


Poglavje 3

Relacijski podatkovni model

Večinoma povzeto po [2]



O relacijskem podatkovnem modelu

- Pojavi se leta 1970, predlaga ga **Edgar Codd**.
- **Pomeni revolucijo**, nadomesti starejše modele.
- Model PB, ki temelji na relacijskem modelu, je predstavljena z množico **relacij**, kjer je vsaka relacija **tabela z vrsticami in stolpci**.
- Je zelo **enostaven za razumevanje**:
 - Tudi neizkušeni lahko razumejo vsebino podatkovne baze;
 - Na voljo so enostavni vendar močni jeziki za poizvedovanje po vsebini PB.



Terminologija pri relacijskem modelu..

- Pri relacijskem modelu uporabljamo določeno terminologijo:
 - Relacija
 - Atribut
 - Domena
 - n-terica
 - Stopnja relacija
 - Števnost relacije
 - Relacijska shema
 - Relacijska PB



Terminologija pri relacijskem modelu..

- **Relacijo** si lahko predstavljamo kot dvodimenzionalno tabelo s stolpci in vrsticami.
 - Velja za logično strukturo podatkovne baze in ne za fizično.
 - Vrstica relacije predstavlja objekt, osebo, dogodek, pravilo – ima nek pomen

Ime	Starost (v letih)	Teža (v kg)
Tine	15	50
Meta	20	45
Jure	40	80
Ana	5	10

→ Relacija



Terminologija pri relacijskem modelu..

- **Atribut** je poimenovani stolpec relacije.
 - Predstavlja lastnost tega (objekt, osebo, dogodek, pravilo), kar predstavlja relacijo
 - Atributu določimo podatkovni tip in dolžino

Ime	Starost (v letih)	Teža (v kg)
Tine	15	50
Meta	20	45
Jure	40	80
Ana	5	10

→ Atribut relacije

Terminologija pri relacijskem modelu..

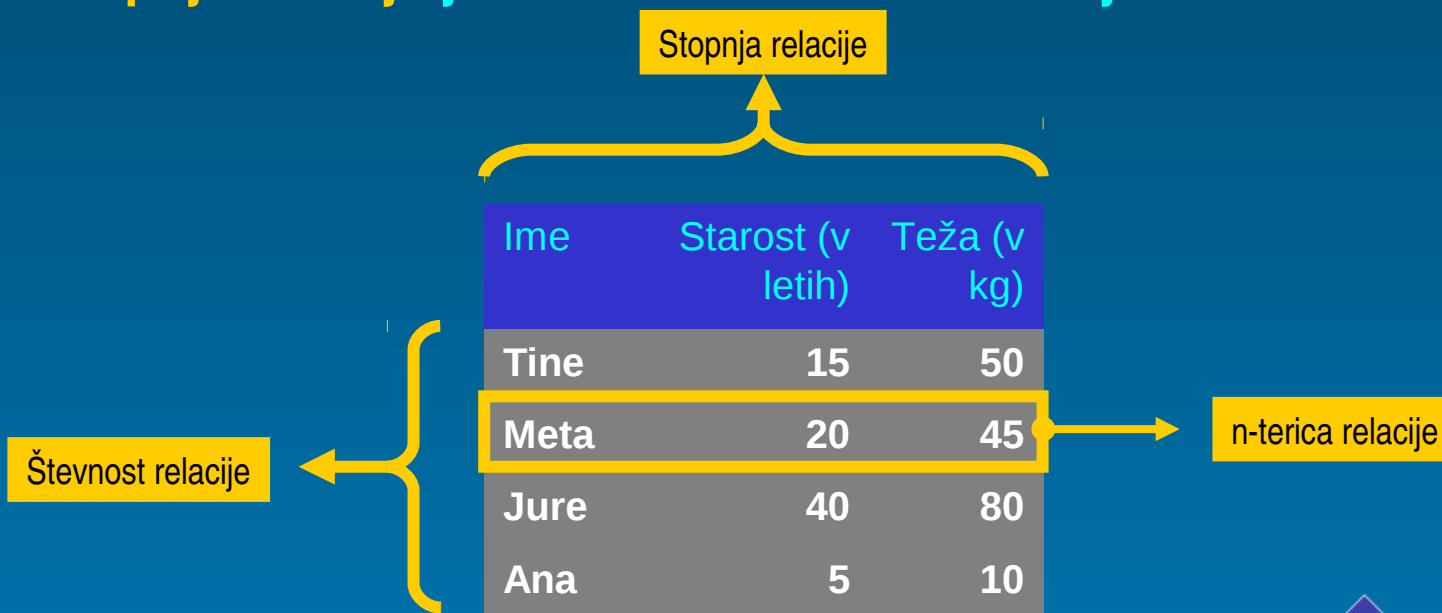
- **Domena** določa poleg podatkovnega tipa in dolžine tudi množico dovoljenih vrednosti enega ali več atributov, ki so vključeni v to domeno.
- **Primeri domen:**

Attribute	Domain Name	Meaning	Domain Definition
branchNo	BranchNumbers	The set of all possible branch numbers	character: size 4, range B001–B999
street	StreetNames	The set of all street names in Britain	character: size 25
city	CityNames	The set of all city names in Britain	character: size 15
postcode	Postcodes	The set of all postcodes in Britain	character: size 8
sex	Sex	The sex of a person	character: size 1, value M or F
DOB	DatesOfBirth	Possible values of staff birth dates	date, range from 1-Jan-20, format dd-mmm-yy
salary	Salaries	Possible values of staff salaries	monetary: 7 digits, range 6000.00–40000.00



Terminologija pri relacijskem modelu..

- N-terica je ena vrstica v relaciji.
- Števnost relacije je število n-teric relacije.
- Stopnja relacije je število atributov v relaciji.



Terminologija pri relacijskem modelu

- **Relacijska podatkovna baza** je množica normaliziranih relacij z enoličnimi imeni.
- Kaj so **normalizirane relacije** se bomo učili v nadaljevanju.



Matematična definicija relacije..

- Vzemimo dve množici D1 in D2, kjer $D1 = \{2, 4\}$ in $D2 = \{1, 3, 5\}$.
- **Kartezijski produkt** $D1 \times D2$ je množica vseh urejenih parov, kjer prvi element pripada množici D1, drugi množici D2: $D1 \times D2 = \{(2, 1), (2, 3), (2, 5), (4, 1), (4, 3), (4, 5)\}$



Matematična definicija relacije..

- Vsaka podmnožica kartezijskega produkta $D1 \times D2$ je relacija: $R \subset (D1 \times D2)$.

- Primer:

$$R = \{(2, 1), (4, 1)\}$$

- Kateri pari so v relaciji lahko določimo s pogoji: na primer vsi pari, kjer je drugi element enak 1:

$$R = \{(x, y) \mid x \in D1, y \in D2 \text{ in } y = 1\}$$

- ali kjer je prvi element dvakrat večji od drugega:

$$S = \{(x, y) \mid x \in D1, y \in D2 \text{ in } x = 2y\}$$



Matematična definicija relacije..

- Vzemimo tri množice D1, D2, D3 s kartezijskim produktom D1 x D2 x D3; Na primer:

$$D1 = \{1, 3\} \quad D2 = \{2, 4\} \quad D3 = \{5, 6\}$$

$$D1 \times D2 \times D3 = \{(1,2,5), (1,2,6), (1,4,5), (1,4,6), (3,2,5), (3,2,6), (3,4,5), (3,4,6)\}$$

- Vsaka podmnožica teh urejenih trojic je relacija!



Matematična definicija relacije..

- Kartezijski produkt n množic (D_1, D_2, \dots, D_n) je:

$$D_1 \times D_2 \times \dots \times D_n = \{(d_1, d_2, \dots, d_n) \mid d_1 \in D_1, d_2 \in D_2, \dots, d_n \in D_n\}$$

- Navadno pišemo kot
$$\prod_{i=1}^n D_i$$

- Vsaka podmnožica n -teric iz tega kartezijskega produkta je relacija teh n množic.



Relacijska shema..

- Vsaki relaciji pripada **relacijska shema**.
- Relacijsko shemo sestavlja oznaka sheme **R** ter lista oznak atributov **A_i** s pripadajočimi oznakami domen **D_i**:

$R (A_1: D_1, A_2: D_2, \dots, A_n: D_n)$

- Relacijska shema predstavlja **semantiko** ali **pomen** relacije.



Relacijska shema..

- Relacijske sheme so **del konceptualnih shem**. Razlagajo pomen relacij.
- Glede na skromno izrazno možnost nudijo informacijo le **poznavalcem podatkovne baze**, ki znajo relacijske sheme pravilno interpretirati.
- Vsaki relaciji pripada natanko ena relacijska shema; neka relacijska shema pa lahko pojasnjuje tudi več relacij:

$$\text{Sh: } \{r\} \rightarrow \{R\} \text{ Sh}(r) = R$$



Relacijska shema

Ime	Starost (v letih)	Teža (v kg)
Tine	15	50
Meta	20	45
Jure	40	80
Ana	5	10

Relacijska shema

Relacija, predstavljena kot tabela

$Sh(r) = Oseba(Ime: I, Starost: C, Teža: C)$

Relacijska shema

Domena, ki obsega imena: $I \equiv \{Tine, Meta, Jure, Ana\}$
Domena, ki obsega interval celih števil: $C \equiv 1, 2, \dots, 200$

Domene atributov
relacije



Lastnosti relacij..

- **Ime relacije je enolično.** V logični enoti (odvisno od SUPB, praviloma gre za shemo) podatkovne baze ni dveh relacij z enakim imenom
- Vsaka celica tabele (polje), ki predstavlja relacijo, vsebuje **natančno eno atomarno vrednost**
- Vsak **atribut relacije ima enolično ime.** V isti relaciji ni dveh atributov, ki bi imela isto ime
- Vrednosti nekega atributa so vse **iz iste domene**



Lastnosti relacij

- Vsaka n -terica relacije je enolična \rightarrow v relaciji ni dveh enakih n -teric.
- Vrstni red atributov v relaciji je nepomemben.
- Vrstni red n -teric v relaciji je nepomemben.



Primeri..

Ime	Starost (v letih), teža (v kg)
Tine	S15_T50
Meta	S20_T45
Jure	S40_T80
Ana	S5_T10



Celice ne vsebujejo atomarnih vrednosti

Zakonca	Leto poroke (celo število)
Tine	1995
Meta	1995
Ana,	1980
Jure	1980



Celice vsebujejo več vrednosti



Primeri

- Dvojni pomen posameznih atributov ni priporočljiv:

Tip	Ime/Priimek
1	Jana
2	Novak
1	Tina
1	Kaja

Če je tip='1' potem ime, sicer priimek



Funkcionalne odvisnosti..

- **Relacija je model nekega stanja v poslovni domeni** → njena vsebina ne more biti poljubna.
- Realne omejitve ne omogočajo, da bi bili odnosi v svetu kakršnikoli; možna so le določena stanja.
- **Odvisnosti** so sredstvo, s katerim lahko v relacijskem modelu povemo, katere vrednosti relacij so veljavne in katere sploh ne morejo obstajati.



Funkcionalne odvisnosti..

- Poznamo več vrst odvisnosti:
 - Funkcionalne odvisnosti (functional dependency)
 - Večvrednostne odvisnosti (multivalued dependency)
 - Stične odvisnosti (join dependency)
- Obravnavali bomo **funkcionalne odvisnosti**



Funkcionalne odvisnosti..

- Predpostavimo, da obstaja relacijska shema R z množico atributov, katere podmnožici sta X in Y .
- V relacijski shemi R velja $X \rightarrow Y$ (X funkcionalno določa Y oziroma Y je funkcionalno odvisen od X), če v nobeni relaciji, ki pripada shemi R , ne obstajata dve n -terici, ki bi se ujemali v vrednostih atributov X in se ne bi ujemali v vrednostih atributov Y .



Funkcionalne odvisnosti

- Množico funkcionalnih odvisnosti, ki veljajo med atributi funkcionalne sheme R in v vseh njenih relacijah, označimo s F

$$X \rightarrow Y \in F \Leftrightarrow \forall r (\text{Sh}(r) = R \Rightarrow \forall t, \forall u (t \in r \wedge u \in r \wedge t.X = u.X \Rightarrow t.Y = u.Y))$$

kjer

$t.X, u.X, t.Y$ in $u.Y$ označujejo vrednosti atributov X oziroma Y v n -tericah t oziroma u .



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 oceno OcenaPisno in OcenaUstno.

- Funkcionalne odvisnosti relacijske sheme Izpit so:

$$F \equiv \{ \text{VpŠt} \rightarrow (\text{Priimek}, \text{Ime}), (\text{VpŠt}, \text{ŠifraPredmeta}, \text{DatumIzpita}) \rightarrow (\text{OcenaPisno}, \text{OcenaUstno}) \}$$


Ključni relacije..

- Ker je relacija množica n -teric, so v njej vse n -terice ločene med seboj (**ne pozabimo: v relaciji ni dveh enakih n -teric**).
- Za sklicevanje na posamezno n -terico ni potrebno poznati vseh vrednosti atributov n -terice, če v shemi nastopajo funkcionalne odvisnosti.
- Množici atributov, ki določajo vsako n -terico, pravimo **ključ relacije** oziroma **ključ relacijske sheme**.



Ključni relacije..

- Predpostavimo, da obstaja relacijska shema z atributi A_1, A_2, \dots, A_n katere podmnožica je množica atributov X .
- Atributi X so ključ relacijske sheme oziroma pripadajočih relacij, če sta izpolnjena naslednja dva pogoja:
 - (1) $X \rightarrow A_1, A_2, \dots, A_n$
 - (2) ne obstaja X' , ki bi bila prava podmnožica od X in ki bi tudi funkcionalno določala A_1, A_2, \dots, A_n



Ključni relacije..

- Poznamo več vrst ključev:
 - Kandidat za ključ (a key candidate)
 - Primarni ključ (primary key)
 - Tuji ključ (foreign key)
- **Kandidat za ključ** je vsaka podmnožica atributov relacije, ki relacijo enolično določa.



Ključni relacije

- **Primarni ključ** je tisti kandidat za ključ, ki ga (med vsemi kandidati za ključ) izberemo za shranjevanje relacij v fizični podatkovni bazi.
- **Tuji ključ** je množica atributov, v okviru ene relacije, ki je enaka primarnemu ključu neke druge ali iste relacije.



Primeri ključev

ARTIKEL		
Šifra	Naziv	Zaloga
A10	Telovadni copati Nike	10
A12	Trenerka Bali	4
BC80	Moška jakna QuickSilver	1
X12	Ženska jakna QuickSilver	0

Primarni ključ v tabeli Artikel

POSTAVKA		
Račun	Šifra artikla	Količina
15/05	A10	1
15/05	X12	1

Primarni ključ v tabeli Postavka

Tuji ključ v tabeli Postavka → kaže na primarni ključ v tabeli Artikel



Omejitve nad podatki..

- **Podatkovni model v širšem smislu je skupek konceptov za opisovanje strukture podatkov, mehanizmov za obdelavo podatkov, povezav med njimi in omejitev nad podatki.**
- **Sestoji se iz treh delov:**
 - **Komponenta za strukturo podatkov:** zajema pravila, po katerih je možno kreirati podatkovno bazo;
 - **Komponenta za obdelavo:** definira tipe operacij, ki so dovoljene nad podatki → vključuje operacije za ažuriranje in iskanje podatkov v bazi ter za spreminjanje strukture;
 - **Komponenta za omejitve:** množica pravil oziroma omejitev, ki skrbijo za celovitost podatkov.



Omejitve nad podatki

- **Poznamo več vrst omejitev:**
 - Omejitve domene (Domain constraints)
 - Pravila za celovitost podatkov (Integrity constraints)
 - Celovitost entitet (Entity Integrity)
 - Celovitost povezav (Referential Integrity)
 - Splošne omejitve (General constraints)



Oznaka “Null”

- Oznaka “Null”:
 - Predstavlja vrednost atributa, ki je trenutno neznana ali irelevantna za n-terico.
 - Gre za nepopolne podatke ali podatke pri izjemnih primerih.
 - Predstavlja odsotnost podatka. Ni enako kot 0 ali prazen znak, kar je dejansko vrednost.
- Oznaka “Null” je problematična pri implementaciji, saj je relacijski model osnovan na predikatnem računu prvega reda (Boolean logic), kjer so edini možni vrednosti true in false.



Omejitve entitet in povezav

- **Omejitev entitete**

- V osnovni relaciji ne sme biti noben atribut, ki je del ključa, enak Null.
- Primarni ključ kot integritetna omejitev
- Ostale omejitve vezane na kombinacijo vrednosti posameznih atributov v n-terici

- **Omejitve povezav**

- Če v relaciji obstajajo tuji ključi, potem morajo:
 - (a) njihove vrednosti ustrezati tistim, ki so v obliki ključa zapisane v eni izmed n-teric neke druge ali iste relacije
 - (b) ali pa mora biti tuji ključ v celoti enak Null.
- Tuji ključ kot integritetna omejitev



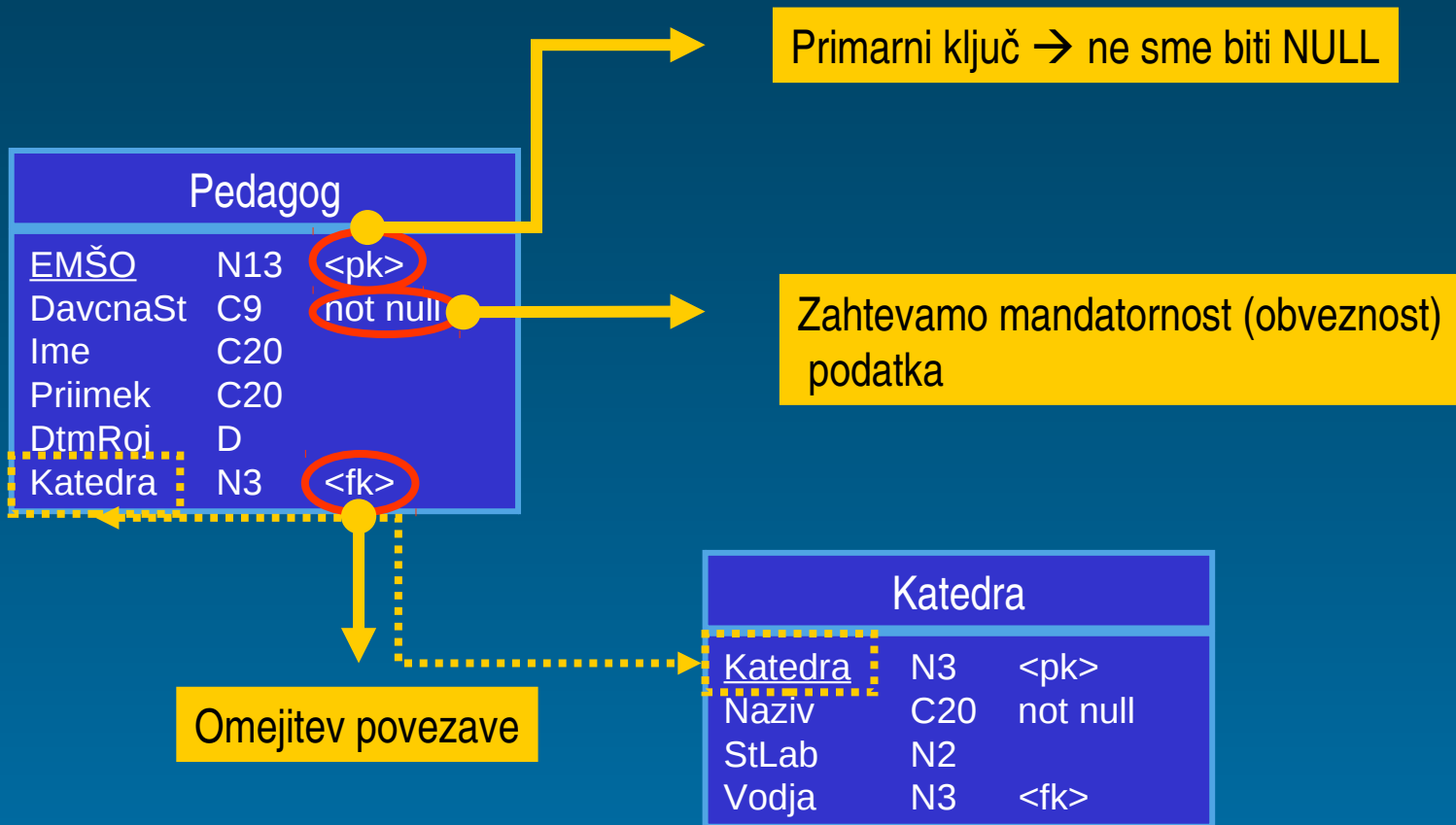
Splošne omejitve

- Splošne omejitve

- Dodatna pravila, ki jih določi uporabnik ali skrbnik podatkovne baze, ki definirajo ali omejujejo nek vidik poslovne domene, za katero je narejena podatkovna baza.



Primeri omejitev



Pogledi..

- **Osnovna relacija (base relation)**
 - Poimenovana relacija, ki ustreza nekemu entitetnemu tipu v konceptualnem modelu, katere n-terice so fizično shranjene v podatkovni bazi.
 - Predstavlja konceptualno shemo
- **Pogled (view)**
 - Rezultat ene ali več operacij nad osnovnimi relacijami z namenom pridobitve nove relacije.
 - Predstavlja zunanjo shemo



Pogledi

- Pogled je **navidezna relacija**, ki ne obstaja v relacijski bazi, temveč se dinamično kreira takrat, ko nekdo po njej povprašuje.
- Vsebina pogleda je **definirana kot poizvedba** nad eno ali več osnovnimi relacijami.
- **Pogledi so dinamični** → spremembe nad osnovnimi relacijami, katerih atributi so zajeti tudi v pogledu, so v pogledu takoj vidne.



Primer pogleda

ARTIKEL			RAČUN		
Šifra	Naziv	Zaloga	Račun	Šifra artikla	Količina
A10	Telovadni copati Nike	10	15/05	A10	1
A12	Trenerka Bali	4	15/05	X12	1
BC80	Moška jakna QuickSilver	1	16/05	A10	3
X12	Ženska jakna QuickSilver	0	17/05	A10	1

```
SELECT A.sifra, A.naziv, sum(R.kolicina) AS Prodanih
FROM artikel A, racun R
WHERE A.sifra = R.sifra
```

Šifra	Naziv	Prodanih
A10	Telovadni copati Nike	5
X12	Ženska jakna QuickSilver	1
...		



Namen uporabe pogledov

- Predstavljajo odličen **mehanizem za zagotavljanje varnosti** → skrivajo posamezne dele konceptualne sheme pred določenimi uporabniki.
- Uporabnikom dajejo možnost, da do podatkov **dostopajo na prilagojen način** → isti podatki so lahko s strani različnih uporabnikov v istem času vidni na različne načine.
- **Poenostavljajo kompleksne operacije nad osnovnimi relacijami.**
- Omogočajo **zunanjo shemo** (nivoji abstrakcije)



Spreminjanje vsebine pogledov..

- Vse spremembe nad osnovnimi relacijami morajo biti takoj vidne tudi v pogledih nad temi relacijami.
- Če spremenimo podatke v pogledu, se morajo spremembe poznati tudi v osnovnih relacijah, na katere se te spremembe nanašajo.



Spreminjanje vsebine pogledov..

- V pogledih niso možne vse spremembe. Veljajo naslednje omejitve:
 - Nad pogledom so možne spremembe, če pogled **zajema eno samo osnovno relacijo** ter vključuje attribute, ki so **kandidat za ključ relacije**.
 - Če pogled **zajema več relacij**, spremembe niso možne (izjeme).
 - Če je pogled pridobljen z **agregacijo** ali **grupiranjem** n-teric, spremembe niso možne.

