Osnove podatkovnih baz

Skripta za (avditorne) vaje pri predmetih iz podatkovnih baz

Lovro Šubelj, Bojan Klemenc, Aljaž Zrnec in Marko Bajec
Laboratorij za podatkovne tehnologije, Fakulteta za računalništvo in informatiko, Univerza v Ljubljani

28. september 2010

Delovna skripta za (avditorne) vaje pri predmetih iz podatkovnih baz na univerzitetnem študiju na Fakulteti za računalništvo in informatiko. Razdelki označeni s **[sOPB]** in **[sPB1]** se **ne** obravnavajo pri predmetih "Osnove podatkovnih baz" in "Podatkovne baze 1" na starem univerzitetnem študiju, razdelki označeni z **[bOPB]** pa **ne** na novem (bolonjskem) univerzitetnem študiju.

Za napake, komentarje ali predloge pišite na *lovro.subelj@fri.uni-lj.si*. Hvala.

Kazalo

Poglavje 1

Relacijski podatkovni model

Temelje *relacijskega podatkovnega modela* je postavil Codd (1970). V relacijskem podatkovnem modelu so podatki logično strukturirani v *relacije*. Osnovni koncepti modela so *relacija*, *atribut*, *vrednostna množica*, *n-terica* (zapis), *relacijska shema*, *odvisnosti med atributi*, *ključi*, itd.

Relacije navadno predstavimo s *tabelo* (glej 1.1).

r

A_1	A_2	...	A_n
...	...	...	...
...	...	...	...

Tabela 1.1: Relacija r z atributi $A_1, A_2 \ldots A_n$.

1.1 Relacija

Naj bo o nek *objekt* iz množice objektov $\mathcal{O}$, $o \in \mathcal{O}$, ter naj bodo $A_i, 1 \leq i \leq n$, (njegove) lastnosti, ki jih imenujemo *atributi*.

$$A_i : \mathcal{O} \to D_{A_i},$$

kjer je D_{A_i} *vrednostna množica* atributa A_i (*območje*, *domena* atributa A_i).

Vsakemu objektu $o \in \mathcal{O}$ tako pripada neka n-terica $z_o \in D_{A_1} \times D_{A_2} \ldots \times D_{A_n}$ definirana kot

$$z_o = (A_1(o), A_2(o) \ldots A_n(o)).$$

Definicija. Relacijo r formalno definiramo kot *preslikavo*, ki slika iz kartezijskega produkta vrednostnih množic atributov v binarno množico $\{\top, \bot\}$ (res, ni res),

$$r : D_{A_1} \times D_{A_2} \ldots \times D_{A_n} \to \{\top, \bot\}.$$

Objekt $o \in \mathcal{O}$, oziroma pripadajoča n-terica z_o, je v relaciji, kadar velja $r(z_o) = \top$. Sicer objekt o ni v relaciji.

Definicija. Relacijo r pa lahko definiramo tudi kot *množico* n-teric, ki pripadajo objektom v relaciji.

$$\{z_o |\ o \in \mathcal{O} \wedge r(z_o) = \top\}$$

Vrstni red n-teric v relaciji je tako nepomemben.

Pojmi. *Stopnja* relacije r je enaka številu atributov, s katerimi opišemo objekte (označimo $deg(r)$). *Moč* ali *števnost* relacije r je enaka številu n-teric v relaciji (označimo $|r|$).

Lastnosti. Lastnosti relacij:

- ime relacije je enolično,
- ime atributa relacije je enolično,
- vsaka n-terica relacije je enolična (po definiciji),
- vrstni red n-teric v relaciji je nepomemben (po definiciji),
- vrstni red atributov relacije je nepomemben (po dogovoru).

1.2 Relacijska shema

Definicija. Imena in domene atributov relacije imenujemo *relacijska shema*. Relacijska shema R predstavlja semantično komponento relacije r in je definirana kot

$$\{A_1 : D_{A_1}, A_2 : D_{A_2} \dots A_n : D_{A_n}\} = R(A_1, A_2 \dots A_n) = A_1 A_2 \dots A_n.$$

Vsaki relaciji pripada natanko ena shema, dočim lahko neki shemi pripada več relacij.

Sh je funkcija, ki vsaki relaciji priredi pripadajočo relacijsko shemo.

$$Sh : \{r\} \rightarrow \{R\}$$

Poglavje 2

Formalni poizvedovalni jeziki

Poizvedovalni jeziki so specializirani jeziki za izvajanje poizvedb nad nekim naborom podatkov. Poznamo dva formalna (teoretična) poizvedovalna jezika:

Relacijska algebra (*RA*): Poizvedbe opredeljujejo zaporedje operacij, ki naj se izvedejo nad neko relacijo (oziroma množico relacij). Poizvedbe so tako opredeljene na *proceduralen način* (opis postopka).

Relacijski račun (*RR*): Poizvedba je zgolj opis želenega rezultata. Poizvedbe so tako opredeljene na *deklarativen način* (opis rezultata).

Relacijska algebra je osnova za *transformacijski jezik SQL* (*Structured Query Language*), relacijski račun pa je osnova za *grafični jezik QBE* (*Query-by-example*).

2.1 Relacijska algebra

2.1.1 Uvod v relacijsko algebro

V relacijski algebri poizvedbe predstavljajo zaporedje operacij, ki naj se izvedejo nad neko relacijo (oziroma množico relacij). Poizvedbe so tako opredeljene na proceduralen način (opis postopka). Rezultati in operandi operacij v relacijski algebri so relacije.

Operacije relacijske algebre razdelimo v naslednje skupine:

Enostavne (unarne) operacije: projekcija, selekcija, preimenovanje

Operacije iz teorije množic: unija, razlika, presek

Operacije množenja: kartezični produkt, θ-stik, ekvistik, naravni stik, deljenje

Dodatne operacije: polstik, odprti stik, agregacija, grupiranje, itd.

2.1.2 Enostavne (unarne) operacije

Projekcija

$$\pi_{\mathbf{S}}(\mathbf{r})$$

Opis. Projekcija relacije r, določena s seznamom atributov S.

Definicija. $\pi_{\mathbf{S}}(\mathbf{r}) = \{\mathbf{z_S} | \mathbf{z_r} \in \mathbf{r} \wedge \mathbf{z_S} \subseteq \mathbf{z_r}\}$, kjer je z_S (pod-)n-terica, sestavljena iz atributov v S
Shema. $Sh(\pi_S(r)) = S$
Pogoji. Atributi v S so v $Sh(r)$.
Zgledi. Glej 2.1.

r

A	B	C
1	1	1
1	1	2
1	2	3

$\rightarrow$

$\pi_{A,B}(r)$

A	B
1	1
1	2

$\pi_A(r)$

A
1

Tabela 2.1: Projekcije relacije r.

Selekcija

$$\sigma_\theta(\mathbf{r})$$

Opis. Selekcija relacije r, določena s pogojem θ[1].

Definicija. $\sigma_\theta(\mathbf{r}) = \{\mathbf{z_r} | \mathbf{z_r} \in \mathbf{r} \wedge \mathbf{z_r} \text{ zadošča pogoju } \theta\}$
Shema. $Sh(\sigma_\theta(r)) = Sh(r)$
Pogoji. Atributi v pogoju θ so v $Sh(r)$.
Zgledi. Glej 2.2.

r

A	B	C
1	1	1
1	1	2
1	2	3

$\rightarrow$

$\sigma_{A \neq C \wedge B < 2}(r)$

A	B	C
1	1	2

Tabela 2.2: Selekcija relacije r.

Preimenovanje

$$\rho_{\mathbf{s(S)}}(\mathbf{r})$$

Opis. Preimenovanje relacije r (s je novo ime relacije, S pa seznam novih imen atributov).

Definicija. $\rho_{\mathbf{s(S)}}(\mathbf{r}) = \mathbf{r}$
Shema. $|Sh(\rho_{s(S)}(r))| = |Sh(r)| \ldots$
Pogoji. Dolžina seznama S je enaka $|Sh(r)|$.
Komentar. Poznamo tudi operacijo prirejanja $\leftarrow$, $s(S) \leftarrow r$.
Zgledi. Glej 2.3.

[1] θ je logični pogoj sestavljen iz operacij primerjanja, $<, >, \leq, \geq, =, \neq$, in osnovnih logičnih operacij, $\neg, \wedge, \vee$.

r

A	B	C
1	1	1
1	1	2
1	2	3

$\rightarrow$

$\rho_{s(D,E,F)}(r)$

D	E	F
1	1	1
1	1	2
1	2	3

Tabela 2.3: Preimenovanje relacije r.

2.1.3 Operacije iz teorije množic

Unija

$\mathbf{r \cup s}$

Opis. Unija relacij r in s.

Definicija. $\mathbf{r \cup s = \{z_{rs} | \, z_{rs} \in r \lor z_{rs} \in s\}}$
Shema. $Sh(r \cup s) = Sh(r)$
Pogoji. Relaciji r in s sta medsebojno kompatibilni[2].
Zgledi. Glej 2.4.

r

A	B	C
1	2	3
4	5	6
7	8	9

s

D	E	F
1	2	3
2	4	6
3	6	9

$\rightarrow$

$r \cup s$

A	B	C
1	2	3
2	4	6
3	6	9
4	5	6
7	8	9

Tabela 2.4: Unija relacij r in s.

Razlika

$\mathbf{r - s}$

Opis. Razlika relacij r in s.

Definicija. $\mathbf{r - s = \{z_r | \, z_r \in r \land z_r \notin s\}}$
Shema. $Sh(r - s) = Sh(r)$
Pogoji. Relaciji r in s sta medsebojno kompatibilni[2].
Zgledi. Glej 2.5.

Presek

$\mathbf{r \cap s}$

[2]Relaciji r in s sta *medsebojno kompatibilni*, če imata enako število atributov ($deg(r) = deg(s)$), istoležni atributi pa imajo enake domene.

r

A	B	C
1	2	3
4	5	6
7	8	9

s

D	E	F
1	2	3
2	4	6
3	6	9

$\rightarrow$

$r - s$

A	B	C
4	5	6
7	8	9

Tabela 2.5: Razlika relacij r in s.

Opis. Presek relacij r in s.

Definicija. $\mathbf{r} \cap \mathbf{s} = \mathbf{r} - (\mathbf{r} - \mathbf{s})$
Shema. $Sh(r \cap s) = Sh(r)$
Pogoji. Relaciji r in s sta medsebojno kompatibilni[2].
Zgledi. Glej 2.6.

r

A	B	C
1	2	3
4	5	6
7	8	9

s

D	E	F
1	2	3
2	4	6
3	6	9

$\rightarrow$

$r \cap s$

A	B	C
1	2	3

Tabela 2.6: Presek relacij r in s.

2.1.4 Operacije množenja

Kartezični produkt

$\mathbf{r} \times \mathbf{s}$

Opis. Kartezični produkt relacij r in s je relacija, ki vsebuje po eno n-terico za vsak par n-teric iz relacij r in s (n-terica je enaka stiku n-teric iz relacij r in s).

Definicija. $\mathbf{r} \times \mathbf{s} = \{\mathbf{z_r}.\mathbf{z_s} | \mathbf{z_r} \in \mathbf{r} \wedge \mathbf{z_s} \in \mathbf{s}\}$
Shema. $Sh(r \times s) = Sh(r).Sh(s)$
Zgledi. Glej 2.7.

r

A	B	C
1	2	3
4	5	6
7	8	9

s

D	E
1	2
3	4

$\rightarrow$

$r \times s$

A	B	C	D	E
1	2	3	1	2
1	2	3	3	4
4	5	6	1	2
4	5	6	3	4
7	8	9	1	2
7	8	9	3	4

Tabela 2.7: Kartezični produkt relacij r in s.

Pogojni stik

$\mathbf{r} \bowtie_\theta \mathbf{s}$

Opis. Pogojni stik relacij r in s je enak kartezičnemu produktu pri čimer pa ohranimo le tiste n-terice, ki zadoščajo pogoju θ[1].

Definicija. $\mathbf{r} \bowtie_\theta \mathbf{s} = \sigma_\theta(\mathbf{r} \times \mathbf{s})$
Shema. $Sh(r \bowtie_\theta s) = Sh(r).Sh(s)$
Pogoji. Atributi v pogoju θ so v $Sh(r) \cup Sh(s)$.
Komentar. Pogojni stik navadno imenujemo kar θ-stik.
Zgledi. Glej 2.8.

r

A	B	C
1	2	3
4	5	6
7	8	9

s

D	E
1	2
3	4

$\rightarrow$

$r \bowtie_{C \geq D \wedge A=E} s$

A	B	C	D	E
4	5	6	3	4

Tabela 2.8: θ-stik relacij r in s.

Ekvistik

$\mathbf{r} \bowtie_{\theta_=} \mathbf{s}$

Opis. Ekvistik relacij r in s je θ-stik pri katerem pa pogoj $\theta_=$ vsebuje le enakosti[3].

Definicija. $\mathbf{r} \bowtie_{\theta_=} \mathbf{s} = \sigma_{\theta_=}(\mathbf{r} \times \mathbf{s})$, kjer $\theta_=$ vsebuje le enakosti
Shema. $Sh(r \bowtie_{\theta_=} s) = Sh(r).Sh(s)$
Pogoji. Atributi v pogoju $\theta_=$ so v $Sh(r) \cup Sh(s)$.
Zgledi. Glej 2.9.

r

A	B	C
1	2	3
4	5	6
7	8	9

s

D	E
1	2
3	4

$\rightarrow$

$r \bowtie_{C=D} s$

A	B	C	D	E
1	2	3	3	4

Tabela 2.9: Ekvistik relacij r in s.

Naravni stik

$\mathbf{r} \bowtie \mathbf{s}$

Opis. Naravni stik relacij r in s je ekvistik po vseh skupnih atributih relacij pri čimer pa ohranimo le eno pojavitev skupnih atributov (ni podvojenih atributov).

[3] $\theta_=$ je logični pogoj sestavljen iz operacij enakosti =, in logičnih operacij $\wedge$.

Definicija. $\mathbf{r} \bowtie \mathbf{s} = \pi_{\mathbf{R} \cup \mathbf{S}}(\sigma_{\mathbf{r.A_1=s.A_1 \wedge r.A_2=s.A_2 \wedge \ldots}}(\mathbf{r} \times \mathbf{s}))$
Shema. $Sh(r \bowtie s) = Sh(r) \cup Sh(s)$
Komentar. V kolikor relaciji r in s nimata skupnih atributov je naravni stik enak kartezičnemu produktu.
Zgledi. Glej 2.10.

r

A	B	C
1	2	3
4	5	6
7	8	9

s

C	D	E
6	1	2
9	3	4

→

$r \bowtie s$

A	B	C	D	E
4	5	6	1	2
7	8	9	3	4

Tabela 2.10: Naravni stik relacij r in s.

Deljenje

$\mathbf{r}/\mathbf{s}$

Opis. Količnik relacij r in s vsebuje vse tiste (pod-)n-terice iz r, sestavljene iz atributov v $Sh(r) - Sh(s)$, ki pokrijejo relacijo s (glej 2.11).

Definicija. $\mathbf{r}/\mathbf{s} = \pi_{\mathbf{R-S}}(\mathbf{r}) - \pi_{\mathbf{R-S}}((\pi_{\mathbf{R-S}}(\mathbf{r}) \times \mathbf{s}) - \mathbf{r})$
Shema. $Sh(r/s) = Sh(r) - Sh(s)$
Pogoji. Velja $Sh(s) \subset Sh(r)$.
Zgledi. Glej 2.11 in 2.12.

opravil

Vpisna	Sifra	...
10000	101	...
10001	101	...
10001	102	...
10002	101	...
10003	102	...
...	...	...

izpit

Sifra	...
101	...
102	...

→

$\pi_{Vpisna,Sifra}(opravil)/\pi_{Sifra}(izpit)$

Vpisna
10001
...

Tabela 2.11: Kdo je že opravil vse izpite? Primer uporabe deljenja v praksi.

2.1.5 Dodatne operacije

Polstik

$\mathbf{r} \rhd \mathbf{s}, \mathbf{r} \rhd_\theta \mathbf{s}$ (polstik, pol-θ-stik)

Opis. Polstik relacij r in s je enak naravnemu stiku pri čimer pa ohranimo le atribute (podatke) iz leve relacije r (v rezultatu so vse n-terice v r, ki se pojavijo v naravnem stiku relacij r in s; podobno za pol-θ-stik).

Definicija. $\mathbf{r} \rhd \mathbf{s} = \pi_{\mathbf{R}}(\mathbf{r} \bowtie \mathbf{s}), \mathbf{r} \rhd_\theta \mathbf{s} = \pi_{\mathbf{R}}(\mathbf{r} \bowtie_\theta \mathbf{s})$
Shema. $Sh(r \rhd s) = Sh(r \rhd_\theta s) = Sh(r)$
Pogoji. Atributi v pogoju θ so v $Sh(r) \cup Sh(s)$.
Zgledi. Glej 2.13 in 2.14.

r

A	B	C	D
1	1	2	1
1	3	4	1
2	3	4	1
3	1	2	1
4	1	2	1
4	3	4	2
5	5	6	1
6	7	8	1

s

B	C
1	2
3	4

$\rightarrow$

r/s

A	D
1	1

Tabela 2.12: Količnik relacij r in s.

r

A	B	C
1	2	3
4	5	6
7	8	9

s

C	D	E
6	1	2
9	3	4

$\rightarrow$

$r \rhd s$

A	B	C
4	5	6
7	8	9

Tabela 2.13: Polstik relacij r in s.

r

A	B	C
1	2	3
4	5	6
7	8	9

s

D	E
1	2
3	4

$\rightarrow$

$r \rhd_{C \geq D \wedge A = E} s$

A	B	C
4	5	6

Tabela 2.14: Pol-θ-stik relacij r in s.

Odprti stik

$$\mathbf{r} \rtimes \mathbf{s}, \mathbf{r} \ltimes \mathbf{s}, \mathbf{r} \times \mathbf{s} \text{ (levi, desni, popolni)}$$

Opis. Levi odprti stik relacij r in s je enak naravnemu stiku pri čimer pa v rezultat dodamo še vse n-terice iz leve relacije r, ki se sicer nebi uvrstile v rezultat (podobno za desni, popolni). Neznane vrednosti atributov pri tem postavimo na *NULL*.

Definicija. $\mathbf{r} \rtimes \mathbf{s} = \mathbf{r} \bowtie \mathbf{s} \cup \{\mathbf{z_r}.\mathbf{z_N} | \mathbf{z_r} \in \mathbf{r} - \pi_{\mathbf{R}}(\mathbf{r} \bowtie \mathbf{s})\}$, kjer je z_N n-terica samih *NULL* vrednosti (podobno za desni, popolni)

Shema. $Sh(r \rtimes s) = Sh(r \ltimes s) = Sh(r \times s) = Sh(r) \cup Sh(s)$

Komentar. Pri levem odprtem stiku tako ohranimo vse podatke iz leve relacije r (podobno za desni, popolni). Odprti stik imenujemo tudi funkcijski ali zunanji stik.

Zgledi. Glej 2.15, 2.16 in 2.17.

r

A	B	C
1	2	3
4	5	6
7	8	9

s

C	D	E
6	1	2
9	3	4
12	5	6

→ $r \rtimes s$

A	B	C	D	E
1	2	3	*NULL*	*NULL*
4	5	6	1	2
7	8	9	3	4

Tabela 2.15: Levi odprti stik relacij r in s.

r

A	B	C
1	2	3
4	5	6
7	8	9

s

C	D	E
6	1	2
9	3	4
12	5	6

→ $r \ltimes s$

A	B	C	D	E
4	5	6	1	2
7	8	9	3	4
NULL	*NULL*	12	5	6

Tabela 2.16: Desni odprti stik relacij r in s.

r

A	B	C
1	2	3
4	5	6
7	8	9

s

C	D	E
6	1	2
9	3	4
12	5	6

→ $r \times s$

A	B	C	D	E
1	2	3	*NULL*	*NULL*
4	5	6	1	2
7	8	9	3	4
NULL	*NULL*	12	5	6

Tabela 2.17: Popolni odprti stik relacij r in s.

Agregacija

$$\tau_{\mathbf{AS}}(\mathbf{r})$$

Opis. Agregacija relacije r uporabi seznam agregacijskih funkcij AS nad r (glej 2.18). AS je oblike $af_1A_1, af_2A_2 \ldots$, kjer so A_i atributi relacije r, af_i pa agregacijske funkcije *COUNT*, *SUM*, *AVG*, *MIN* in *MAX*.

Definicija. ...
Shema. $|Sh(\tau_{AS}(r))| = |AS| \ldots$
Pogoji. Atributi v seznamu AS so v $Sh(r)$.
Zgledi. Glej 2.18.

r

A	B	C	D
1	a	100	1
1	a	200	1
1	b	200	1
2	a	100	1
2	b	100	1
3	a	100	1
3	*NULL*	100	1
4	b	100	1
4	b	200	1
4	*NULL*	200	1

→

$\tau_{MAX\ A, COUNT\ B, SUM\ C}(r)$

...	...	...
4	8	1400

Tabela 2.18: Agregacija relacije r.

Grupiranje

$$_{\mathbf{GA}}\tau_{\mathbf{AS}}(\mathbf{r})$$

Opis. Agregacija z grupiranjem nad relacijo r uporabi seznam agregacijskih funkcij AS nad grupami relacije r, določenimi s seznamom grupirnih atributov GA (glej 2.19). GA je oblike $A_1, A_2 \ldots$, kjer so A_i atributi relacije r.

Definicija. ...
Shema. $|Sh(_{GA}\tau_{AS}(r))| = |GA| + |AS| \ldots$
Pogoji. Atributi v seznamih GA in AS so v $Sh(r)$, vsak atribut pa je lahko le v enem od seznamov.
Zgledi. Glej 2.19 in 2.20.

opravil

Vpisna	Sifra	Ocena	...
10000	101	10	...
10001	101	8	...
10001	102	9	...
10002	101	7	...
10003	102	10	...
...	...	...	...

→

$\rho_{(Vpisna, Stevilo, Povprecje)}(_{Vpisna}\tau_{COUNT\ Sifra, AVG\ Ocena}(opravil))$

Vpisna	Stevilo	Povprecje
10000	1	10
10001	2	8.5
10002	1	7
10003	1	10
...	...	...

Tabela 2.19: Povprečne ocene študentov. Primer uporabe agregacije z grupiranjem v praksi.

2.1.6 Prioriteta operacij

Prioriteta operacij je naslednja (Elmasri & Navathe, 2007) (od najvišje prioritete do najnižje prioritete):

r

A	B	C	D
1	a	100	1
1	a	200	1
1	b	200	1
2	a	100	1
2	b	100	1
3	a	100	1
3	*NULL*	100	1
4	b	100	1
4	b	200	1
4	*NULL*	200	1

$\rightarrow$

$_{A,D}\tau_{COUNT\ B, SUM\ C}(r)$

A	*D*	...	...
1	1	3	500
2	1	2	200
3	1	1	200
4	1	2	500

Tabela 2.20: Agregacija z grupiranjem nad relacijo r.

1. enostavne (unarne) operacije: projekcija, selekcija, preimenovanje

2. operacije množenja: kartezični produkt, θ-stik, naravni stik, deljenje

3. operacije iz teorije množic: presek, unija, razlika

4. dodatne operacije: agregacija (z grupiranjem)

2.1.7 Naloge

OPERATER Glej B.1.

1. Katere stranke kupujejo pri operaterju Mobitel?

2. Pri katerih operaterjih prodajajo Janezov najljubši telefon?

3. Katere telefone lahko kupuje Petra, glede na to, da kupuje le pri nekaterih operaterjih, kateri prodajajo le nekatere telefone?

4. Katere stranke kupujejo pri vseh operaterjih? Predpostavimo, da so vsi operaterji navedeni v relaciji p.

5. Pri katerih operaterjih prodajajo vse Janezove najljubše telefone? Predpostavimo, da je v relaciji n več n-teric, ki ustrezajo Janezu.

6. Katere stranke kupujejo zgolj pri enem operaterju?

GSM Glej B.2.

1. Poiščite imena in priimke strank, ki so iz Kranja in so stare več kot 18 let.

2. Poiščite imena in priimke strank, ki so kadarkoli kaj kupile.

3. Poiščite imena in priimke strank, ki niso še nikoli nič kupile. Predpostavimo, da imamo stranke že vnaprej podane v relaciji s.

4. Poiščite imena in priimke prodajalcev, ki so do sedaj prodali le GSM aparate tipa Nokia.

5. Za vsak tip GSM aparata izpišite skupno število prodanih kosov.

NAROCILO Glej B.3.

1. Poiščite imena in priimke agentov, ki niso še nikoli nič prodali.

2. Poiščite imena in priimke agentov, ki so kdaj prodali izdelke stranki iz Kopra ali Ljubljane.

3. Poiščite imena izdelkov, ki so jih kupile stranke, katerim je odobren vsaj 20% popust.

HOTEL Glej B.5.

1. Poiščite številke vseh enoposteljnih sob ($RType$ = 1), katerih cena je pod 50€ na dan.

2. Poiščite številke, cene ter tipe sob v hotelu Lev.

3. Poiščite imena gostov, ki se trenutno nahajajo v hotelu Lev (spremenljivka *today*). Izpišite tudi cene in tipe sob, v katerih se nahajajo.

4. Za vsak hotel izpišite ime, skupno število sob ter povprečno ceno sobe.

5. Izpišite vse podatke o vseh sobah v hotelu Lev (*RNo*, *RType* in *RPrice*), vključno z imenom gosta v sobi, v kolikor je soba trenutno zasedena (sicer *NULL*).

AIRCRAFT Glej B.6.

1. Poiščite vse podatke o letalih z imenom Boeing.
2. Poiščite vse podatke o letalih Boeing 737.
3. Poiščite številke pilotov, ki imajo certifikat za letenje z Boeing-i.
4. Poiščite številke in imena letal, ki lahko brez postankov letijo od Ljubljane do New York-a ($ARange \geq FDistance$). Predpostavimo, da obstaja let od Ljubljane do New York-a.
5. Poiščite številke zaposlenih, ki imajo najvišjo plačo.
6. Poiščite številke zaposlenih, ki imajo certifikat za letenje za vsaj 5 letal.

Drugo

1. Naj bosta r, s neki relaciji ter $R = ABC$, $S = CDE$ pripadajoči relacijski shemi. Moč relacij r, s je enaka N_r, N_s zaporedoma (velja $N_r > N_s > 0$).

   (i) $r \bowtie_{A<D} s$
   (ii) $r \bowtie_{r.C=s.C} s$
   (iii) $r \bowtie s$
   (iv) $r \rhd s$
   (v) $r \rtimes s$

   Podajte relacijske sheme relacij, ki jih dobite kot rezultat zgornjih izrazov relacijske algebre.

   Ali so naslednje trditve (vedno) resnične?

   (a) Moč relacij (ii) in (iii) je lahko različna.
   (b) Moč relacij (iii) in (iv) je lahko različna.
   (c) Vsaka od relacij je lahko prazna.
   (d) Relacija (v) lahko ne vsebuje *NULL* vrednosti.
   (e) Moč relacije (v) je večja ali enaka moči relacije r.

2. Naj bosta r, s neki relaciji ter R, S pripadajoči relacijski shemi. Moč relacij r, s je enaka N_r, N_s zaporedoma (velja $0 < N_r < N_s$). Podajte najmanjšo in največjo možno moč relacij, ki jih dobite kot rezultat spodnjih izrazov relacijske algebre. Za vsak izraz podajte tudi morebitne pogoje, ki morajo veljati, da je izraz smiseln.

   (a) $\tau_{COUNT\ A} r$

   (b) $\rho_{t(A,B,C)} r$

   (c) $r \cup s$

   (d) $r \bowtie s$

   (e) $r \rtimes s$

   (f) $r \ltimes s$

   (g) $r \rhd s$

   (h) r/s

   (i) s/r

   (j) $r \rhd s \cup r$

   (k) $r \rhd (s \cup r)$

   (l) $r \rhd s - r \cap s$

   (m) $r \cup s - r \rhd s$

(n) $r \rtimes r \rtimes s$

(o) $r \rtimes (r \rtimes s)$

(p) $r \rtimes r \ltimes r$

(q) $r \rtimes s - r \bowtie s$

2.2 Relacijski račun

2.2.1 Uvod v relacijski račun

V relacijskem računu poizvedbe predstavljajo opis lastnosti relacije, ki jo želimo v rezultatu. Poizvedbe so tako opredeljene na *deklarativen način* (opis rezultata).

Ločimo dve vrsti relacijskega računa (glede na vrsto uporabljenih spremenljivk):

n-terični relacijski račun (*NRR*): Spremenljivke predstavljajo n-terice relacije (n-terične spremenljivke).

Domenski relacijski račun (*DRR*): Spremenljivke predstavljajo atribute relacije (domenske ali komponentne spremenljivke).

2.2.2 n-terični relacijski račun

V n-teričnem relacijskem računu uporabljamo spremenljivke, ki predstavljajo n-terice neke relacije (predstavljajo zapise v tabeli). Imenujemo jih *n-terične spremenljivke*.

Zgledi. Relacija *oseba*, relacijska shema $OSEBA(Ime, Priimek, Starost)$.

$$\{O|\ oseba(O)\}$$

$$\{O.Ime, O.Priimek|\ oseba(O) \land (O.Starost < 18 \lor O.Starost \geq 64)\}$$

$$\{O|\ oseba(O) \land \neg\exists O1(oseba(O1) \land O1.Starost < O.Starost)\}$$

$$\{O|\ oseba(O) \land \neg\exists O1(O1 \in oseba \land O1[Starost] < O[3])\}$$

Definicija. Poizvedbe v n-teričnem relacijskem računu so oblike

$$\{\mathbf{X}|\ \mathbf{F}(\mathbf{X})\},$$

kjer je

X n-terična spremenljivka, katere domena je določena z relacijsko shemo ustrezne relacije,

F(**X**) formula predikatnega računa prvega reda $(\forall, \exists, \neg, \wedge, \vee)$.

Iskano relacijo $\{X|\ F(X)\}$ tako sestavljajo vse tiste n-terice X, ki zadoščajo formuli F.
Spremenljivke na levi strani izraza $\{X|\ F(X)\}$ imenujemo *proste spremenljivke*. Vse ostale spremenljivke morajo biti vezane znotraj formule F z uporabo $\forall, \exists$ (*vezane spremenljivke*)!

2.2.3 Domenski relacijski račun

V domenskem relacijskem računu uporabljamo spremenljivke, ki predstavljajo vrednosti atributov neke relacije (predstavljajo celice v tabeli). Imenujemo jih *domenske* ali *komponentne spremenljivke*.

Zgledi. Relacija *oseba*, relacijska shema $OSEBA(Ime, Priimek, Starost)$.

$$\{IME, PRIIMEK, STAROST|\ oseba(IME, PRIIMEK, STAROST)\}$$

$$\{IME, PRIIMEK|\ \exists STAROST(oseba(IME, PRIIMEK, STAROST) \wedge (STAROST < 18 \vee STAROST \geq 64))\}$$

$$\{IME, PRIIMEK, STAROST|\ oseba(IME, PRIIMEK, STAROST) \wedge \neg \exists IME1\ \exists PRIIMEK1\ \exists STAROST1(oseba(IME1, PRIIMEK1, STAROST1) \wedge STAROST1 < STAROST)\}$$

2.2.4 Nevarni izrazi

Izrazi relacijskega računa omogočajo generiranje neskončnih relacij. Takim izrazom pravimo *nevarni izrazi*.

Zgledi.

$$\{O|\ \neg oseba(O)\}$$

$$\{O|\ oseba(O) \vee O[7] > 100\}$$

$$\{X|\ \exists Y(Y[1] = X[1] \wedge Y[7] > 100)\}$$

Definicija. Izraz je *varen*, če velja, da se vrednosti, ki se nahajajo v rezultatu, nahajajo tudi v *domeni izraza*, ter pa, da nam pri formulah z $\forall, \exists$ ni potrebno preiskati neskončno mnogo možnosti.

2.2.5 Naloge

NESRECA Glej B.4.

1. Poiščite številke policistov iz policijske postaje Ljubljana-Moste.

2. Poiščite imena in priimke oseb, ki so mlajše od 18 let ter niso bile še nikoli udeležene v nobeni prometni nesreči. Predpostavimo, da imamo osebe že vnaprej podane v relaciji o.

3. Poiščite imena in priimke oseb, ki so se zaletele pred 1.1.2000 v Ljubljani.

4. Poiščite številke oseb, ki so bile udeležene v kakšni nesreči, ki jo je obravnaval policist Bruno Pendrek.

5. Poiščite imena in priimke oseb, ki so se zaletele z vozilom znamke BMW.

6. Poiščite imena in priimke najmlajših oseb.

AIRCRAFT Glej B.6.

1. Poiščite vse podatke o letalih z imenom Boeing.

2. Poiščite vse podatke o letalih Boeing 737.

3. Poiščite številke pilotov, ki imajo certifikat za letenje z Boeing-i.

4. Poiščite številke in imena letal, ki lahko brez postankov letijo od Ljubljane do New York-a ($ARange \geq FDistance$). Predpostavimo, da obstaja let od Ljubljane do New York-a.

5. Poiščite številke zaposlenih, ki imajo najvišjo plačo.

6. Poiščite številke zaposlenih, ki imajo certifikat za letenje za natanko 2 letali.

Poglavje 3

Funkcionalne odvisnosti

Odvisnosti (med atributi) relacijske sheme so sredstvo s katerim povemo, katere vrednosti v relaciji so možne in katere sploh ne morejo obstajati. Poznamo tri vrste:

Funkcionalne odvisnosti.

Večvrednostne odvisnosti.

Stične odvisnosti.

3.1 Uvod v funkcionalne odvisnosti

3.1.1 Definicija funkcionalnih odvisnosti

Funkcionalne odvisnosti predstavljajo razmerje med atributi relacijske sheme, ki omejuje dopustne vrednosti atributov v n-tericah relacije.

Definicija. Naj bosta X in Y neprazni podmnožici atributov relacijske sheme R, $X, Y \subseteq R$, $X, Y \neq \emptyset$. Atributi X *funkcionalno določajo* atribute Y (označimo $X \to Y$), če v nobeni relaciji s shemo R ne moreta obstajati dve n-terici, ki bi se ujemali v vrednostih atributov X in se nebi ujemali v vrednostih atributov Y. Formalno,

$$X \to Y \in F_R \Leftrightarrow \forall r(Sh(r) = R \Rightarrow \forall z_1 \forall z_2((z_1 \in r \wedge z_2 \in r) \Rightarrow (z_1.X = z_2.X \Rightarrow z_1.Y = z_2.Y))),$$

kjer je F_R množica funkcionalnih odvisnosti, ki veljajo med atributi sheme R.

Levi strani funkcionalne odvisnosti običajno pravimo *determinanta*.

Definicija. Naj bosta X in Y podmnožici relacijske sheme R, $X, Y \subseteq R$. Množica odvisnosti F_R *logično implicira* odvisnost $X \to Y$, če vsaka relacija s shemo R, ki zadošča odvisnostim F_R, zadošča tudi odvisnosti $X \to Y$. Označimo,

$$F_R \models X \to Y.$$

Definicija. *Zaprtje množice funkcionalnih odvisnosti* F_R je množica vseh odvisnosti, ki so logično implicirane z odvisnostmi F_R. Označimo,

$$F_R^+ = \{f \mid F_R \models f\}.$$

3.1.2 Identifikacija funkcionalnih odvisnosti

Funkcionalne odvisnosti lahko identificiramo na dva načina:

- na podlagi razumevanja relacijske sheme (zdrav razum, dokumentacija, opis),
- na podlagi **reprezentativne** množice podatkov.

Pri identifikaciji pazimo na naslednje zahteve:

- med determinanto in desno stranjo funkcionalne odvisnosti mora obstajati 1 : 1 preslikava (funkcija),
- funkcionalne odvisnosti morajo veljati ves čas,
- funkcionalne odvisnosti naj bodo *popolne* (vse determinante so minimalne).

3.1.3 Izpeljava funkcionalnih odvisnosti

Pri izpeljavi funkcionalnih odvisnosti si pomagamo z *Armstrongovimi aksiomi* in *inferenčnimi pravili*.

Naj bodo X, Y in Z podmnožice atributov relacijske sheme R, $X, Y, Z \subseteq R$.

Armstrongovi aksiomi

Veljajo naslednji aksiomi.

Refleksivnost (A1): $Y \subseteq X \Rightarrow X \rightarrow Y$

Razširitev (A2): $\{X \rightarrow Y\} \models XZ \rightarrow YZ$

Tranzitivnost (A3): $\{X \rightarrow Y, Y \rightarrow Z\} \models X \rightarrow Z$

Armstrongovi aksiomi so *smiselni* in *popolni*.

Inferenčna pravila

Iz Armstrongovih aksiomov je moč izpeljati naslednja pravila.

Dekompozicija (I1): $\{X \rightarrow YZ\} \models X \rightarrow Y$

Dokaz.
$\{X \rightarrow YZ\}$
$A1 : YZ \rightarrow Y$
$A3 : \{X \rightarrow YZ, YZ \rightarrow Y\} \models X \rightarrow Y$

□

Združitev (I2): $\{X \rightarrow Y, X \rightarrow Z\} \models X \rightarrow YZ$

Dokaz.
$\{X \rightarrow Y, X \rightarrow Z\}$
$A2 : \{X \rightarrow Y\} \models X \rightarrow YX$
$A2 : \{X \rightarrow Z\} \models YX \rightarrow YZ$
$A3 : \{X \rightarrow YX, YX \rightarrow YZ\} \models X \rightarrow YZ$

□

Psevdotranzitivnost (I3): $\{X \to Y, WY \to Z\} \models WX \to Z$

Dokaz.

$\{X \to Y, WY \to Z\}$

$A2 : \{X \to Y\} \models WX \to WY$

$A3 : \{WX \to WY, WY \to Z\} \models WX \to Z$

□

Zgled. Naj bo $R = ABCDXYZW$ neka relacijska shema ter F_R, E_R pripadajoči množici funkcionalnih odvisnosti. Ali množica F_R logično implicira vse odvisnosti v množici E_R? Odgovor utemeljite le z uporabo Armstrongovih aksiomov.

$$F_R = \{A \to BC, B \to D, C \to D, X \to ZW, X \to Y, YW \to Z\}$$

$$E_R = \{A \to BCD, B \to D, AC \to BD, X \to W\}$$

Vsako odvisnost v množici E_R želimo izpeljati iz odvisnosti v množici F_R.

$A \to BCD$
$A2 : \{C \to D\} \models BC \to BCD$
$A3 : \{A \to BC, BC \to BCD\} \models A \to BCD$

$B \to D$
$B \to D$

$AC \to BD$
$A2 : \{A \to BC\} \models AC \to BC$
$A2 : \{C \to D\} \models BC \to BD$
$A3 : \{AC \to BC, BC \to BD\} \models AC \to BD$

$X \to W$
$A1 : ZW \to W$
$A3 : \{X \to ZW, ZW \to W\} \models X \to W$

Množica F_R res logično implicira vse odvisnosti v množici E_R.

3.1.4 Zaprtje množice atributov

Definicija. Naj bo X podmnožica atributov relacijske sheme R, $X \subseteq R$. *Zaprtje množice atributov* X, glede na množico funkcionalnih odvisnosti F_R, je množica vseh atributov sheme R, ki so funkcionalno odvisni od X. Formalno,

$$X^+_{F_R} = \{Y | \, F_R \models X \to Y\}.$$

Sledi $X \to X^+_{F_R}$ (uporabimo $I2$).

Algoritem.

```
INPUT: X, F_R
OUTPUT: X^+_{F_R}

X^+_{F_R} = X;
DO
  FOR EACH Y → Z ∈ F_R DO
    IF Y ⊆ X^+_{F_R} THEN
      X^+_{F_R} = X^+_{F_R} ∪ Z;
WHILE X^+_{F_R} has changed;
```

Zgled. Naj bo $R = ABCDEFG$ neka relacijska shema ter F_R pripadajoča množica funkcionalnih odvisnosti. Poiščite zaprtje množice atributov $X = EF$.

$$F_R = \{A \to B, BE \to G, EF \to A, D \to AC\}$$

Z uporabo algoritma dobimo $EF \rightarrow AEF \rightarrow ABEF \rightarrow ABEFG$. Sledi $X^+_{F_R} = ABEFG$.

3.1.5 Pokritje množice funkcionalnih odvisnosti

Definicija. Množica funkcionalnih odvisnosti F_R *pokriva* množico funkcionalnih odvisnosti E_R, če je vsaka odvisnost $f \in E_R$ logično implicirana z množico odvisnosti F_R. Formalno,

$$F_R \text{ } pokriva \text{ } E_R \Leftrightarrow \forall f (f \in E_R \Rightarrow F_R \models f).$$

Definicija. Množica funkcionalnih odvisnosti F_R *pokriva* množico funkcionalnih odvisnosti E_R, če je zaprtje odvisnosti E_R podmnožica zaprtja odvisnosti F_R. Formalno,

$$F_R \text{ } pokriva \text{ } E_R \Leftrightarrow E_R^+ \subseteq F_R^+.$$

Če F_R in E_R pokrivata ena drugo pravimo, da sta *ekvivalentni*. Njuni zaprtji sta tedaj enaki. Označimo,

$$F_R \equiv E_R \Leftrightarrow F_R^+ = E_R^+.$$

Minimalno pokritje množice funkcionalnih odvisnosti

Navadno želimo, da je množica funkcionalnih odvisnosti podana na čim bolj neredundanten način – da vsebuje zgolj tiste odvisnosti, iz katerih je še moč dobiti F_R^+. Taki (minimalni) množici pravimo *minimalno pokritje množice funkcionalnih odvisnosti*.

Definicija. Minimalno pokritje F_R^{min}, množice funkcionalnih odvisnosti F_R, je množica, za katero velja:

- F_R^{min} je pokritje F_R,
- vsaka odvisnost v F_R^{min} ima le en atribut na desni strani (*kanonična oblika funkcionalnih odvisnosti*),
- iz F_R^{min} ni moč odstraniti nobene odvisnosti tako, da bi nova množica še vedno pokrivala F_R,
- v F_R^{min} ni moč zamenjati nobene odvisnosti $X \to Y$ z $X' \to Y$, kjer je $X' \subset X$, tako, da bi nova množica še vedno pokrivala F_R (*popolne funkcionalne odvisnosti*).

Minimalno pokritje lahko za enostavne primere določimo z *grafično metodo*.

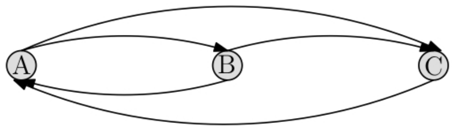
Zgled. Naj bo $R = ABC$ neka relacijska shema ter F_R pripadajoča množica funkcionalnih odvisnosti. Poiščite minimalno pokritje množice F_R.

$$F_R = \{A \to BC, B \to AC, C \to A\}$$

1. Odvisnosti pretvorimo v kanonično obliko (uporabimo I1),

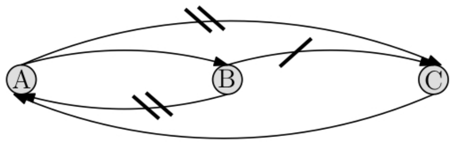
$$F_R^{can} = \{A \to B, A \to C, B \to A, B \to C, C \to A\}$$

2. narišemo graf odvisnosti F_R^{can} (glej sliko 3.1),



Slika 3.1: Graf odvisnosti F_R^{can}.

3. naredimo tranzitivno redukcijo grafa odvisnosti F_R^{can} (uporabimo A3; glej sliko 3.2) in



Slika 3.2: Tranzitivna redukcija grafa odvisnosti F_R^{can}.

4. iz grafa preberemo F_R^{min}.

$$F_R^{min} = \{A \to B, B \to C, C \to A\} \text{ ali } \{A \to B, A \to C, B \to A, C \to A\}$$

3.1.6 Naloge

Identifikacija funkcionalnih odvisnosti

1. Naj bo *zaposleni* neka relacija z relacijsko shemo *ZAPOSLENI* (glej 3.1). Identificirajte funkcionalne odvisnosti na podlagi razumevanja relacijske sheme.

   Opis sheme *ZAPOSLENI*: vsak zaposleni ima enolično številko zaposlenega. Vsaka podružnica se nahaja na natanko eni lokaciji, na vsaki lokaciji pa je največ ena podružnica. Plača zaposlenega je odvisna od položaja zaposlenega ter od podružnice, v kateri je zaposlen.

---

2. Naj bo $R = ABCDE$ neka relacijska shema. Identificirajte funkcionalne odvisnosti na podlagi reprezentativne množice podatkov (glej 3.2).

ID	ImePriimek	Polozaj	Placa	PodruznicaID	PodruznicaNaslov
L21	Janez Novak	direktor	40000	P002	Slovenska 93, Ljubljana
N37	Luka Luka	programer	12000	P007	Tržaška 25, Ljubljana
...	...	...	...	...	...

Tabela 3.1: Relacijska shema *ZAPOSLENI*.

A	B	C	D	E
a	a	b	a	a
b	a	a	a	b
a	b	b	a	c
b	b	a	a	d
a	c	b	b	e
b	c	a	b	f

Tabela 3.2: Reprezentativna množica podatkov.

Izpeljava funkcionalnih odvisnosti

1. Naj bo $R = ABCD$ neka relacijska shema ter F_R pripadajoča množica funkcionalnih odvisnosti.

$$F_R = \{A \to B, B \to CD, C \to D\}$$

(a) Poiščite 5 odvisnosti, ki veljajo med atributi sheme R.

(b) Poiščite kakšno odvisnost v F_R^+, ki jo je moč izpeljati le z uporabo $I1$.

(c) Ali iz množice odvisnosti F_R moč izpeljati odvisnost $A \to ABCD$?

---

2. Naj bo $R = ABCDEFGH$ neka relacijska shema ter F_R pripadajoča množica funkcionalnih odvisnosti.

$$F_R = \{AB \to D, AC \to FG, B \to G, D \to B\}$$

Ali množica F_R logično implicira odvisnost f? Odgovor utemeljite z uporabo Armstrongovih aksiomov ali inferenčnih pravil.

(a) $f = AC \to C$

(b) $f = AC \to G$

(c) $f = H \to DE$

(d) $f = AB \rightarrow DG$

(e) $f = AD \rightarrow BG$

(f) $f = AEC \rightarrow BH$

3. Naj bo $R = ABCDE$ neka relacijska shema ter F_R pripadajoča množica funkcionalnih odvisnosti.

$$F_R = \{A \rightarrow B, B \rightarrow C, AB \rightarrow D, D \rightarrow E\}$$

Ali množica F_R logično implicira odvisnost f? Odgovor utemeljite le z uporabo Armstrongovih aksiomov.

(a) $f = A \rightarrow D$

(b) $f = AD \rightarrow BE$

(c) $f = BCE \rightarrow AD$

(d) $f = AB \rightarrow D$

(e) $f = AB \rightarrow DE$

(f) $f = A \rightarrow ABCDE$

Zaprtje množice atributov

1. Naj bo $R = ABCDEF$ neka relacijska shema ter F_R pripadajoča množica funkcionalnih odvisnosti.

$$F_R = \{A \rightarrow CDE, BE \rightarrow CF, C \rightarrow E, D \rightarrow B, E \rightarrow BCD\}$$

Poiščite zaprtje množice atributov X.

(a) $X = ABCDEF$
(b) $X = BF$
(c) $X = E$
(d) $X = AD$
(e) $X = ABD$

---

2. Naj bo $R = ABCDEFG$ neka relacijska shema ter F_R pripadajoča množica funkcionalnih odvisnosti.

$$F_R = \{AB \to G, BC \to E, C \to B, F \to EG, G \to ABC\}$$

   Poiščite zaprtje množice atributov X.

   (a) $X = C$
   (b) $X = AG$
   (c) $X = DF$

---

3. Naj bo $R = ABCDE$ neka relacijska shema ter F_R pripadajoča množica funkcionalnih odvisnosti.

$$F_R = \{AB \to CE, B \to D, D \to A\}$$

   Katere od naslednjih odvisnosti ni moč implicirati z množico odvisnosti F_R?

   - $E \to E$
   - $BC \to DE$
   - $AD \to CE$
   - $B \to E$
   - $AB \to A$
   - nobene

---

4. Naj bo $R = ABCDEF$ neka relacijska shema ter F_R pripadajoča množica funkcionalnih odvisnosti.

$$F_R = \{AB \to CE, B \to D, D \to A, A \to F\}$$

   Koliko funkcionalnih odvisnosti, ki veljajo za shemo R, ima determinanto enako X?

   (a) $X = AF$

   (b) $X = BC$

   (c) $X = F$

Pokritje množice funkcionalnih odvisnosti

1. Naj bo R neka relacijska shema ter F_R pripadajoča množica funkcionalnih odvisnosti. Poiščite minimalno pokritje množice F_R.

   (a) $R = ABCD$, $F_R = \{B \to ACD, C \to ABD\}$

(b) $R = ABCDE$, $F_R = \{A \to BCDE, B \to CD, D \to BC\}$

(c) $R = ABCDEF$, $F_R = \{A \to BCDEF, B \to DE, E \to D, G \to ABCE\}$

2. Naj bo R neka relacijska shema ter F_R, E_R pripadajoči množici funkcionalnih odvisnosti. Ali sta množici F_R in E_R ekvivalentni?

   (a) $R = ABC$, $F_R = \{A \to B, B \to C\}$, $E_R = \{A \to C, C \to B\}$

   (b) $R = ABCD$, $F_R = \{A \to CD, B \to AC\}$, $E_R = \{A \to CD, B \to AD\}$

   (c) $R = ABCDEFG$, $F_R = \{A \to BCF, B \to E, E \to D, G \to A\}$, $E_R = \{B \to ACD, C \to ABD\}$

3.2 Ključi

3.2.1 Definicija ključev

Definicija. Naj bo X podmnožica atributov relacijske sheme R, $X \subseteq R$. Atributi X so *ključ* sheme R, oziroma pripadajočih relacij, če sta izpolnjena naslednja dva pogoja:

- atributi X funkcionalno določajo vse atribute sheme R,

$$X \to R$$

- ne obstaja prava podmnožica $X' \subset X$, ki bi prav tako funkcionalno določala vse atribute sheme R.

$$\neg \exists X' (X' \subset X \wedge X' \to R)$$

Shema ima lahko več takih ključev, ki jih imenujemo *kandidati za ključ*. Pri realizaciji izberemo enega izmed njih, ki ga imenujemo *primarni ključ*.

Podmnožico atributov relacijske sheme, ki je ključ v neki drugi shemi, imenujemo *tuji ključ*.

3.2.2 Identifikacija kandidatov za ključ

Preveriti moramo vse možne kandidate za ključ (vse možne podmnožice R). Prvi pogoj iz definicije preverjamo s pomočjo zaprtja množice atributov, drugi pogoj pa nam dejansko postopoma omejuje prostor možnih kandidatov za ključ, ki ga še moramo preiskati.

Zgled. Naj bo $R = ABC$ neka relacijska shema in F_R pripadajoča množica funkcionalnih odvisnosti. Določite kandidate za ključ.

$$F_R = \{A \to B, B \to A, AB \to C, BC \to A\}$$

Najprej opazimo, da velja $(A)^+_{F_R} = ABC$. Takoj sledi, da je A kandidat za ključ, saj je drugi pogoj iz definicije trivialno izpolnjen. Podobno opazimo, da je tudi B kandidat za ključ.

Zaradi drugega pogoja noben kandidat za ključ ne vsebuje A ali B. Možen kandidat je tako edino še C, vendar pa velja $(C)^+_{F_R} = C$. Kandidata za ključ sta tako le A in B.

3.2.3 Naloge

1. Naj bo $STUDENT = \{Vpisna, Ime, Priimek, Letnik, Posta, Kraj\}$ neka relacijska shema in $F_{STUDENT}$ pripadajoča množica funkcionalnih odvisnosti. Poiščite vse kandidate za ključ.

$$F_{STUDENT} = \{Vpisna \rightarrow \{Ime, Priimek, Letnik, Posta\}, Posta \rightarrow Kraj\}$$

2. Naj bo R neka relacijska shema, X podmnožica atributov sheme R, $X \subseteq R$, in F_R pripadajoča množica funkcionalnih odvisnosti. Ali je X kandidat za ključ?

   (a) $R = ABCDEFG, X = EF, F_R = \{A \rightarrow B, BE \rightarrow G, D \rightarrow AC, EF \rightarrow A\}$

   (b) $R = ABCDE, X = ABE, F_R = \{AB \rightarrow C, C \rightarrow A, CD \rightarrow E, E \rightarrow D\}$

   (c) $R = ABCDEFG, X = AG, F_R = \{A \rightarrow BCF, B \rightarrow E, E \rightarrow D, G \rightarrow A\}$

3. Naj bo R neka relacijska shema in F_R pripadajoča množica funkcionalnih odvisnosti. Poiščite vse kandidate za ključ.

   (a) $R = AB, F_R = \{A \rightarrow B, B \rightarrow A\}$

   (b) $R = ABC, F_R = \{\}$

   (c) $R = ABC, F_R = \{A \rightarrow B, B \rightarrow A, AB \rightarrow C, BC \rightarrow A, AC \rightarrow B\}$

   (d) $R = ABCD, F_R = \{A \rightarrow B, B \rightarrow D, D \rightarrow C, C \rightarrow A\}$

   (e) $R = ABCD, F_R = \{A \rightarrow BCD, AB \rightarrow C, AC \rightarrow B, BC \rightarrow A\}$

   (f) $R = ABCDE, F_R = \{A \rightarrow B, B \rightarrow A\}$

   (g) $R = ABCDE, F_R = \{A \rightarrow BCDE, BCDE \rightarrow A\}$

(h) $R = ABCDEFG, F_R = \{A \to B, B \to A, ABC \to DEFG\}$

3.3 [sPB1] Normalizacija

3.3.1 Uvod v normalizacijo

Relacije navadno preoblikujemo v obliko, pri kateri ne more priti do *ažurnih anomalij*. Postopku preoblikovanja pravimo *normalizacija*.

Normalizacija uporablja pristop *od spodaj navzgor*. To pomeni, da načrtovanje podatkovne baze pričnemo z eno relacijo, ki vsebuje vse pomembne atribute. Relacijo nato postopoma preoblikujemo v želeno obliko.

Obstaja več normalnih oblik:

1.NO: prva normalna oblika

2.NO: druga normalna oblika

3.NO: tretja normalna oblika

BCNO: Boyce-Codd-ova normalna oblika

4.NO: četrta normalna oblika

5.NO: peta normalna oblika

1.NO, 2.NO, 3.NO, BCNO temeljijo na funkcionalnih odvisnostih, 4.NO temelji na večvrednostnih odvisnostih, 5.NO pa na stičnih odvisnostih. Gre za *formalno normalizacijo*, obstaja pa tudi *poslovna normalizacija* (1.NO, 2.NO, 3.NO, nato 4.PNO in 5.PNO).

Definicija. *Each attribute is placed in an entity where it is dependent on the key, the whole key, and nothing but the key.*

3.3.2 Prva normalna oblika

Relacija je v 1.NO, če:

1. ima določene funkcionalne odvisnosti,

2. ima izbran (primarni) ključ,

3. nima *večvrednostnih atributov*.

Večvrednostne atribute odpravimo tako, da vnesemo manjkajoče vrednosti ali pa atribute prestavimo v novo relacijo (skupaj s ključem).

3.3.3 Druga normalna oblika

Relacija je v 2.NO, če:

1. je v 1.NO,

2. nima *parcialnih (delnih) odvisnosti*.

Parcialne odvisnosti odpravimo tako, da relacijo razbijemo na več novih relacij.

3.3.4 Tretja normalna oblika

Relacija je v 3.NO, če:

1. je v 2.NO,

2. nima *tranzitivnih odvisnosti*.

Tranzitivne odvisnosti odpravimo tako, da relacijo razbijemo na več novih relacij.

Zgled. Naj bo $VPIS$ neka relacijska shema, ki predstavlja vpis študentov v posamezen letnik. Normalizirajte shemo $VPIS$ do 3.NO.

$$VPIS(Vpisna, Ime, Priimek, (ID, ProgramID, ProgramIme, Letnik, Vpisnina))$$

1.NO Identificiramo funkcionalne odvisnosti,

$$F_{VPIS} = \{\{Vpisna, ID\} \rightarrow \{ProgramID, Letnik\}, Vpisna \rightarrow \{Ime, Priimek\}, ProgramID \rightarrow ProgramIme,$$

$$\{ProgramID, Letnik\} \rightarrow Vpisnina\}$$

poiščemo kandidate za ključ,

$$\{Vpisna, ID\}$$

izberemo (primarni) ključ in

$$\{Vpisna, ID\}$$

odpravimo večvrednostne atribute.

$$VPIS(\underline{Vpisna, ID}, Ime, Priimek, ProgramID, ProgramIme, Letnik, Vpisnina)$$

2.NO Odpravimo parcialne odvisnosti ($\{Vpisna \rightarrow \{Ime, Priimek\}\}$).

$$VPIS(\underline{\#Vpisna, ID}, ProgramID, ProgramIme, Letnik, Vpisnina)$$

$$F_{VPIS} = \{\{Vpisna, ID\} \rightarrow \{ProgramID, Letnik\}, ProgramID \rightarrow ProgramIme, \{ProgramID, Letnik\} \rightarrow Vpisnina\}$$

$$STUDENT(\underline{Vpisna}, Ime, Priimek)$$

$$F_{STUDENT} = \{Vpisna \rightarrow \{Ime, Priimek\}\}$$

3.NO Odpravimo tranzitivne odvisnosti ($\{ProgramID \rightarrow ProgramIme, \{ProgramID, Letnik\} \rightarrow Vpisnina\}$).

$$VPIS(\underline{\#Vpisna, ID}, \#ProgramID, \#Letnik)$$

$$F_{VPIS} = \{\{Vpisna, ID\} \rightarrow \{ProgramID, Letnik\}\}$$

$$STUDENT(\underline{Vpisna}, Ime, Priimek)$$

$$F_{STUDENT} = \{Vpisna \rightarrow \{Ime, Priimek\}\}$$

$$PROGRAM(\underline{ProgramID}, ProgramIme)$$

$$F_{PROGRAM} = \{ProgramID \rightarrow ProgramIme\}$$

$$VPISNINA(\underline{ProgramID, Letnik}, Vpisnina)$$

$$F_{VPISNINA} = \{\{ProgramID, Letnik\} \rightarrow Vpisnina\}$$

3.3.5 Naloge

1. Naj bo $R = ABCD$ neka relacijska shema, F_R pripadajoča množica funkcionalnih odvisnosti ter AB (primarni) ključ. V kateri normalni obliki je shema R? Shemo postopoma normalizirajte do 3.NO. Za vsako (novo) shemo podajte ključe ter pripadajoče funkcionalne odvisnosti.

$$F_R = \{AB \rightarrow CD, C \rightarrow D\}$$

---

2. Naj bo $R = ABCDE$ neka relacijska shema ter F_R pripadajoča množica funkcionalnih odvisnosti. V kateri normalni obliki je shema R? Shemo postopoma normalizirajte do 3.NO. Za vsako (novo) shemo podajte ključe ter pripadajoče funkcionalne odvisnosti.

$$F_R = \{A \rightarrow D, B \rightarrow E, E \rightarrow B\}$$

---

3. Naj bo $R = ABCDEF$ neka relacijska shema, F_R pripadajoča množica funkcionalnih odvisnosti ter BC (primarni) ključ. V kateri normalni obliki je shema R? Shemo postopoma normalizirajte do 3.NO. Za vsako (novo) shemo podajte ključe ter pripadajoče funkcionalne odvisnosti.

$$F_R = \{B \rightarrow DE, C \rightarrow F, E \rightarrow AF\}$$

Poglavje 4

[sOPB] Datotečne organizacije in indeksiranje

4.1 Indeksiranje

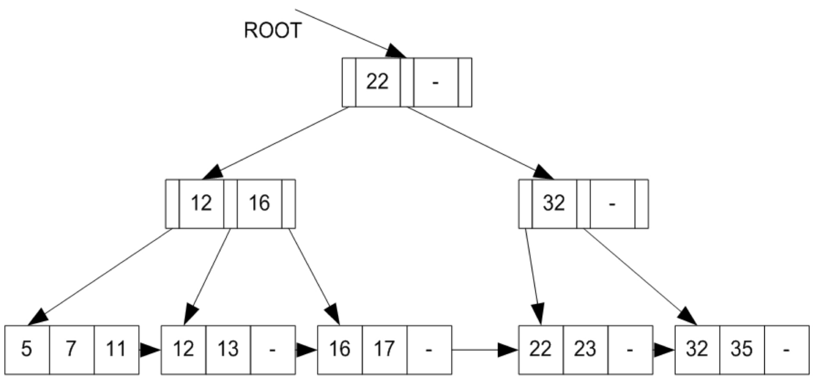
Indeks je podatkovna struktura, ki omogoča hiter dostop do zapisov v datoteki (hitrejši kot pa ga omogoča sama organizacija datoteke) ter pa tudi zaporeden (urejen) dostop do zapisov.

V nadaljevanju predpostavimo, da podatkovna datoteka ne vsebuje duplikatov.

4.1.1 B$^+$-indeks

B$^+$-indeksiranje je dinamično večnivojsko indeksiranje z uporabo *B$^+$-dreves*. *B$^+$-indeks* je sestavljen iz *B-indeksa* in *gostega indeksa* (Mohoric, 2002).

Definicija. B$^+$-drevesa so variacija *B-dreves*, kjer zapise hranimo le v listih, v notranjih vozliščih pa hranimo ključe, ki omogočajo iskanje. Drevo je uravnoteženo, listi drevesa pa so še dodatno povezani v (urejen) dvosmerni seznam.



Slika 4.1: B$^+$-drevo reda 1 (pri $a = 3$).

Naj bo $|v|$ število *elementov* (ključev ali zapisov) v nekem vozlišču v.

Definicija. Naj bo d *red* B$^+$-drevesa, a največje število zapisov v listih ter v neko vozlišče. Velja

$$1 \leq |v| \leq 2d, \text{ če je } v \text{ koren,}$$

$$d \leq |v| \leq 2d, \text{ če je } v \text{ notranje vozlišče,}$$

$$a/2 \leq |v| \leq a, \text{ če je } v \text{ list.}$$

Vsako vozlišče v, ki ni list, ima natanko $|v| + 1$ sinov (poddreves). Naj bodo k_i ključi v v,

$$k_1 < k_2 < \ldots < k_{|v|},$$

ter naj bo x vrednost poljubnega ključa v i-tem poddrevesu v. Velja

$$x < k_1, \text{ če je } i = 0,$$

$$k_i \leq x < k_{i+1}, \text{ če je } 0 < i < |v|,$$

$$k_{|v|} \leq x, \text{ če je } i = |v|.$$

Naj bo h *višina* B^+-drevesa. Vsak B^+-indeks indeksira najmanj $(d+1)^{h-1}a$ in največ $(2d+1)^h a$ zapisov.

Zgled. Glej sliko 4.1.

Operacije

Iskanje. Sledimo kazalcem do lista, kjer naj bi se nahajal zapis.

Dodajanje. Sledimo kazalcem do vozlišča, kamor spada element. Če je v vozlišču prostor, element dodamo in zaključimo, v nasprotnem primeru pa

- [če ima vozlišče brata v in $|v| < 2d$ (oz. $|v| < a$), izvedemo **redistribucijo**],
- sicer vozlišče **razpolovimo**, delitveni ključ pa **dodamo** očetu (rekurzija).

Šele nato dodamo element.

Brisanje. Sledimo kazalcem do vozlišča, kjer se nahaja element, in ga zbrišemo. Če zasedenost vozlišča pade pod d (oz. $a/2$)

- če ima vozlišče brata v in $|v| > d$ (oz. $|v| > a/2$), izvedemo **redistribucijo**,
- sicer **zlijemo** vozlišče in morebitnega brata, ter **zbrišemo** ustrezen delitveni ključ očetu (rekurzija).

Bulk loading

Bulk loading je postopek za izgradnjo B^+-indeksa iz obstoječih podatkov. Na podlagi zapisov, ki jih želimo indeksirati, se najprej kreirajo bloki (strani) in pripadajoči delitveni ključi (listi B^+-drevesa). Le-te se nato postopoma doda v indeksno strukturo.

4.1.2 ISAM indeks

Indeksiranje z ISAM (Indexed Sequential Access Method) je statično večnivojsko indeksiranje.

Definicija. *ISAM indeks* je drevesna struktura, kjer zapise hranimo le v listih (ter v *prelivnih straneh*), v notranjih vozliščih pa hranimo ključe, ki omogočajo iskanje. Elementi v listih in prelivnih straneh običajno niso urejeni (zaradi hitrejšega dodajanja).

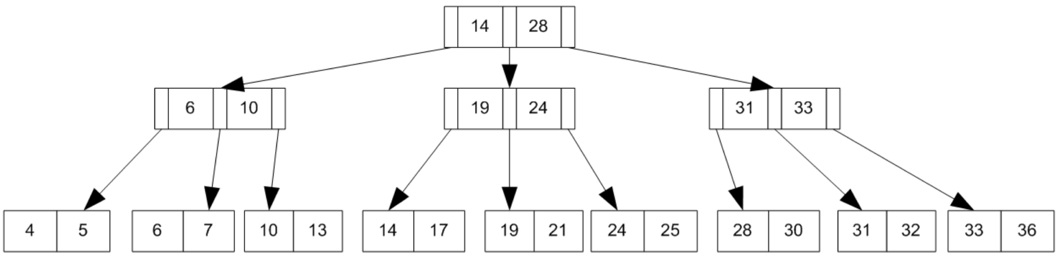
Zgled. Glej sliko 4.2.

Operacije

Iskanje. Sledimo kazalcem do lista, kjer naj bi se nahajal zapis. Če zapisa ni v listu, preiščemo še prelivne strani.

Dodajanje. Sledimo kazalcem do lista, kamor spada zapis. Dodamo ga na prvo prosto mesto ali pa kreiramo novo prelivno stran.

Brisanje. Sledimo kazalcem do lista, kjer se nahaja zapis, ter ga zbrišemo. Če tako dobimo prazno prelivno stran, jo zbrišemo.



Slika 4.2: ISAM indeks.

4.1.3 Bitni indeks

Namesto indeksiranja dejanske vrednosti nekega atributa, za vsak zapis hranimo le *bitni vektor*. Slednji ima postavljen bit, ki ustreza vrednosti atributa zapisa.

Zgled. Glej 4.1 in 4.2.

ID	ImePriimek	Polozaj	Placa	PodruznicaID	PodruznicaNaslov
...	Janez Novak	direktor	40000	P002	Slovenska 93, Ljubljana
...	Luka Luka	programer	12000	P007	Tržaška 25, Ljubljana
...	...	programer	12000	P002	Slovenska 93, Ljubljana
...	...	direktor	30000	P007	Tržaška 25, Ljubljana
...	...	direktor	36000	P006	Celovška 113, Ljubljana
...	...	programer	12000	P002	Slovenska 93, Ljubljana
...	...	programer	12000	P002	Slovenska 93, Ljubljana
...	...	programer	11000	P006	Celovška 113, Ljubljana

Tabela 4.1: Relacija *zaposleni*.

4.2 Naloge

B^+-indeks

1. V osnovno datoteko, ki jo indeksira B^+-indeks na sliki 4.1, zaporedoma dodajte zapise s ključi 8, 24 in 25 (redistribucija se uporablja). Narišite indeks po koncu vsakega ažuriranja.

direktor	programer
1	0
0	1
0	1
1	0
1	0
0	1
0	1
0	1

P002	P006	P007
1	0	0
0	0	1
1	0	0
0	0	1
0	1	0
1	0	0
1	0	0
0	1	0

Tabela 4.2: Bitni indeks za atributa *Polozaj* in *PodruznicaID*.

---

2. V osnovno datoteko, ki jo indeksira B$^+$-indeks na sliki 4.1, najprej dodajte zapis s ključem 8 in nato zbrišite zapis s ključem 32 (redistribucija se uporablja). Narišite indeks po koncu vsakega ažuriranja.

---

3. V osnovno datoteko, ki jo indeksira B$^+$-indeks na sliki C.1, dodajte zapis s ključem 9 (redistribucija se ne uporablja). Narišite le indeks po koncu vseh ažuriranj.

---

4. V osnovno datoteko, ki jo indeksira B$^+$-indeks na sliki C.1, dodajte zapis s ključem 3 (pri redistribuciji se uporablja le desni brat). Narišite le indeks po koncu vseh ažuriranj.

---

5. V osnovno datoteko, ki jo indeksira B$^+$-indeks na sliki C.1, najprej dodajte zapis s ključem 13 in nato še zapis s ključem 3 (redistribucija se ne uporablja). Narišite le indeks po koncu vseh ažuriranj.

---

6. V osnovno datoteko, ki jo indeksira B$^+$-indeks na sliki C.1, zaporedoma dodajte zapise s ključi 46, 47, 48, 49 in 50 (pri redistribuciji se uporablja le desni brat). Narišite le indeks po koncu vseh ažuriranj.

---

7. Iz osnovne datoteke, ki jo indeksira B$^+$-indeks na sliki C.1, zbrišite zapis s ključem 8 (pri redistribuciji se najprej uporabi desni brat, nato levi). Narišite le indeks po koncu vseh ažuriranj.

---

8. Iz osnovne datoteke, ki jo indeksira B$^+$-indeks na sliki C.1, zbrišite zapis s ključem 8 (pri redistribuciji se najprej uporabi desni brat, nato levi). Narišite le indeks po koncu vseh ažuriranj.

---

9. Iz osnovne datoteke, ki jo indeksira B^+-indeks na sliki C.1, zbrišite zapis s ključem 91 (pri redistribuciji se najprej uporabi levi brat, nato desni). Narišite le indeks po koncu vseh ažuriranj.

---

10. Iz osnovne datoteke, ki jo indeksira B^+-indeks na sliki C.1, zaporedoma zbrišite zapise s ključi 2, 32, 41, 5, 39 in 45 (pri redistribuciji se najprej uporabi desni brat, nato levi). Narišite le indeks po koncu vseh ažuriranj.

---

11. V osnovno datoteko, ki jo indeksira B^+-indeks na sliki C.1, najprej dodajte zapis s ključem 46 in nato zbrišite zapis s ključem 52 (pri redistribuciji se najprej uporabi levi brat, nato desni). Narišite le indeks po koncu vseh ažuriranj.

---

12. V osnovno datoteko, ki jo indeksira B^+-indeks na sliki C.1, najprej zaporedoma dodajte zapisa s ključema 28 in 30 ter nato zbrišite zapis s ključem 39 (pri redistribuciji se najprej uporabi desni brat, nato levi). Narišite le indeks po koncu vseh ažuriranj.

ISAM indeks

1. V osnovno datoteko, ki jo indeksira ISAM indeks na sliki 4.2, zaporedoma dodajte zapise s ključi 22, 20, 26, 23 in 18. Narišite le indeks po koncu vseh ažuriranj.

---

2. V osnovno datoteko, ki jo indeksira ISAM indeks na sliki 4.2, najprej zaporedoma dodajte zapise s ključi 29, 35 in 100 ter nato zaporedoma zbrišite zapise s ključi 19, 29, 28, 32, 30, 28 in 36. Narišite le indeks po koncu vseh ažuriranj.

---

3. V osnovno datoteko, ki jo indeksira ISAM indeks na sliki C.2, najprej zaporedoma dodajte zapise s ključi 1, 2, 3, 4 in 25 ter nato zaporedoma zbrišite zapise s ključi 10, 15, 27, 37, 25, 1, 3 in 20. Narišite le indeks po koncu vseh ažuriranj.

---

4. Iz osnovne datoteke, ki jo indeksira ISAM indeks na sliki C.2, zaporedoma zbrišite zapise s ključi 37, 33, 40, 51, 55 in 10. Narišite le indeks po koncu vseh ažuriranj.

Dodatek A

Spremenljivke

Spremenljivke	**Opis**
$\mathcal{O}$	množica objektov
o	objekt ($o \in \mathcal{O}$)
$r, s, t \ldots$	relacije
$N_r, N_s, N_t \ldots$	moč relacij ($N_r = \lvert r \rvert, N_s = \lvert s \rvert, N_t = \lvert t \rvert \ldots$)
$R, S, T \ldots$	relacijske sheme ($R = Sh(r), S = Sh(s), T = Sh(t) \ldots$)
$A, B, C \ldots$	atributi
$D_A, D_B, D_C \ldots$	domene atributov ($A : D_A, B : D_B, C : D_C \ldots$)
z_o	n-terica (zapis), ki pripada objektu o
$z_r, z_s, z_t \ldots$	n-terice (zapisi) v relacijah ($z_r \in r, z_s \in s, z_t \in t \ldots$)
z_N	n-terica (zapis) samih $NULL$ vrednosti
$F_R, F_S, F_T \ldots$	funkcionalne odvisnosti med atributi shem $R, S, T \ldots$
θ	logični pogoj

Dodatek B

Relacije

OPERATER

Glej B.1.

Relacija. *k kupuje*

Stranka	Operater
Marko	Mobitel
Marko	Simobil
Meta	Mobitel
Meta	Vega
Meta	Simobil
Janez	Simobil
Petra	Mobitel

Relacija. *p prodaja*

Operater	Telefon
Mobitel	Nokia
Mobitel	Siemens
Vega	Nokia
Vega	SE
Simobil	Nokia
Simobil	Siemens
Simobil	SE

Relacija. *n najraje*

Stranka	Telefon
Marko	Siemens
Meta	Nokia
Janez	Siemens
Petra	Nokia

Tabela B.1: Relacije *OPERATER*.

GSM

Glej B.2.

Relacija	**Relacijska shema**
s stranka	$STRANKA(\underline{SID}, SIme, SPriimek, SStarost, SKraj)$
p prodajalec	$PRODAJALEC(\underline{PID}, PIme, PPriimek, PStarost, PPopust)$
g gsm	$GSM(\underline{GID}, GTip, GCena)$
k kupcija	$KUPCIJA(\#\underline{SID}, \#\underline{PID}, \#\underline{GID}, KDatum, KKosov)$

Tabela B.2: Relacije *GSM*.

NAROCILO

Glej B.3.

Relacija	**Relacijska shema**
s stranka	$STRANKA(\underline{SID}, SIme, SPriimek, SMesto, SPopust)$
a agent	$AGENT(\underline{AID}, AIme, APriimek, AMesto)$
i izdelek	$IZDELEK(\underline{IID}, IIme, IZaloga, ICena)$
n narocilo	$NAROCILO(\underline{NID}, \#SID, \#AID, \#IID, NDatum, NKosov)$

Tabela B.3: Relacije *NAROCILO*.

NESRECA

Glej B.4.

Relacija	**Relacijska shema**
o oseba	$OSEBA(\underline{OID}, OIme, OPriimek, OSpol, OStarost)$
p policist	$POLICIST(\underline{PID}, PIme, PPriimek, PPostaja)$
v vozilo	$VOZILO(\underline{VRegistrska}, VZnamka, VModel)$
n nesreca	$NESRECA(\underline{NID}, \#OID_1, \#OID_2, \#VRegistrska_1, \#VRegistrska_2, \#PID, NDatum, NKraj)$

Tabela B.4: Relacije *NESRECA*.

HOTEL

Glej B.5.

Relacija	**Relacijska shema**
g guest	$GUEST(\underline{GNo}, GName, GAddress)$
h hotel	$HOTEL(\underline{HNo}, HName, HCity)$
r room	$ROOM(\underline{RNo}, \#\underline{HNo}, RType, RPrice)$
b booking	$BOOKING(\#\underline{HNo}, \#\underline{RNo}, \#\underline{GNo}, \underline{BFrom}, BTo)$

Tabela B.5: Relacije *HOTEL*.

AIRCRAFT

Glej B.6.

Relacija	**Relacijska shema**	**Komentar**
e employee	$EMPLOYEE(\underline{ENo}, EName, ESalary, EPosition)$	tudi ne-piloti
a aircraft	$AIRCRAFT(\underline{ANo}, AName, AModel, ARange)$	
f flight	$FLIGHT(\underline{FNo}, FFrom, FTo, FDistance, FDepart, FArrive)$	
c certified	$CERTIFIED(\#\underline{ENo}, \#\underline{ANo})$	certifikati za letenje

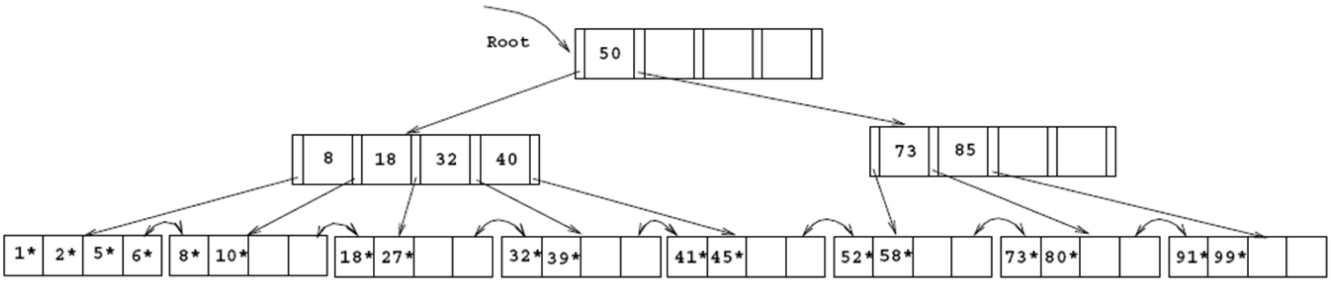
Tabela B.6: Relacije *AIRCRAFT*.

Dodatek C

Indeksi

B$^+$-indeks

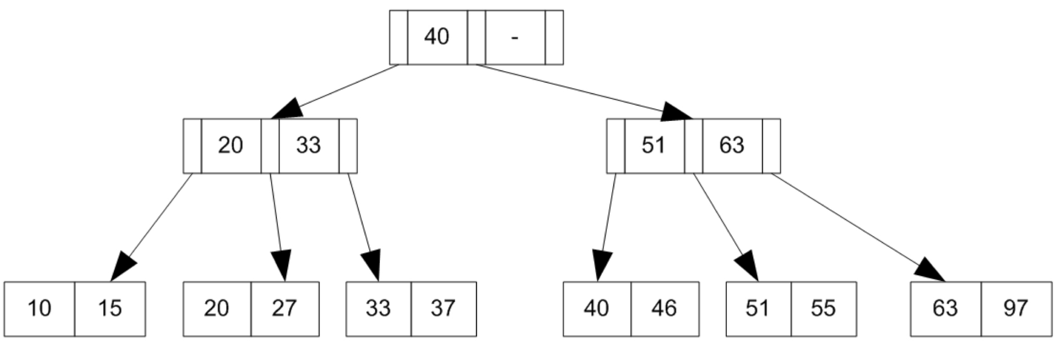
Glej sliko C.1.



Slika C.1: B$^+$-indeks reda 2.

ISAM indeks

Glej sliko C.2.



Slika C.2: ISAM indeks.

Literatura

Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM*, *13*(6), 377–387.

Elmasri, R., & Navathe, S. (2007). *Fundamentals of database systems*. Addison-Wesley.

Mohoric, T. (2002). *Podatkovne baze*. Bi-TIM.

Tabele

Slike