

Fyzická implementácia DB

- 1) Súbory
- 2) Page, Index
- 3) Tabuľky
- 4) B-strom
- 5) Návrh indexov (či indexovať a či klastrový index)
- (6) Príklady: ZS)

1) Súbory

SQL Server mapuje DB na systémové súbory.

SQL Server rozlišuje 3 typy súborov:

- **primárny dátový súbor** - *.**mdf**
 - ukazuje na ďalšie súbory + master DB
 - každá DB ma iba jeden
- **sekundárny dátový súbor** - *.**ndf**
- **Log súbor** - *.**ldf**
 - slúži na obnovenie DB po transakcii

3) Page/stránka

Tabuľky a indexy sú uložené ako kolekcia 8-KB stránok.

Stránka

- je základná jednotka ukladania dát (tabuliek a indexov)
- operácie I/O sú vykonané pomocou stránok
- 1 page = 8kb
- stránka sa začína hlavičkou za ktorou nasledujú seriové riadky
- 8 typov strán – **dáta, indexy**, image/text, info o voľných miest + 4 ďalšie

Index v knihách je zoznam dvojíc, dôležitých pojmov a čísiel strán, v abecednom poradí umiestnený na konci knihy. Index zrýchli vyhľadávanie pojmov. Indexy v DB tiež slúžia na rýchle vyhľadávanie záznamov, riadkov tabuľky.

Indexový súbor/tabuľka je v podstate **zoznam s dvomi poliami**: prvé pole je **index** ⇔ **klúč** a druhé pole môže byť **dvojaké**: buď obsahuje samotný záznam alebo smerník na blok dátového súboru, kde sú záznamy s príslušnou hodnotou indexového poľa. Indexový súbor je oveľa **menší** ako dátový súbor.

Rozlišujeme klastrový a neklastrový index, pozri nižšie.

klastrový index ⇔ **zotriedený index** aj fyzicky

3) Tabuľky

Dátové stránky tabuľky môžu byť organizované ako
- klastrové tabuľky s klastrovým indexom (B-stromy)
alebo

- HEAP tabuľky (halda) bez klastrového indexu

Kým klastrové tabuľky sú zoradené, halda je bez zoradenia.

Klastrový a neklastrový index a ich štruktúra

- Ak tabuľka obsahuje *klastrový index*, potom dáta (presnejšie *údajové stránky*) sú *v listových uzloch*. Tabuľka môže mať iba jeden kl.index.

Pri vytvorení klastrového indexu:

- vytvorí sa index (indexové stránky vo vnútorných uzloch
– jeden indexový riadok je dvojica: [kľúč, pointer na dáta])
- tabuľka sa *zotriedi* podľa kľúča

(veľa IO operácií a intenzívne využitie systémových a pamäťových zdrojov)

- *Neklastrový index* je tiež organizovaný ako B-strom, ale

- *v listových uzloch* sú *indexové stránky* namiesto údajových stránok.

Tabuľka môže mať až 249 neklastrových indexov. Indexové stránky obsahujú indexový riadok, ktorý sa skladá z kľúča a z **lokátora** riadku.

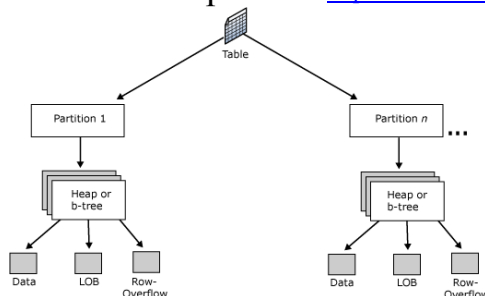
Neklastrový index môže byť definovaný na

- tabuľke alebo view s klastrovým indexom, potom **lokátor** je kľúč klastrového indexu na dátový riadok
- halde, potom **lokátor** je ukazovateľ (pointer) na dátový riadok.

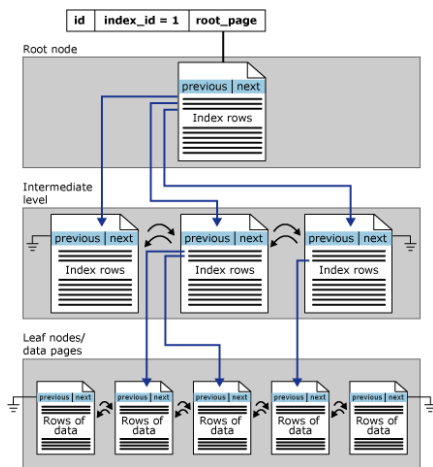
Na rozdiel od klastrových indexov, neklastrové indexy nemajú vplyv na poradie dátových riadkov.

Partícia je horizontálne rozdelenie tabuľky alebo indexu (ich riadkov) do menších jednotiek, definovaných užívateľom! (MS SQL Server 2005 Enterprise, Developer editions!). Vďaka partícii veľké tabuľky a indexy sa chovajú optimálnejšie.

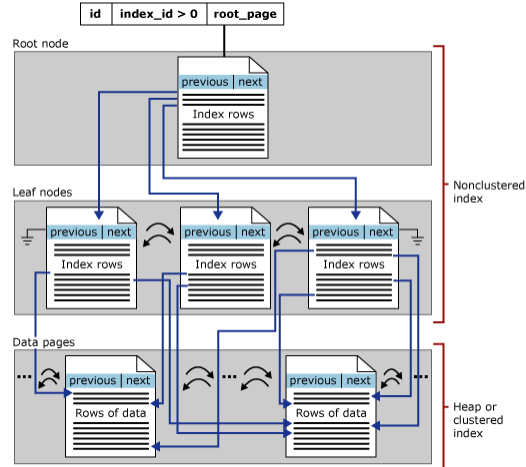
- tabuľka => partície [http://msdn.microsoft.com/en-us/library/ms188706\(SQL.90\).aspx](http://msdn.microsoft.com/en-us/library/ms188706(SQL.90).aspx)



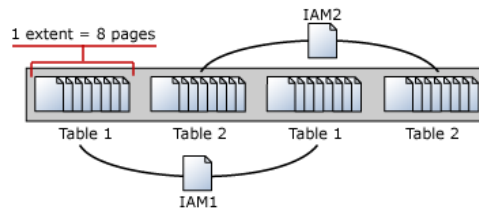
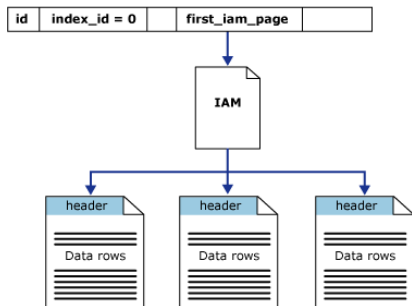
- partícia: riadky => pages – **klastrová tabuľka s klastrovým indexom**
- **heap (halda) tabuľka bez kl. indexu**



Štruktúra klastrového indexu v jednej partícii.
[http://msdn.microsoft.com/en-US/library/ms177443\(v=sql.90\).aspx](http://msdn.microsoft.com/en-US/library/ms177443(v=sql.90).aspx)



Štruktúra neklastrového indexu v jednej partícii.
[http://msdn.microsoft.com/en-us/library/ms177484\(v=sql.90\).aspx](http://msdn.microsoft.com/en-us/library/ms177484(v=sql.90).aspx)



SQL Server používa IAM (Index Allocation Map) stránky na získanie dátových riadkov v jednej **heap** partícii. [http://msdn.microsoft.com/en-us/library/ms188270\(v=sql.90\).aspx](http://msdn.microsoft.com/en-us/library/ms188270(v=sql.90).aspx) [http://msdn.microsoft.com/en-us/library/ms187501\(SQL.90\).aspx](http://msdn.microsoft.com/en-us/library/ms187501(SQL.90).aspx)

4) B-stromy

- **Strom**
- **B-strom**
- **Výška B-stromu**
- **Operácie na B-stromoch**

Terminológia **stromu**

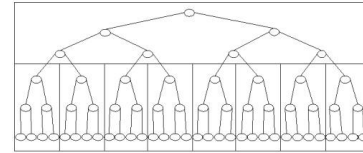
Strom je tvorený **uzlami**/vrcholmi a hranami. Rozlišujeme **3 typy** uzlov:

- koreňový
- vnútorný, nelistový
- vonkajší, list(ový)

Každý uzol, okrem koreňa má jeden **rodičovský** uzol a niekoľko – nula a viac – **potomkov**.

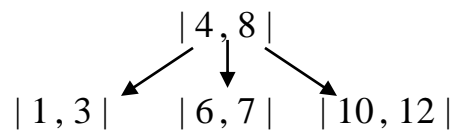
Hĺbka, **výška** uzla je vždy o jeden viac ako úroveň rodičovského uzla, pričom úroveň koreňa je nula. **Podstrom** uzla sa skladá z uzla a všetkých jeho potomkov.

Binárny strom rozdelený na **stránky** (po 7 uzlov):

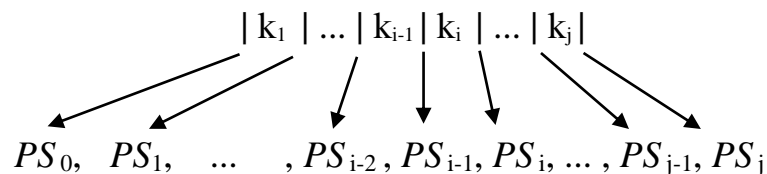


B-strom

B-strom je stromová dátová štruktúra, vhodná na ukladanie a čítanie veľkých celkov dát a zabezpečuje vyhľadanie, vkladanie a vymazanie za logaritmickej čas. B-strom zaviedol Bayer (1972) a McCreight.



Každý interný uzol v **B-strome** má tvar



alebo

$$\langle p_1, k_1, \dots, p_{i-1}, k_{i-1}, p_i, \dots, p_l, k_l, p_{l+1} \rangle$$

kde p_i je smerník (šípka) na iný uzol v strome / podstrom / PS_i , k_i je hodnota kľúča, l je počet kľúčov / prvkov k_i v uzle, $i=1,2,\dots,l$ a $l=1,2,\dots,m-1$, kde m je **rád** stromu.

Definícia B-stromu ():

B-strom **rádu** m je strom s vlastnosťami:

- 1) Koreň buď má aspoň **dvoch potomkov** alebo je **list**.
- 2) Každý vnútorný uzol má **minimálne** $\lceil m/2 \rceil$ a **maximálne** m **potomkov**.
- 3) Vnútorný uzol s $j+1$ potomkami obsahuje j kľúčov/prvkov.
- 4) Kľúče v uzle sú usporiadané

$$k_1 < \dots < k_{i-1} < k_i < \dots < k_j.$$

- 5) Pre kľúče *podstromo* v_{j-1} a v_j platí:

$$k_{j-1} < k < k_j.$$

- 6) Každý list je v tej istej hĺbke od koraňa.

B-stromy rádu 4 sú známe ako 2-3-4 stromy.

Výška B-stromu

Počet operácií v B-strome je proporcionálny k hĺbke stromu, ktorá rastie pomaly.

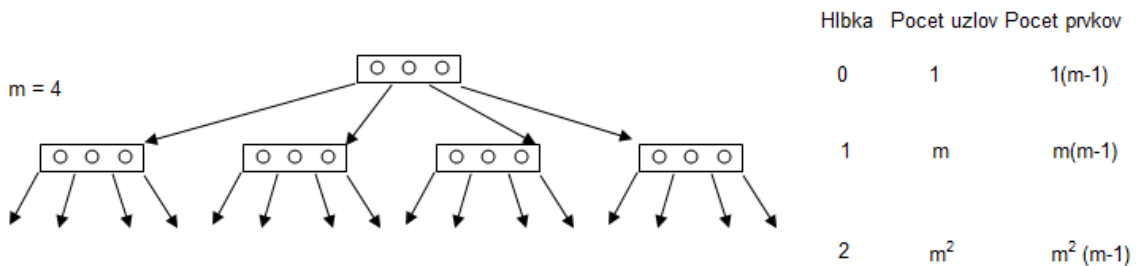
Veta

Úplny B-strom rádu $m \geq 2$ s $n \geq 1$ prvkami má hĺbku

$$h = \log_m(n+1) - 1.$$

Dôkaz:

Vieme, že m zadá maximálny počet potomkov.



Teda pre počet **prvkov** n az do hĺbky h platí:

$$n = (m-1) \sum_{i=0}^h m^i = (m-1)(m^{h+1}-1)/(m-1) = m^{h+1}-1$$

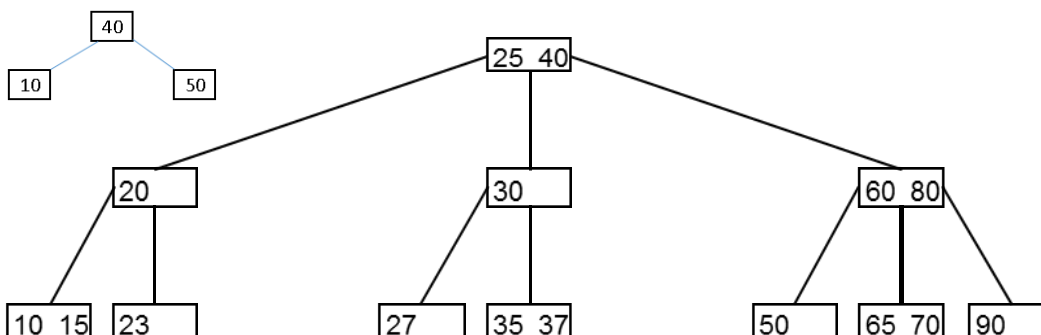
z čoho plynie: $h = \log_m(n+1) - 1$.

Základné operácie na B-stromoch

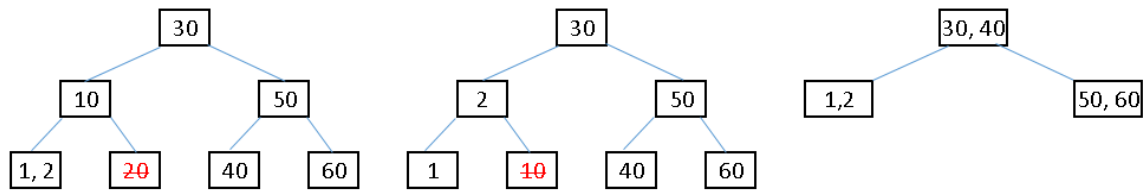
- Vyhľadávanie
- Vkladanie
- Vymazávanie

Vkladanie B₃:

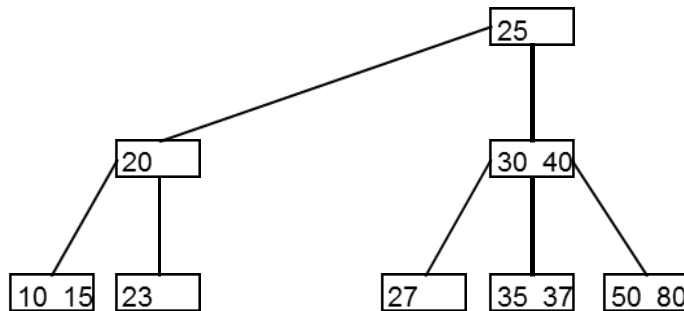
10, 40, 50, 20, 60, 30, 70, 35, 25, 15, 27,37, 23, 80, 90, 65



Vymazávanie B₃:



60, 70, 90, 65



5) Návrh indexov

Význam indexu

- výhody
- nevýhody (pozri príklady)

Návrh indexu

Výber indexu a jeho typu treba dôkladne premyslieť. Existujú všeobecné odporúčania, kedy (ne)použiť index a aký typ. SQL Server má index-tuning system: Database Tuning Advisor (DTA).

Dobrý index

- ak tabuľka je viac dopytovaná ako modifikovaná
- stĺpce, ktoré nie sú často modifikované
- stĺpec/ce s cudzím kľúčom
- stĺpec, ktorý má veľa odlišných hodnôt
- stĺpce vo WHERE klauzule
- dopyty s order by/ join/ agregáčné

Zlý index

- ak tabuľka je viac modifikovaná ako dopytovaná
- tabuľka s malým počtom riadkov
- stĺpec, ktorý má málo odlišných hodnôt
- stĺpec, ktorý sa v dopyte nepoužíva
- stĺpce s dátovými typmi: **text, ntext, image, a bit**

Treba mať na pamäti, že ak index sa skladá z viacerých stĺpcov, potom pri vkladaní nových riadkov alebo modifikovaní daných stĺpcov prebieha viac dodatočných úkonov, súvisiacich s údržbou indexu.

6) Príklady

Definícia:

0_index_Create.sql

Nevýhoda:

1_CrInd_Insert__Insert_CrInd.sql

Uloženie:

2_ClIndex_Ukladanie.sql

Výhoda – hľadanie:

3a_RAND_SEED_KERNEL.sql

3b_Search_RandTab.sql

3c_Search.sql

[https://msdn.microsoft.com/en-us/library/ms188783\(v=sql.120\).aspx](https://msdn.microsoft.com/en-us/library/ms188783(v=sql.120).aspx)