalne i formalne specifikacije modela
  - neformalne: multimedijalni zapisi, tekstualni zapisi putem prirodnog jezika
  - semi-formalne: dijagramske reprezentacije modela
  - formalne: tekstualni zapisi putem formalnog jezika, sve više u upotrebi su jezici namenski za domen (*Domain Specific Language*), namenski jezici za kreiranje specifikacija korisničkih zahteva
- modeli funkcionalne, organizacione, menadžerske i prostorne strukture realnog sistema
- modeli procesa poslovanja realnog sistema
- modeli pravila poslovanja i ograničenja u realnom sistemu
- modeli dokumenata i tokova dokumenata u realnom sistemu
- modeli arhitekture postojećeg informacionog sistema
- modeli zahteva prema novom informacionom sistemu

## Konceptualno projektovanje

Obuhvata konceptualno projektovanje:

- baze podataka
- aplikativne softverske podrške

## Konceptualno projektovanje baze podataka

Obuhvata konceptualno projektovanje šeme i eksternih šema.

Vrši se putem modela podataka koji poseduje:

- semantičko bogatstvo
- izražajnost
- jednostavnost
- minimalnost
- dijagramsku reprezentaciju
- formalnost

Pogodan izbor je prošireni model tipova entiteta i poveznika, odnosno EER model.

Postoje dva različita pristupa:

- direktni postupak
- postupak postupne integracije

### Direktni postupak

Direktno se formira konceptualna šema baze podatka na osnovu specificiranih korisničkih zahteva.

Iz konceptualne šeme izdvajaju se potrebne eksterne šeme.

Primeren u slučaju projektovanja šema baze podataka manjeg obima.

### Postupak postupne integracije

Obuhvata projektovanje i integraciju eksternih šema

Primenjuje se pri projektovanju šema baze podataka većeg obima.

### Projektovanje eksternih šema

Iterativni postupak, dolazi se do potrebnog nivoa detalja u projektu u više prolaza:

Može se realizovati korišćenjem jednim od ponuđenih ili kombinacijom sledeća dva postupka:

- s vrha ka dnu ("top-down")
- sa dna ka vrhu ("bottom-up")

Ulazne specifikacije postupka:

- specifikacije korisničkih zahteva
- specifikacije arhitekture razvijanog informacionog sistema
  - model funkcionalne strukture informacionog sistema
    - pogled sa stanovišta poslovne arhitekture
  - model toka podataka informacionog sistema
    - pogled sa stanovišta arhitekture podataka

### Postupak s vrha ka dnu

Projektuje se prva verzija eksterne šeme, pomoću koncepata na visokom nivou apstrakcije, poput tipova entiteta i poveznika.

Verzije se iterativno modifikuju i specijalizuju:

- kardinaliteti tipova poveznika
- skupovi obeležja tipova entiteta i poveznika i njihovi domeni
- ključevi tipova entiteta
- integriteti pojave tipa i integriteti domena
- uloge tipova u eksternoj šemi
- skupovi obeležja za modifikaciju tipova u eksternoj šemi

Primena postupka integracije može dovesti do novih modifikacija eksterne šeme.

Konačna verifikacija eksterne šeme se vrši pri projektovanju transakcionih programa koji treba da rade nad posmatranom eksternom šemom.

### Postupak sa dna ka vrhu

Koristi se uglavnom u reinženjeringu postojećeg modela.

### Integracija eksternih šema

Vrši se analizom semantike (sadržaja) i struktura eksternih šema u cilju:

- otkrivanja kolizija
- otklanjanja kolizija, modifikacija eksternih šema
- konačne integracije u objedinjenu konceptualnu šemu baze podataka

Tipovi kolizija:

- kolizija imenovanja
  - sinonimi i homonimi
- kolizija tipa
  - predstavljanje istih realnih objekata putem različitih koncepata
- kolizija kardinaliteta
  - neusaglašenost kardinaliteta istih tipova poveznika

- kolizija primarnih ključeva
  - neusaglašenost primarnih ključeva istih tipova
- kolizija sadržavanja obeležja
  - ista obeležja u različitim tipovima
- kolizija integriteta pojave tipa
  - neusaglašenost integriteta pojave tipa istih tipova
  - koje operacije upitnog jezika i jezika za manipulaciju podacima su moguće (read, insert, update, delete)
  - kolizije u domenu, uslovu, null svojstvu ili predefinisanoj vrednosti

## Konceptualno projektovanje softvera

Obuhvata konceptualno projektovanje:

- transakcionih programa
- aplikacija informacionog sistema

## Konceptualno projektovanje transakcionih programa

Transakcija je najmanja jedinica obrade podataka koja:

- prevodi bazu podataka iz jednog u drugo (ne nužno različito) konzistentno stanje
  - s obzirom na implementirana ograničenja
- sadrži operacije upita ili/i operacije ažuriranja podataka u bazi podataka

Efekti izvođenja transakcije se na kraju u celosti:

- potvrđuju (*commit*-uju)
  - postaju vidljivi ostalim korisnicima u sistemu
- poništavaju (*rollback*-uju)
  - ostavljaju obrađivani deo baze podataka u stanju kakvo je važno neposredno pre početka izvođenja transakcije

Transakcioni program sadrži formalni opisi jedne ili više transakcija.

Jedno izvršenje transakcionog programa može pokrenuti neke, ili sve od formalno opisanih transakcija.

Konceptualno projektovanje transakcionih programa obuhvata:

- izradu platformski nezavisnih modela transakcionog programa
- kreiranje formalnih specifikacija visokog nivoa putem namenski kreiranih jezika
  - specifikaciju korisničkog interfejsa
  - specifikaciju (referenciranje) eksterne šeme
  - specifikaciju funkcionalnosti (logike) programa
    - moguće putem strukturnog dijagrama
      - kroz dijagram se kreće kroz postupke od gore ka dole, s leva na desno
      - svaki postupak može imati ulaz (praktično uvek) i izlaz

- moguće je više puta izvršiti neke postupke

Vrši se na osnovu:

- modela procesa poslovanja
- modela tokova podataka
- eksternih šema

Ciljevi:

- formiranje platformski nezavisnog modela transakcionih programa
- obezbeđivanje podloge za implementaciono projektovanje transakcionih programa
- verifikacija korektnosti eksternih šema

### **Konceptualno projektovanje aplikacija informacionog sistema**

Aplikacija je skup transakcionih programa, snabdeven strukturom, namenjen za izvršavanje na jednom tipu radnog mesta od strane grupe korisnika sa ličnim ili istim zahtevima.

Konceptualno projektovanje aplikacija informacionog sistema obuhvata:

- izradu platformski nezavisnog modela aplikacija informacionog sistema
- specifikaciju naziva aplikacije informacionog sistema
- specifikaciju strukture (sadržaja) aplikacije informacionog sistema:
  - specifikaciju skupa transakcionih programa koji čine aplikaciju
  - specifikaciju veza između transakcionih programa
- specifikaciju prava pristupa na nivou uloga korisnika

Važna aktivnost u cilju obezbeđenja dugovečnosti aplikacija informacionog sistema.

Praktično nezaobilazna u pristupu razvoju softvera zasnovan na modelima - *Model Driven Software Development*-u.

U praksi se nažalost često u potpunosti izostavlja:

- inicijalno se čini nepotrebnom
- verifikacija eksternih šema putem programa ostavlja se za fazu implementacionog projektovanja softvera
- dugoročne negativne posledice mogu biti ozbiljne
- efekti se ne primećuju odmah, već nakon proteka dužeg perioda vremena