

TÉMATICKÝ OKRUH

Počítače, sítě a operační systémy

Číslo otázky : 12.
Otázka : Metody fyzické organizace dat

Obsah :

- 1.Úvod
- 2.Vnější paměti
- 3.Sekvenční soubory
- 3.1 Setříděné sekvenční soubory
- 4.Zřetěžené organizace záznamů
- 5.Soubory s přímým adresováním
- 6.Indexované a indexové soubory
- 7.Hierarchické indexování, B-stromy
- 7.1 Hierarchické indexování
- 7.2 B-stromy
- 8.Indexování pomocí binární matice
- 9.Datové soubory s proměnnou délkou záznamu
- 9.1 Pseudoproměnná délka záznamu
- 9.2 Proměnná délka záznamu v sekvenčním souboru
- 9.3 Proměnná délka záznamu s jinou organizací

1. Úvod

Připomeňme si, že hlavní důvodem existence databází je evidovat v nich rozsáhlá data. Ale samotná evidence by nebyla k velkému užítku, kdyby se uložené informace nedaly podle okamžitých potřeb vyhledávat a zpracovávat. U rozsahem malých databází (stovky až tisíce záznamů) v době rychlých počítačů příliš na způsobu uložení dat možná nezáleží. Ale z rozsáhlých databází ani rychlé počítače nevyhledají potřebná data dostatečně rychle, pokud jejich uložení nebude vhodně organizováno.

Výkon a tím i efektivita používání databáze je velmi závislá na implementaci operací na fyzické úrovni. Komunikaci s databází na logické úrovni zajišťují čtyři základní databázové operace:

- vyhledávání záznamu podle hodnoty jedné nebo více položek SELECT nebo FIND
- vložení nového záznamu INSERT
- modifikace záznamu UPDATE nebo MODIFY
- rušení záznamu DELETE nebo ERASE.

Při modifikaci nebo rušení záznamu musíme nejprve příslušný záznam vyhledat a potom s ním provést příslušnou operaci. Uložení nových záznamů bude u různých organizací různě rychlé, ale na čas obvykle méně náročné. Tedy klíčovou operací z hlediska rychlosti je vyhledání záznamu.

2. Vnější paměti

Diskem nazýváme jednu nebo celý svazek kruhových desek pokrytých magnetickou vrstvou. Každou desku pokrývají soustředné kružnice nazývané stopy, do těchto stop se zapisují data. Deska je dále rozdělena na kruhové výseče. Část stopy v jedné výseči se nazývá sektorem a je to nejmenší adresovatelná jednotka na disku. Pokud disk tvoří několik desek nad sebou, všechny stopy nad sebou nazýváme válec či cylindr.

Diskovou adresu na fyzické úrovni tedy tvoří označení diskové jednotky, číslo cylindru, stopy a sektoru.

Blok (stránka) je základní jednotku informace pro přenos dat mezi pamětí vnitřní a vnější. Rychlost takového přenosu i u rychlých systémů je řádově 1000x pomalejší, než rychlost operací ve vnitřní paměti. Tedy rychlost zpracování dat záleží na počtu přenosů dat mezi diskem a pamětí a prakticky nezáleží na počtu operací v paměti. Počet přenosů dále velmi závisí na organizaci uložení dat v diskových souborech.

Vyhledat záznam na disku tedy znamená tři etapy zpracování:

- Uživatel zadá logickou podmínku na hledaný záznam v rámci aplikačního programu nebo pomocí dotazovacího jazyka
- Na základě logické podmínky pro záznam určí SŘBD logickou adresu záznamu, tj. dle okolností pořadové číslo záznamu pro soubor s pevnou délkou nebo relativní adresu v rámci souboru. V obou případech SŘBD na základě znalosti své organizace dat spočítá číslo bloku, ve kterém je záznam uložen a určí začátek záznamu v bloku.
- Z čísla bloku se určí číslo válce, stopy a sektoru, tedy určit fyzickou adresu záznamu. Tuto etapu řeší obvykle součást operačního systému - subsystém ovládání souborů. Ten na základě zadaného čísla bloku v souboru spočítá absolutní číslo válce, stopy a sektoru. Pak na základě adresy záznamu v bloku a jeho délky najde hledaný záznam.

3. Sekvenční soubory

Nejjednodušší organizací pro implementaci, vycházející z přirozeného uspořádání záznamů podle pořadí jejich uložení, jsou sekvenční soubory. Věty jsou uloženy v souboru v libovolném pořadí v blocích, ty mohou následovat fyzicky za sebou. Pokud bloky nenásledují za sebou, jsou propojeny ukazateli, nebo jsou adresy bloků souboru někde uloženy, to obvykle řeší OS.

Základní databázové operace se provádějí následovně:

Nový záznam se uloží na konec souboru.

Vyhledání záznamu se zadaným pořadím se určí z pořadí a délky záznamu adresa záznamu přímo. Při vyhledávání podle hodnot položek je nutno prohledávat datový soubor sekvenčně: každý záznam postupně načíst a otestovat, zda vyhovuje podmínce. Složitost takového vyhledávání je $n / (2 * B)$ přístupů na disk (n ...počet záznamů, B ...počet záznamů v bloku).

Modifikace záznamu znamená provést tyto operace: nalézt záznam, načíst, opravit a na stejnou adresu znovu zapsat.

Zrušení záznamu u sekvenčních souborů se obvykle provádí ne vymazáním záznamu, ale pouze označením jeho neplatnosti. K označení neplatnosti se obvykle vyhradí místo v záznamu (bit, byte) a vlastní položky záznamu zůstanou zachovány. Dochází tak k narůstání velikosti souboru, proto je vhodné občas provést překopírování pouze platných záznamů do nového souboru.

3.1. Setříděné sekvenční soubory

Pokud se v souboru často vyhledává podle některého klíče a provádí se relativně málo změn těchto klíčů, je vhodné uchovávat soubor v setříděném tvaru podle vyhledávacího klíče. Znamená to po každé změně znovu soubor setřídít, což výrazně zpomaluje tyto operace.

Pak se ale dá vyhledávat podle klíče mnohem rychleji. Počet přenosů pro binární hledání je průměrně $\log_2 n$. S výhodou se tedy tento způsob uložení používá u souborů s velmi řídkými změnami.

4. Zřetěžené organizace záznamů

Proti sekvenčnímu souboru obsahuje každý záznam navíc jednu položku. V ní je ukazatel na následující záznam v souboru podle daného uspořádání. V souboru tak je vytvořen seznam či řetěz vzájemně propojených záznamů.

Základní DB operace:

SELECT - provádí se tak, že se seznam prohledává postupně pomocí ukazatelů a testuje, zda záznam vyhovuje vyhledávací podmínce. Vyhledávání v seznamu bude vyžadovat častější přechody mezi bloky souboru a proto více přenosů mezi diskem a pamětí.

INSERT - provede se tak, že se záznam fyzicky zapíše kamkoliv, pak se v seznamu vyhledají sousední záznamy dle udržovaného pořadí a přesměrují se ukazatele.

DELETE – provede se tak, že se vyhledá umístění záznamu v seznamu a přesměrují se ukazatele.

UPDATE - znamená jen vyhledání záznamu a po modifikaci jeho zápis zpět. Jen při modifikaci položek, které mají vliv na uspořádání seznamu, se provede modifikace jako DELETE a INSERT.

Obvykle se zřetěžených organizací užívá v kombinaci s některou jinou metodou. Seznamy pak jsou dostatečně krátké, aby byly zachovány výhody zřetěžení.

5. Soubory s přímým adresováním

Jedná se o velmi rychlý přístup k záznamům. Jednoznačný klíč záznamu se zakóduje do jednoznačného čísla, toto číslo pak přímo určuje adresu záznamu v souboru. Tak je možné jediným přístupem na disk záznam načíst, příp. druhým přístupem záznam po modifikaci zapsat.

Interval získaných adres však může být ohromný a velmi řídké obsazený. Tak by docházelo k velkému plýtvání kapacitou paměti. Proto se užívá speciální funkce zvané hašovací, která transformuje původní interval adres do číselného intervalu požadované velikosti. Velikost výsledného intervalu je zvolena tak, aby zhruba odpovídala skutečnému počtu záznamů.

Pokud několika různým záznamům odpovídá jedna adresa získaná pomocí hash funkce, ukládají se záznamy pomocí zřetězení. Protože již jde o krátké seznamy, je přístup k nim rychlý.

Základní DB operace:

Vyhledání záznamu podle klíče je tedy velmi rychlé a odtud jsou rychlé i ostatní databázové operace. Čas potřebný k výpočtu adresy pomocí hašovací funkce je zanedbatelný.

Vyhledávání podle neklíčové hodnoty se však naopak prodlužuje, protože je potřeba sekvenčně procházet prázdná místa a seznamy. Také požadavek na seřídění záznamů znamená komplikaci.

Pro nový záznam se spočítá adresa skupiny záznamů, v ní se prohledají záznamy (pro kontrolu jednoznačnosti klíče), nový záznam se pak uloží na první volné místo ve skupině.

Rušení se provede vyhledáním, označením neplatnosti a přesměrováním ukazatelů z předchůdce na následníka.

Modifikace znamená vyhledání, modifikaci a uložení zpět. Jen při modifikaci klíče se provede nejprve zrušení a pak nový záznam.

6. Indexované a indexové soubory

Základem indexované organizace je sekvenční datový soubor, ke kterému existuje jedna nebo několik dalších pomocných tabulek, pomocí nichž můžeme v datovém souboru rychleji hledat. Základní datový soubor, který je doplněn takovou pomocnou tabulkou, nazýváme *indexovaným souborem*, pomocnou tabulku nazýváme *indexovým souborem*.

Do pomocné tabulky se umístí hodnota vyhledávacího klíče a adresa záznamu. Tabulka je organizována tak, aby vyhledání klíčové hodnoty proběhlo rychle pro libovolnou část datového souboru. Například při seřídění pomocné tabulky dle indexu se hledá binárně hodnota klíče, na vyhledaném řádku v indexovém souboru se zjistí hodnota adresy v datovém souboru a jediným přístupem do dat se načte hledaný datový záznam.

Hlavní myšlenkou indexování tedy je, že ukazatele nemusí být vždy součástí záznamů (jako u zřetězených organizací), ale mohou být uloženy ve zvláštním (indexovém) souboru.

Pokud je indexem klíč, mluvíme o *primárním indexování*, v ostatních případech o *sekundárním indexování*. Při sekundárním indexování musíme počítat s tím, že stejné hodnoty indexu v datech nabývá více záznamů a tak jedné hodnotě položky odpovídá více adres.

Pro sekundární indexování se také používá název *invertovaný soubor*. Jestliže jsou tabulky indexů vytvořeny pro všechny atributy říkáme, že je soubor *úplně invertovaný*. V tomto případě je zapotřebí více než dvakrát tolik paměti, než potřebuje sám datový soubor.

Základní DB operace:

Vkládání nového záznamu - záznam uloží na konec datového souboru, přidá se do všech indexových souborů a ty se seřídí.

Vyhledávání - podle některého existujícího indexu se provádí binárním vyhledáním hodnoty indexu v příslušném indexovém souboru a odtud adresy záznamu v datovém souboru.

Vyhledávání (i modifikace a rušení) záznamu - podle neindexovaných položek se provádí jako u sekvenčního souboru. Ovšem při změně indexovaných položek je opět nutné upravit indexové soubory.

Modifikace položek a rušení záznamu, vyhledávaného pomocí indexu je rychlá. Vyhledají se a po modifikaci zapíší zpět, případně označí za neplatné. Při modifikaci některé indexované položky je nutné přetřídit příslušný indexový soubor.

V některých aplikacích mohou být klíče velmi dlouhé. Se zvětšováním délky klíče se prodlužují seznamy záznamů, odkazující na shodné hodnoty pod klíče. Pak se prodlužuje i hledání v takových indexech. Proto se někdy ukládají klíče v indexech zkráceným způsobem tak, že je v tabulce uložena předpona klíče a je připojen seznam přípon s adresami, potom další předpona atd. Stejná metoda je používána v jazykových slovnících.

Technika indexových souborů zrychluje přístup k tabulkám, ale za cenu větší spotřeby místa na disku. Je také zapotřebí větší režie pro údržbu indexových souborů. Proto je indexování výhodné především pro soubory s malým objemem změn a pro malé a střední rozsahy dat.

7. Hierarchické indexování, B-stromy

7.1 Hierarchické indexování

Pro zvláště velké soubory je možno pro hledání použít opět indexový soubor a sestavit tak celou hierarchii indexových souborů. Pak se na vyšších úrovních indexace objevují indexy skupin indexů nižší úrovně. V rámci nalezené skupiny se dohledávají záznamy sekvenčně nebo opět binárně. Na indexy 1. úrovně mohou odkazovat indexy 2. úrovně atd. Hledá se od nejvyšší úrovně. V indexech vyšších úrovní se nevyhledává přesná hodnota indexovaného atributu, ale interval, do kterého padne.

7.2 B-stromy

Pro uspořádání úrovní indexů se velmi často používá varianta hierarchického indexování pomocí tzv. B-stromů (Balanced-tree). Listy stromu vzniknou tak, že se indexové soubory všech úrovní (mimo kořen) rozsekají na části. Listy odpovídají velikosti bloku a jsou tak načteny jediným přenosem z disku. V rámci malého a seřazeného uzlu se najde nejbližší hodnota (nejblíže nižší nebo nejbliže vyšší podle konstrukce stromu) hledaného vyhledávacího klíče velmi rychle. Platí, že všechny cesty od kořene stromu do libovolného listu jsou stejně dlouhé – stejným počtem přenosů se dojde k datům.

Základní DB operace:

Hledání - najdeme cestu ve stromě od kořene až k listu, v němž by měl být hledaný záznam. V každém uzlu najdeme správnou větev porovnáním hledaného klíče s klíči v uzlu. Klíče v uzlu mohou udávat minimální (příp. maximální) hodnotu klíče, která je příslušnou větví dosažitelná.

Modifikace záznamu - záznam se vyhledá, změní a zapíše zpět. Při modifikaci klíče se provede zrušení původního záznamu a zavedení nového v indexu0 (případně vyšších, viz vkládání).

Vkládání nového záznamu – najde se příslušný blok a mohou nastat dvě možnosti: buď v nalezeném bloku je místo, takže se může přidat vkládaný záznam, nebo nalezený blok je plný, takže se musí vytvořit nový blok. Z původního plného bloku se vytvoří dva bloky, každý obsahuje polovinu záznamů a zbylá místa jsou prázdná. Do vyšší úrovně musíme nový blok nižší úrovně zaznamenat. Proces se opakuje až do kořene stromu a případně se musí kořen rozdějit. Pak se přidá nový kořen a vznikne o úroveň víc v indexové struktuře.

Rušení záznamů – provádí se opačně, než vkládání. Při zrušení posledního záznamu listu stromu se zruší i odkaz na něj, totéž se promítne do vyšších úrovní, případně se v krajním případě může hierarchie indexů o jednu úroveň snížit.

8. Indexování pomocí binární matice

Pro sekundární indexování se někdy z důvodů ušetření kapacity používá jiného typu indexů - binárních matic. Poloha záznamu se bude zaznamenávat polohou jedničkového bitu v posloupnosti, která má tolik bitů, kolik má soubor záznamů. První bit odpovídá prvnímu záznamu, druhý druhému atd. Pro každou hodnotu sekundárního atributu je zaznamenána nová posloupnost. Zřejmě je tato metoda vhodná pro takové atributy, které nabývají jen několika málo různých hodnot.

Binární matice jsou zvláště výhodné v případech, že se hodnoty sekundárních atributů nemění, když se nemění záznamy, případně se jen přidávají sériově na konec souboru. Další výhodou binárních matic je snadná realizace kombinovaných dotazů pomocí logických operátorů negace, konjunkce a disjunkce.

V případech, které nevyužijí jmenovaných výhod, se používají normální indexy.

9. Datové soubory s proměnnou délkou záznamu

Používání souborů s proměnnou délkou záznamu vede k řadě nových problémů. Často se úvahy o datových typech vedoucích k proměnné délce záznamu vyskytují jen v logickém modelu a implementace se provádí pomocí záznamů pevné délky. Novější SŘBD však stále častěji připouštějí různé datové typy proměnné délky a také je tak implementují.

9.1 Pseudoproměnná délka záznamu

Takto nazveme případ, kdy se logicky proměnná délka záznamu implementuje jako záznam s pevnou délkou.

Používají se k tomu následující způsoby:

- pole se známým počtem opakování*: pole atomických položek se rozloží na jednotlivé položky, každá dostane vlastní jméno a pracuje se s ní samostatně. Při zpracování to ale komplikuje přístup k opakovaným položkám jako celku.
- pole s neznámým počtem opakování*: odhadne se shora počet výskytů prvků pole a tak se převede na předcházející případ. Může se stát, že značná část prvků pole bude nevyužitá. Dalším problémem je pak rozpoznání prázdného prvku – může se řešit například další položkou „počet“.
- místo opakující se položky* uvedeme odkaz na seznam jejích prvků, ty mohou tvořit záznamy jiného souboru; soubor proměnné délky se převede na dva soubory pevné délky.
- pro záznamy s alternativními skupinami položek*: buď se proměnná část "překrývá" a záznam zabírá velikost nejdelší z proměnných částí, avšak v záznamu se musí rozlišovat typ proměnné části a implementace je složitější, nebo se všechny rozdílné atributy zaznamenají "za sebou" a pro každý typ se vyplňují jen odpovídající atributy. Implementace je jednodušší, záznam však obsahuje vždy řadu prázdných položek.

9.2 Proměnná délka záznamu v sekvenčním souboru

Při sekvenční organizaci souborů s proměnnou délkou záznamu je nutno od sebe jednotlivé záznamy rozlišit. Používají se např. tyto metody :

- systém oddělovačů*: záznamy jsou odděleny oddělovačem (viz např. textové soubory), uvnitř záznamu se atributy oddělují jiným typem oddělovače, opakující se položky dalším typem apod.
- zaznamenání délky aktuálního záznamu na začátku záznamu* (pro jednosměrný průchod souborem), či na začátku i konci záznamu (pro obousměrný průchod souborem)

9.3 Proměnná délka záznamu s jinou organizací

Zřetěžené organizace, přímé adresování, indexování, případně další organizace je možno

implementovat podobně, jako při pevné délce záznamu. Rozdíl je pouze v tom, že místo pořadového čísla záznamu je nutno zaznamenávat skutečnou adresu záznamu v souboru, což obecně zabírá více místa.