

Fizično načrtovanje

Vsebina:

- Sistematična izvedba fizičnega načrtovanja
- Kreiranje osnovnih relacij
- Izbira datotečne organizacije
- Indeksiranje
- Izvedba omejitev
- Določitev varnostnih mehanizmov
- Dokumentiranje fizičnega modela

Načrtovanje fizične podatkovne baze

- Fizično načrtovanje PB opredeljuje proces, s katerim izdelamo opis implementacije PB na sekundarnem pomnilnem mediju.
- Fizični načrt opisuje
 - osnovne relacije,
 - datotečno organizacijo,
 - indekse za doseg učinkovitega dostopa do podatkov,
 - povezane omejitve in
 - varnostne mehanizme.

Metoda načrtovanja fizične PB...

- Možni koraki načrtovanja fizične PB:
 - K3 – Pretvori logični model v jezik za ciljni SUPB
 - K3.1 – Izdelaj načrt osnovnih relacij
 - K3.2 – Izdelaj načrt predstavitve izpeljanih atributov
 - K3.3 – Izdelaj načrt splošnih omejitev
 - K4 – Izdelaj načrt datotečne organizacije ter indeksov
 - K4.1 – Analiziraj transakcije
 - K4.2 – Izberi datotečno organizacijo
 - K4.3 – Določi indekse
 - K4.4 – Oцени velikost podatkovne baze

Koraka K1 in K2 se nanašata na izdelavo konceptualnega in logičnega podatkovnega modela

Metoda načrtovanja fizične PB...

- Možni koraki načrtovanja fizične PB (nadaljevanje):
 - K5 – Izdelaj načrt uporabniških pogledov
 - K6 – Izdelaj načrt varnostnih mehanizmov
 - K7 – Preveri smiselnost uvedbe nadzorovane redundance podatkov (denormalizacija)

K3 – Pretvorba v jezik za SUPB

- Namen koraka: iz logičnega modela izdelati podatkovno shemo za ciljni SUPB.
- Poznati moramo funkcionalnosti ciljnega SUPB, npr.:
 - Kako izdelati osnovne relacije?
 - Ali ciljni SUPB podpira primarne, tuje in alternativne ključe?
 - Ali podpira obveznost podatkov (NOT NULL)?
 - Ali podpira domene?
 - Ali podpira pravila omejitve podatkov?
 - Ali podpira sprožilce (triggers) in bazne programe (stored procedures)?

K3.1 – Izdelava osnovnih relacij...

- Namen: določiti, kako bodo osnovne relacije predstavljene v ciljnem SUPB.
- Vir podatkov je podatkovni slovar, jezik za opis pa DBDL (database definition language)
- Za vsako relacijo definiramo:
 - Naziv relacije;
 - Listo osnovnih atributov;
 - Primarni ključ ter kjer smiselno alternativne in tuje ključe;
 - Omejitve povezav za vsak tuji ključ.

K3.1 – Izdelava osnovnih relacij

- V podatkovnem slovarju imamo za vsak atribut :
 - Domeno, ki se sestoji iz podatkovnega tipa, dolžine in omejitev domene;
 - Morebitno privzeto vrednost;
 - Podatek o obveznosti atributa;
 - Podatek, ali je atribut izpeljan in če je, kako ga izračunamo.

Primer opisa osnovnih relacij z DBDL

```
Domain PropertyNumber:    variable length character string, length 5
Domain Street:            variable length character string, length 25
Domain City:              variable length character string, length 15
Domain Postcode:         variable length character string, length 8
Domain PropertyType:     single character, must be one of 'B', 'C', 'D', 'E', 'F', 'H', 'M', 'S'
Domain PropertyRooms:    integer, in the range 1-15
Domain PropertyRent:     monetary value, in the range 0.00-9999.99
Domain OwnerNumber:      variable length character string, length 5
Domain StaffNumber:      variable length character string, length 5
Domain BranchNumber:     fixed length character string, length 4

PropertyForRent(
  propertyNo PropertyNumber NOT NULL,
  street      Street        NOT NULL,
  city        City          NOT NULL,
  postcode   Postcode,
  type        PropertyType  NOT NULL DEFAULT 'F',
  rooms      PropertyRooms  NOT NULL DEFAULT 4,
  rent        PropertyRent  NOT NULL DEFAULT 600,
  ownerNo     OwnerNumber   NOT NULL,
  staffNo     StaffNumber,
  branchNo    BranchNumber  NOT NULL,
  PRIMARY KEY (propertyNo),
  FOREIGN KEY (staffNo) REFERENCES Staff(staffNo) ON UPDATE CASCADE ON DELETE SET NULL,
  FOREIGN KEY (ownerNo) REFERENCES PrivateOwner(ownerNo) and BusinessOwner(ownerNo)
                                ON UPDATE CASCADE ON DELETE NO ACTION,
  FOREIGN KEY (branchNo) REFERENCES Branch(branchNo)
                                ON UPDATE CASCADE ON DELETE NO ACTION);
```

K3.2 – Predstavitev izpeljanih atributov...

- Namen: določiti, kako bodo v SUPB predstavljeni izpeljani atributi.
- Preučiti logični podatkovni model in podatkovni slovar; izdelaj seznam izpeljanih atributov.
- Za vsak izpeljani atribut določi:
 - Atribut je shranjen v podatkovni bazi ali
 - Atribut se vsakokrat posebej izračuna in se ne hrani v podatkovni bazi.

K3.2 – Predstavitev izpeljanih atributov...

- Pri odločitvi, kako predstaviti izpeljane attribute, upoštevaj:
 - "strošek" shranjevanja in vzdrževanja skladnosti izpeljanih atributov z osnovnimi atributi, iz katerih je izpeljan;
 - "strošek" vsakokratnega izračunavanja izpeljanega atributa.
- Izberi stroškovno ugodnejšo rešitev.

Primer hranjenja izpeljanega atributa

PropertyForRent

propertyNo	street	city	postcode	type	rooms	rent	ownerNo	staffNo	branchNo
PA14	16 Holhead	Aberdeen	AB7 5SU	House	6	650	CO46	SA9	B007
PL94	6 Argyll St	London	NW2	Flat	4	400	CO87	SL41	B005
PG4	6 Lawrence St	Glasgow	G11 9QX	Flat	3	350	CO40		B003
PG36	2 Manor Rd	Glasgow	G32 4QX	Flat	3	375	CO93	SG37	B003
PG21	18 Dale Rd	Glasgow	G12	House	5	600	CO87	SG37	B003
PG16	5 Novar Dr	Glasgow	G12 9AX	Flat	4	450	CO93	SG14	B003

Staff

staffNo	fName	lName	branchNo	noOfProperties
SL21	John	White	B005	0
SG37	Ann	Beech	B003	2
SG14	David	Ford	B003	1
SA9	Mary	Howe	B007	1
SG5	Susan	Brand	B003	0
SL41	Julie	Lee	B005	1

K3.3 – Načrt splošnih omejitev

- Namen: izdelava načrta splošnih omejitev za ciljni SUPB.
- Glede podpore splošnim omejitvam obstajajo velike razlike med SUPB-ji. Če SUPB podpira standardni SQL, potem je deklaracija omejitve

```
CONSTRAINT StaffNotHandlingTooMuch  
CHECK (NOT EXISTS (SELECT staffNo  
                        FROM PropertyForRent  
                        GROUP BY staffNo  
                        HAVING COUNT(*) > 100))
```

Pomen omejitve: nihče od zaposlenih ne sme skrbeti za več kot 100 nepremičnin

K4 – Datotečna organizacija in indeksi

- Namen: izbrati optimalno datotečno organizacijo za shranjevanje osnovnih relacij ter potrebne indekse za doseganje ustrezne učinkovitosti.
- Načrtovalec mora dobro poznati, kakšne strukture in organizacije SUPB podpira ter kako deluje.
- Pomen certifikatov za obvladovanje SUPB
- Ključnega pomena so uporabniške zahteve v zvezi z željeno/pričakovano učinkovitostjo transakcij.
- Med SUPB-ji obstajajo velike razlike.

K4.1 – Analiza transakcij...

- Namen: razumeti namen transakcij, ki bodo tekle na SUPB ter analizirati tiste, ki so najpomembnejše.
- Najpomembnejše izberemo na podlagi različnih kriterijev.
- Poskušaj določiti kriterije učinkovitosti:
 - Pogoste transakcije, ki imajo velik vpliv na učinkovitost;
 - Transakcije, ki so kritičnega pomena za poslovanje;
 - Pričakovana obdobja (v dnevu/ tednu), ko bo SUPB najbolj obremenjen (peak load).
- Preveri tudi:
 - Attribute, ki jih transakcije spreminjajo
 - "Iskalne" pogoje v transakcijah...

K4.1 – Analiza transakcij...

- Pogosto ni moč analizirati vseh transakcij. Pregledamo zgolj najpomembnejše.
- Za identifikacijo najpomembnejših transakcij lahko uporabimo:
 - Matriko transakcija/relacija, ki kaže, katere relacije se v transakcijah uporabljajo.
 - Diagram uporabe transakcij, ki kaže, katere transakcije se bodo potencialno zelo frekventno izvajale.

K4.1 – Analiza transakcij...

- Možen pristop k obravnavi potencialno problematičnih delov modela:
 - Izdelamo matriko povezav transakcija/relacija,
 - Ugotovimo, katere relacije se najpogosteje uporabljajo v transakcijah,
 - Analiziramo, kateri podatki se uporabljajo v transakcijah, ki te relacije uporabljajo.

Primer matrice transakcija/relacija

Table 17.1 Cross-referencing transactions and relations.

Transaction/ Relation	(A)				(B)				(C)				(D)				(E)				(F)			
	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D
Branch									X				X											X
Telephone																								
Staff	X				X				X								X				X			
Manager																								
PrivateOwner	X																							
BusinessOwner	X																							
PropertyForRent	X					X	X	X									X							X
Viewing																								
Client																								
Registration																								
Lease																								
Newspaper																								
Advert																								

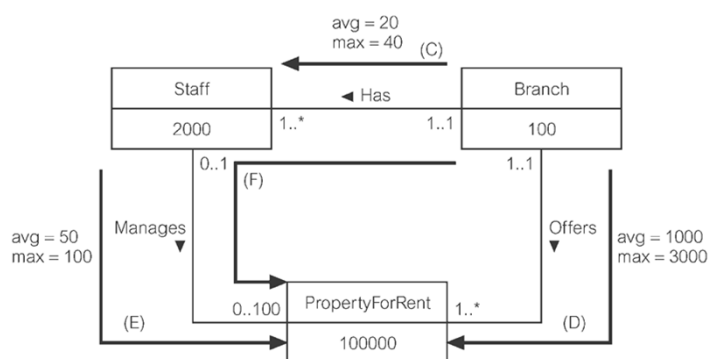
Dodatno lahko zapišemo število operacij na časovno enoto

I = Insert; R = Read; U = Update; D = Delete

Primer transakcije - F: Vrni skupno število nepremičnin (PropertyForRent), ki so dodeljene določenemu zaposlenemu (Staff) v dani agenciji (Branch).

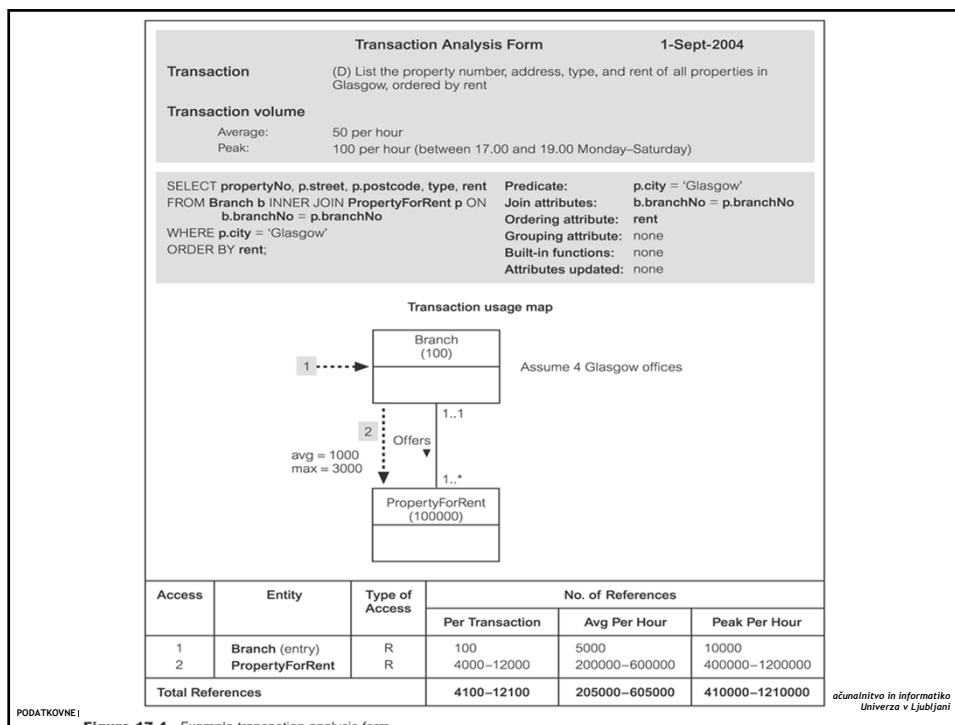
- 186 -

Primer diagrama uporabe transakcij



V specifikaciji zahtev je ocenjeno:

- 100.000 nepremičnin;
- 2.000 zaposlenih v 100 agencijah;
- Vsaka agencija ima v povprečju 1000 in maksimalno do 3.000 nepremičnin



K4.2 – Izbira datotečne organizacije

- Namen: izbrati učinkovito datotečno organizacijo za vse osnovne relacije.
- Datotečne organizacije:
 - Kopica (Heap),
 - Hash,
 - Metoda indeksiranega zaporednega dostopa (Indexed Sequential Access Method - ISAM),
 - Drevo B+,
 - Gruča (Cluster).
- Nekateri SUPB-ji ne podpirajo (ali ne dopuščajo izbire) vseh datotečnih organizacij.

K4.3 – Izbira indeksov...

- Namen: ugotoviti, ali lahko z dodatnimi indeksi povečamo učinkovitost sistema.
- Možen pristop:
 - Zapise pustimo neurejene.
 - Izdelamo toliko sekundarnih indeksov, kolikor je potrebno.

Sekundarni indeks = indeks po atributu, ki ni obenem tudi atribut, po katerem je urejena relacija

K4.3 – Izbira indeksov...

- Alternativni pristop
 - Zapise uredimo po primarnem ključu ali po indeksu gruče. V tem primeru kot atribut za sortiranje izberemo:
 - Atribut, ki se največkrat uporablja za povezovanja ali
 - Atribut, ki se najpogosteje uporablja za dostop do podatkov v relaciji.
 - Če je izbrani atribut za sortiranje primarni ključ, potem gre za primarni indeks sicer za indeks gruče.
 - Relacija ima lahko primarni indeks ali indeks gruče

Primarni indeks = indeks po primarnem ključu, po katerem je urejena relacija. Vsak zapis ima svojo vrednost.

Indeks gruče = indeks po atributu, ki je obenem tudi atribut, po katerem je urejena relacija, ni pa primarni ključ. Ključ ni unikaten!

K4.3 – Izbira indeksov...

- Sekundarni indeksi so način, kako omogočiti učinkovito iskanje s pomočjo dodatnih ključev.
- Pri določanju sekundarnih indeksov upoštevamo:
 - Povečanje učinkovitosti (predvsem pri iskanju po PB)
 - Dodatno delo, ki ga mora sistem opravljati za vzdrževanje indeksov. To vključuje:
 - Dodajanje zapisa v vsak sekundarni indeks, kadarkoli dodamo nek zapis v osnovno relacijo
 - Spreminjanje sekundarnega indeksa vsakokrat, ko se osnovna relacija spremeni
 - Povečanje porabe prostora v sekundarnem pomnilniku
 - Povečanje časovnega obsega za optimizacijo poizvedb zaradi preverjanja vseh sekundarnih indeksov.

K4.3 – Izbira indeksov...

- Nekaj smernic za uporabo sekundarnih indeksov:
 - Ne indeksiraj majhnih relacij. Zakaj?
 - Če datoteka ni urejena po primarnem ključu, potem kreiraj indeks na osnovi primarnega ključa.
 - Če je tuji ključ pogosto v uporabi, dodaj sekundarni indeks na tuji ključ.
 - Sekundarni indeks dodaj vsakemu atributu, ki se pogosto uporablja kot sekundarni ključ.
 - Sekundarne indekse dodaj atributom, ki nastopajo v pogojih za selekcijo ali stik: ORDER BY; GROUP BY ali v drugih operacijah, ki vključujejo sortiranje (npr. UNION ali DISTINCT).

K4.3 – Izbira indeksov

- Nekaj smernic za uporabo sekundarnih indeksov: (nadaljevanje):
 - Dodaj sekundarni indeks atributom, ki nastopajo v vgrajenih funkcijah, vključno z vgrajenimi agregatnimi funkcijami (npr. AVG, SUM, itd.);
 - Izogibaj se indeksiranju atributov, ki se pogosto spreminjajo.
 - Izogibaj se indeksiranju atributov v relacijah, nad katerimi se bodo pogosto izvajale poizvedbe, ki bodo vrstile večji del zapisov. Zakaj?
 - Izogibaj se indeksiranju atributov, ki so predstavljeni z daljšimi stringi.

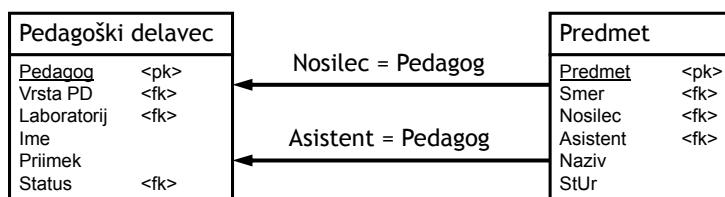
K4.4 – Ocena velikosti podatkovne baze

- Namen: oceniti, koliko prostora v sekundarnem pomnilniku zahteva načrtovana podatkovna baza.
- Ocena je odvisna
 - od velikosti in števila zapisov ter
 - od ciljnega SUPB (metapodatki).
- Primer: ocena velikosti podatkovne baze s pomočjo orodja PowerDesigner.

K5 – Načrt uporabniških pogledov

- Namen: izdelati načrt uporabniških pogledov, ki so bili opredeljeni v okviru zajema uporabniških zahtev.
- Uporabimo mehanizem pogledov (view).
- Pogled je navidezna relacija, ki fizično ne obstaja v PB, temveč se vsakokratno kreira s pomočjo poizvedbe.

Primer pogleda



```
create or replace view OBREMENITEV as
select PMET.PEDAGOG as Predavatelj,
       (PDG.PRIIMEK + PDG.IME) as Naziv_predavatelja,
       count(*) as Stevilo_Predmetov,
       sum(PMET.STUR) as Stevilo_ur_tedensko
from PREDMET PMET, PEDAGOŠKI DELAVEC PDG
where PMET.NOSILEC = PDG.PEDAGOG
group by PMET.NOSILEC, (PDG.PRIIMEK + PDG.IME);
```

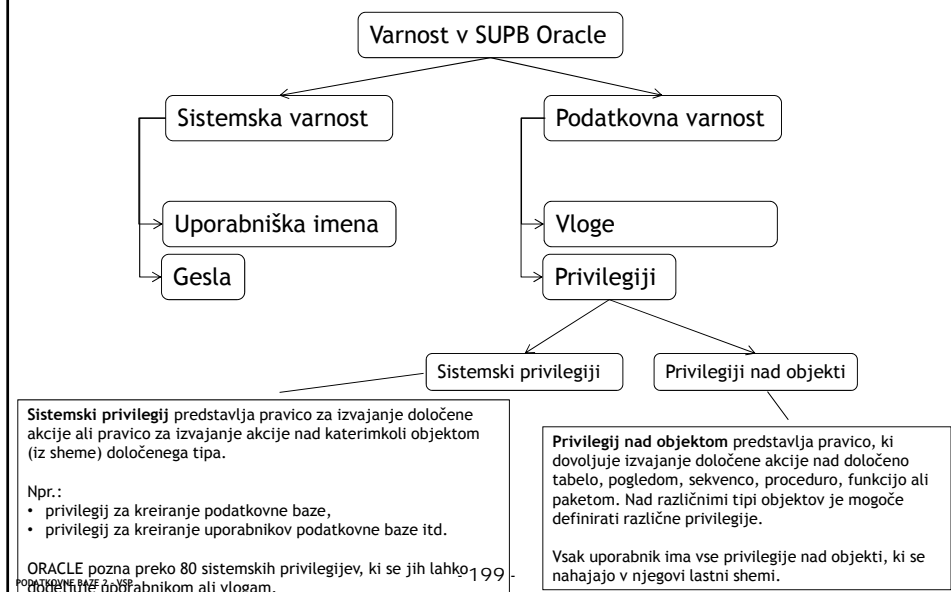
Obremenitev

Predavatelj
Naziv_predavatelja
Stevilo_Predmetov
Stevilo_ur_tedensko

K6 – Načrt varnostnih mehanizmov

- Namen: izdelati načrt varnostnih mehanizmov skladno z zahtevami naročnika.
- SUPB-ji tipično podpirajo dve vrsti varnosti:
 - Sistemsko varnost: varnost dostopa in uporabe podatkovne baze (navadno zagotovljeno s pomočjo uporabniških imen in gesel);
 - Podatkovno varnost: varnost uporabe podatkov – kdo lahko uporablja določene relacije ter kako.
- Med SUPB-ji so velike razlike v mehanizmih, ki jih imajo na voljo za doseg varnosti.

K6 – Primer - Oracle



K7 – Uvedba nadzorovane redundance...

- Namen: ugotoviti, ali je smiselno dopustiti določeno mero redundance (denormalizacija) ter tako izboljšati učinkovitost.
- Rezultat normalizacije je načrt, ki je po strukturi konsistenten ter minimalen.
- Včasih normalizirane relacije ne dajo zadovoljive učinkovitosti.
- Razmislimo, ali se zavoľjo izboljšanja učinkovitosti odpovemo določenim koristim, ki jih prinaša normalizacija.

K7 – Uvedba nadzorovane redundance...

- Upoštevamo tudi:
 - Implementacija denormaliziranih relacij je težja;
 - Z denormalizacijo velikokrat zgubimo na prilagodljivosti modela;
 - Denormalizacija navadno pospeši poizvedbe, vendar upočasni spreminjanje podatkov → potrebno je skrbeti še za redundančne podatke.

K7 – Uvedba nadzorovane redundance...

▪ Denormalizacija:

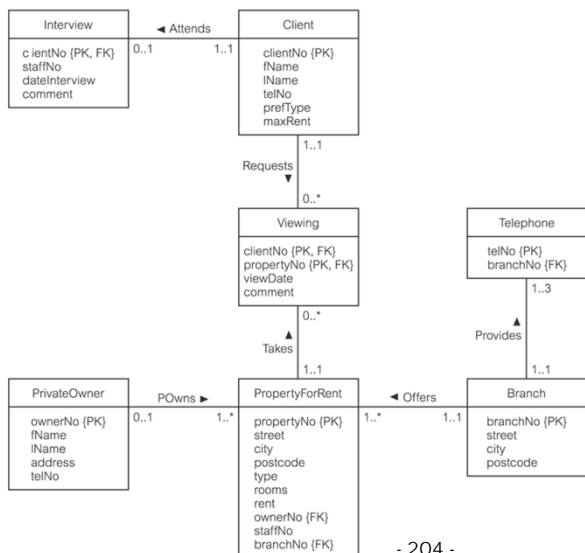
- Denormalizacija se nanaša na dopolnitev relacijske sheme, tako da eni ali več relacij znižamo stopnjo normalne oblike (npr. 3NO → 2NO).
- Nanaša se tudi na primere, ko dve normalizirani relaciji združimo v eno, ki je še vedno normalizirana, vendar zaradi združitve vsebuje več nedefiniranih vrednosti (NULL). (4PNO → 3NO).

K7 – Uvedba nadzorovane redundance...

▪ Koraki denormalizacije:

- K7.1 – združevanje 1:1 povezav
- K7.2 – Podvajanje neosnovnih atributov v povezavah 1:* za zmanjšanje potrebnih stikov.
- K7.3 – Podvajanje tujih ključev v 1:* povezavah za zmanjšanje potrebnih stikov.
- K7.4 – Podvajanje atributov v *: * povezavah za zmanjšanje potrebnih stikov.
- K7.5 – Uvedba ponavljajočih skupin atributov
- K7.6 – Kreiranje tabel, ki predstavljajo izveček osnovne tabele.
- K7.7 – Razbitje relacij.

Primer normaliziranega modela



PODATKOVNE BAZE 2 - VSP

- 204 -

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

Primeri relacij

Branch

branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

Telephone

telNo	branchNo
0207-886-1212	B005
0207-886-1300	B005
0207-886-4100	B005
01224-67125	B007
0141-339-2178	B003
0141-339-4439	B003
0117-916-1170	B004
0208-963-1030	B002

Client

clientNo	fName	lName	telNo	prefType	maxRent
CR76	John	Kay	0207-774-5632	Flat	425
CR56	Aline	Stewart	0141-848-1825	Flat	350
CR74	Mike	Ritchie	01475-392178	House	750
CR62	Mary	Tregear	01224-196720	Flat	600

Interview

clientNo	staffNo	dateInterview	comment
CR56	SG37	11-Apr-00	current lease ends in June
CR62	SA9	7-Mar-00	needs property urgently

PropertyForRent

propertyNo	street	city	postcode	type	rooms	rent	ownerNo	staffNo	branchNo
PA14	16 Holthead	Aberdeen	AB7 5SU	House	6	650	CO46	SA9	B007
PL94	6 Argyll St	London	NW2	Flat	4	400	CO87	SL41	B005
PG4	6 Lawrence St	Glasgow	G11 9QX	Flat	3	350	CO40		B003
PG36	2 Manor Rd	Glasgow	G32 4QX	Flat	3	375	CO93	SG37	B003
PG21	18 Dale Rd	Glasgow	G12	House	5	600	CO87	SG37	B003
PG16	5 Novar Dr	Glasgow	G12 9AX	Flat	4	450	CO93	SG14	B003

PrivateOwner

ownerNo	fName	lName	address	telNo
CO46	Joe	Keogh	2 Fergus Dr, Aberdeen AB2 7SX	01224-861212
CO87	Carol	Farrel	6 Achray St, Glasgow G32 9DX	0141-357-7419
CO40	Tina	Murphy	63 Well St, Glasgow G42	0141-943-1728
CO93	Tony	Shaw	12 Park Pl, Glasgow G4 0QR	0141-225-7025

Viewing

clientNo	propertyNo	viewDate	comment
CR56	PA14	24-May-01	too small
CR76	PG4	20-Apr-01	too remote
CR56	PG4	26-May-01	
CR62	PA14	14-May-01	
CR56	PG36	28-Apr-01	no dining room

PODATKOVNE BAZE 2 - VSP

- 205 -

(b)
Univerza v Ljubljani

K7.1 – Združevanje 1:1 povezav

Če podatke, ki so v različnih relacijah obravnavamo skupaj, je združitev primerna.

ClientInterview
clientNo {PK}
fName
lName
telNo
prefType
maxRent
staffNo
dateInterview
comment

(a)

ClientInterview

clientNo	fName	lName	telNo	prefType	maxRent	staffNo	dateInterview	comment
CR76	John	Kay	0207-774-5632	Flat	425			
CR56	Aline	Stewart	0141-848-1825	Flat	350	SG37	11-Apr-03	current lease ends in June
CR74	Mike	Ritchie	01475-392178	House	750			
CR62	Mary	Tregear	01224-196720	Flat	600	SA9	7-Mar-03	needs property urgently

(b)

Figure 18.2 Combined Client and Interview: (a) revised extract from the relation diagram; (b) combined relation.

K7.2 – Podvajanje neosnovnih atributov

PropertyForRent

propertyNo	street	city	postcode	type	rooms	rent	ownerNo	lName	staffNo	branchNo
PA14	16 Holhead	Aberdeen	AB7 5SU	House	6	650	CO46	Keogh	SA9	B007
PL94	6 Argyll St	London	NW2	Flat	4	400	CO87	Farrel	SL41	B005
PG4	6 Lawrence St	Glasgow	G11 9QX	Flat	3	350	CO40	Murphy		B003
PG36	2 Manor Rd	Glasgow	G32 4QX	Flat	3	375	CO93	Shaw	SG37	B003
PG21	18 Dale Rd	Glasgow	G12	House	5	600	CO87	Farrel	SG37	B003
PG16	5 Novar Dr	Glasgow	G12 9AX	Flat	4	450	CO93	Shaw	SG14	B003

Ko dostopamo do podatkov o nepremičninah, nas v večini primerov zanima tudi ime lastnika (lName)

“Lookup tabele”...

PropertyType

type	description
H	House
F	Flat

Lookup tabele se sestojijo zgolj iz šifre in naziva. Prednosti uporabe Lookup tabel so mnoge. Vseeno obstajajo primeri, ko je smiselno (Lookup tabele ukiniti in) podatke podvajati v osnovnih relacijah. Taki primeri so:

- Ko se do Lookup tabele frekventno dostopa
- Ko so Lookup tabele uporabljene v kritičnih poizvedbah
- Ko je majhna verjetnost, da se bodo po vrednosti spreminjale

PropertyForRent

propertyNo	street	city	postcode	type	rooms	rent	ownerNo	staffNo	branchNo
PA14	16 Holhead	Aberdeen	AB7 5SU	H	6	650	CO46	SA9	B007
PL94	6 Argyll St	London	NW2	F	4	400	CO87	SL41	B005
PG4	6 Lawrence St	Glasgow	G11 9QX	F	3	350	CO40		B003
PG36	2 Manor Rd	Glasgow	G32 4QX	F	3	375	CO93	SG37	B003
PG21	18 Dale Rd	Glasgow	G12	H	5	600	CO87	SG37	B003
PG16	5 Novar Dr	Glasgow	G12 9AX	F	4	450	CO93	SG14	B003

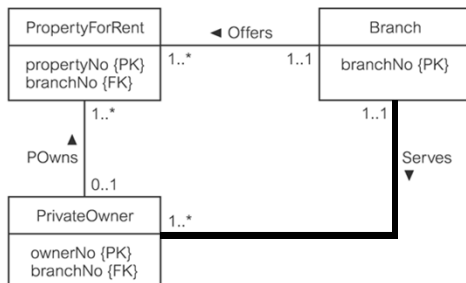
“Lookup tabele”

PropertyForRent

propertyNo	street	city	postcode	type	description	rooms	rent	ownerNo	staffNo	branchNo
PA14	16 Holhead	Aberdeen	AB7 5SU	H	House	6	650	CO46	SA9	B007
PL94	6 Argyll St	London	NW2	F	Flat	4	400	CO87	SL41	B005
PG4	6 Lawrence St	Glasgow	G11 9QX	F	Flat	3	350	CO40		B003
PG36	2 Manor Rd	Glasgow	G32 4QX	F	Flat	3	375	CO93	SG37	B003
PG21	18 Dale Rd	Glasgow	G12	H	House	5	600	CO87	SG37	B003
PG16	5 Novar Dr	Glasgow	G12 9AX	F	Flat	4	450	CO93	SG14	B003

Figure 18.5 Modified PropertyForRent relation with duplicated description attribute.

K7.3 - Podvajanje tujih ključev



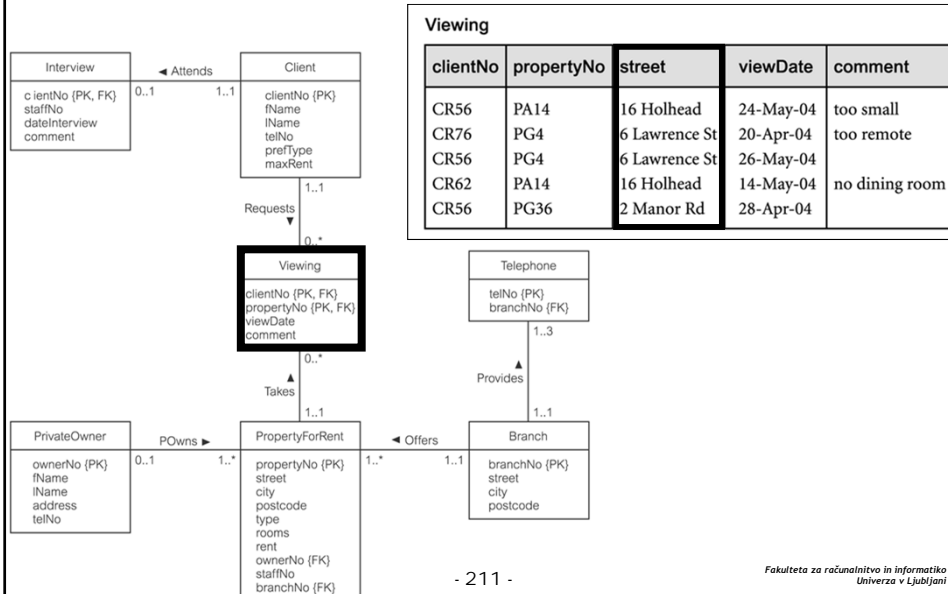
(a)

PrivateOwner

ownerNo	fName	lName	address	telNo	branchNo
CO46	Joe	Keogh	2 Fergus Dr, Aberdeen AB2 7SX	01224-861212	B007
CO87	Carol	Farrel	6 Achray St, Glasgow G32 9DX	0141-357-7419	B003
CO40	Tina	Murphy	63 Well St, Glasgow G42	0141-943-1728	B003
CO93	Tony	Shaw	12 Park Pl, Glasgow G4 0QR	0141-225-7025	B003

(b)

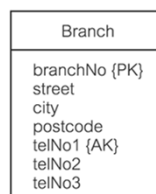
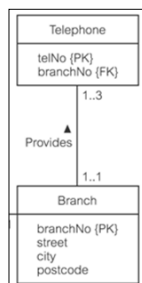
K7.4 - Podvajanje atributov v povez. *:*



Viewing

clientNo	propertyNo	street	viewDate	comment
CR56	PA14	16 Holhead	24-May-04	too small
CR76	PG4	6 Lawrence St	20-Apr-04	too remote
CR56	PG4	6 Lawrence St	26-May-04	
CR62	PA14	16 Holhead	14-May-04	no dining room
CR56	PG36	2 Manor Rd	28-Apr-04	

K7.5 - Uvedba ponavljajočih skupin



(a)

Branch

branchNo	street	city	postcode	telNo1	telNo2	telNo3
B005	22 Deer Rd	London	SW1 4EH	0207-886-1212	0207-886-1300	0207-886-4100
B007	16 Argyll St	Aberdeen	AB2 3SU	01224-67125		
B003	163 Main St	Glasgow	G11 9QX	0141-339-2178	0141-339-4439	
B004	32 Manse Rd	Bristol	BS99 1NZ	0117-916-1170		
B002	56 Clover Dr	London	NW10 6EU	0208-963-1030		

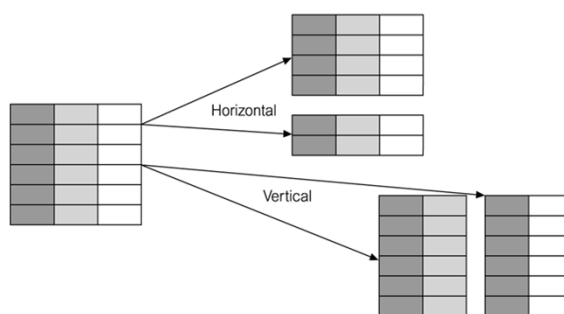
(b)

K7.6 – Uporaba izvlečkov

- Številne poizvedbe in poročila zahtevajo dostop do več osnovnih relacij ter kompleksne povezave med njimi.
- Z namenom izboljšanja učinkovitosti je v določenih primerih smiselno uvesti novo denormalizirano relacijo, ki vsebuje podatke iz več osnovnih relacij.

K7.7 – Razbitje relacij

- Za povečanje učinkovitosti nad relacijami z zelo velikim številom n-teric uporabimo pristop, kjer relacijo razbijemo na manjše dele - particije.
- Dva načina delitve sta:
 - Horizontalni in
 - Vertikalni.



Primer – particije v SUPB Oracle

```
CREATE TABLE ArchivedPropertyForRentPartition(  
  propertyNo VARCHAR2(5) NOT NULL,  
  street VARCHAR2(25) NOT NULL,  
  ...  
  PRIMARY KEY (propertyNo),  
  FOREIGN KEY (ownerNo) REFERENCES PrivateOwner(ownerNo),  
  ...  
  PARTITION BY HASH (branchNo) (  
    PARTITION b1 TABLESPACE TB01,  
    PARTITION b2 TABLESPACE TB02,  
    ...  
  );
```

Prednosti razbitja relacije na particije

- Uporaba particioniranja prinaša številne prednosti:
 - Boljša porazdelitev vnosa (load balancing)
 - Večja učinkovitosti (manj podatkov za obdelavo, paralelnost izvajanja,...)
 - Boljša razpoložljivost
 - Boljša obnovljivost
 - Več možnosti za zagotavljanje varnosti

Slabosti razbitja relacije na particije

- Particioniranje ima tudi slabosti:
 - Kompleksnost (particije niso vedno transparentne za uporabnike...)
 - Slabša učinkovitost (pri poizvedbah, kjer je potrebno poizvedovati po več particijah, je učinkovitost slabša)
 - Podvajanje podatkov (pri vertikalnem particioniranju)