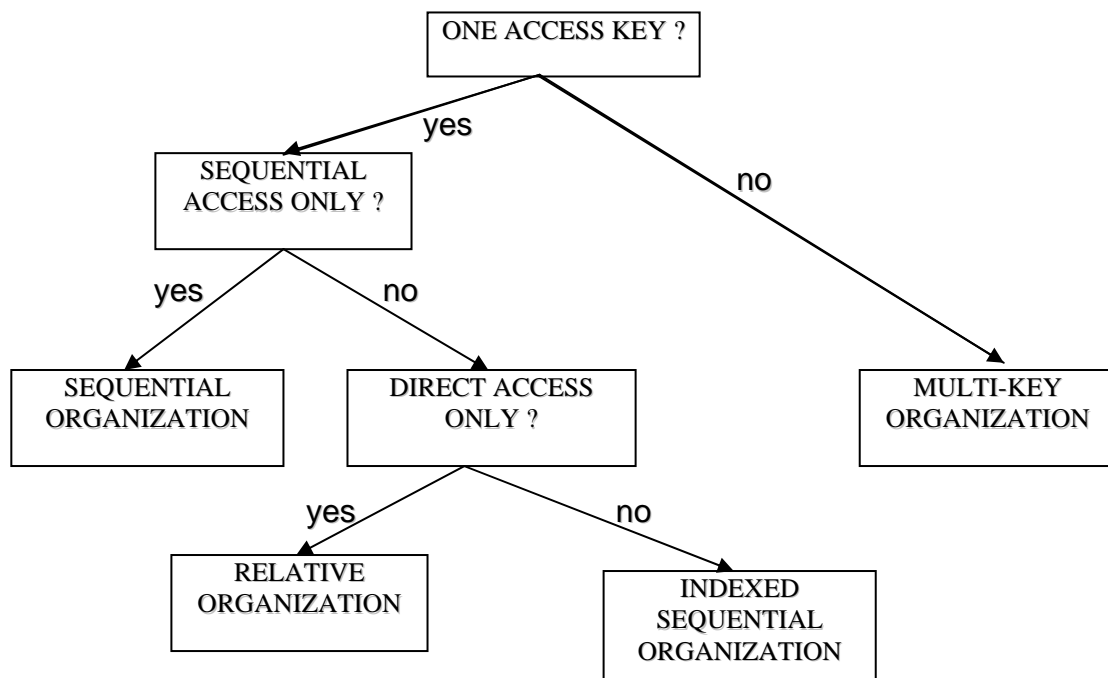


ORGANISASI BERKAS DENGAN BANYAK KEY

PENGERTIAN ORGANISASI BERKAS DENGAN BANYAK KEY

Organisasi berkas yang memperbolehkan record diakses oleh lebih dari satu key field disebut organisasi berkas dengan banyak key.



Gambar 1. Dasar Organisasi Berkas

Ada banyak teknik yang dipakai untuk organisasi berkas dengan banyak key ini. Hampir semua pendekatan bergantung pada pembentukan indeks yang dapat memberi akses langsung dengan nilai key.

Ada 2 teknik dasar untuk pemberian hubungan antara sebuah indeks dan data record dari berkas, yaitu:

- Inversion
- Multi-list

DEFINISI DAN APLIKASI BERKAS DENGAN BANYAK KEY

Banyak sistem informasi interaktif memerlukan dukungan dari berkas banyak key.

Contoh:

Sebuah sistem perbankan yang mempunyai beberapa pemakai (user), seperti kasir, pegawai kredit, manajer cabang, pegawai bank, nasabah, dll.

Semuanya memerlukan akses data yang sama dengan format record:

ACCOUNT

ID	NAME		GROUP-CODE		SOCNO	BALANCE	OVERDRAW LIMIT
	LAST	FIRST	BRANCH	TYPE			

Gambar 2. Contoh Format Record

Adanya pemakai yang berbeda memerlukan akses record-record ini dalam cara yang berbeda.

Kasir → Mengidentifikasi record account menurut nilai ID.

Kredit → Akses semua record menurut nilai OVERDRAW LIMIT atau semua record account dengan nilai SOCNO.

Manajer Cabang → Akses semua record menurut Branch dan Type.

Pegawai Bank → Membuat laporan berkala untuk semua record account yang disortir berdasarkan ID.

Nasabah → Memerlukan akses recordnya dengan memberikan ID yang dimilikinya atau kombinasi dari NAME, SOCNO dan Type.

Satu pendekatan yang dapat mendukung semua jenis akses adalah dimilikinya banyak berkas yang berbeda. Setiap berkas di organisasi untuk melayani satu jenis keperluan.

Maka untuk contoh sistem perbankan di atas harus ada:

- File account yang organisasinya indeks sekuensial dengan nilai key:
ID → untuk melayani kasir, pegawai bank & nasabah.
- File account yang organisasinya sekuensial dengan record diurut menurut:
OVERDRAW LIMIT → untuk melayani pegawai kredit.
- File account yang organisasinya relatif dengan nilai key:
SOCNO → untuk melayani pegawai kredit.
- File account yang organisasinya sekuensial dengan record diurut menurut:
GROUP-CODE → untuk melayani manajer cabang.
- File account yang organisasinya relatif dengan nilai key:
NAME, SOCNO dan TYPE → untuk melayani nasabah.

Jadi kita mempunyai 5 file, semuanya mempunyai record yang sama. Kelima file itu hanya berbeda dalam organisasi dan cara aksesnya.

Pengulangan data dari beberapa file bukan merupakan cara yang baik untuk mengakses record dengan berbagai cara. Dan cara ini memerlukan ruang (space) yang besar di storage dan kesulitan pada waktu peng-*update*-an record secara serentak.

Untuk mengatasi masalah di atas, maka digunakan organisasi berkas banyak key yang umumnya diimplementasikan dengan pembentukan banyak indeks untuk memberikan akses yang berbeda terhadap record data.

Mungkin juga cara ini memakai banyak link-list terhadap record. Dan sebuah indeks dapat dibentuk dengan beberapa cara, misal sebagai tabel binary search tree atau B-tree.

ORGANISASI INVERTER FILE

Satu pendekatan dasar untuk memberikan hubungan antara sebuah indeks dan data record dari file adalah inversi.

Sebuah key pada indeks inversi mempunyai semua nilai key dimana masing-masing nilai key mempunyai penunjuk ke record yang bersangkutan. File yang demikian disebut inverted file.

Indeks inversi yang sederhana dibentuk sebagai sebuah tabel.

Contoh:

Inversi file ACCOUNT terhadap SOCNO menghasilkan indeks inversi.

SOCNO	ADDRESS
001234123	6
123456789	5
213823456	14
299167305	16
399042131	4
411625331	13
412631467	7
426135447	20
445062681	17
459463001	1
513014265	8
612305116	9
626311660	18
692122222	12
713214622	3
798392289	15
821346211	2
912346789	19
923146154	11
987654321	10

Gambar 3. Indeks Inversi Record dengan key SOCNO

Indeks inversi tersebut telah di sortir menurut SOCNO. Penambahan record file menyebabkan indeks inversi juga harus diubah.

Sebuah indeks inversi dapat dibuat bersama sebuah relatif file atau sebuah indeks sekuensial. Sebuah indeks inversi dengan key SOCNO untuk sebuah relatif file dengan nilai key ID akan memberikan sebuah file yang dapat diakses langsung oleh sebuah ID atau SOCNO.

SOCNO	ID
001234123	291821
123456789	208432
213823456	486212
299167305	534216
399042131	201431
411625331	415631
412631467	300123
426135447	696132
445062681	535218
459463001	111111
513014265	310103
612305116	324196
626311660	641230
692122222	387122
713214622	198121
798392289	511211
821346211	112131
912346789	686116
923146154	386132
987654321	335812

Gambar 4. Indeks Inversi Record dengan key SOCNO Menggunakan Indirect Addressing

Jika sebuah key dipakai untuk menentukan struktur storage dari file disebut primary key, sedangkan key yang lainya disebut secondary key.

File yang mempunyai indeks inversi untuk setiap data field disebut completely inverted. File yang bukan completely inverted tapi paling sedikit mempunyai satu indeks inversi disebut partialy inverted file.

Sebuah variasi dari struktur indeks inversi adalah pemakaian secondary key dan primary key dari indirect addressing. Pendekatan ini membiarkan file yang direorganisasi dan restructure secara fisik tanpa menyebabkan indeks file. Pada contoh di atas, nilai SOCNO adalah unik. Setiap SOCNO dalam indeks inversi mempunyai record. Indeks inversi dapat juga dipakai dengan key yang bukan unik.

GROUP-CODE BRANCH TYPE		ID
DT	001	111111, 201431, 310103, 686116
DT	002	198121, 335812, 486212, 641230
EA	001	324196, 534216, 535218
NE	001	415631
NE	002	300123, 696123
NW	001	112131, 208432, 511211
NW	002	291821
WE	002	386132, 387122

Gambar 5. Indeks Inversi Record dengan key GROUP-CODE

Satu hal yang baik dari inversi file adalah beberapa pertanyaan di bawah ini dapat dijawab tanpa akses data file, cukup dengan indeks inversinya.

- Apakah ada indeks untuk SOCNO = '414556233' ?
- Berapa banyak account dengan GROUP-CODE = 'EA001' ?
- Berapa banyak account dengan BRANCH = 'NE' ?
- Apakah BRANCH = 'NE' mempunyai TYPE = '001' ?

ORGANISASI MULTI-LIST FILE

Suatu pendekatan lain yang memberikan hubungan antara sebuah indeks dan data record dari sebuah file disebut organisasi multi-list file.

Seperti sebuah inverted file, sebuah multi-list file mempunyai sebuah indeks untuk setiap secondary key.

Organisasi multi-list file berbeda dengan inverted file, dimana dalam indeks inversi untuk sebuah nilai key mempunyai sebuah penunjuk untuk sebuah data record dengan nilai key, sedangkan dalam indeks multi-list untuk sebuah nilai key mempunyai hanya sebuah penunjuk untuk data record pertama dengan nilai key. Data record mempunyai sebuah penunjuk untuk data record selanjutnya dengan nilai key dan seterusnya. Maka terdapat sebuah linked-list dari data record untuk setiap nilai dari secondary key.

GROUP-CODE BRANCH TYPE		ID
DT	001	111111
DT	002	198121
EA	001	324196
NE	001	415631
NE	002	300123
NW	001	112131
NW	002	291821
WE	002	386132

Gambar 6. Multi-list Index for GROUP-CODE secondary key

OVERDRAW LIMIT	ID
0.	111111
100.	112131
200.	198121
250.	511211

Gambar 7. Multi-list Index for OVERDRAW-LIMIT secondary key
 Gambar 6 dan 7, menunjukkan indeks multi-list untuk secondary key GROUP-CODE dan OVERDRAW LIMIT, sedangkan gambar 8

menunjukkan data file. Setiap data record mempunyai penunjuk untuk mengakses record selanjutnya.

Record Address	ID	NAME		GROUP-CODE		NEXT	SOCNO	BALANCE	OVERD. LIMIT
		LAST	FIRST	BRANCH	TYPE				
1	111111	DEAN	JOHN	DT	001	201431	459463001	100.50	0.
2	112131	AMORE	CAROL	NW	001	208432	821346211	2311.20	100.
3	198121	RICE	BESS	DT	002	335812	713214622	-191.87	200.
4	201431	PARKER	OLA	DT	001	310103	399042131	3142.93	100.
5	208432	ANDERSON	BARBA	NW	001	511211	123456789	95.26	0.
6	291821	ANDERSON	BRENT	NW	002	0	001234123	146.92	0.
7	300123	HART	ALLIS	NE	002	696123	412631467	3000.00	100.
8	310103	TAHOE	SEREN	DT	001	686116	513014265	243.63	0.
9	324196	BONES	IGOR	EA	001	534216	612305116	0.02	0.
10	335812	BEE	VERA	DT	002	486212	987654321	88.72	0.
11	386132	ARC	JEAN	WE	002	387122	923146154	-89.98	100.
12	387122	ELLS	JOSHU	WE	002	0	692122222	10.95	0.
13	415631	BELLE	ELLIE	NE	001	0	411625331	426.86	100.
14	486212	MATTHERS	ANN	DT	002	641230	213823456	893.92	100.
15	511211	SNUGGS	JOHN	NW	001	0	798392289	2814.12	250.
16	534216	LOOMIN	TIM	EA	001	535218	299167305	296.12	250.
17	535218	ELLSWORT	NANCY	EA	001	0	445062681	2814.12	250.
18	641230	WELLS	CHUCK	DT	002	0	626311660	-98.14	100.
19	686116	SWORTH	LAUR	DT	001	0	912346789	3216.16	100.
20	696132	GUNNESS	MARIO	NE	002	0	426135447	95.93	25.

Gambar 8. Contoh Data File

Nilai key harus di urut, struktur indeks adalah tabel dengan indirect addressing dan mempunyai hubungan data record yang disusun menurut ID secara ascending.

GROUP-CODE		
BRANCH TYPE	ID	LENGTH

DT	001	111111	4
DT	002	198121	4
EA	001	324196	3
NE	001	415631	1
NE	002	300123	2
NW	001	112131	3
NW	002	291821	1
WE	002	386132	2

Gambar 9. Variasi Index pada gambar 6, menunjukkan length of linked-list

OVERDRAW LIMIT	ID	LENGTH
0.	111111	7
100.	112131	8
200.	198121	1
250.	511211	4

Gambar 10. Variasi Index pada gambar 7, menunjukkan length of linked-list

Informasi tentang banyak record dalam link-list juga berguna untuk mendapatkan cara yang terbaik dalam pengaksesan.

Contoh:

Ada 3 cara akses yang potensial dalam menjawab pertanyaan:

Daftar nilai ID untuk account dengan GROUP-CODE ='EA001' dan OVERDRAW-LIMIT = 100 ?

Data file dapat dicari secara sekuensial, indeks GROUP-CODE atau dengan memakai OVERDRAW-LIMIT.

Mana cara yang terbaik ?

Pencarian secara sekuensial memerlukan akses sampai 20 data record, menggunakan indeks GROUP-CODE berarti akses sampai 3

data record dan memakai indeks OVERDRAW-LIMIT berarti akses sampai 8 data record.

Dari ketiga cara di atas, yang terbaik adalah menggunakan indeks GROUP-CODE.

DEKLARASI BERKAS DALAM COBOL

```
SELECT ACCOUNT-FILE ASSIGN TO DISK
  ORGANIZATION IS INDEXED
  ACCESS MODE IS SEQUENTIAL OR RANDOM OR DYNAMIC
  RECORD KEY IS ID
  ALTERNATE RECORD KEY IS SOCNO
  ALTERNATE RECORD KEY IS GROUP-CODE
  WITH DUPLICATES.
```

SOCNO	ID
459463001	111111
821346211	112131
713214622	198121
399042131	201431
123456789	208432
001234123	291821
412631467	300123
513014265	310103
612305116	324196
987654321	335812
923146154	386132
692122222	387122
411625331	415631
213823456	486212
798392289	511211
299167305	534216
445062681	535218
626311660	641230
912346789	686116
426135447	696132

Gambar 11. Contoh Hasil Program COBOL