

DBS - Fyzický pohled na data

Michal Valenta

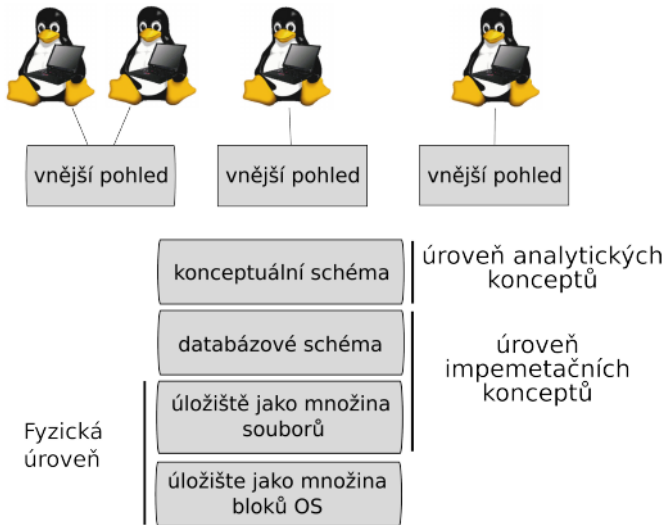
Katedra softwarového inženýrství FIT
České vysoké učení technické v Praze
©Michal Valenta, 2010

BI-DBS, ZS 2010/11

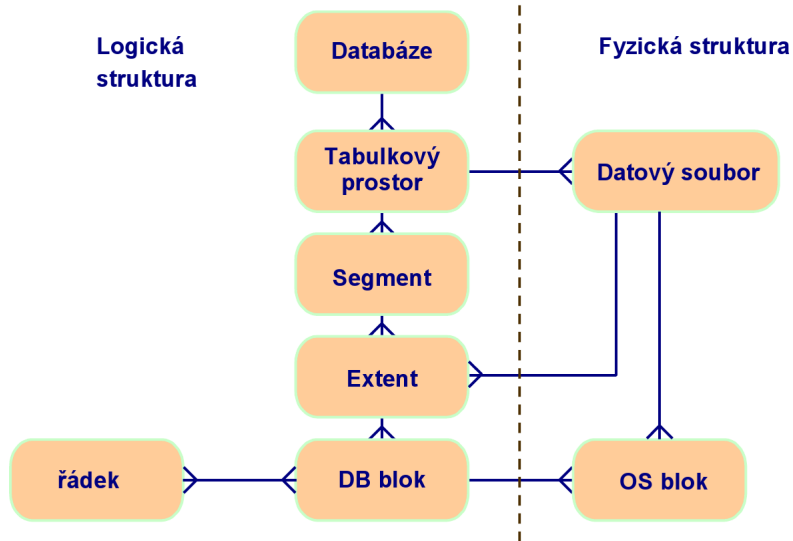
<https://edux.fit.cvut.cz/courses/BI-DBS/>



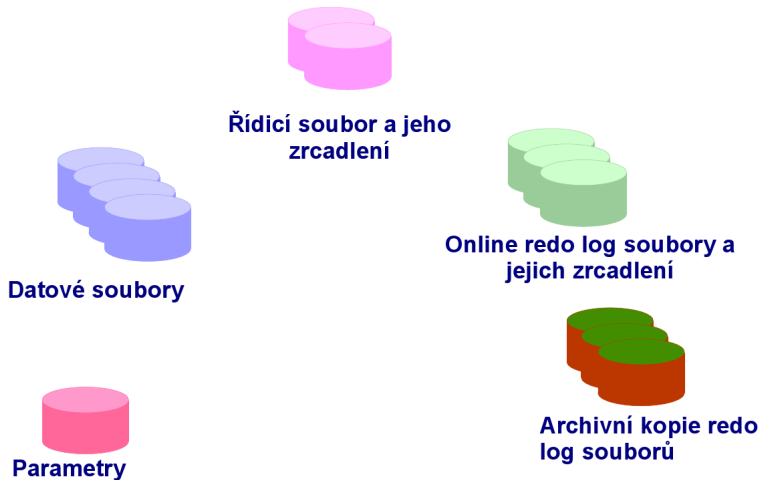
Různé úrovně pohledu na data



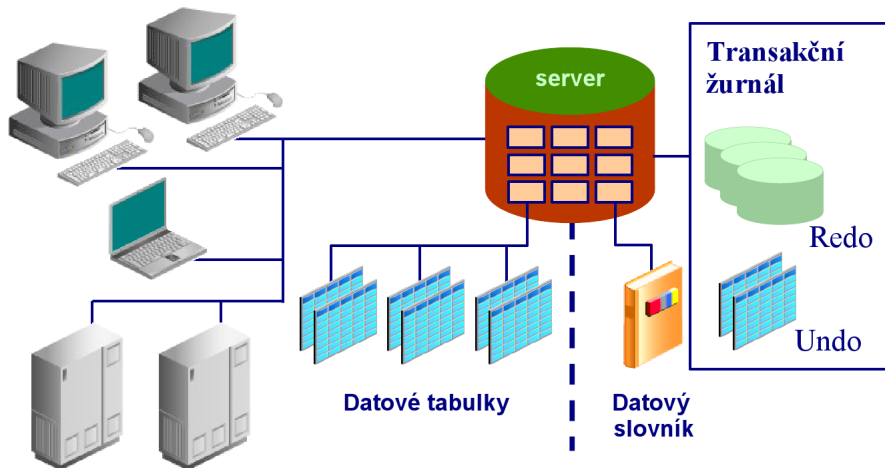
Struktura databáze



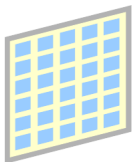
Transakční zpracování



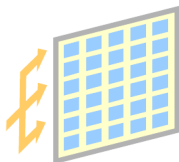
Klient - (aplikační server) - Server



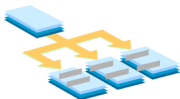
Fyzická organizace relační tabulky



**HEAP
tabulka**



**HEAP
tabulka s indexy**



**Tabulka s indexovou
organizací**



**tabulka ve shluku
(cluster)**

Adresa řádku

```
select rowid, t.* from Titul t
```

| ROWID | TITUL_ID | NAZEV | ROK_VYROBY |
|--------------------|----------|-----------------|------------|
| ----- | ----- | ----- | ----- |
| AAAsaTAAHAAA656AAA | 1 | Název_titulu _1 | 01.01.2005 |
| AAAsaTAAHAAA656AAB | 2 | Název_titulu _2 | 01.01.2005 |
| AAAsaTAAHAAA656AAC | 3 | Název_titulu _3 | 01.01.2005 |
| ... | | | |

Asociativní výběr – „prohrabání hromady“

```
select * from Titul t where NAZEV='Název_titulu _2'
```

Adresní výběr (nadrelační rys):

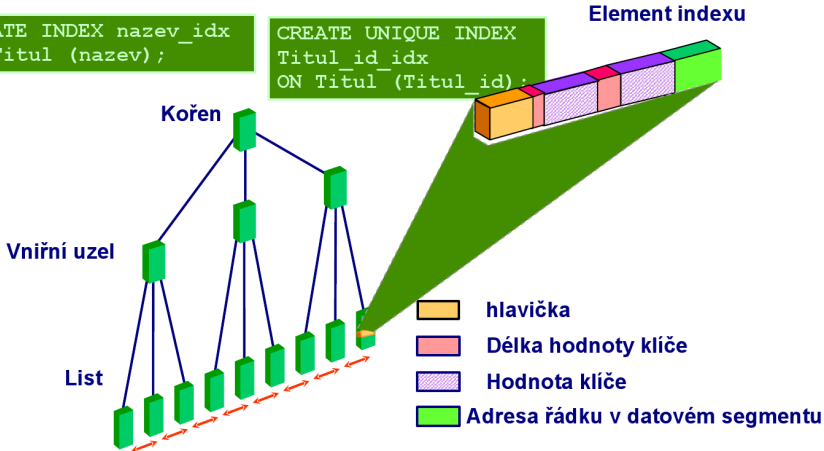
```
select * from Titul t where ROWID ='AAAsaTAAHAAA656AAB'
```

Index typu B*-Tree

```
CREATE INDEX nazev_idx on Titul (nazev);  
CREATE UNIQUE INDEX titul_id_idx on Titul (titul_id);
```

```
CREATE INDEX nazev_idx  
ON Titul (nazev);
```

```
CREATE UNIQUE INDEX  
Titul_id_idx  
ON Titul (Titul_id);
```



Index typu B-Tree - charakteristika

B - strom (řádu m) je m -ární strom, splňující následující omezení

- Kořen má nejméně 2 potomky, pokud není listem
- každý uzel kromě kořene a listu má nejméně $\lceil m/2 \rceil$ a nejvýše m potomků
- každý uzel má nejméně $\lceil m/2 \rceil - 1$ a nejvíce $m - 1$ datových záznamů (většinou jen klíčů)
- všechny cesty ve stromě jsou stejně dlouhé
- data v nelistovém uzlu jsou organizována $p_0(k_1, p_1), (k_2, p_2), \dots, (k_n, p_n), u$
 - ▶ kde p_0, p_1, \dots, p_n jsou ukazatele na potomky
 - ▶ k_0, k_1, \dots, k_n jsou klíče
 - ▶ u je nevyužitý prostor
 - ▶ záznamy (k_i, p_i) jsou uspořádány vzestupně podle klíčů
 - ▶ $\lceil m/2 \rceil - 1 \leq n \leq m - 1$
- odpovídá-li ukazateli p_i , kde $i \in \langle 1, n \rangle$ podstrom $U(p_i)$ platí:
 - ▶ (i) pro každé $k \in U(p_{i-1})$ je $k \leq k_i$
 - ▶ (ii) pro každé $k \in U(p_i)$ je $k > k_i$
- listy obsahují úplnou množinu klíčů a mohou se lišit strukturou.

Bitmapové indexy

```
CREATE BITMAP INDEX rok_id_bix  
ON Titul (rok_vyroby);
```

| | Rok_vyroby | | | | |
|----------|------------|------|------|------|-----|
| | 2001 | 2002 | 2003 | 2004 | ... |
| Titul_id | | | | | |
| 'titul1' | 1 | 0 | 0 | 0 | |
| 'titul2' | 0 | 1 | 0 | 0 | |
| 'titul3' | 0 | 0 | 1 | 0 | |
| 'titul4' | 0 | 0 | 0 | 1 | |
| 'titul5' | 0 | 1 | 0 | 0 | |
| 'titul6' | 0 | 0 | 1 | 0 | |
| . | . | . | . | . | |
| . | . | . | . | . | |

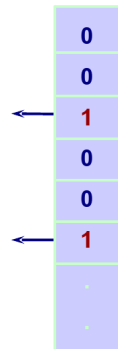
Použití bitmapového indexu při vyhodnocení dotazu

```
SELECT *  
FROM Titul  
WHERE Rok_vyroby = 2003;
```

Konverze na množinu adres řádků

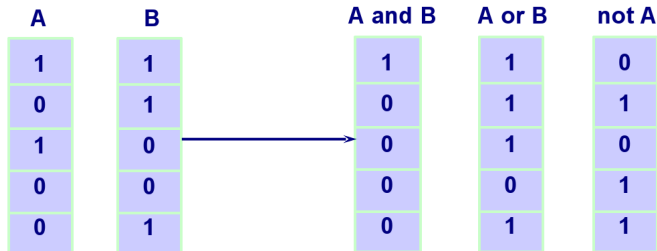


Rok_vyroby = 2003



Použití bitmapového indexu při vyhodnocení dotazu

- Výběrová podmínka s operátorem IN
- Výberová podmínka s operátorem AND/OR

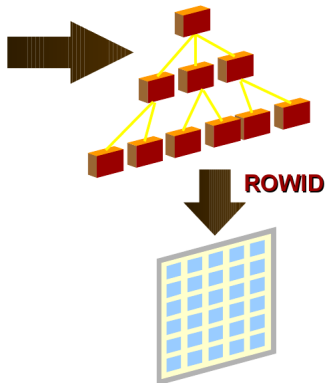


Porovnání indexů B-strom a Bitmap

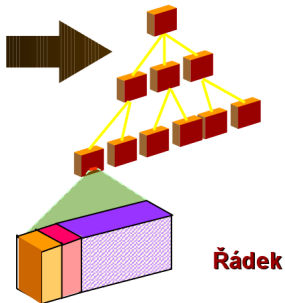
| B-strom | Bitová mapa |
|---------------------------------------|---|
| Sloupce s vysokou kardinalitou | Sloupce s nízkou kardinalitou |
| DML operace relativně drahé | DML operace velmi drahé |
| Vhodné pro OLTP | Vhodné pro ad hoc dotazy v datových skladech DSS |

Indexově organizovaná tabulka

Heap tabulka s indexem



Indexově organizovaná tabulka



Shluk (Cluster)

| TITUL_ID | KOPIE_ID | DATUM |
|----------|----------|-------|
| 148590 | 00001 | YYYY |
| 148969 | 00001 | YYYY |
| 148969 | 00002 | YYYY |
| 148590 | 00002 | YYYY |
| 155791 | 00001 | YYYY |
| 148590 | 00003 | YYYY |

| TITUL_ID | NAZEV | ROK_VYROBY |
|----------|-------|------------|
| 148590 | xxx1 | 1997 |
| 148969 | XXX2 | 2001 |
| 155791 | XXX3 | 2005 |

**Samostatné tabulky s vazbou
přes cizí klíč**

Cluster Key: TITUL_ID

| | | |
|--------|-------|------|
| 148590 | xxx1 | 1997 |
| | 00001 | YYYY |
| | 00002 | YYYY |
| | 00003 | YYYY |
| 148969 | xxx2 | 2001 |
| | 00001 | YYYY |
| | 00002 | YYYY |
| 155791 | xxx3 | 2005 |
| | 00001 | YYYY |

Tabulky ve shluku

Další techniky zrychlení přístupu k datům

Připomenutí pojmů:

- OLTP versus DSS systémy
- optimalizace dotazu, prováděcí plán, cena dotazu
- fáze zpracování dotazu (příkazu)

Používané techniky:

- zavedení redundance, materializované pohledy
- využití vyrovnávacích pamětí pro data, ale také pro parsované SQL příkazy a kurzory

Optimalizace aplikace a databáze:

- návrh struktury (normalizace vs. denormalizace)
- speciální struktury pro uložení dat
- optimalizace dotazů (zjištění dotazů, které je třeba optimalizovat)
- systémové zdroje (paměť - velikost a struktura)
- konfigurace serveru (disky, parametry databáze, zálohování)