# Adatbázisok

## Fizikai adatszervezés
A háromrétegű modell alsó rétege

**Soros/közvetlen** hozzáférésű tárolóeszköz

**HDD esete: rekordszerű struktúrák kezelése**
Tárfoglalást műveletek: Keresés, törlés, beszúrás, módosítás
A DBMS számára a fizikai réteg fájlként jelenik meg.

**Főcél: blokkműveletek számának minimalizálása.**

## Témák

- DBMS
- Fizikai adatbázis
- Relációs adatmodell
- Relációs lekérdezések optimalizálása
- Relációs adatbázis logikai tervezése
- Tranzakciók adatbázis-kezelő rendszerekben

## Relációs adatbázisok logikai tervezése, normalizálás

## Relációs adatbázisok fizikai szervezése, indexelése

## Relációs adatmodell

## Relációs lekérdezések optimalizálása

## Tranzakciókezelés

Eddig: **1 CPU, 1 felhasználó, 1 háttértár**
→ fut-van, ha a kódunk

Tőlük felhasználói: 10 Rendszerszintű: MISC
Nem elég gyors? Szinkron I/Ok: MISC
Nem elég rákacsolva? Szinkron I/Ok: Más-tárgy
Nem elég halálosra? Szinkron I/Ok: Kiváltási home

**Transakció**: Egy program egyszeri futtatása, amely egy teljesen végzőjét, vagy teljesen semmi sem szabad hogy megszakítsák bele

Egyszerűsítés hogy tranzakciók problémáival oszlunk meg:
- konzisztenciaszint
- izolációszint

### lost update, non-repeatable read, phantom read, dirty data

Problémák **4** alapesete,

Mindig a kézben használt adatokat van a probléma.

### Zárak; paltok és éhezés

Zárak: adatok/táblák/oldalak/stb, mely adhatók és visszavonhatók.

Egyik Demóznánk: Egy olyan demóprogram, mely szándékosan okoz problémákat a tranzakciókban.

Input Demóznánk: Olyan demóprogram, mely szándékosan okoz problémákat a bemenetben.

Nem fog sallerni az egészet hybrid reprezentáció

### Relációs lekérdezések költségbecslés alapú optimalizálása

Elemzés (statisztikák) fontossága

Költségoptimalizáló

Kérdezés

Olyan ezek Microsoft Informix megoldott, hogy az operátor mögött van egy statisztikai szerver, amely ezeket a statisztikákat szolgáltatja.
Heurisztikus költségalapú optimalizáláso a megoldás

### Katalógusban tárolt információk

Katalógusban van egy sémaleíró táblánk, ahol van egy rekordunk minden táblánkra, és van egy sémaleíró táblánk, ahol van egy rekordunk minden attribútumunkra.

Szükséges lehet a katalógusban tárolt adatszótár is a szinonimák miatt.

szinonimák: ugyanazt a dolgot más-más néven szólítjuk

homonimák: ugyanazt a dolgot ugyanazon néven szólítjuk

Ez a szinonimák és homonimák problémája a szinonimák miatt van szükségünk az adatszótárra.

Ez a szinonimák és homonimák problémája a szinonimák miatt van szükségünk az adatszótárra.

Ez a szinonimák és homonimák problémája a szinonimák miatt van szükségünk az adatszótárra.

### Materializáció és pipelining

Materializáció: Egy adott relációt egy ideiglenes táblába mentünk el, majd ezt az ideiglenes táblát olvasunk tovább.

Egy operátor szerintünk annyit csinál, amennyit a szülője kérni fog tőle.

Pipelining: Egy operátor addig dolgozik, amíg van olyan szülője, aki kérni szeretne tőle valamit.

### Kiértékelési terv kiválasztása

Megoldás:
- Szabályalapú optimalizáció
- Heurisztikus optimalizáció
- Költségalapú optimalizáció

### RLOCK-WLOCK modell

RLOCK: olvasáshoz szükséges zár. Más tranzakció is elhelyezhet egyidejűleg egy olvasási zárat.

WLOCK: íráshoz szükséges zár. Más tranzakció nem helyezhet el egyszerre más zárat az adatbázisunkon.

A szintek az egyszerű Tranzakciókezelés: Egy olvasási szint szintjeinek.

Szinonimák és homonimák miatt van szükségünk az adatszótárra.

Szinonimák és homonimák miatt van szükségünk az adatszótárra.

Szinonimák és homonimák miatt van szükségünk az adatszótárra.

### Tranzakciókhoz kapcsolódó adatok

Tranzakciókezelés szintjei

Szinonimák és homonimák miatt van szükségünk az adatszótárra.

Szinonimák és homonimák miatt van szükségünk az adatszótárra.

### Tranzakciókezelés, idő előtti befejeződés okai

1. user abort
2. rendszerhiba (szerver is)
3. clienthiba (kliens is)
4. rendszerhiba (szerver is) és szerver szerver
5. mélyhiba (szerver szerver is)

szinkronizáláshoz kellenek

- szinkronizáció
- naplózás
- kétfázisú protokoll
- kétfázisú zárolás