



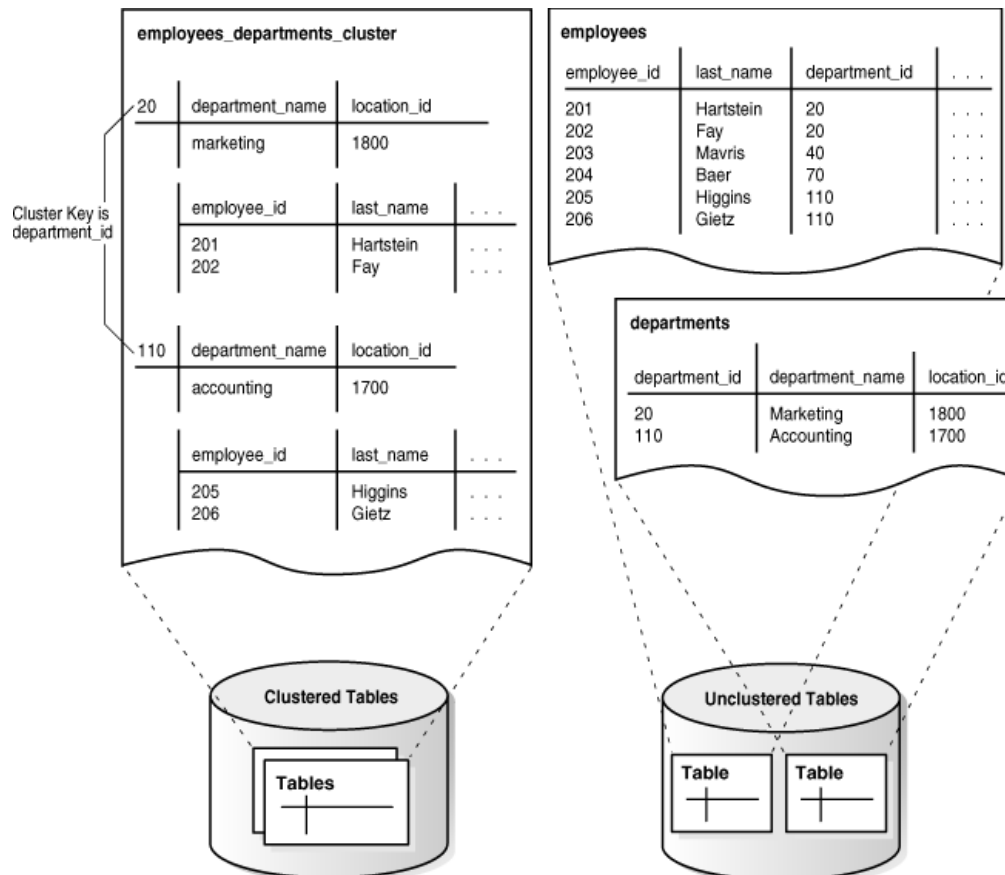
ELTE | IK
INFORMATIKAI KAR

Adatbázisok 2.

6. gyakorlat

Klaszterek (1)

- A klaszter táblák egy csoportja, amelyeknek közös oszlopaik vannak és fizikailag egy helyen kerülnek tárolásra



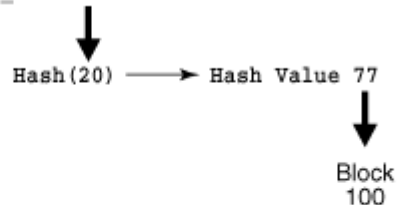
Klaszterek (2)

- Előnyök:
 - Kevesebb lemez I/O összekapcsolás esetén
 - Kevesebb tárhely szükséges, mivel a klaszter kulcs csak egyszer tárolódik
- Nem ajánlott, ha:
 - A tábla gyakran van módosítva, sok az új rekord
 - Sokszor van szükséges FULL TABLE SCAN-re
 - Ha a klaszter kulcshoz nagyon eltérő mennyiségű rekordok tartoznak

Klaszter típusok

- Indexelt klaszter
 - B-fa készül a klaszter kulcsra
 - Index nélkül nem tudunk értékeket sem beszúrni
- Hash klaszter
 - Hash kulcs alapján határozza meg hová kerüljenek a sorok
 - Egy tábla esetén is használható (SINGLE CLUSTER)
 - Akkor előnyös ha gyakran kérdezzük le egyenlőségvizsgálat alapján

```
SELECT * FROM employees  
WHERE department_id = 20
```

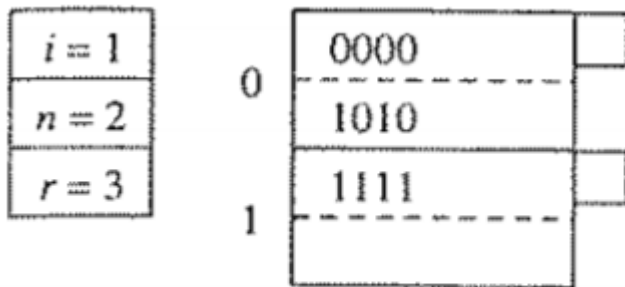


Data Blocks in Cluster Segment

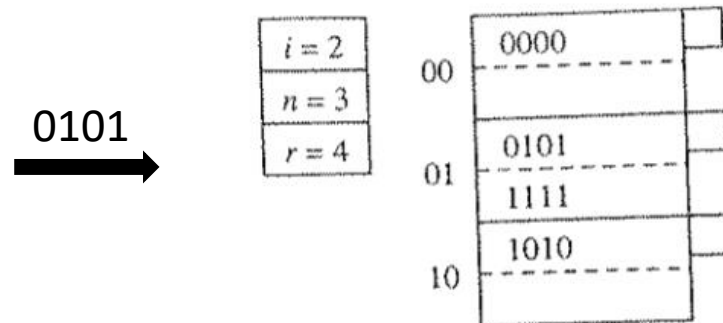


Lineáris hashelés (tördelőtáblázat)

- Fontos paraméterek:
 - i - kosárindexhez használt bitek száma
 - n - kosarak száma
 - r - rekordok száma
- Ha átlépünk egy határértéket (pl. $r/n > 1.7$), akkor megnöveljük a kosarak számát
- Ha nem lépjük át a határértéket, de nincs hely, akkor használhatunk túlcsordulás blokkot



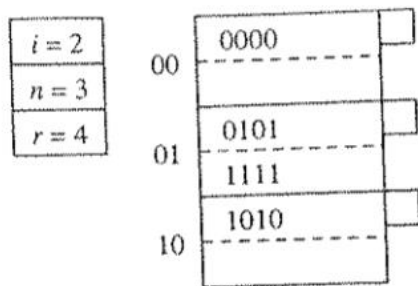
4.36. ábra. Lineáris tördelőtáblázat



4.37. ábra. Egy harmadik kosár hozzáadása

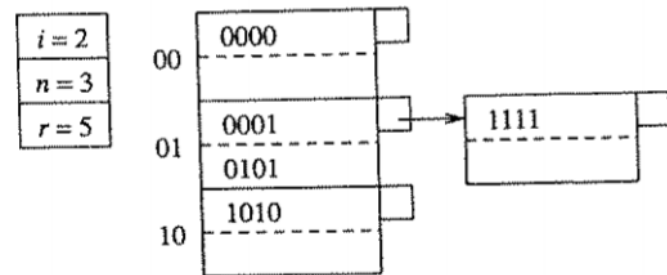
Lineáris hashelés (tördelőtáblázat)

- Az utolsó i bit alapján határozzuk meg egy rekord helyét
- Ha nincs olyan kosár, ahol a rekordnak lennie kellene, akkor abba rakjuk, amely csak az első bitben különbözik



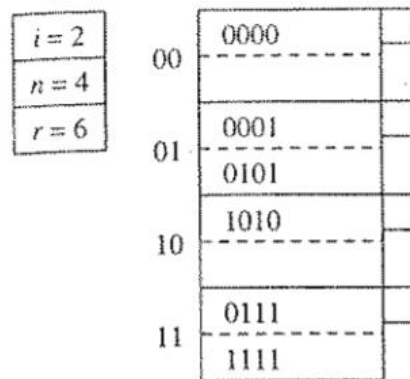
4.37. ábra. Egy harmadik kosár hozzáadása

0001
→



4.38. ábra. Szükség esetén túlsordulásblokkokat használunk

0111
→



4.39. ábra. Egy negyedik kosár hozzáadása

Lineáris hashelés feladat

- Tegyük fel, hogy egy blokkba 2 rekord fér el
- A határértékünk 2,4 (rekordok száma/kosarak száma)
- Szűrjük be az alábbi hasító értékkel rendelkező sorokat egymás után:
- 0001, 0110, 1011, 0111, 1100, 1111, 0100
- Kiinduló állapot:

i = 1
r = 3
n = 2

0	<table border="1"><tr><td>0000</td></tr><tr><td>1110</td></tr></table>	0000	1110
0000			
1110			
1	<table border="1"><tr><td>0101</td></tr><tr><td> </td></tr></table>	0101	
0101			