

# Adatbázisműveletek és lekérdezésoptimalizálás

## *Nagyméretű adathalmazok kezelése*

Kazi Sándor

2010. február 24.

- 1 Bevezetés
- 2 Optimalizáció
  - Oracle Server optimalizációja
  - MS SQL Server optimalizációja
- 3 Adatbázisműveletek
- 4 Indexek
  - Oracle Server indexei
  - MS SQL indexek
- 5 Tábla-elérési módok
  - Oracle Server tábla-elérési módjai
  - MS SQL Server tábla-elérési módjai
- 6 Lekérdezések megfogalmazása
- 7 Források

# Relációs algebrai alpműveletek

- Descartes-szorzat:  $R \times S$
- Illesztések:  $R \bowtie S$ ,  $R \bowtie_{\Theta} S$ ,  $R \ltimes S$ ,  $R \rtimes S$
- Projekció:  $\pi_L(R)$
- Szelekció:  $\sigma_{\Theta}(R)$
- Ismétlődések szűrése:  $\delta(R)$
- Csoportosítás:  $\gamma_L(R)$
- Rendezés:  $\tau_L(R)$

# Mire optimalizálunk és hogyan?

*Általában válaszdőre:*

- I/O költség
- CPU használat
- Memória-használat

A három közül (mágneselemek esetén) jellemzően az I/O költség a legnagyobb, így a legjellemzőbb.

*Alapos tervezéssel:*

- Adatbázisszerkezet
- Statisztikák
- Heurisztikák
- Korábbi tapasztalatok

# Elemzőfa

Az elemzőfa egy szemléltetési módja a végső tervnek és kialakulásának:

- Levelei a relációk
- Belső csomópontjai a műveletek
- Gyökere az "eredmény"
- Az adatok a relációk felől a gyökér felé "áramlanak"

A végrehajtási tervnek megfelelő elemzési fát az eredeti fa átrendezésével kaphatjuk meg.

- 1 Bevezetés
- 2 Optimalizáció**
  - Oracle Server optimalizációja
  - MS SQL Server optimalizációja
- 3 Adatbázisműveletek
- 4 Indexek
  - Oracle Server indexei
  - MS SQL indexek
- 5 Tábla-elérési módok
  - Oracle Server tábla-elérési módjai
  - MS SQL Server tábla-elérési módjai
- 6 Lekérdezések megfogalmazása
- 7 Források

# Optimalizálni nehéz

Tétel:

A kérdés, hogy néhány reláció illesztésének van-e sora, NP-teljes.

Bizonyítás:

NP-beli a tanú-tétel szerint, mert egy sor az illesztés eredményéből egy polinom időben ellenőrizhető tanú. Emellett  $JOIN > 3SZÍN$ . Vegyük a  $G$  gráfot, amelyről tudni szeretnénk, hogy 3-színezhető-e, minden éléhez vegyünk fel egy kétattribútumú relációt, az attribútum legyen a két végpont azonosítója, amelyhez az él tartozik. Ezután az összes relációba vegyünk fel a C1-C2, C1-C3, C2-C1, C2-C3, C3-C1, C3-C2 sorokat. A  $G$  gráf így pontosan akkor 3-színezhető, ha a relációk természetes illesztésének van sora (és az meg is határoz egy színezést).

# Lekérdezésfeldolgozás

## Általános séma

### 1 Elemző

- Lekérdezés fordítása
- Logikai végrehajtási terv készítése

### 2 Optimalizáló

- Fizikai végrehajtási terv készítése
- Illesztési mód kiválasztása
- Táblához-záférési mód kiválasztása

### 3 Sorfordító

- Fizikai terv leképzése I/O műveletekre

### 4 Végrehajtó

- I/O művelet sor végrehajtása



- 1 Bevezetés
- 2 Optimalizáció**
  - Oracle Server optimalizációja
  - MS SQL Server optimalizációja
- 3 Adatbázisműveletek
- 4 Indexek
  - Oracle Server indexei
  - MS SQL indexek
- 5 Tábla-elérési módok
  - Oracle Server tábla-elérési módjai
  - MS SQL Server tábla-elérési módjai
- 6 Lekérdezések megfogalmazása
- 7 Források

# Oracle Server lekérdezésfeldolgozása

- 1 Elemző
- 2 Optimalizáló
  - 1 Lekérdezésátalakító
  - 2 Becslő (és adatszótár)
  - 3 Tervkészítő
- 3 Sorfordító
- 4 Végrehajtó

A tervekészítő által készített terv is bekerül az adatszótárba a becslőn keresztül.

# Elemző és lekérdezésátalakító

*Ha az elemző feladata a logikai terv kialakítása, akkor mire való a lekérdezésátalakító?*

## Elemző

- Logikai terv készítése
- Kiválasztás műveletek lefelé mozgatása az elemzőfában
- Vetítések felfelé mozgatása
- Illesztések struktúrája és sorrendje ( $O(n!)$  lehetőség)
- Szelekciós feltételek szétvágása/cseréje

## Lekérdezésátalakító

- Nézetek befésülése (nem mindig optimális)
- Feltételek lefelé tolása az elemzőfában
- Allekérdezések felemelése
- Lekérdezések újraírása materializált nézetekre

# Becslő és tervkészítő

## Becslő

- Kapcsolat az adatszótárral
- Statisztikák használata
  - Oszlopérték-hisztogramok
  - Indexek telítettsége
  - Dinamikus mintavételezés

Statisztikák nélkül nagyon rossz terv is születhet.

## Tervkészítő

RBO Rule Based Optimizer: Szabályalapú optimalizáló

CBO Cost Based Optimizer: Költségalapú optimalizáló

Az optimalizáló befolyásolása:

- Indexek deklarációjával
- SQL hintek használatával
- Lekérdezés átírásával

- 1 Bevezetés
- 2 Optimalizáció**
  - Oracle Server optimalizációja
  - **MS SQL Server optimalizációja**
- 3 Adatbázisműveletek
- 4 Indexek
  - Oracle Server indexei
  - MS SQL indexek
- 5 Tábla-elérési módok
  - Oracle Server tábla-elérési módjai
  - MS SQL Server tábla-elérési módjai
- 6 Lekérdezések megfogalmazása
- 7 Források

# MS SQL Server optimalizációja

- 1 Triviális terv elkészítése
  - Egyszerűbb lekérdezésekre
  - Szabály alapú
- 2 3 fázisú optimalizáló
  - Nem készíthető triviális terv

Statisztikák alapján

Az optimalizáló befolyásolása:

- Indexek deklaráálásával
- SQL hintek használatával
- Lekérdezés átírásával

# Háromfázisú optimalizáló

## 0 fázis

- Egyszerű átalakítások
- $Cost < 0.2 \rightarrow$  KÉSZ

## 1 fázis

- Kibővített átalakítások
- $Cost < 1 \rightarrow$  KÉSZ

## 2 fázis

- Párhuzamosíthatóság vizsgálata

- 1 Bevezetés
- 2 Optimalizáció
  - Oracle Server optimalizációja
  - MS SQL Server optimalizációja
- 3 Adatbázisműveletek**
- 4 Indexek
  - Oracle Server indexei
  - MS SQL indexek
- 5 Tábla-elérési módok
  - Oracle Server tábla-elérési módjai
  - MS SQL Server tábla-elérési módjai
- 6 Lekérdezések megfogalmazása
- 7 Források



# Adatbázisműveletek elvégzési módjai

Fokozatok a feladat "nehézsége" szerint:

- 1 Egymenetes algoritmusok
- 2 Kétmenetes algoritmusok
- 3 Adatok méretétől független módszerek

Módszerek alapja szerint:

- 1 Rendezésen alapuló módszerek
- 2 Tördelésen alapuló módszerek
- 3 Index alapú módszerek

Operátorok szerint:

- 1 Soronkénti, unáris műveletek
- 2 Unáris, teljes relációs műveletek
- 3 Bináris, teljes relációs műveletek

# Adatbázisműveletek elvégzési módjai

## Fokozatok a feladat "nehézsége" szerint

- 1 Egymenetes algoritmusok
  - Egyszer kell végigolvasni a táblát
  - Általában csak akkor, ha az egyik tábla elfér a memóriában
- 2 Kétmenetes algoritmusok
  - "Kétszer" kell beolvasni: műveletvégzésre, összefésülésre
  - Nagyobb táblákra is működik
- 3 Adatok méretétől független módszerek
  - Három vagy több menet
  - A kétmenetes algoritmusok általánosításai

# Egymenetes algoritmusok

- 1 Soronkénti műveletek ( $\pi$ ,  $\sigma$ )
  - Soronkénti olvasás, módosítás, kiírás
- 2 Unáris, teljes relációs műveletek ( $\delta$ ,  $\gamma$ )
  - Olvasás - Hasonlítások, számlálások - Írás
- 3 Bináris, teljes relációs műveletek
  - Az egyik tábla megfelelően kicsi kell, hogy legyen
  - A kisebbik tábla alapján struktúrát építünk
  - A nagyobbik táblát a struktúra segítségével kezeljük

# Tábla illesztési módok

## Oracle és MS SQL

### Oracle / MS SQL join operátorok

- 1 Nested loops join / Nested loops
  - illesztés kettős ciklusba szervezve
  - ha a külső tábla "kicsi"
  - ◇ Nyalábolt adatokon ( $O(\text{Blokkok}(R) \cdot \text{Blokkok}(S))$ )
  - ◇ Nem nyalábolt adatokon ( $O(\text{Sorok}(R) \cdot \text{Sorok}(S))$ )
- 2 Hash join költsége
  - a kisebb tábla segítségével a memóriában hash-struktúra
  - a struktúra alapján a nagyobb tábla illesztés
  - ◇ ( $O(\_ (R) + \_ (S))$ )
- 3 Sort merge join
  - rendezés, majd illesztés
  - jobb mint a hash-join, ha rendezett a struktúra, vagy nincs rendezésre szükség
  - ◇ ( $O(\_ (R) + \_ (S))$ )

- 1 Bevezetés
- 2 Optimalizáció
  - Oracle Server optimalizációja
  - MS SQL Server optimalizációja
- 3 Adatbázisműveletek
- 4 Indexek**
  - Oracle Server indexei
  - MS SQL indexek
- 5 Táblaelési módok
  - Oracle Server táblaelési módjai
  - MS SQL Server táblaelési módjai
- 6 Lekérdezések megfogalmazása
- 7 Források

# Indexek pro és kontra

## Indexeljünk / Ne indexeljünk, mert:

- + Gyorsabban érhetjük el az adatokat
- + Egyediséget biztosíthatunk
- Diszk területet használ
- Többletmunkát igényel
- Túl sok index használata nem gyorsít, hanem lassít

## Indexelendő / Nem indexelendő oszlopok:

- + Elsődleges és idegen kulcsok (join)
- + Gyakori adattartomány keresés
- + Rendezett elérés
- Ritkán használt
- Kevés különböző értékkel rendelkező oszlopok
- BLOB

- 1 Bevezetés
- 2 Optimalizáció
  - Oracle Server optimalizációja
  - MS SQL Server optimalizációja
- 3 Adatbázisműveletek
- 4 Indexek**
  - Oracle Server indexei
  - MS SQL indexek
- 5 Táblaelési módok
  - Oracle Server táblaelési módjai
  - MS SQL Server táblaelési módjai
- 6 Lekérdezések megfogalmazása
- 7 Források

# Oracle indexek 1

- 1 B-fa indexek
  - Egyszerű és összetett indexek
  - Függvények használata nem támogatott
  - Sok különböző értéket felvevő oszlopra érdemes felvenni
- 2 Függvény alapú indexek
  - Matematikai függvény és determinisztikus Oracle függvények
  - Értékek előre kiszámolva
- 3 Bittérkép (bitmap) indexek
  - Kis doménnel rendelkező oszlopokra
  - 0-1 értékek
  - Több oszlopra felvett bitmax indexek együtt hatékonyak
  - Feldolgozás: "Bitmapped index merge"
- 4 Fordított kulcsú (reverse key) indexek
  - Oszlopon belül a bájtok sorrendje fordítva
  - Gyorsan és párhuzamosan számítható
  - Közeli értékek egymástól távol
  - Nincs tartománykeresés



## Oracle indexek 2

- 5 Alkalmazás-specifikus (application domain) indexek
- 6 Csoport (Cluster) indexek
  - Gyakran összekapcsolt táblák esetén
  - Két tábla fizikailag egymáshoz közel
  - Nélküle nem érhető el a hozzá tartozó tábla
- Indexelt szervezésű táblák
  - Adatok tárolása az indexszel együtt B-fában
  - Csak elsődleges kulccsal rendelkező táblára
  - Az index ilyenkor nem az index szegmensben jön létre
  - Célszerű, ha legtöbbször az elsődleges kulcs szerint keresünk
  - Gyors tartománykeresés az elsődleges kulcsra
  - Helyet spórolunk meg

- 1 Bevezetés
- 2 Optimalizáció
  - Oracle Server optimalizációja
  - MS SQL Server optimalizációja
- 3 Adatbázisműveletek
- 4 Indexek**
  - Oracle Server indexei
  - MS SQL indexek**
- 5 Tábla-elérési módok
  - Oracle Server tábla-elérési módjai
  - MS SQL Server tábla-elérési módjai
- 6 Lekérdezések megfogalmazása
- 7 Források

# MS SQL indexek

- 1 Egyszerű index
  - Egy attribútumértéket és blokkcímeket tartalmaz
- 2 Összetett index
  - Több attribútumértéket és címeket tartalmaz
- 1 Csoportosított (Clustered) index
  - Az adatok az index szerint tárolva
  - Egy táblán csak egy lehet, de több táblán lehet ugyanaz
  - Gyakran összeillesztett táblák tárolhatók az illesztés szerint
  - Elsődleges kulcs mentén létrejön
- 2 Nem csoportosított (Nonclustered) index
  - Cover index
    - Az indexbe felvett plusz attribútumok
    - Nem a tábla gyors elérését szolgálják
    - Cél, hogy ezek az adatok gyorsan elérhetőek legyenek

- 1 Bevezetés
- 2 Optimalizáció
  - Oracle Server optimalizációja
  - MS SQL Server optimalizációja
- 3 Adatbázisműveletek
- 4 Indexek
  - Oracle Server indexei
  - MS SQL indexek
- 5 **Tábla-elérési módok**
  - Oracle Server tábla-elérési módjai
  - MS SQL Server tábla-elérési módjai
- 6 Lekérdezések megfogalmazása
- 7 Források

# A legalapvetőbb tábla-elérési módok

## Teljes végigolvasás (*Full Table Scan*)

- Nyalábolt esetben  $O(\text{Blokkok}(R))$
- Nem nyalábolt esetben  $O(\text{Sorok}(R))$

## Index szerinti végigolvasás (*Index Scan*)

- Rendezési költség
- +
- A  $\sigma$  (szelekció) operátor megvalósítása

- 1 Bevezetés
- 2 Optimalizáció
  - Oracle Server optimalizációja
  - MS SQL Server optimalizációja
- 3 Adatbázisműveletek
- 4 Indexek
  - Oracle Server indexei
  - MS SQL indexek
- 5 Tábla-elérési módok**
  - Oracle Server tábla-elérési módjai**
  - MS SQL Server tábla-elérési módjai
- 6 Lekérdezések megfogalmazása
- 7 Források

# Oracle Server tábla-elérési módjai 1

- 1 Full table scan
  - Nincs index a táblán
  - Várhatóan sok rekord kell
  - Kicsi a tábla és egyszerűbb beolvasni
- 2 Rowid scan
  - Indexből nyert sorazonosító szerinti olvasás
- 3 Unique index scan
  - Elsődleges kulcs használata
  - Ha az index összes oszlopára szűrünk
  - *Preferált*
- 4 Index range scan
  - Index alapú szűrés
  - =, <, >, *LIKE*
  - Index szerint rendezett eredményhalmaz
- 5 Full index scan
  - Index szerint a teljes tábla kiolvasása
  - Kulcs szerint rendezett eredményhalmaz

## Oracle Server tábla-elérési módjai 2

- 6 Fast full index scan
  - Ha csak az indexben szereplő attribútumok kelljenek
  - A táblát nem is éri el
- 7 Index join
  - Ha az attribútumok több indexben elosztva
  - Indexek összekapcsolásával (általában hash alapon)
- 8 Bitmap index scan
  - Bitmap index alapú Boole-kifejezés kiértékelése
  - Bitmapped index merge



- 1 Bevezetés
- 2 Optimalizáció
  - Oracle Server optimalizációja
  - MS SQL Server optimalizációja
- 3 Adatbázisműveletek
- 4 Indexek
  - Oracle Server indexei
  - MS SQL indexek
- 5 **Tábla-elérési módok**
  - Oracle Server tábla-elérési módjai
  - **MS SQL Server tábla-elérési módjai**
- 6 Lekérdezések megfogalmazása
- 7 Források

# MS SQL Server tábla-elérési módjai

- 1 Table scan
  - Csak ha nincs semmilyen index a táblán
  - Szűrési feltételt is kiértékeli
- 2 Clustered index scan
  - Van clustered index: adatblokkok az index szerint rendezve
  - Nyalábolt adatolvasás
  - Preferált a Table scannel szemben
- 3 Nonclustered index scan
  - Elsősorban = operátor kiértékelésére
- 4 Clustered index seek
  - <, BETWEEN, > operátorok kiértékelésére
- 5 Nonclustered index seek
  - <, BETWEEN, > operátorok kiértékelésére

- 1 Bevezetés
- 2 Optimalizáció
  - Oracle Server optimalizációja
  - MS SQL Server optimalizációja
- 3 Adatbázisműveletek
- 4 Indexek
  - Oracle Server indexei
  - MS SQL indexek
- 5 Tábla-elérési módok
  - Oracle Server tábla-elérési módjai
  - MS SQL Server tábla-elérési módjai
- 6 Lekérdezések megfogalmazása**
- 7 Források

# Egyszerű és összetett lekérdezések

Lekérdezések altípusai:

- Egyszerű lekérdezések
- Allekérdezést alkalmazó lekérdezések
  - Nem korrelatív: Az allekérdezés független a fő kérdéstől
  - Korrelatív: Az allekérdezés nem független a fő kérdéstől
- Nézetet tartalmazó lekérdezések

Nem korrelatív lekérdezések jól optimalizálhatók, míg a korrelatívak nehezen. Törekedjünk nem korrelatív allekérdezések megfogalmazására.

Az allekérdezésekben használt nézeteket általában külön optimalizálják az adatbáziskezelők, és a tárolt tervet használják fel. Lehetőleg ne használjunk nézeteket allekérdezőként, főleg ne kapcsoljuk őket össze.

# Indexek és függvények

## Indexek

Egy táblán egy lekérdezés általában csak egyet tud használni, ezt többnyire a join műveletre meg is teszi.

Ha a kulcs kifejezésben szerepel már nem tudjuk az indexet használni ( $ID + 0$ ).

## Függvények

Select listában használva nem befolyásolja a végrehajtási tervet.  
Where feltételben lehetőleg ne alkalmazzuk mert:

- Minden rekordra le kell futtatni
- Nehezen mozgatható a fában
- Kimenetére nem készül statisztika, ezért nehéz optimalizálni

- 1 Bevezetés
- 2 Optimalizáció
  - Oracle Server optimalizációja
  - MS SQL Server optimalizációja
- 3 Adatbázisműveletek
- 4 Indexek
  - Oracle Server indexei
  - MS SQL indexek
- 5 Tábla-elérési módok
  - Oracle Server tábla-elérési módjai
  - MS SQL Server tábla-elérési módjai
- 6 Lekérdezések megfogalmazása
- 7 **Források**

# Források

- 1 Gajdos Sándor: Lekérdezés optimalizálás (egyetemi jegyzet 2008)
- 2 Garcia-Molina-Ullman-Widom: Adatbázisrendszerek megvalósítása
- 3 Kovács Ferenc: Adatbáziskezelő rendszerek, (egyetemi jegyzet 2005-2008)
- 4 [www.dba-oracle.com](http://www.dba-oracle.com)