

## 01. Fizična podatkovna baza!

Podatkovni del podatkovne baze, ki ga sestavljata fizična in metapodatkovna baza, se shranjuje v zunanjem pomnilniku računalniškega sistema. Predvsem fizična podatkovna baza utegne biti po obsegu dokaj velika, zato je pomembno, da je shranjena v takem pomnilniku in v taki obliki, ki oba skupaj omogočata hiter dostop in ažuriranje podatkov. Od zunanjega pomnilnika pričakujemo predvsem naslednje lastnosti:

- sposobnost shraniti razmeroma velike količine podatkov;
- trajno pomnjenje (ob izklopu napajanja se podatki ne smejo izgubiti);
- hitro vpisovanje, spreminjanje in dostop do podatkov;
- nizko ceno na enoto shranjenih podatkov

Pomnilnik, ki v današnjem času najbolje izpolnjuje omenjene zahteve in ga zato tudi najpogosteje srečujemo kot zunanji pomnilnik, je magnetni disk. V primerih, ko tretja od zgoraj navedenih omejenih lastnosti ni bistvena, se kot zunanji pomnilnik uporablja tudi magnetni trak. Na magnetnem traku se npr. pogosto shranjujejo podatki, ki služijo za obnavljanje fizične podatkovne baze po nesrečah, pri katerih se vsa ali del njene vsebine izgubi.

Podatki iz fizične podatkovne baze so dostopni SUPB (sistem za upravljanje podatkonih baz) šele potem, ko jih OS prebere z diska in prenese v notranji pomnilnik, kar pa seveda zahteva določen čas. Čas dostopa do podatkov v notranjem pomnilniku merimo v mikrosekundah, do bloka podatkov na disku pa v desetih milisekund. Razmerje med dostopnima časoma je torej približno 1:10.000, kar pomeni, da je hiter dostop do podatkov potrebno posvetiti posebno pozornost predvsem ugotavljanju, v katerem bloku se iskani podatki nahajajo. Ta problem se rešuje z uporabo različnih vrst datotečnih organizacij.

## 02. Zaporedna datoteka!

Pri zaporedni datoteki se zapisi logične datoteke preslikajo v zapise fizične datoteke tako, da so v njej urejeni po vrednosti ključa – zapis z manjšo vrednostjo ključa se nahaja pred zapisom z večjo vrednostjo ključa. Kaj je večja in kaj je manjša vrednost ključa, se določi v okviru njegove deklaracije. Zapisi so lahko urejeni po naraščajoči ali padajoči vrednosti ključa. Ker je ključ lahko sestavljen iz več podatkovnih elementov, je važna tudi njihova razvrstitev od najpomembnejšega do njemanj pomembnega, pri čemer lahko vsak od njih vpliva na vrednost ključa z ozirom na padajočo ali naraščajočo vrednost. Pri ureditvi po padajoči vrednosti ključa, bomo tako upoštevali, da je ključ z vrednostjo 100 manjši od ključa z vrednostjo 90.

## 03. Relacijska algebra!

Operacije relacijske algebre delimo na osnovne in izvedene. Med osnovne operacije prištevamo:

- Unijo,
- Razliko,
- Kartezijski produkt,
- Projekcijo,
- Selekcijo.

Izvedene operacije, ki jih lahko izrazimo z osnovnimi operacijami, pa so naslednje:

- Presek,
- Naravni stik,
- $\theta$ -stik,
- Količnik.

Povpraševalni jeziki običajno ne dosegajo vseh naštetih operacij, pač pa le njihov izbor, ki pa mora biti tolikšen, da omogoča poljubno preslikavo vhodnih podatkov (relacij) v rezultat.

Če v izrazu relacijske algebre z oklepaji ni drugače določeno, je prioriteta izvajanja operacij naslednja:

1. Najvišjo prioriteto imata selekcija in projekcija,
2. Slede operacije kartezijski produkt,  $\theta$ -stik, naravni stik in količnik,
3. Nato presek,
4. Najnižja prioriteta pa pripada uniji in razliki.

## 04. Relacija!

Matematična osnova relacijskega modela je relacija, ki je podmnožica kartezijskega produkta liste domen. Na listi domen se posamezna domena lahko pojavlja tudi večkrat. Predpostavimo, da listo sestavljajo domene  $D_1, D_2, \dots, D_n$ , pri čemer dopoščamo, da je  $D_i = D_j$ , pri  $1 \leq i < j \leq n$ . Relacija  $r$  je torej poljubna podmnožica kartezijskega produkta ene ali večih domen. Drugače povedano – relacija je množica urejenih  $n$ -teric, pri čemer je vsaka  $n$ -terica sestavljena iz komponent, ki so elementi domen. V okviru relacijskih podatkovnih baz nas zanimajo le relacije, katerih moč ali kardinalnost  $m$  je končno število, saj lahko le take relacije shranimo v končno velikem pomnilniku. Številu domen  $n$ , nad katerimi je relacija definirana, pravimo stopnja relacije.

Vsaka izmed domen igra v relaciji določeno vlogo, oziroma ima določen pomen. Če se na listi domen pojavlja ena domena večkrat, potem nastopa očitno vsakič v drugi vlogi. Zato, da bi eksplicitno prikazali pomen domen v relaciji, lahko relacijo predstavimo tudi kot množico preslikav.

Relacija v podatkovni bazi je model nekega dogajanja ali stanja v svetu. Predpostavimo, da vsaka  $n$ -terica relacije predstavlja posamezno

entitet ali povezavo entitet ali oboje hkrati, za kar uporabimo skupno ime objekt. Relacija naj torej predstavlja množico objektov. Vsak objekt lahko okarakteriziramo z njegovimi lastnostmi, pri čemer so zaloge vrednosti posameznih lastnosti domene relacije  $D_1, D_2, \dots, D_n$ . Posamezno lastnost objekta imenujemo atribut  $A_i$ . Atribut je preslikava množice objektov  $O$  v pripadajočo domeno  $D_i$ .

#### 05. Relacijska shema!

Vsaki relaciji pripada relacijska shema. Sestavljata jo oznaka relacijske sheme  $R$  in lista oznak atributov  $A_i$  s pripadajočimi oznakami domen  $D_i$ .

Relacijske sheme predstavljajo semantično komponento podatkovne baze in so del konceptualne oziroma zunanjih shem. Iz njih naj bi bil razviden pomen relacij, ki tem shemam pripadajo. Oznaka relacijske sheme naj bi posredovala informacijo o pomenu relacije, oznake atributov pa pomen posameznih komponent v n-tericah relacije. Z ozirom na dokaj skromne izrazne možnosti nudi relacijska shema informacijo le poznavalcem podatkovne baze, ki znajo oznake iz relacijske sheme pravilno interpretirati.

Vsaki relaciji pripada natanko ena relacijska shema, medtem, ko lahko posamezna relacijska shema pojasnuje tudi več relacij.

#### 06. Podatkovni tipi!

#### 07. Podatkovni model!

Posamezne vrste shem opišemo s pomočjo strukturnega in opisanega mehanizma – podatkovnega modela, imenovanega tudi jezik za opis podatkov. Sestavni del podatkovnega modela je tudi množica operatorjev, ki služijo za rokovanje s tako predstavljenimi podatki. Ti operatorji tvorijo jezik za manipulacijo s podatki oziroma povpraševalni jezik.

Za modeliranje podobe sveta s podatkovno bazo se uporabljajo podatkovni modeli. Obsegajo podatkovno strukturo, operacije nad podatkovno strukturo, operacije nad podatkovno strukturo in integritetne omejitve. Pri modeliranju s podatkovno bazo se na konceptualnem in zunanjem nivoju uporabljajo logični modeli, na notranjem nivoju pa fizični modeli. Logične modele delimo nadalje v površinske in globinske. Med površinske podatkovne modele, ki so zapisno orientirani, štejemo relacijski, mrežni in hierarhični podatkovni model. Poglavitni globinski modeli so podatkovni model »entiteta – razmerje«, binarni podatkovni modeli, podatkovni modeli na osnovi semantičnih mrež in infofolški podatkovni modeli.

Problem sožitja med površinskimi in globinskimi podatkovnimi modeli rešuje koeksistenčni princip, po katerem je konceptualni nivo podatkovne baze razdeljen na infofolški in datološki del.

#### 08. Kartezični produkt domen!

#### 09. Indeksiranje!

Indeks je datoteka, katere vsebina omogoča hiter dostop do zapisov v osnovni fizični datoteki, ki je z indeksom indeksirana. Pod hitrim dostopom razumemo dostop do podatkov osnovne datoteke, ki je hitrejši, kot ga sicer omogoča organizacija osnovne datoteke. Indeks se uporablja tudi za zaporedni dostop do zapisov nezaporedne datoteke. Indeks je možno vedno znova kreirati na osnovi podatkov v osnovni datoteki in je s tega stališča redundanten.

Indeks je praviloma zaporedna datoteka, v kateri so indeksni zapisi urejeni po svojem ključu in sicer s fizičnim zaporedjem, lahko pa je tudi razpršena datoteka z razpršitvijo v skupine.

#### 10. Indeksiranje B+

Pri večnivojskem indeksiranju je poseben problem ažuriranje indeksov pri dodajanju in brisanju zapisov osnovne datoteke. Če se ažuriranju indeksov odrečemo, potem govorimo o statičnem indeksiranju. V takem primeru ostajajo indeksi nespremenjeni ne glede na to, da se je spremenila vsebina osnovne datoteke. To pa pomeni, da se lahko posamezne skupine zapisov v osnovni datoteki zelo razrastejo, druge pa močno zmanjšajo. Posledica so različni iskalni časi za različne vrednosti ključev. Če se iskani zapis nahaja v veliki skupini, bo potrebno preiskati več blokov osnovne datoteke, kot pa če se nahaja v majhni skupini.

V primeru, ko sistem indeksov prilagajamo vsakokratni vsebini datoteke, govorimo o dinamičnem indeksiranju. Eden izmed pogosto uporabljenih načinov dinamičnega indeksiranja je indeksiranje, pri katerem je celoten sistem indeksov zgrajen v bliki uravnovešenega drevesa, od koder izvira njegova oznaka indeksiranje B.

Kot uravnovešeno drevo je zgrajen sistem redkih indeksov, ki je shranjen v posebni fizični datoteki (indeks B). Najnižji nivo indeksa B lahko neposredno indeksira osnovno datoteko, ki mora biti seveda zaporedna datoteka. V tem primeru govorimo o indeksiranju B. Če se med indeksom B in osnovno datoteko nahaja še gosti indeks na osnovno datoteko, kar pomeni, da je lahko le-ta tudi neurejena, govorimo o indeksiranju B+.

#### 11. Indeksiranje B (kako je sestavljeno)!

#### 12. Ključi (lastnost, kaj je to ključ...)

#### 13. Uravnovešeno drevo!

14. **Disk (kako je sestavljen)!**
15. **Delitvene enote!**
16. **Relacijski račun!**
17. **4 poizvedovanja (iskanje v zaporedni datoteki)!**
18. **Indeksi (kaj je indeks, kaj je gosti index, kaj je redki indeks, večnivojsko indeksiranje, zakaj imamo gosti indeks...)!**
19. **Podatkovne nesreče!**  
Str. 207
20. **Kdaj nastane sistemska nesreča?**  
Str. 208
21. **Kako obnovimo PB po taki nesreči?**
22. **Kontrolna točka!**
23. **PXC in PSC protokoli!**
24. **Kaj je to zaseženje?**