# Az oracle architektúrája

# Elmélet

## Indexek:

A lekérdezést optimalizálni kell, jó lenne, ha nem kellene az egész táblát lekérni (ehhez lehet használni indexeket)

Az index lehet:

* unique (ha levél elemekben lévő kulcs értékek egyediek (unique / primary key az oszlop a táblában))
* nem unique (egy érték többször is előfordul az oszlopban)

2 fajtája van:

* B-fa index: ( fa struktújájú adat halmaz, gyökér elem fent van, levél elemben sorrendbe rendezve vannak az értékek (by default növekvőben))

összekötés van a levélelemek között 🡪 a lekérdezés tud ugrálni a levélelemek között

* Bitmap index: kulcs érték található 2 row id-val (Start ROWID és End ROWID) és egy bitmap sorozattal (melyik sor kell az adatblokkból)

Amikor sok az ismétlődő elem az oszlopban akkor érdemes ezt használni

(Jól tudja támogatni, ahol aggregációk történnek)

Index előny: Ha egy oszlopra felépül akkor tudja gyorsítani a keresést

Ha tábla mellé van index struktúránk akkor a táblát karban kell tartani + tárolási többletköltsége is van

## Adatszótár:

Metaadatok az adatszótárban (data dictionary: központi, csak olvashtó táblák és nézetek) tárolódnak

Data dictionary szerkezete:

* Base tables:
* User-Accessible Views

Data dictionary típusai:

* DBA (az adminnisztrátor tudja olvasni)
* ALL (amihez jogom van)
* USER (azokról az objektumokról, aminek én vagyok a tulaja)

## Adatbázis architektúra:

3 nyelv, amit megért:

* SQL (deklaratív típusú nyelv)
* Procedurális nyelvek: PL/SQL, Java

Architektúra:

* amikor a kliens kiadja az első utasítást akkor létrejön egy user process, ami kommunikálni fog a szerveren ezáltal létrejövő server processel

(van egy listener a kettő között, ami figyeli mindig, hogy van e bejövő kérés)

(ez egy munkafoylamat (session), aminek vége, ha a felhasználó kilép belőle/abort történik/...)

* adatbázis szerver több Instance (szolgáltatásból) épül fel
  + Memory Structures (System global Area)
  + Process Structure
* adatbázist alkotó fájlok struktúrája (fizikai rész) Storage Structures

Egy adatbázis példány mindig egy adatbázishoz kötődik

Lehet, hogy több adatbázis példány egy adatbázishoz kötődik

Egy példány (Instance) nem kötődhet több adatbázishoz (Database-hez)

Kliens kapcsolódása az adatbázishoz:

* Van az adatbázis szerver, oda belép a felhasználó a szerver operációs rendszerén keresztül
* Külön gépen van a kliens és külön gépen a szerver és hálózaton keresztül kapcsolódik a kettő
* Háromrétegű kapcsolaton keresztül – programon keresztül kérjük le, ami kapcsolódik az alkalmazás szerveréhez, ami lekéri az adatbázisból

Az Oracle struktúra három fő csoportra oszthatók:

* Tároló struktúrák
* Memória struktúrák
* Szerverfolyamatok

SGA (System Global Area), PGA (Program Global Area - szerver folyamatokhoz tartozik)

### SGA:

Database buffer cache – Beolvasott adat kerül ide a memóriába

Redo log buffer – Memóriából a módosított adat bekerül ide naplózásra utána a háttértárra kiírásra történik

Shared pool:

* Library cache – Shared SQL area – Követi a folyamatot

(beérkezett SQL (dekleratív nyelven) utasítás, ezt átalakítja)

Végrehajtási terv itt tárolódik

* Data dictionary cache – Adatszótár, leggyakrabban kiolvasott adatok itt tárolódnak
* Server result cache

### PGA:

* Private SQL area:
* Session memory: Az adott session-höz tartozó ideiglenes adatok találhatóak meg

## Adatbázis tranzakciók:

Összetartozó DML (módosító) utasítások sorozata

Van egy eleje és egy vége, feltételezzük, hogy a tranzakció kezdetekor az adatok megfelelnek az elvárásainknak és a végén is van egy konzisztens állapot. Közte lehet egy nem konzisztens állapot.

Egy egységként értelmezzük, konzisztens állapotok között mozgunk

ACID:

* Atomicity -
* Consistency -
* Izolation – Azt várja le, hogy amikor pl 2 tranzakció lefutott akkor utána úgy nézzen ki az adatbázis állapota, mintha külön futott volna le. (Tud várakoztatni, eldönti mely tranzakciókat fésülhet össze)
* Durability – Lefutott egy tranzakció, sikeres volt, ...

# Gyakorlat

OLTP:

Szervezet mindennapos működéséhez tartozó folyamatok (pl boltban való fizetés = módosítási kérések), ezekből sok van, gyakran használt műveletek, általában kevés adatot érintenek.

DSS/DWW:

Egy napban csak egyszer töltünk be adatokat, de akkor nagyon sokat. Több sort, táblát, nagyon sok adatokat érintünk ilyenkor.

Lekérdezések: Nem teszünk fel sok lekérdezést.

Olvasás és adatok kombinálása jelenti a fő feladatot.

Aggregáció(s művelet): Több sort érint, több művelet, több adatból állítjuk elő a választ.

Adatszótár után:

„

SELECT user\_indexes.table\_name, index\_name,

column\_name, column\_position, uniqueness

FROM user\_indexes

INNER JOIN user\_ind\_columns USING (index\_name)

ORDER BY 1,2,4;

„