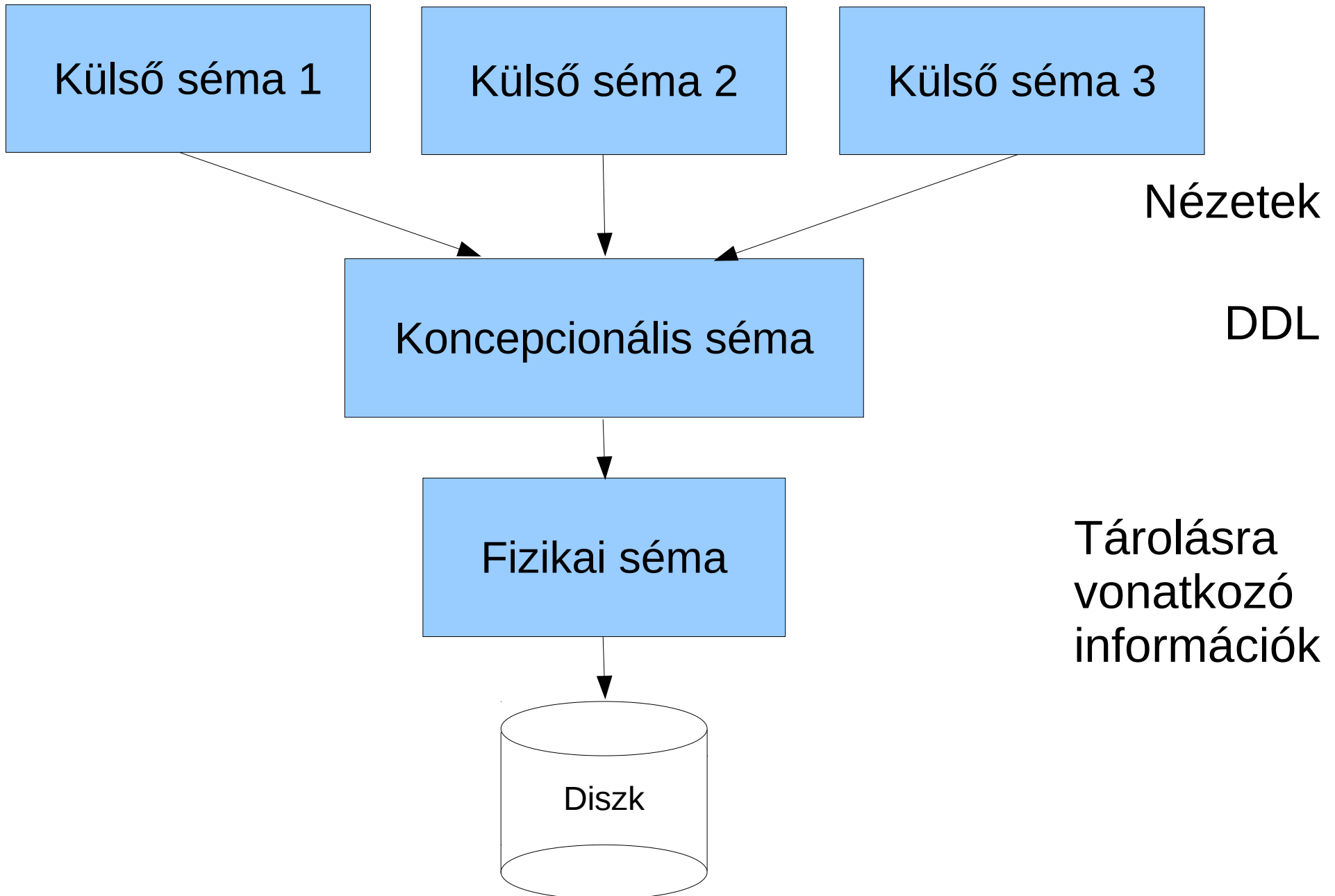


# Fizikai tervezés

## Hatékonysági kérdések

# Adatbázis absztrakciós szintek



# Fizikai tervezés

Tárolásra vonatkozó további részletek

Logikai (konceptcionális) séma közvetlenül leképezhető a relációs adatbázis-kezelőkben

Adattípusok konkrét meghatározása (domain)  
char, varchar, date, datetime, integer (16, 32, 64 bit), stb.

Indexek tervezése, létrehozása

Elsődleges kulcs (primary key) – leghatékonyabb visszakeresés

Egyedi indexek = kulcsok

Egyéb indexek a gyakori keresések gyorsítása

A hatékonyság érdekében redundanciák dokumentált elhelyezése

Triggerek tervezése, elkészítése

# Indexek

Rendezett adathalmazban a keresések gyorsabban végrehajthatók illetve a rendezettségnek megfelelő megjelenítés is gyorsul

Egy reláció egy vagy több attribútumára vonatkozik

Relációkban nem az adatok sorrendjét módosítjuk a rendezettséghez, hanem azoktól függetlenül több sorrend szerinti rendezettséget tárolunk

Indexeket a reláció adatoktól függetlenül tároljuk általában

Hash kód indexek és bináris fa indexek

Bináris keresés (intervallum felezéssel)  $\log_2(n)$

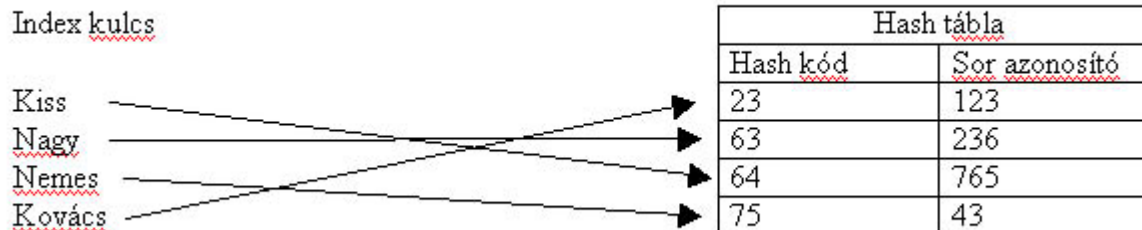
# Hash kód

Leggyorsabb keresés

Hash kód képző algoritmus

Hash kód ütközés kezelése

Csak a teljes mezőre alkalmazható, rész keresés nem lehet



# Indexelési példa (dense index)

## Index

Index	RowID
Bíró	6
Fehér	7
Kovács	2
Nagy	1
Nagy	3
Szabó	4
Takács	5

## Reláció

RowID	Név	
1	Nagy	...
2	Kovács	...
3	Nagy	...
4	Szabó	...
5	Takács	...
6	Bíró	...
7	Fehér	...

# Bináris fa indexek

Növekvő vagy csökkenő sorrend előállítása

Több mezőből összetett értékre is használható, eltérő adattípusok esetén is

Nem kell a teljes tartalmat ismerni a kereséshez, elegendő a keresendő minta elejét ismerni

Kiegyensúlyozott bináris fák



# Triggerek

Az adatbázisban bekövetkezett események hatására végrehajtott programrészletek  
pl. új rekord beszúrása, rekord aktualizálása, rekord törlése

Az adatbázisban nem leképezhető kényszerek érvényesítésére,  
az inkonzisztencia elkerülése érdekében

A módosítás előtt (before) vagy után (after) alkalmazott  
triggerek

Adatbázis-kezelőnként eltérő nyelv (pl. PL/SQL, PostgreSQL)

```
mysql> CREATE TRIGGER ins_sum BEFORE INSERT ON account  
-> FOR EACH ROW SET @sum = @sum + NEW.amount;
```



# Példa

Termek(épület, terem, férőhely, kategória)  
épület, terem → férőhely, férőhely → kategória

Termek(épület, terem, férőhely)  
épület, terem → férőhely

kategóriák(férőhely, kategória)  
férőhely → kategória

kategóriák(min\_férőhely, max\_férőhely, kategória)  
min\_férőhely → max\_férőhely, kategória  
max\_férőhely → min\_férőhely, kategória  
Kategória → min\_férőhely, max\_férőhely

kategóriák(max\_férőhely, kategória)  
max\_férőhely <-> kategória

