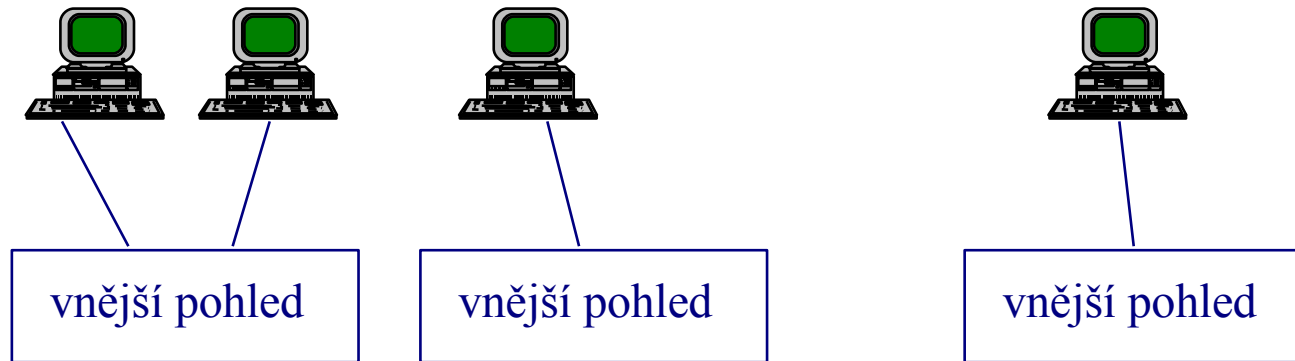


# *Fyzický pohled na data*

# Různé úrovně pohledu na data



Fyzická  
úroveň

konceptuální schéma

Databázové schéma

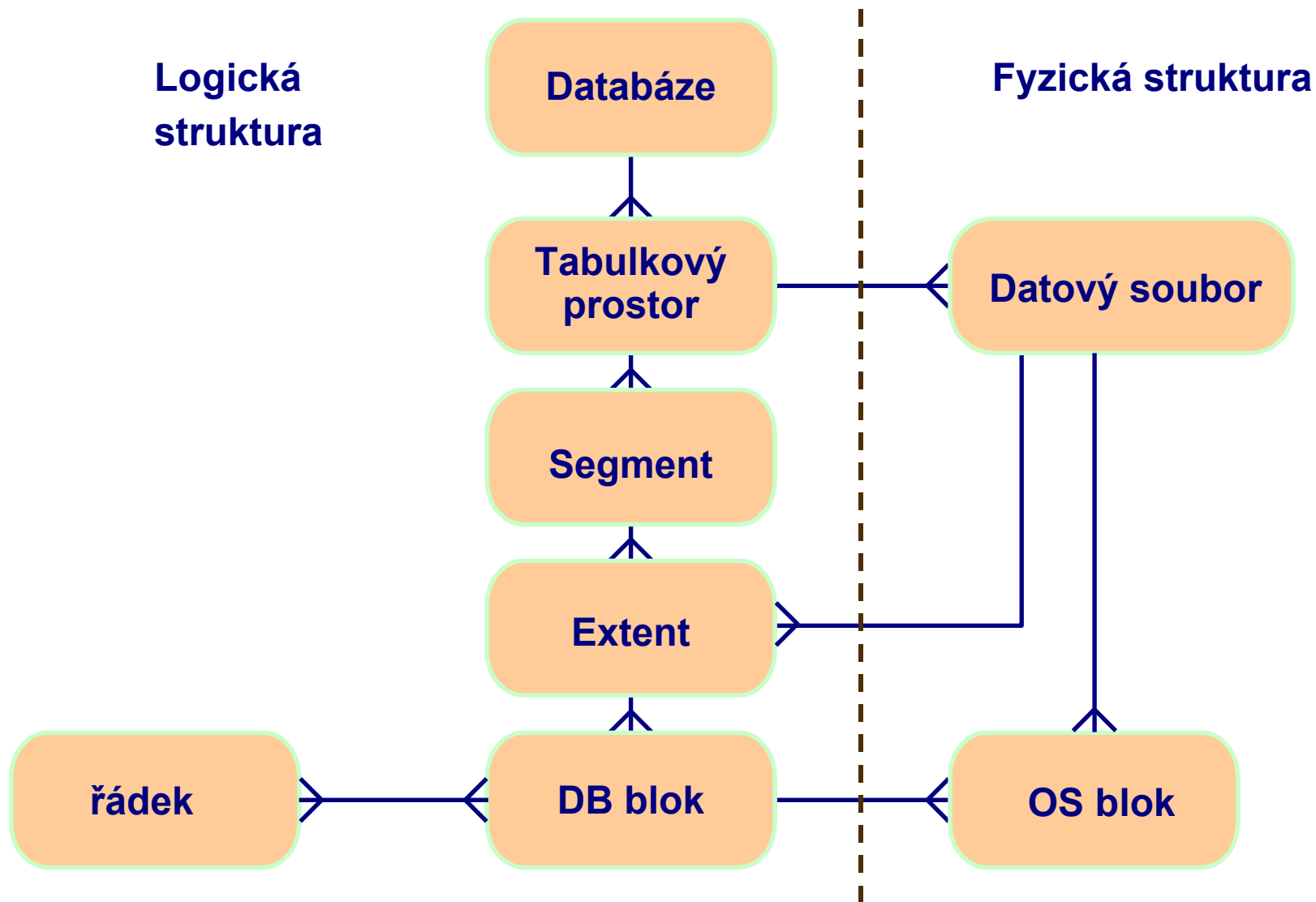
úložiště jako množina  
souborů

úložiště jako  
množina BOIS bloků

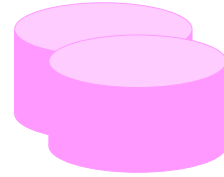
Úroveň analytických  
konceptů

Úroveň  
implementačních  
konceptů

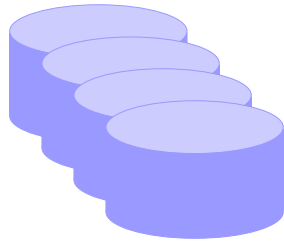
# Struktura databáze



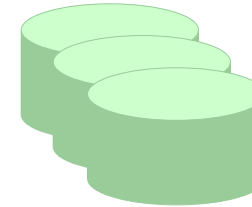
# *Databázové soubory*



**Řídicí soubor a jeho zrcadlení**



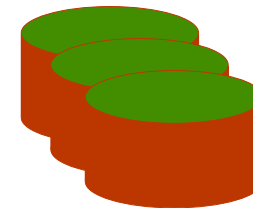
**Datové soubory**



**Online redo log soubory a jejich zrcadlení**

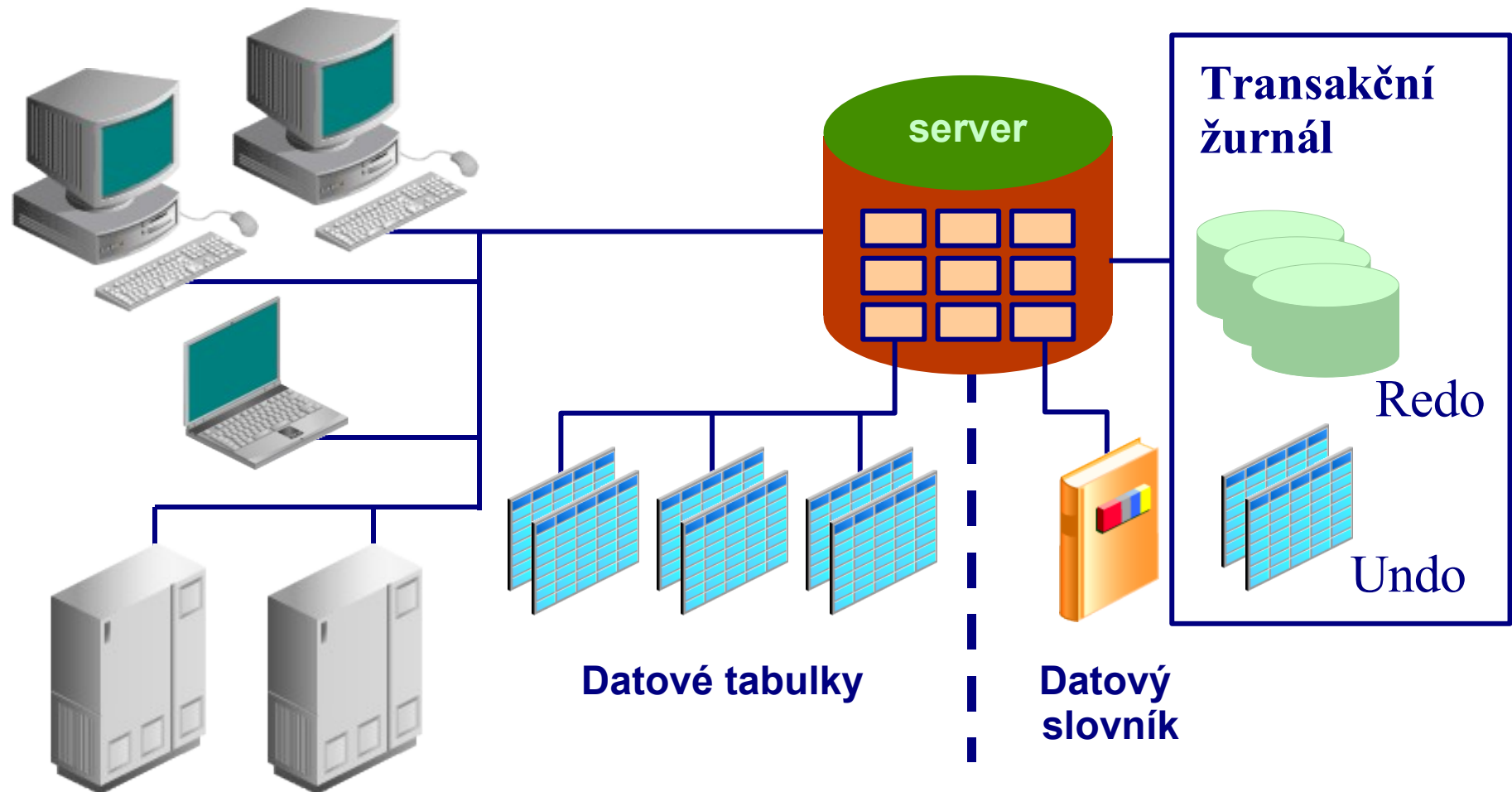


**Parametry**

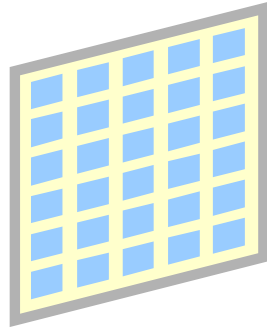


**Archivní kopie redo log souborů**

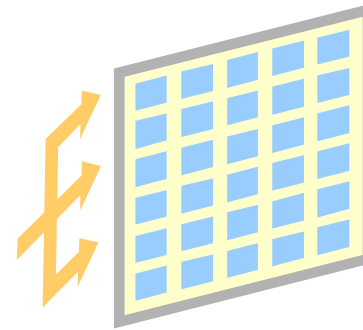
# *Klient – (aplikační server) - Server*



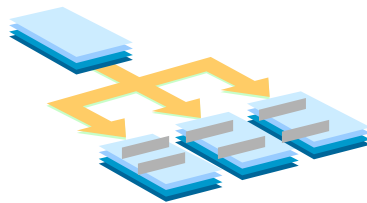
# Fyzická organizace relační tabulky



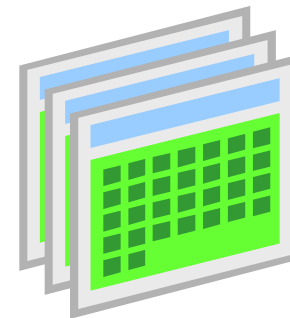
**HEAP  
tabulka**



**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):**

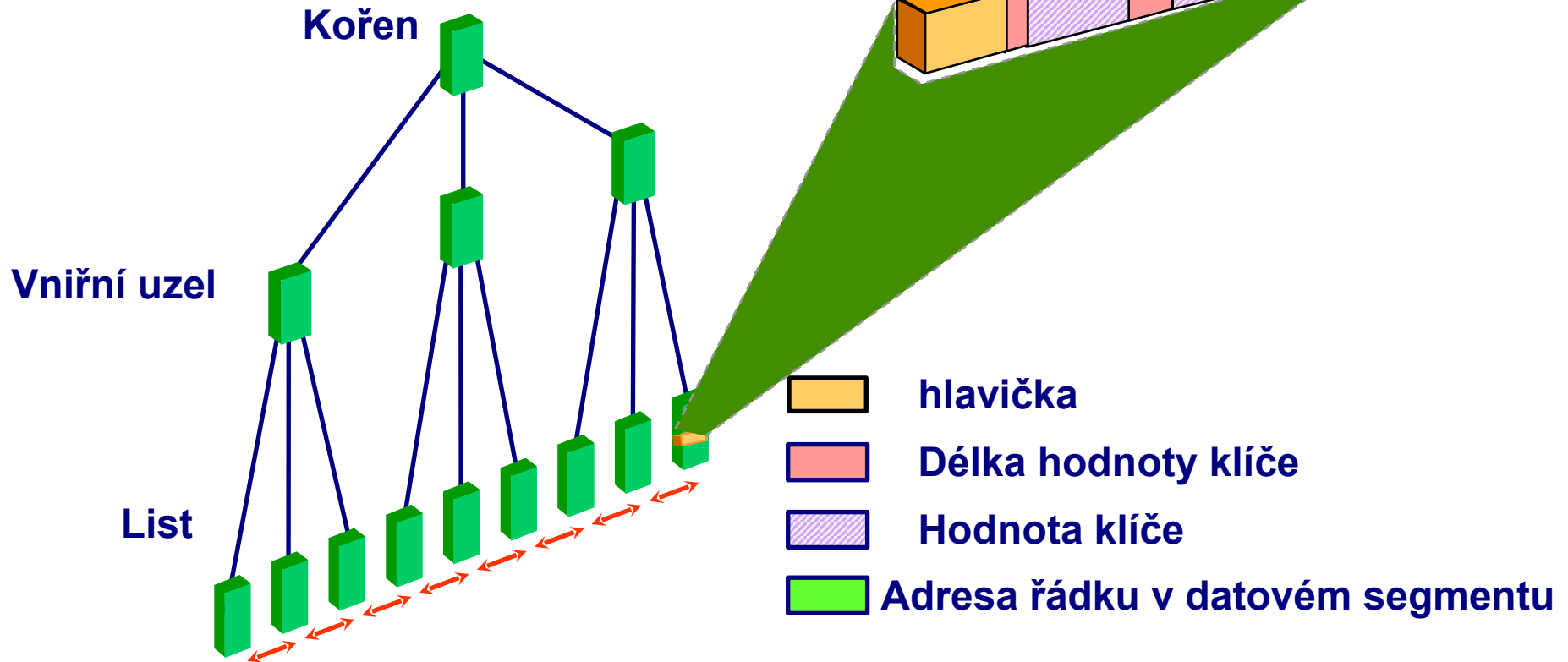
```
select * from Titul t where ROWID ='AAAsaTAAHAAA656AAB'
```

# Index typu B\*-Tree

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

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

Element indexu





# *B-strom*

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

- Kořen má nejméně dva 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 pouze klíčů),
- všechny cesty ve stromě jsou stejně dlouhé,
- data v nelistovém uzlu jsou organizována následovně:
  - $p_0, (k_1, p_1), (k_2, p_2), \dots, (k_n, p_n), u$
  - kde  $p_0, p_1, \dots, p_n$  jsou ukazatele na potomky,  $k_1, k_2, \dots, k_n$  jsou klíče,  $u$  je nevyužitý prostor a záznamy  $(k_i, p_i)$  jsou uspořádány vzestupně podle klíčů, přičemž  $\lceil m/2 \rceil - 1 \leq n \leq m-1$ .
- odpovídá-li ukazateli  $p_i$ , kde  $i \in \langle 1, n \rangle$ , podstrom  $U(p_i)$ , potom platí:
  - (i) pro každé  $k$  v  $U(p_{i-1})$  je  $k \leq k_i$ ,
  - (ii) pro každé  $k$  v  $U(p_i)$  je  $k > k_i$ ,
- listy obsahují úplnou množinu klíčů a mohou mít odlišnou strukturu.

# Bitmapové indexy

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

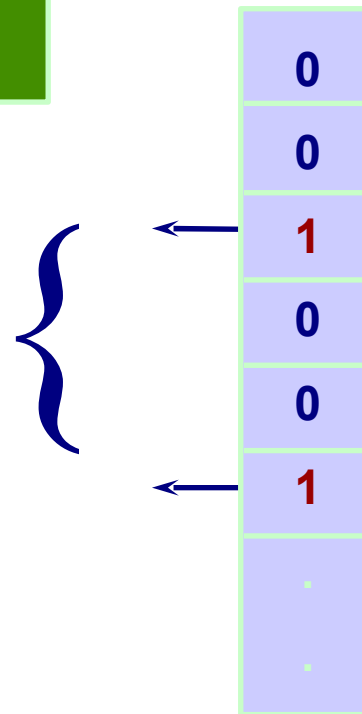
	Rok_vyroby				
	2001	2002	2003	2004	...
Titul_id	'titul1'	1	0	0	0
	'titul2'	0	1	0	0
	'titul3'	0	0	1	0
	'titul4'	0	0	0	1
	'titul5'	0	1	0	0
	'titul6'	0	0	1	0
	.	.	.	.	.
.	.	.	.	.	

# Použití bitmapového indexu při vyhodnocení dotazu

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

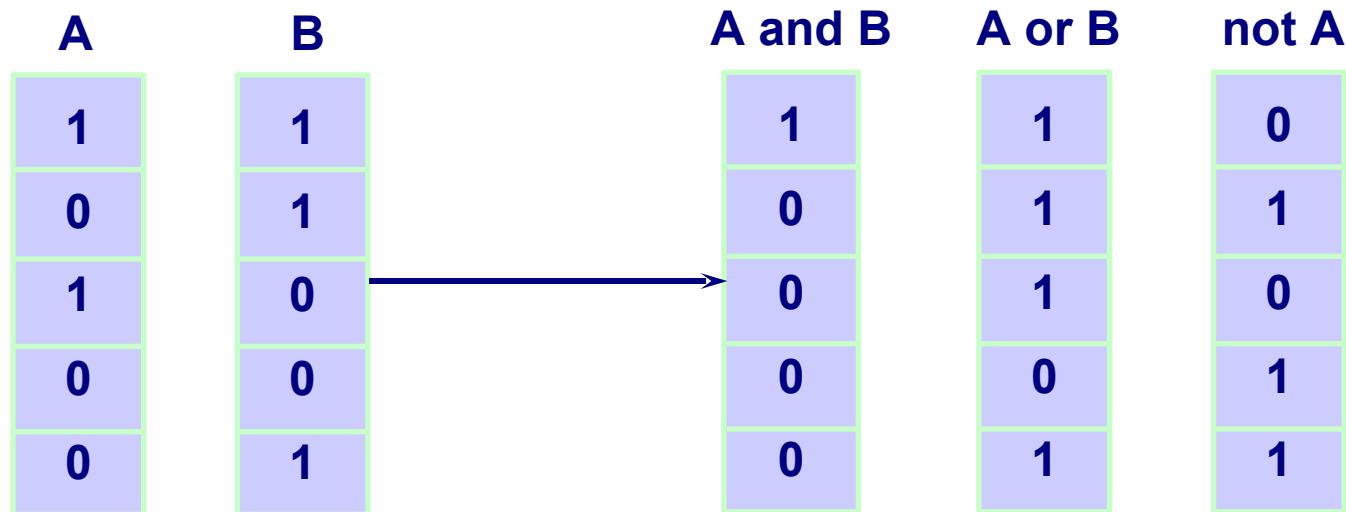
Konverze na množinu adres řádků

Rok\_vyroby = 2003



# Kombinování několika map

- Výběrová podmínka s operátorem **IN**
- Výběrová podmínka s operátory **AND/OR**

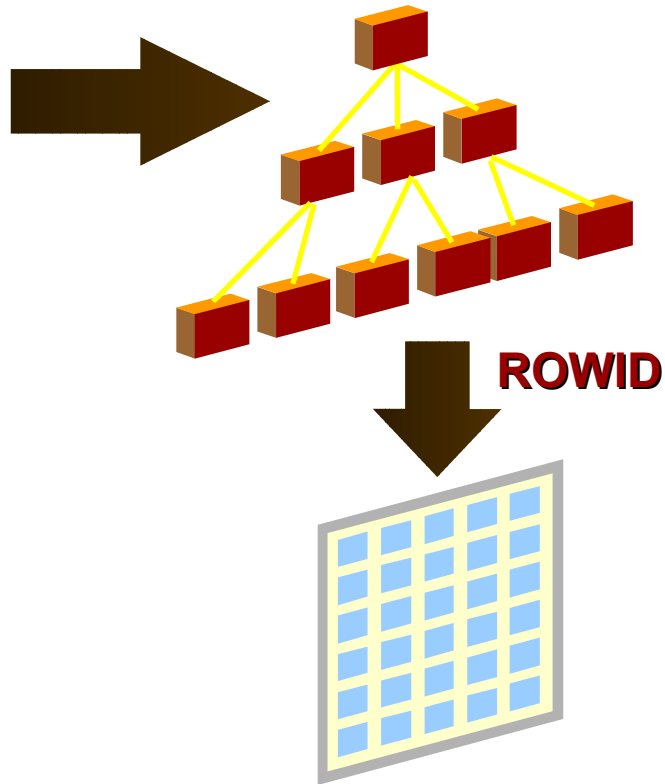


# Porovnání indexů B-strom a Bitmap

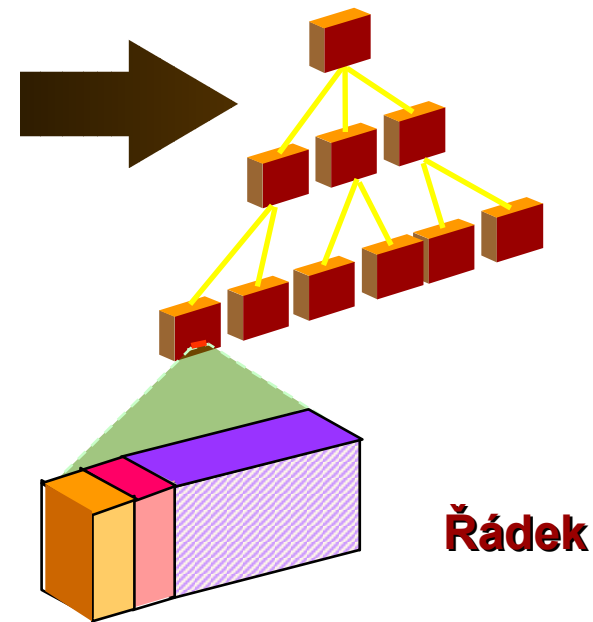
<b>B-strom</b>	<b>Bitová mapa</b>
<b>Sloupce s vysokou kardinalitou</b>	<b>Sloupce s nízkou kardinalitou</b>
<b>DML operace relativně drahé</b>	<b>DML operace velmi drahé</b>
<b>Vhodné pro OLTP</b>	<b>Vhodné pro ad hoc dotazy v datových skladech DSS</b>

# *Indexově organizovaná tabulka*

**Heap tabulka s idexem**



**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

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

**Samostatné tabulky s vazbou  
přes cizí klíč**

**Tabulky ve shluku**