

Fyzická implementace – zadání a některá řešení⁽¹⁾

1. Z kolika a jakých částí se skládá čas pro vstupně výstupní operaci? Ze tří částí:

- *Seek time* je čas, než se hlava disku dostane nad správnou stopu.
- *Rotational delay* je čas, než se čtecí hlava dostane v rámci cílové stopy na cílový sektor.
- *Data transfer time* – již načtení daných dat z nalezené adresy do paměti.

2. Jaký je rozdíl mezi shlukovaným a neshlukovaným indexem? Výhody a nevýhody?

Indexování je pomocná datová struktura (ve zvláštním souboru), která urychluje vyhledávání v databázi dle jednoho či více klíčů, ke kterým obsahuje řid *record id*. Podle toho, jak je tato struktura setříděná, rozlišujeme dva typy indexů:

- *Shlukované indexy* jsou setříděné v pořadí odpovídajícím pořadí dat (na které tyto indexy odkazují) ve stránkách datového souboru. Shlukované mohou být pouze stromové indexy či indexy obsahující celé záznamy (i hašované), což jsou ale vlastně přímo ta data. Výhodou je, že stránky se záznamy se pak dají načítat sekvenčně, což je výhodné (zrychlení) speciálně při dotazech na rozsah, resp. interval.
- *Neshlukované indexy* jsou setříděné nějak jinak než data v datovém souboru. Výhodou je režie na udržování této datové struktury. (Udržovat indexy setříděné dle klíčů stejně jako data je náročné.)

3. Proč používáme stránkování?

Pro stránkování existuje mnoho důvodů. Nejdříve jak to funguje. Data jsou uložena na disku v blocích pevné velikosti (několik kB), ke kterým je HW schopen přistupovat. (Načte vždy celý blok). Jednotlivé bloky (*rámce*) mohou být po disku rozházené neuspořádaně. Nicméně uživatel databáze (program atd.) nepřistupuje přímo k těmto datům na disku, ale ke stránkám, což jsou bloky stejné velikosti jako rámce na disku, ale jsou setříděné logicky podle uspořádání dat. Existuje mapování ze stránek virtuální paměti na rámce na disku.

Obecných výhod stránkování je několik (oddělení virtuální paměti od fyzické (ochrana dat), možnost odebrání přidělené paměti, možnost skrytí

⁽¹⁾ *Za nic neručím. Jsou to pouze pracovní poznámky.*

skutečné velikosti paměti, možnost sdílení dat, atd.) Zde je asi nejdůležitější to, že je pro HW jednodušší načíst blok dané velikosti, než k datům přistupovat po jednotlivých bytech. A podstatné asi také je, že takto nemusí být data na disku nikterak za sebou (pouze vždy v rámci jednoho rámce).

4. **Jaké jsou výhody organizace dat v haldě?**

Záznamy ve stránkách jsou uloženy neuspořádaně sekvenčně tak, jak byly postupně vkládány. Výhodou je rychlé vkládání (v konstatním čase). Naopak pomalejší je hledání a mazání (pro hledání je v nejhorším případě nutno projít všechna data na stránce). Mazáním mohou vznikat kusy prázdného prostoru.

5. **Mějme tabulku *Student(jméno, věk, pohlaví)*. Jaké druhy indexů se hodí pro jednotlivé atributy v tabulce?**

Na pohlaví se hodí bitová mapa. Index na jméno by mohl být hashovaný (pokud bychom očekávali intervalové dotazy, tak B^+ strom) a na věk B^+ strom, neboť tam by mohly být intervalové dotazy poměrně časté.

6. **Co je to „seek time“?**

Čas, než se hlava disku dostane nad správnou stopu.

7. **Mějme tabulku s atributy (*jméno, pohlaví, barvaOčí*) a v ní (Karel, muž, hnědá), (Eva, žena, modrá), (Jan, muž, černá), (Hana, žena, hnědá). Vytvořte bitové indexy pro *poohlaví* a *barvaOčí* nad použitými hodnotami.**

	pohlaví	barvaOčí		
	muž	hnědá	modrá	černá
1	1	0	0	0
0	0	1	0	0
1	0	0	1	1
0	1	0	0	0

8. **Vysvětlete použití indexů z (1) při vyhodnocení dotazu „Které ženy mají modré či černé oči?“**

Dotaz bude vypadat následovně:

(not bitmap(muž) AND (bitmap(modré) OR bitmap(černé)))

$$\neg(1, 0, 1, 0) \& ((0, 1, 0, 0) \vee (0, 0, 1, 0)) = (0, 1, 0, 1) \& (0, 1, 1, 0) = (0, 1, 0, 0)$$

Výsledek je tedy Eva.

9. **Jaké jsou výhody organizace dat v uspořádaném souboru?**

V uspořádaném souboru jsou záznamy ve stránkách uložené setříděně dle vyhledávacího klíče.

Výhody: rychlé vyhledávání podle klíče (na rovnost i na rozsah). A prostor je zaplněn bez prázdných meziprostorů.

Nevýhody: pomalu se vkládá a maže, často je potřeba přesouvat i ostatní data.

10. **Kde není výhodné použít bitmapový index?**

Když nabývá atribut mnoha hodnot (má velkou doménu). Špatně se vyhledává na rozsah (neexistuje uspořádání).

11. **Kdy a na jaký typ je vhodné použít prostorové indexy? Jaké je jejich omezení?**

Když předpokládáme vícerozměrné dotazy, tedy hledání dle n -tice atributů. Vhodné i pro vyhledávání na rozsah. Nicméně náročné na paměť a vhodné pouze do dimenze cca 10.

12. **Popište, jak funguje buffer diskových stránek.**

Paměť, do které se „cachují“ diskové stránky (v poměru 1:1). Ušetří se tím mnoho pomalých přístupů na disk, které se nahradí mnohem rychlejšími přístupy do paměti. Buffer obsahuje ke každé stránce dva příznaky – *pincount*, tj. počet žádostí na danou stránku a *dirty*, který říká, zda byla stránka v paměti upravena, ale na disku odpovídající rámec ještě ne.

13. **Jaké typy indexů používáme pro prostorové dotazy?**

Pro prostorové indexování se používají indexy založené na stromové struktuře, hašování, či i sekvenčním průchodu.

14. **Jakým způsobem mohou být data organizována na disku, jaké jsou základní operace a jejich složitosti?**

operace	halda	uspořádaný soubor	hašovaný soubor
sekvenční načtení záznamu	N	N	N
vyhledání záznamu na rovnost	$\frac{N}{2}$ či N	$\log_2 N$ či N	$\frac{N}{K}$ ideálně
vyhledání záznamu na rozsah	N	$\log_2 N + M$	$\frac{N}{K}$
vložení záznamu	1	N	$\frac{N}{K}$ ideálně
smazání záznamu dle rid	2	?	?
smazání záznamu na shodu	N či $2N$	$\log_2 N + M$ či $\frac{3 \cdot N}{2}$	$\frac{N}{K} + 1$ ideálně
smazání záznamu na rozsah	$2N$?	?

15. **K jakému typu indexů se používá B^+ strom? Pro jaké typy dotazu je tento index vhodný?**

B^+ strom se používá pro jednoatributové indexy (?) a je vhodný pro intervalové dotazy.

16. **Jak se liší víceatributové indexování pomocí samostatných indexů (jeden klíč \rightarrow jeden index) a zřetězených klíčů (všechny klíče \rightarrow jeden index)? U obou popište vyhodnocení rozsahového intervalového, tj. „kvádrového“ dotazu.**

17. **Proč se pro atributy s malým oborem hodnot používá bitmapový index častěji než B-stromový?**

Operace s ním jsou rychlejší (bitové operace jsou rychlejší) a úspornější (jsou efektivně komprimovat). Dotazy nad mapami lze jednoduše paralelizovat.

18. **V jaké situaci se může vyplatit použít sekvenční průchod oproti použití indexu?**

19. **Index jakého typu a pro jaký(é) atributy by bylo nejvhodnější použít pro dotaz „SELECT * FROM učitelé WHERE věk < 50 AND věk > 40“. Odůvodněte.**

B^+ strom, neboť je rychlý pro vyhledávání na rozsah.

20. **Index jakého typu a pro jaký(é) atributy by bylo nejvhodnější použít pro dotaz „SELECT * FROM občan WHERE barvaOčí = 'modrá' OR barvaOčí = 'zelená““. Odůvodněte.**

Bitovou mapu, neboť barvaOčí má malou doménu (jen několik, např. 5 barev).

21. **Proč používáme databázové indexy? Ukažte dvě typické situace, kdy je využíváme.**

Indexování je pomocná datová struktura (ve zvláštním souboru), která urychluje vyhledávání v databázi dle jednoho či více klíčů, ke kterým obsahuje řid *record id*. 2 typické situace???

22. **Kdy je výhodné použít bitmapový index? Viz 10.**

23. **K čemu slouží vícehodnotové indexy?**

K rychlejšímu vyhledávání prostorových dotazů.

24. **Jaké jsou rozdíly použití (výhody/nevýhody) mezi bitmapovým indexem a B-stromem?**

Bitmapové indexy jsou rychlejší a prostorově výhodnější pro atributy s malou doménou. Pro ostatní jsou výhodnější B^+ stromy, speciálně pro vyhledávání na rozsah.

25. **Kdy je výhodné použít hašovaný index a kdy B^+ strom?**

Pokud se nám podaří zvolit vhodnou hašovací funkci, že bude malý počet kolizí, tak v hašovaném indexu je rychlejší vyhledávání na rovnost, v B^+ stromu naopak na rozsah.

26. **Jaké jsou rozdíly mezi haldou a setříděným souborem? (sorted file)**

Do haldy se rychleji vkládá, ale pomalu vyhledává na shodu i rozsah a při mazání vznikají díry. V setříděném souboru se dobře vyhledává, ale pomalu vkládá a maže.

27. **Nakreslete příklad B^+ stromu indexujícího 11 záznamů s kapacitou uzlů 3 klíče.**

28. Jaké jsou výhody a nevýhody hašovaného indexu?
29. Jak se u víceatributového indexování používá index zřetězených klíčů?
30. Co ovlivňuje rychlost přečtení/zápisu dat z/na disk(u) a jaká strategie vede k nejrychlejšímu čtení/zápisu?
31. Jaké jsou náklady základních operací na setříděném datovém souboru (vyhledání, vložení, vymazání záznamu, rozsahový dotaz)?
32. K čemu slouží bitové mapy?
33. Jak funguje buffer?
34. K čemu slouží víceatributové indexy?
35. Jaký je rozdíl mezi haldou a setříděným souborem?
36. Jaký je rozdíl mezi B -stromem a B^+ stromem?
37. Jak se udržují prázdné stránky haldy?
38. K čemu slouží databázové indexy?
39. Jaké jsou rozdíly použití (výhody/nevýhody) mezi hašovaným indexem a B -stromem?
40. Proč se k databázi přistupuje po diskových stránkách a ne např. po bajtech?
41. Co je to shlukovaný index a jaké má výhody/nevýhody oproti neshlukovanému?

Transakce a protokoly – zadání

- Pro výše uvedené transakce sestavte rozvrh podle striktního 2PL protokolu (dvoufázový) tak, aby obsahoval deadlock. Dokažte existenci deadlocku grafem pro detekci uváznutí.
- Na DB modelu z prvního příkladu definujte rozvrh tří transakcí (podle striktního 2PL protokolu), na kterých simulujte uváznutí (přes všechny 3 transakce). Dokažte grafem pro detekci deadlocků.
- Definujte 2 transakce na tabulce Zbozi(VyrobekId, TypVyrobkuID, Cena) a sestavte rozvrh podle 2PL protokolu (dvoufázový) tak, aby obsahoval fantoma (a slovně popište, jak se projevuje). K rozvrhu sestrojte graf pro detekci uváznutí (deadlock).
- Definujte 3 transakce na tabulce Poslanec a sestavte rozvrh podle 2PL protokolu (dvoufázový) tak, aby obsahoval fantoma. K rozvrhu sestrojte graf pro detekci uváznutí (deadlock).
- Definujte 3 (smysluplné) transakce na pomyslné tabulce NávštěvaHosta (ve smyslu prvního příkladu a sestavte rozvrh podle 2PL protokolu (dvoufázového) tak, aby obsahoval fantoma. K rozvrhu sestrojte graf pro detekci uváznutí (deadlock).
- Definujte 3 transakce na tabulce Obsluhuje(ZamID, KlientID) a sestavte rozvrh podle 2PL protokolu (dvoufázový) tak, aby obsahoval fantoma (a slovně popište, jak se projevuje). K rozvrhu sestrojte graf pro detekci uváznutí (deadlock).
- Modifikujte uvedené 4 transakce tak aby vznikl maximálně paralelní rozvrh podle 2PL protokolu (nestriktního dvoufázového). Se kterým sériovým rozvrhem je výsledný rozvrh ekvivalentní?
- Upravte uvedený rozvrh tak, aby se dosáhlo maximálního paralelismu podle SQL urovně izolace REPEATABLE READ (S2PL). K rozvrhu sestrojte graf pro detekci uváznutí (deadlock).
- Definujte 3 transakce na tabulce Zaměstnanec(ZID, Jméno) a sestavte rozvrh podle konzervativního 2PL protokolu (dvoufázový) tak, aby obsahoval fantoma. Může nastat uváznutí? Zkonstruujte graf pro detekci uváznutí.