Adatelemek ábrázolása

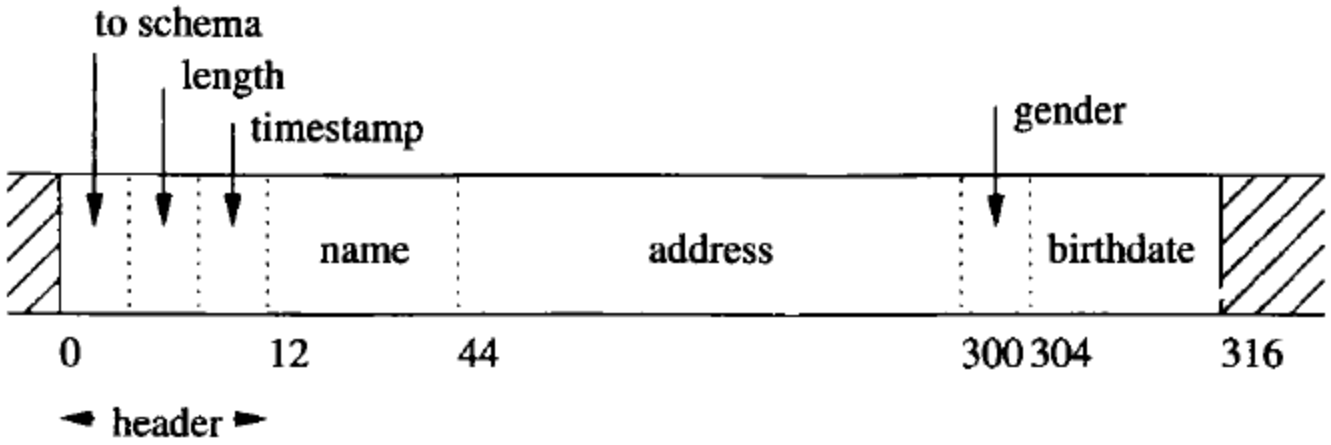
Adatbázisok 2.

Rögzített hosszúságú rekordok

- A relációsorokat rekordokkal ábrázoljuk.
- Legegyszerűbb esetben a rekord minden mezője rögzített hosszúságú.
- Egyes számítógépek hatékonyabban tudják írni és olvasni az olyan adatokat, amelyek a központi memóriában 4-gyel vagy 8-cal osztható címen kezdődnek. Ezt érdemes lehet a másodlagos háttértárolón való tárolásnál is figyelembe venni, hiszen a központi memóriában a rekord első bitje biztos, hogy egy ilyen 4-gyal vagy 8-cal osztható címre kerül.

Példa

```
CREATE TABLE színész (
   név                  CHAR(30),
   cím                  VARCHAR(255),
   neme                 CHAR(1),
   születési_idő        DATE);
```

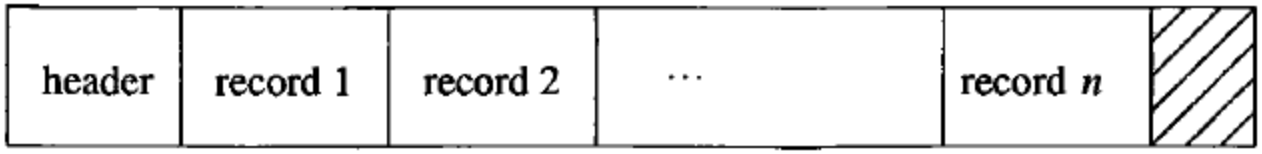


Rekordfejlécek

- A rekordokhoz gyakran tartozik fejléc
  - a rekordhoz tartozó tábla sémájával (vagy a sémainformációra mutató mutatóval),
  - a rekord hosszával,
  - időbélyegzőkkel, amelyek azt mutatják, hogy a rekordot mikor olvasták vagy módosították utoljára.
- A sémainformáció ismerete például akkor lehet fontos, amikor egy blokkban nem csupán egy relációhoz tartozó sorok tárolódhatnak.

Blokkok

- A rekordokat blokkokba szervezve tároljuk.
- A blokkokhoz is tartozik fejléc, amely tartalmazhat
  - linkeket más blokkokra egy-egy blokkhálózat esetén (pl. túlcsordulás blokkok),
  - információt a blokk blokkhálózatbéli szerepére vonatkozóan,
  - információt arról, hogy a blokk rekordjai milyen reláció(k)hoz tartoznak,
  - egy a blokkok eltolási értékeire vonatkozó jegyzéket,
  - egy blokkazonosítót,
  - a blokk utolsó olvasási, módosítási idejét megadó időbélyegzőt.



Blokk- és rekordcímek a központi memóriában

- Ha egy blokkot betöltünk a központi memória egy pufferébe, a blokk első bájtjának memóriacíme lesz a blokk virtuális memóriacíme.
- Hasonlóképpen egy blokkon belüli rekord esetén a rekord első bájtjának memóriacíme lesz a rekord virtuális memóriacíme.

Kliens-szerver rendszerek

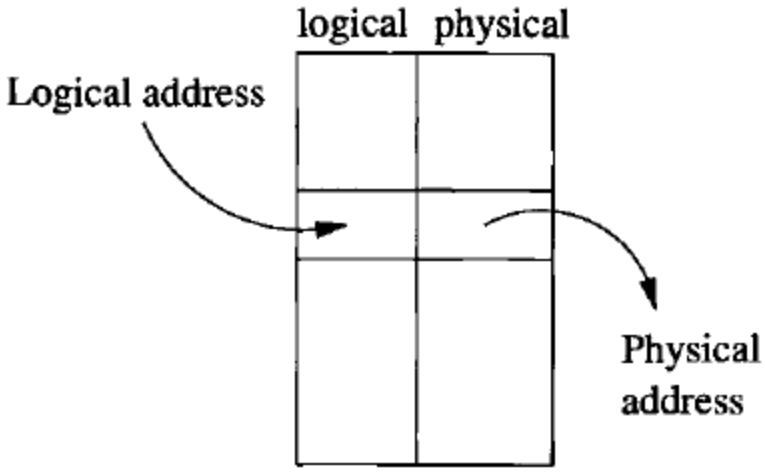
- Az adatbázisrendszereknél rendszerint egy szerver folyamat gondoskodik arról, hogy az adatok a másodlagos tárolóról eljussanak a kliens folyamatokhoz.
- A kliens folyamatok a virtuális címterületet használják.
- A szerver adatai ezzel szemben az adatbázis címterületén helyezkednek el.

Fizikai címek

- Egy rekord fizikai címe olyan bájtokból álló lánc, mely az alábbi információkat tartalmazhatja:
  - mely géphez tartozik a rekord (osztott adatbázis esetén)
  - a blokkot tartalmazó lemez vagy egyéb eszköz azonosítója
  - a cilinder sorszáma
  - a cilinderen belül a sáv sorszáma
  - a blokk sorszáma a sávon belül
  - a rekord eltolási értéke a blokkon belül.

Logikai cím

- Egy rekord logikai címe tetszőleges bájtsorozat lehet.
- A fizikai és logikai cím között a leképezési tábla (map table) teremti meg a kapcsolatot.
- Ez a szisztéma például adatszervezésnél lehet hasznos, amikor a rekordot ide-oda kell mozgatnunk. A rekordra mutató mutatók valójában a logikai címre hivatkoznak, így ezeket nem kell karban tartanunk, egyedül a leképezési táblában kell a fizikai címet egy-egy változtatásnál kiigazítani.

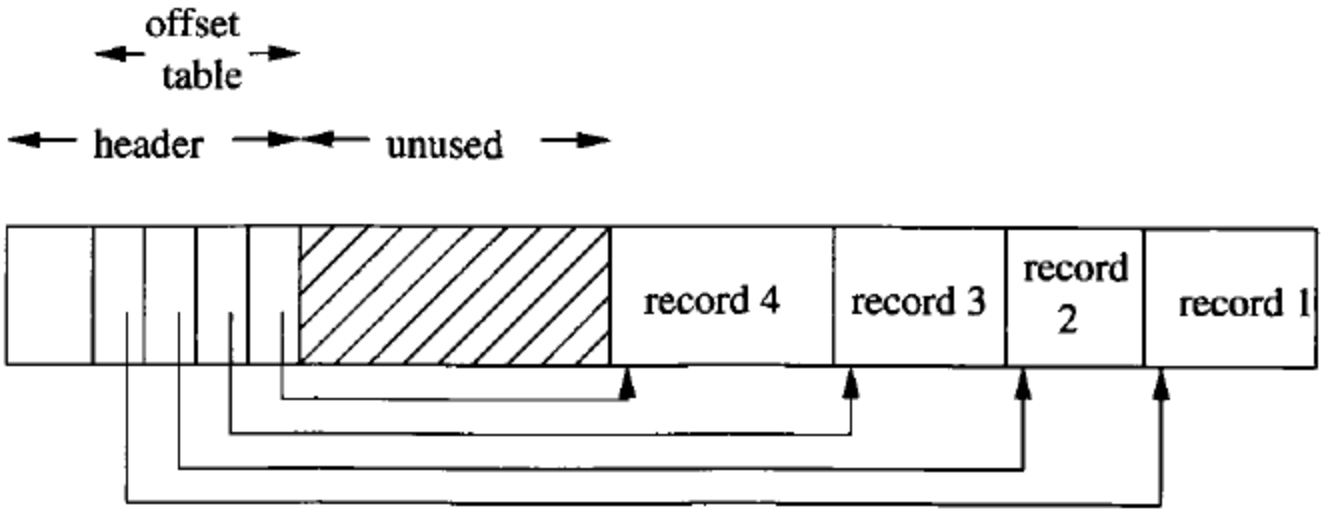


Strukturált címséma

- A strukturált címsémák a logikai és fizikai címek valamilyen kombinációi.
- Például a fizikai címben csak a blokk címét tároljuk, amihez a rekord kulcsértékét tesszük hozzá.
- Amikor a blokkot beolvassuk, a központi memóriában keressük meg a megfelelő kulcsértékű rekordot.

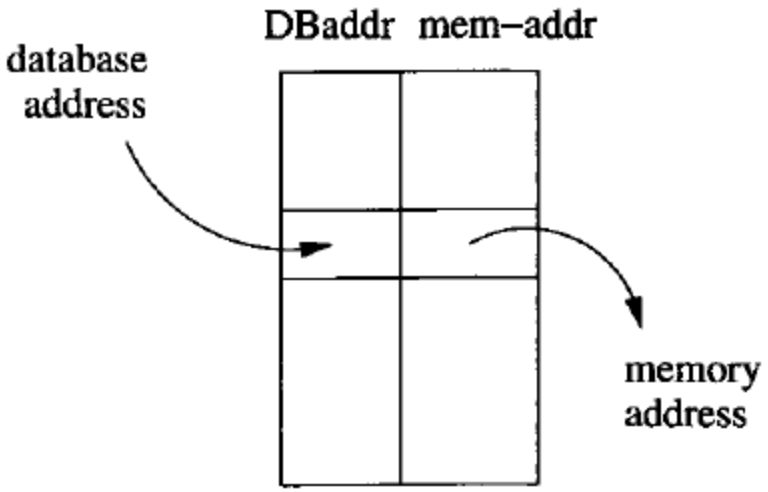
Eltolásiérték-tábla

- Egy másik hasznos kombináció lehet, amikor a blokkokban egy eltolásiérték-táblát tárolunk a rekordok eltolási értékeivel.
- Ha nincs leképezési tábla
  - a rekordot a blokkon belül szabadon mozgathatjuk
  - ha elegendő hely van az eltolásiérték-táblában egy másik blokk címének tárolására, akkor a rekordot akár egy másik blokkba is mozgathatjuk
  - ha törlünk egy rekordot az eltolásiérték-táblában a helyére egy sírkő kerül.



Fordítási tábla

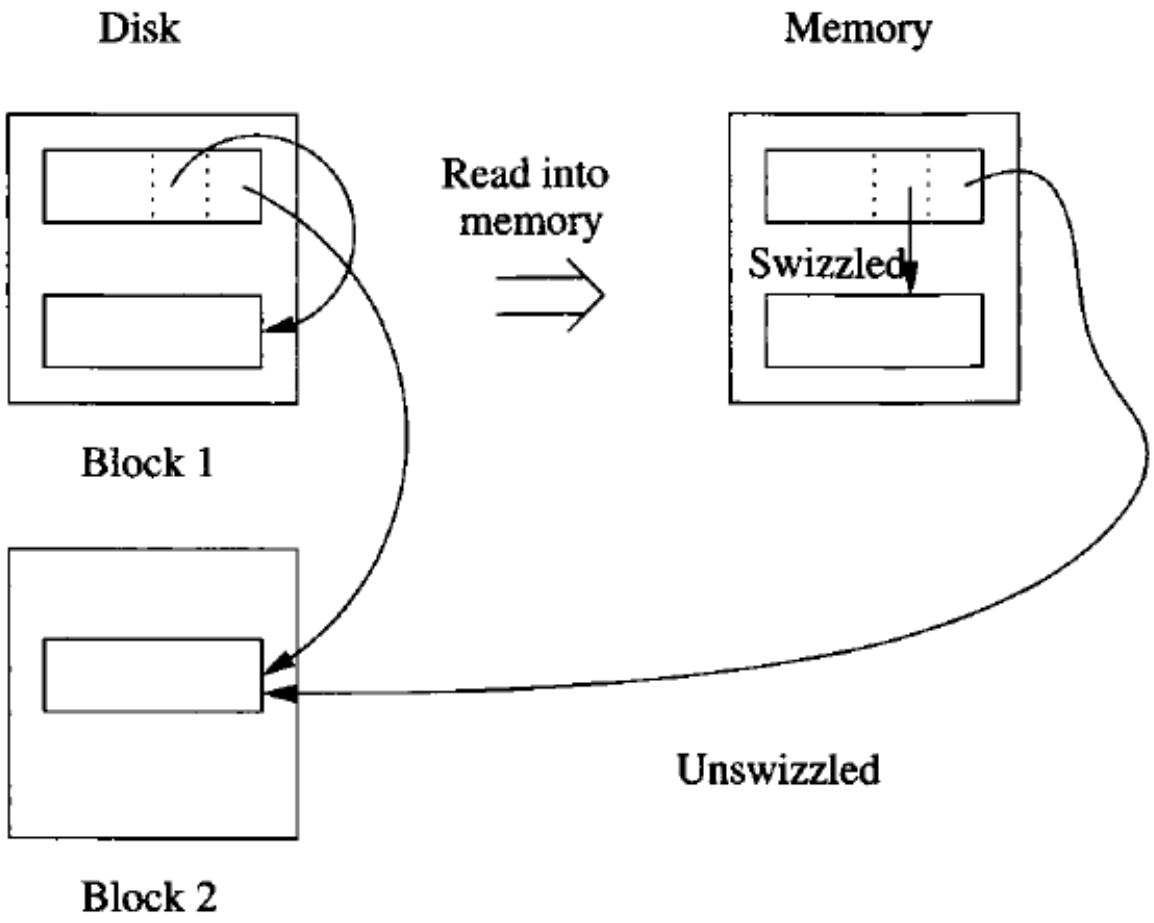
- Több esetben (objektumrelációs rendszerek, indexek) a rekordok mutatókat is tartalmazhatnak.
- Ha egy rekord a háttértárolón van épp az adatbáziscímet használjuk.
- Ellenkező esetben használhatjuk az adatbázis- és a memóriacímet is.
- Szükségünk lesz egy fordítási táblára, amely az adatbázis címhez az aktuális memóriacímet tárolja és megfordítva.



Mutatók helyreigazítása (pointer swizzling)

- A fordítási tábla állandó használata igen költségessé válhat, emiatt érdemes lehet a központi memóriában lévő blokkokban az adatbáziscímeket memóriacímekre cserélni.
- Így egy mutató két részből áll tulajdonképpen:
  - egy bit jelzi, hogy memória- vagy adatbázis címet tartalmaz-e adott mutató,
  - s ezután következik maga a mutató.

Mutatók helyreigazítása példa



Automatikus helyreigazítás

- Amint egy blokkot betöltünk a memóriába az összes mutatót megpróbáljuk helyreigazítani.
  - A beolvasott rekordok memóriacímét ismerjük, ezeket beszúrjuk a fordítási táblába (a megfelelő adatbáziscímmel).
  - A rekordok mutatóit megpróbáljuk a fordítási tábla alapján helyreigazítani, azaz az adatbáziscímeket memóriacímekre cserélni.
- Ha egy beolvasott rekord mutatóját kell követnünk a későbbiekben, ami nem lett helyreigazítva, akkor először mindig a fordítási táblában kell megnéznünk, hogy azóta nem került-e beolvasásra a hivatkozott másik rekord.

Igény szerinti helyreigazítás

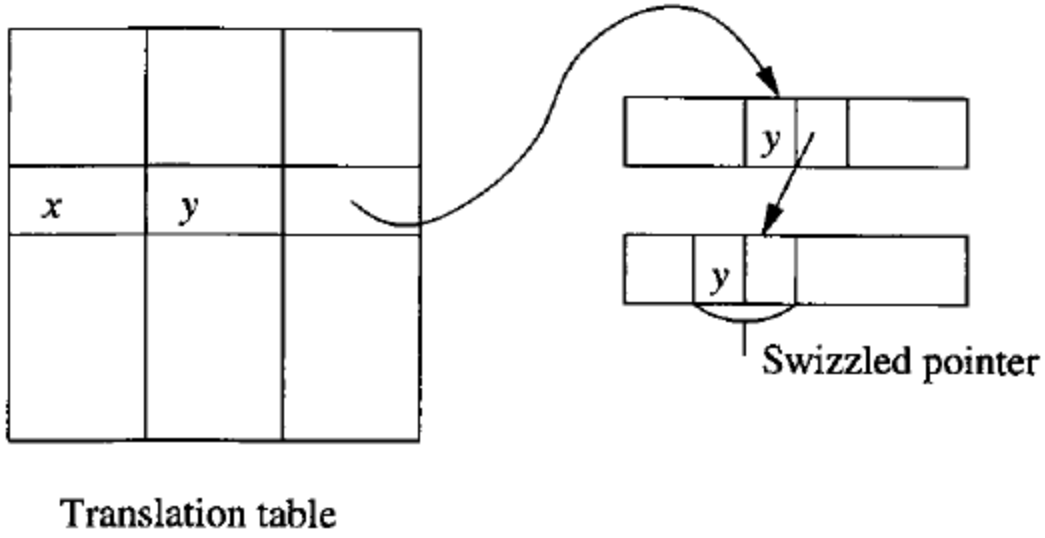
- Amikor beolvassuk a blokkot, egyik mutatót sem igazítjuk helyre.
- Mutatót csak akkor igazítunk helyre, ha egy memóriában lévő rekord mutatóját kell követnünk. Ebben az esetben ugyanúgy járunk el, mint az automatikus helyreigazítás esetében.
- A fordítási táblákhoz általában indexeket is szoktak készíteni, mind a memória- mind az adatbáziscímekhez.

Feltűzött rekordok és blokkok I.

- A memóriában egy blokkot feltűzöttnek (pinned) mondunk, ha nem lehet pillanatnyilag biztonságban visszaírni a lemezre.
- Ennek egyik oka lehet a naplózás.
- Egy másik ok, ha a B_1 blokkon belül van egy B_2 blokkra vonatkozó helyreigazított mutató. Ebben az esetben B_2 szintén feltűzött.
- A feltűzöttség megszűntetéséhez az összes B_2 –re vonatkozó helyreállított mutatót vissza kell cserélni a megfelelő adatbáziscímre.

Feltűzött rekordok és blokkok II.

- Emiatt a fordítási táblában egy-egy rekord esetében a rekordnak megfelelő sornál a rekordra vonatkozó helyreigazított mutatók elhelyezkedését is tárolni kell valamiképp.

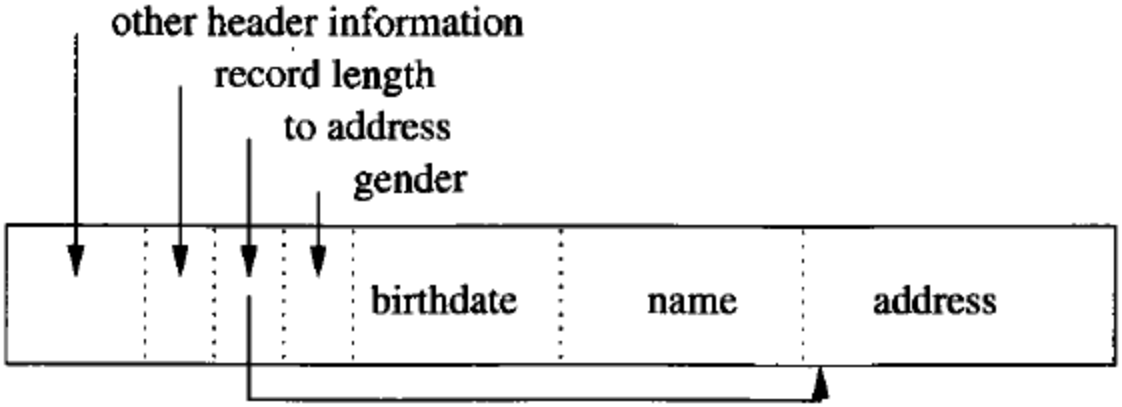


Változó hosszúságú rekordok

- Egy tábla rekordjai nem feltétlen azonos hosszúságúak. Ennek oka lehet:
  - ha változó hosszúságú típusok szerepelnek a rekordban (pl. VARCHAR)
  - valamilyen típusú értékből több is szerepelhet egy rekordban (pl. egy író által írt könyvekre mutató mutatók)
  - a rekord formátuma sem ismert előre (pl. XML adatok relációkban történő tárolásánál)
  - nagyméretű adatok (pl. BLOB).

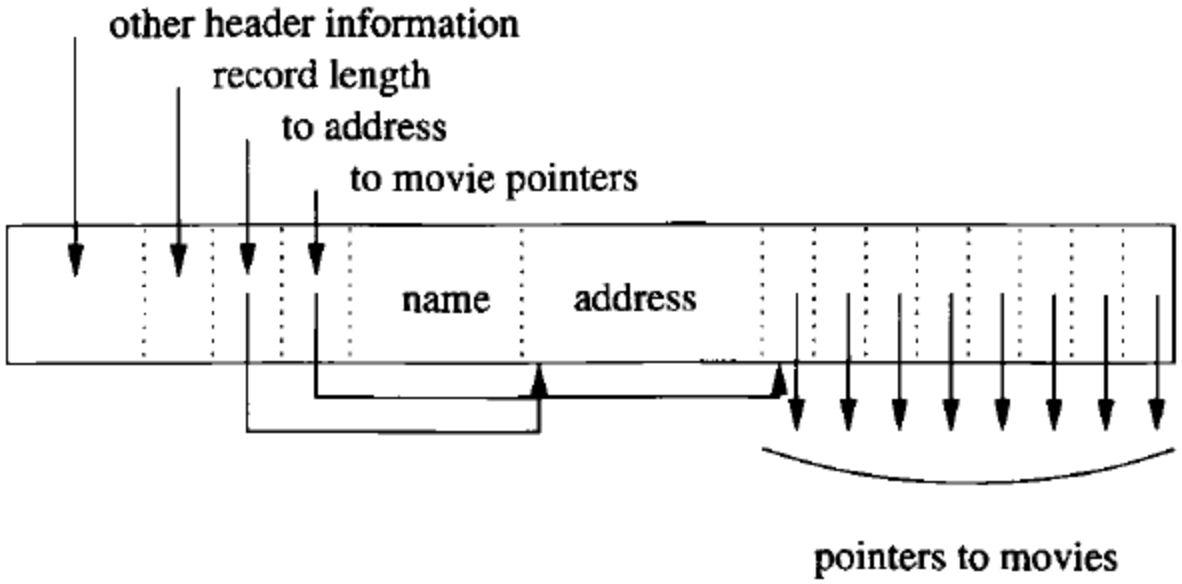
Változó hosszú mezők

- Ebben az esetben egy hatékony stratégia, ha a rögzített hosszúságú mezők szerepelnek a rekord elején.
- A rekord fejlécében ekkor tároljuk
  - a rekord hosszát
  - a változó hosszúságú mezők elejére mutató mutatókat (eltolási értékeket).
- A stratégia null értékek tárolásánál is hasznos lehet. Az eltolási érték egy null értékű mezőnél lehet 0.



Ismétlődő mezők

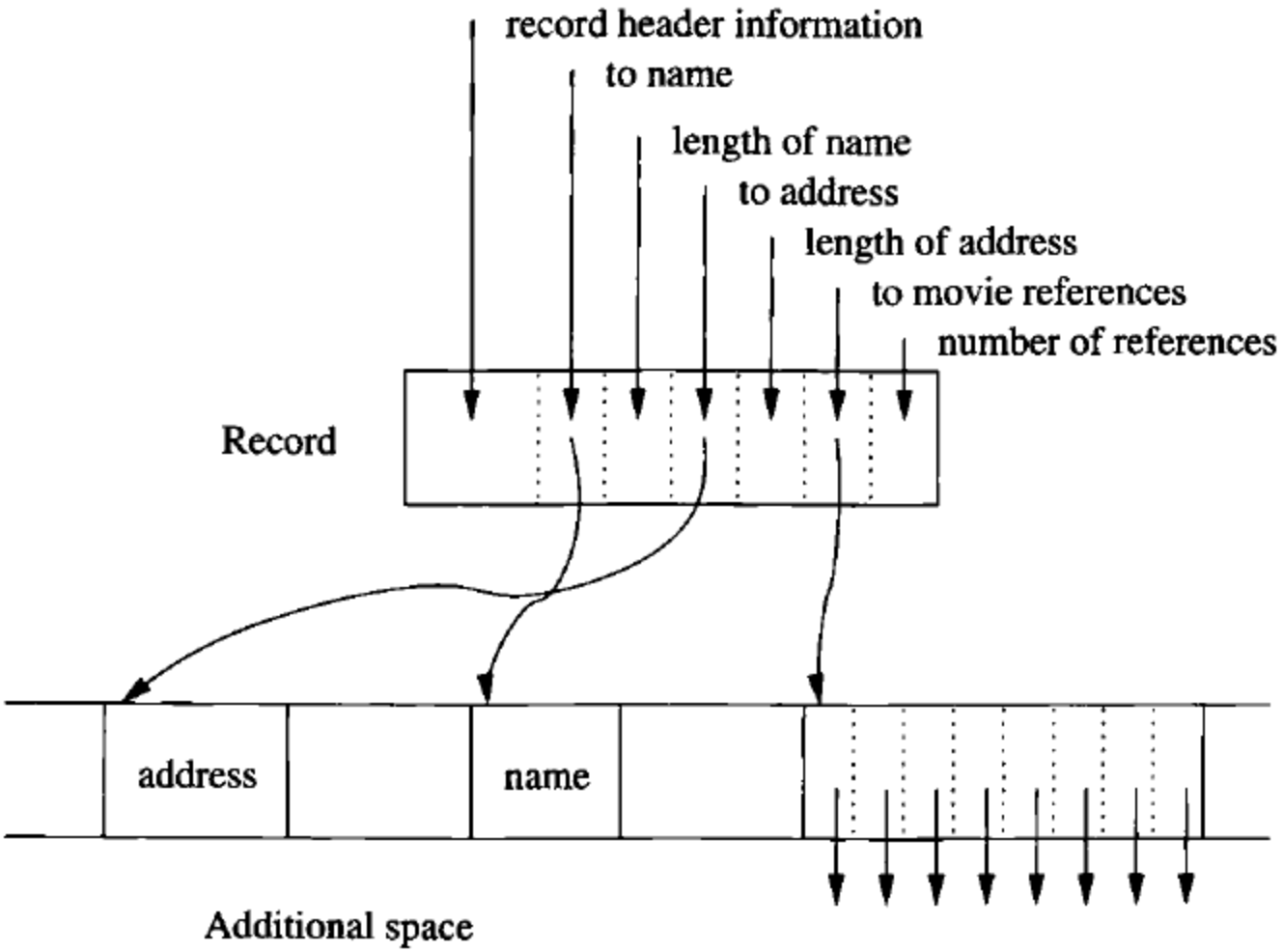
- Ha egy rögzített méretű mezőből szerepel ismeretlen számú előfordulás, akkor a mező összes előfordulásából egy csoportot képezhetünk, s a fejlécben a csoport elejére mutató mutatót tárolhatjuk.
- Az eltolási értékek a mező méretének ismeretében könnyedén számíthatók.



Egy másik megoldás I.

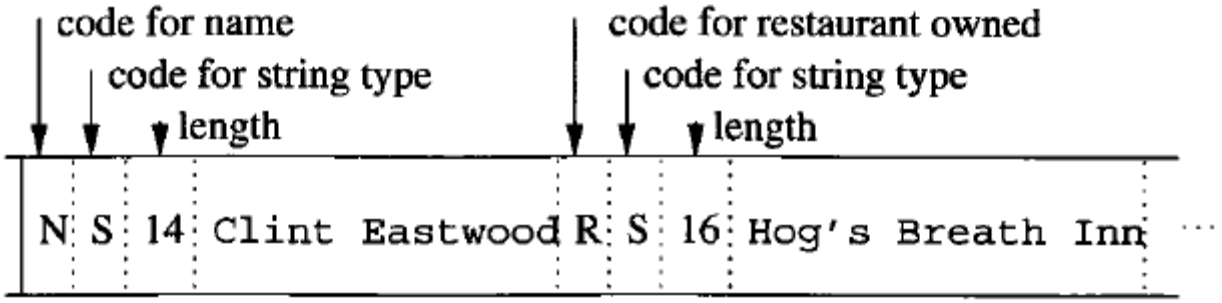
- Egy másik lehetséges stratégia, ha a rögzített hosszúságú mezőket a rekordban tároljuk, a változó hosszúságú vagy ismétlődő mezőket pedig egy másik blokkba helyezzük.
- A rekordok keresése így sokkal hatékonyabb, hiszen azok rögzített hosszúságúak.
- A másik oldalról viszont egy rekord beolvasása két blokk beolvasását igényli.
- Kompromisszumos megoldásként az ismétlődő mezőkből ésszerű számú előfordulást szintén az első rekordban tárolhatunk.

Egy másik megoldás II.



Változó formátumú rekordok

- Változó formátumú rekordok legegyszerűbb ábrázolási módja, mikor címkézett mezők (tagged fields) sorozatát adjuk meg.
- Egy címkézett mező a következőkből áll:
  - mező neve,
  - mező típusa (ha ez nem nyilvánvaló valamilyen könnyen elérhető sémából)
  - a mező hossza.
- Mező értéke

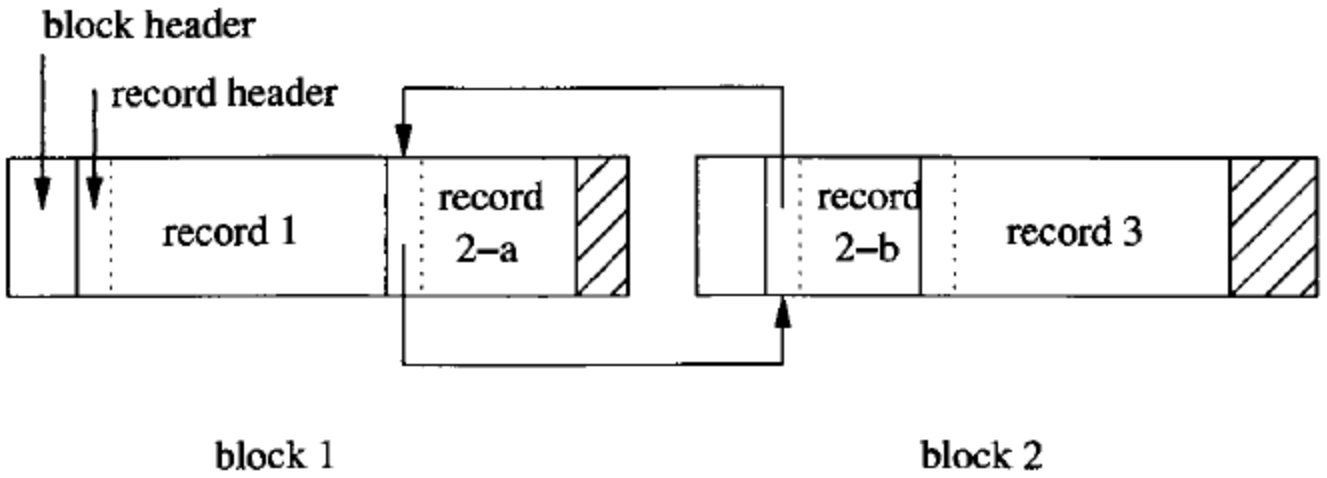


Blokknál hosszabb rekordok

- Ha egy rekord nem fér el a blokkban, akkor „tördelni” kell.
- Egy rekord azon részét, amely egy adott blokkba kerül, rekordtöredéknek nevezzük.
- A tördelt rekord neve átnyúló (spanned) rekord. Ennek megfelelően vannak át nem nyúló rekordok is.
- Átnyúló rekordok esetén több információt kell tárolni a rekord(töredék) fejlécben:
  - egy bit, ami megmondja, hogy a rekord töredék-e vagy sem
  - ha töredék, egy-egy bit tárolhatja, hogy ez a „tördelt” rekord első vagy utolsó eleme-e
  - az előző és következő töredékekre mutató mutatók elhelyezése is szükséges.

Egy másik alkalmazás

- Átnyúló rekordokat akkor is használhatunk, ha egy-egy rekord önmagában beleférne a blokkba, de mellette túl sok hely maradna kihasználatlanul a blokkban.
  - Például a rekord mérete kicsivel több, mint a blokk méretének fele.



BLOB

- BLOB: a binary large object, magyarul: bináris, nagy objektum.
- Tipikusan kép, videófájlok stb.
- Általában előnyös, ha a BLOB típusú adatot tároló blokkok folytonosan helyezkednek el a lemez egy vagy több cilinderén.
- Ha gyorsan kell visszanyerni az egymás után következő blokkokat, akkor a BLOB-ot csíkokra bonthatjuk és ezeket a részeket több lemezen helyezzük el.
- Sok alkalmazásnál fontos lehet, hogy a BLOB belsejéből igényelhessünk adatot a BLOB teljes beolvasása nélkül. Ebben a megfelelő indexek segíthetnek. Például egy BLOB típusú film másodperceire épülő index.

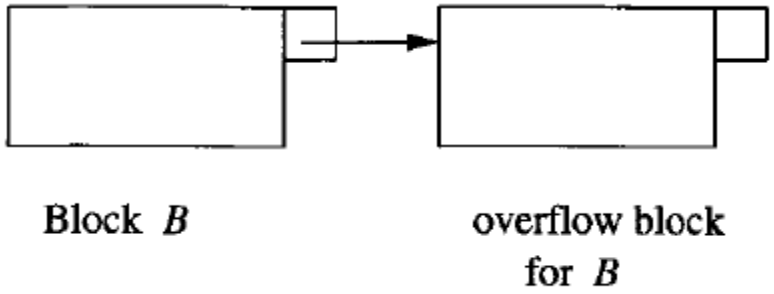
Oszlopok tárolása

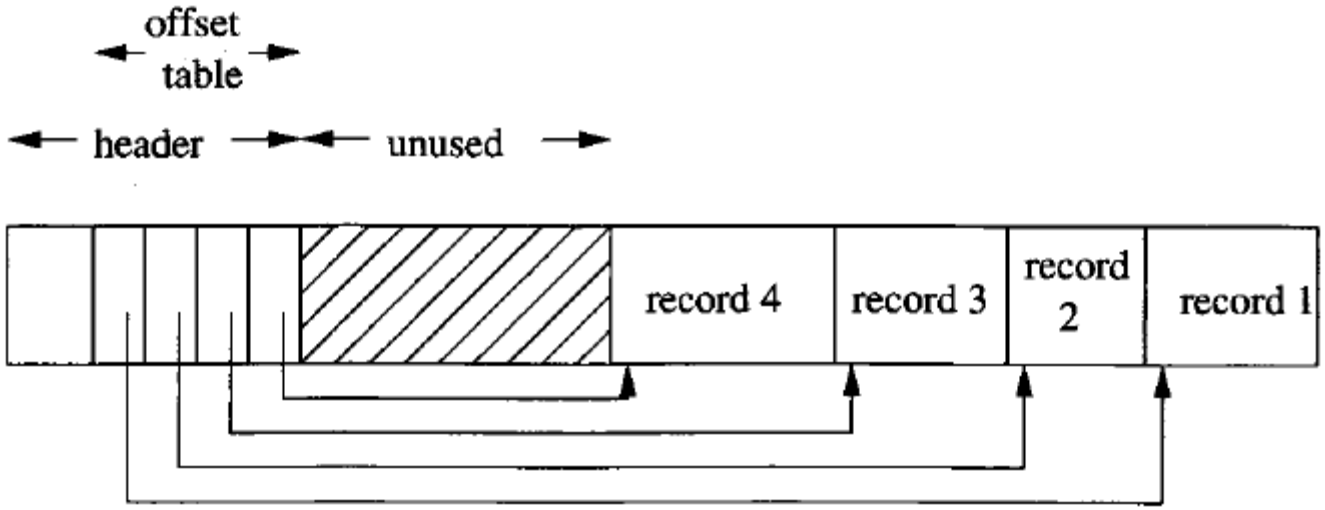
- A sorok helyett az oszlopokat is tárolhatjuk.
- Ilyenkor az A oszlopnak megfelelő rekord: (a, d, a) vagy ((1,a), (2,d), (3,a)), a B oszlop reprezentációja: (c, e, b) vagy ((1,c), (2,e), (3,b)).
- Hasznos lehet tömörítésnél. Például, ha egy 1 bites értéket 4 bájton tároltunk (lásd korábban).
- Akkor lehet érdemes ezt a tárolási stratégiát választani, ha a lekérdezések nagyobb hányada egy-egy oszlop összes vagy majdnem összes értékét felhasználja (pl. OLAP).

A	B
a	c
d	e
a	b

Beszúrás

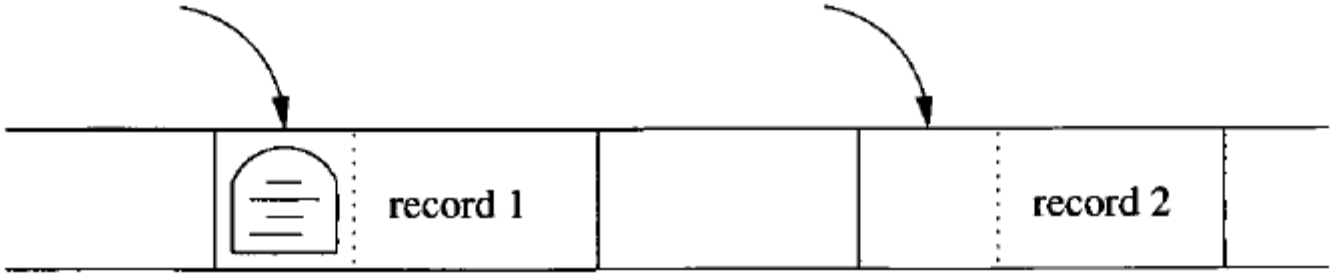
- Ha a relációt nem rendezetten tároljuk, a beszúrás nem jelent problémát.
- Rendezett tárolás esetén, ha a beszúrandó sor nem fér be a megfelelő blokkba:
  - csúsztatnunk kell a sorokat
  - vagy túlcsordulási blokkokat kell létrehoznunk.





Törlés

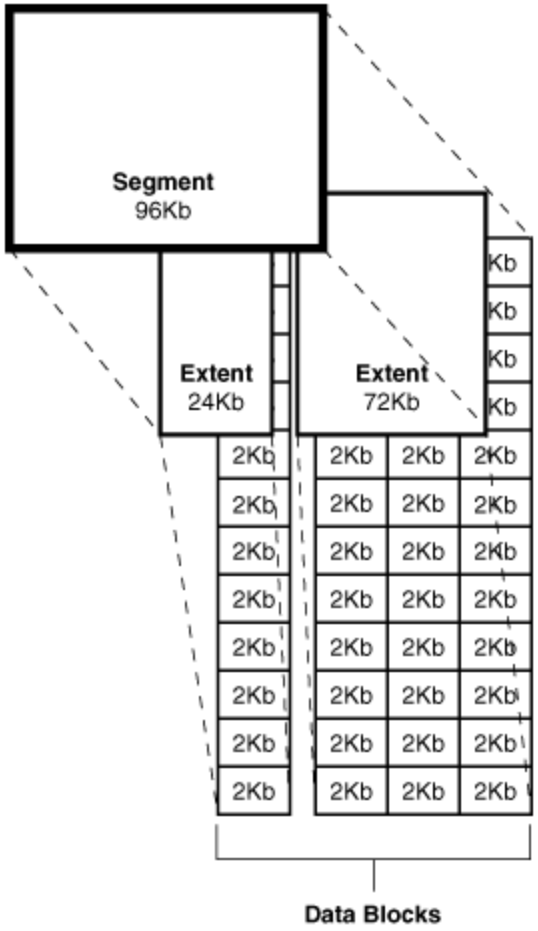
- Ha a rekordokat tudjuk csúsztatni egy blokkon belül (pl. eltolásiérték-táblát használunk), akkor a törlés nem jelent problémát.
- Más esetben napra készen kell tartani egy listát, amely a szabad helyeket mutatja meg.
- Ilyenkor nem a blokk fejlécében kell tárolni a teljes listát, hanem a felszabadult helyeket használhatjuk a listának tagjait megadó hivatkozások elhelyezésére.
- Mutatók használata esetén sok esetben egy sírkövet tesznek a törölt rekord „helyére”. A sírkő állhat:
  - az eltolási-érték táblázat megfelelő helyén
  - a leképezési táblában a fizikai cím helyén,
  - a törölt rekord helyének 1. bájtjában, a megmaradt helyen új rekordot tárolhatunk.



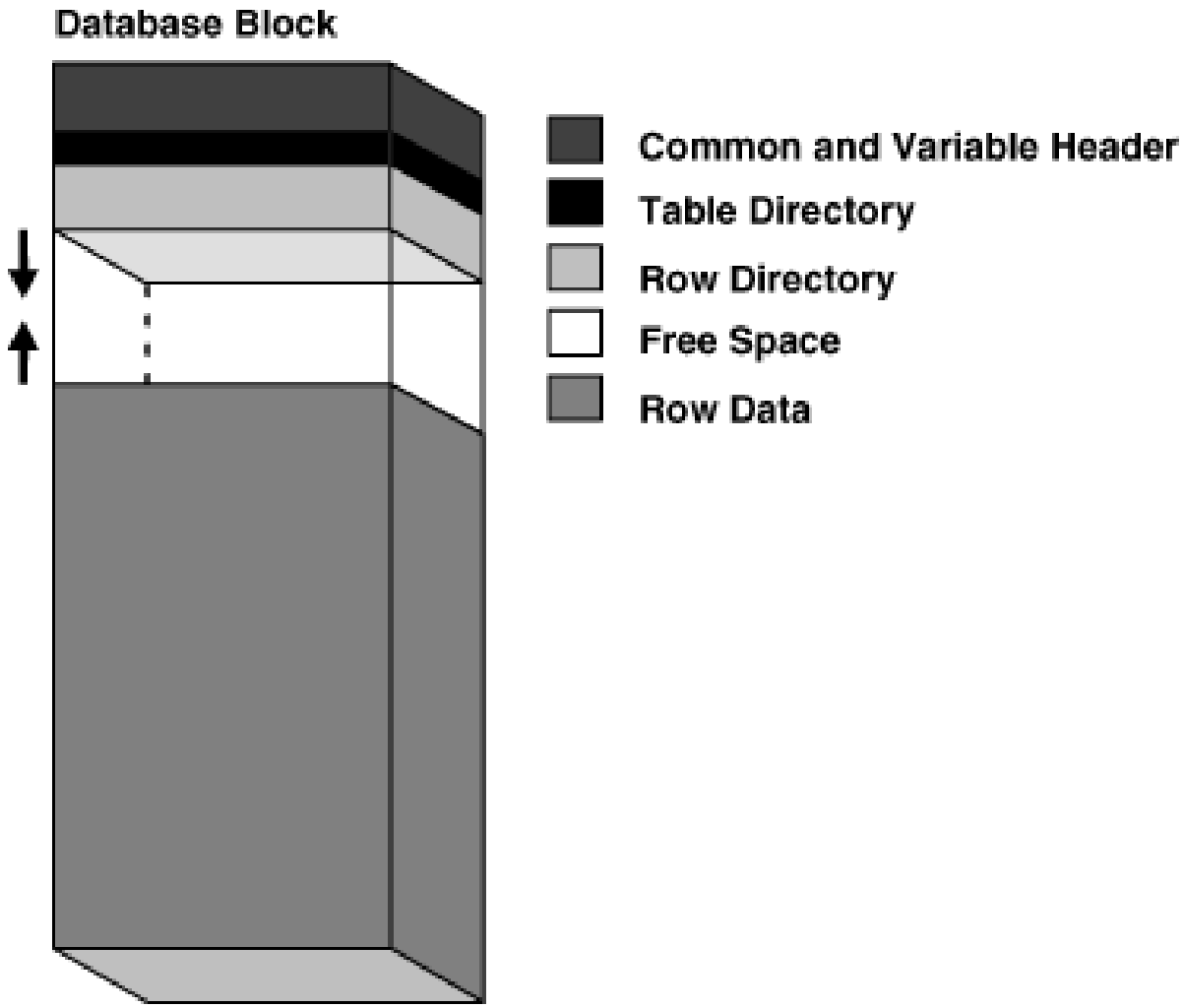
Adattárolás fizikai megvalósítása (Oracle)

- A legkisebb adategység, amit az Oracle kezelni képes a blokk
- Az Oracle által kezelt blokkok mérete különbözhet a háttérben működő operációs rendszer blokkméretétől, viszont annak csak többszöröse lehet.
- *Az extensek a háttértárolón folytonosan elhelyezkedő adatblokkokból állnak.*
- Egy-egy szegmens egy-egy adatobjektumnak felel meg, azaz a szegmens a fizikailag tárolt objektum. Particionált táblák és indexek esetén minden partíció külön szegmensen helyezkedik el.
- Az Oracle a létrehozáskor egyetlen extensen tárolja a szegmenst, ám ahogy az bővül, a rendszer újabb és újabb extenseket foglal le.

Segments, Extents, and Data Blocks



Blokkok (Oracle) I.



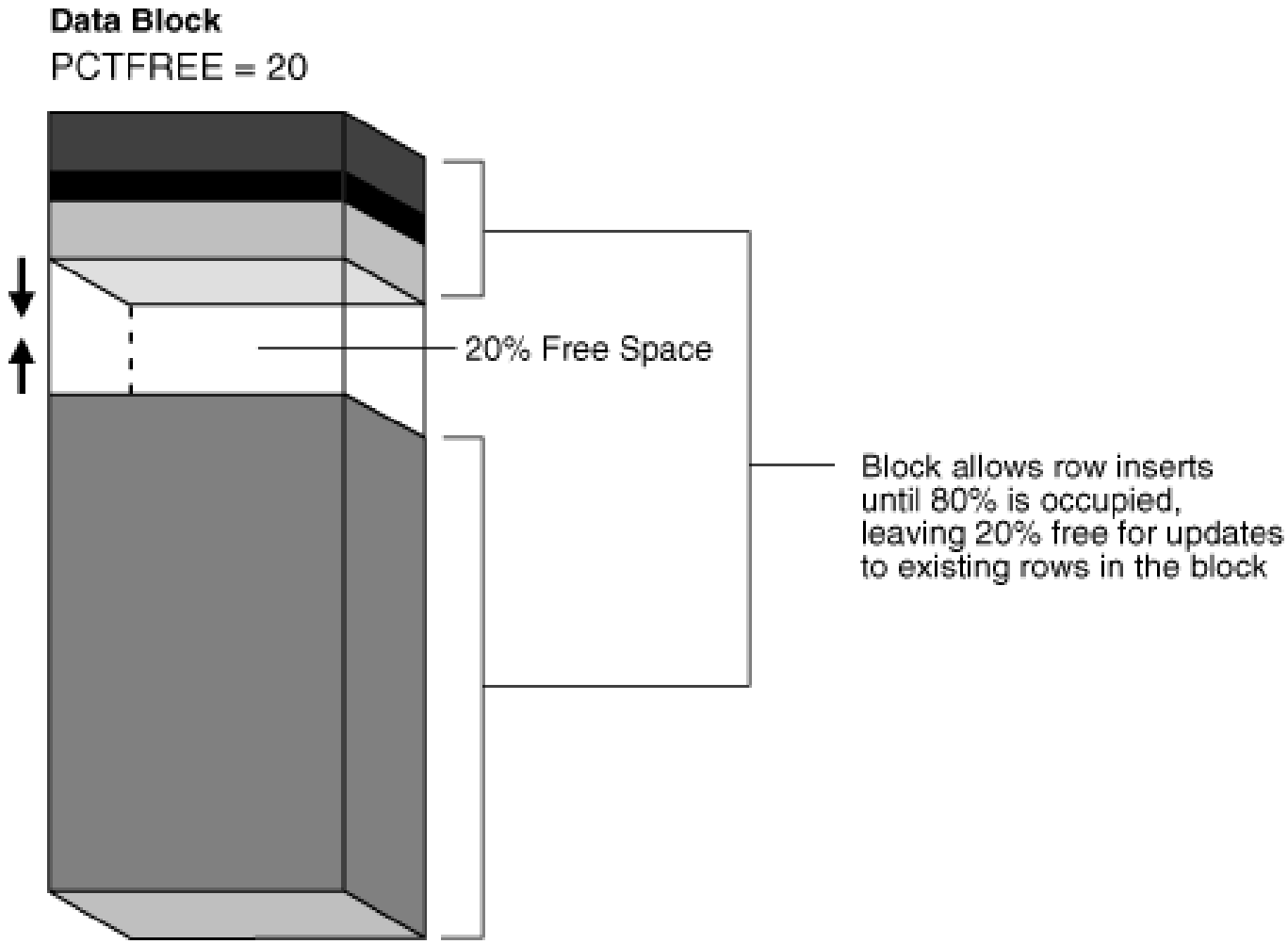
Blokkok (Oracle) II.

- A fejléc általános adatokat tárol, mint például a blokk címe, annak a szegmensnek a típusa (adat, index stb.), amihez a blokkban tárolt adat tartozik stb.
- A tábla könyvtár arról a tábláról tartalmaz információkat, amelyeknek sorai a blokkban tárolódnak.
- A rekord könyvtár (row directory) a blokkban található sorokról tárol információkat, például a címüket. Ha törlődnek a blokk sorai, az Oracle nem szabadítja fel ezt a területet. Csak akkor hasznosítja újra, amikor új sorok kerülnek a blokkba.
- A rekord adat rész tartalmazza magukat a sorokat. A sorok „átlóghatnak” más adatblokkokba.
- A szabad területet módosításnál, illetve új sorok beszúrásánál használja a rendszer. A tranzakciók a sorok *lock-olásánál* (zárolásánál) szintén használják ezt a területet.

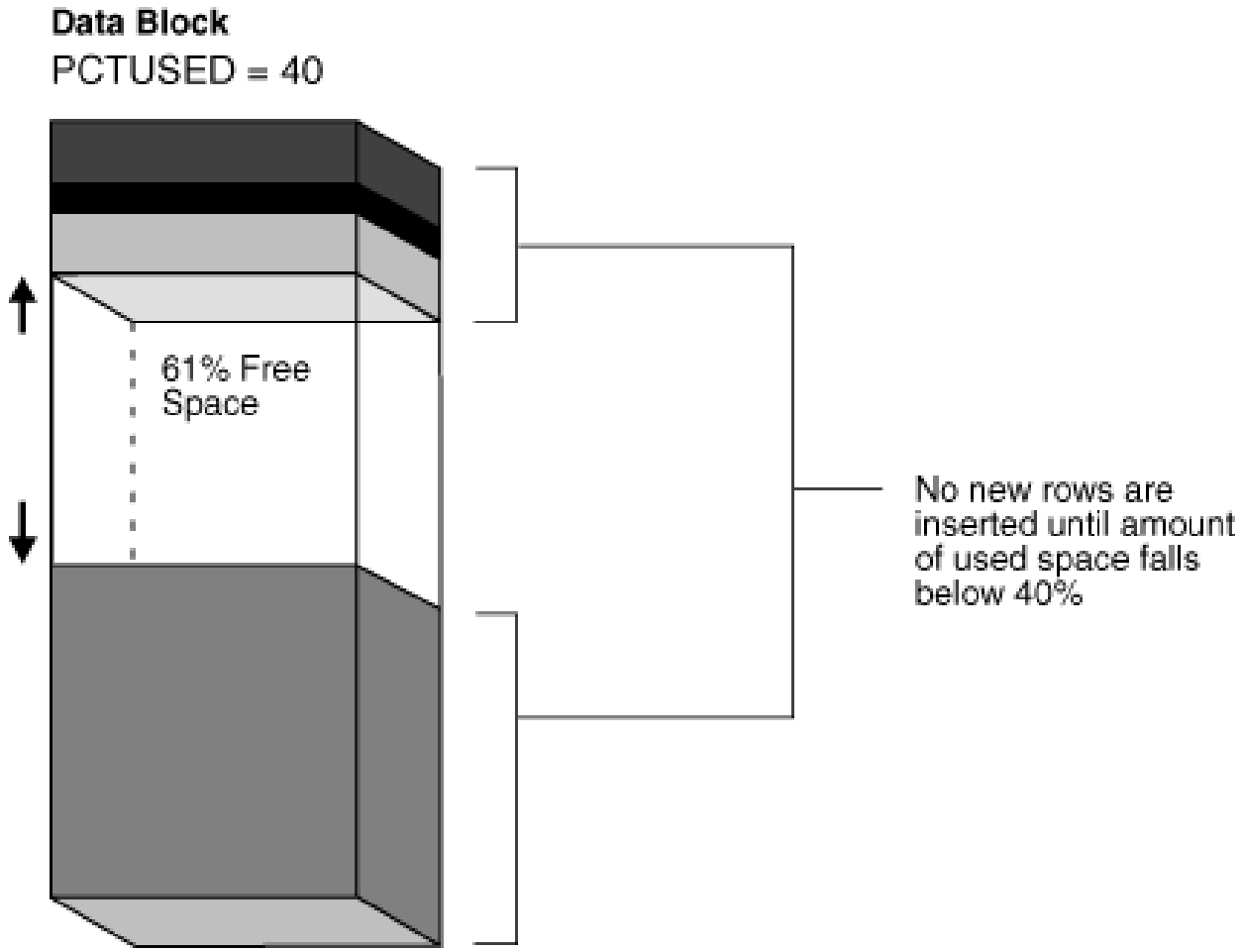
Sorok láncolása és vándorlása

- Ha egy sor nem fér be egy adatblokkba, például médiafájlok esetén, az Oracle a szegmenshez tartozó láncolt adatblokkokban tárolja az egyes részeket.
- Ha egy sor módosítás hatására túl nagyra nő, s már a szabad területen sem fér el, akkor a rendszer az egész sort egy új adatblokkba mozgatja. Ezt nevezik sorok vándorlásának (migrating).
- Az eredeti helyen csak a sor új helyének címe tárolódik. A vándorlás növelheti az I/O műveletek számát.

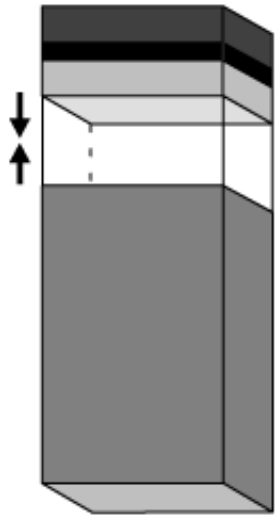
PCTFREE paraméter



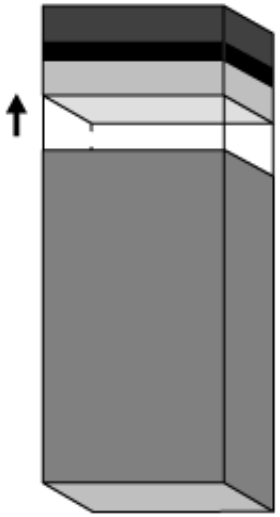
PCTUSED paraméter



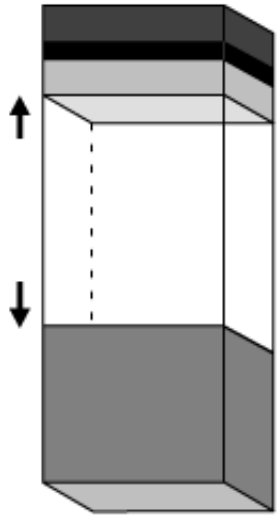
Data Block
PCTFREE = 20, PCTUSED = 40



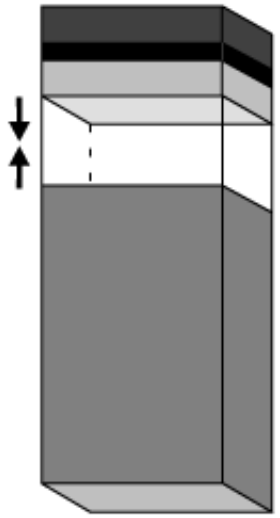
1 Rows are inserted up to 80% only, because PCTFREE specifies that 20% of the block must remain open for updates of existing rows.



2 Updates to exisiting rows use the free space reserved in the block. No new rows can be inserted into the block until the amount of used space is 39% or less.

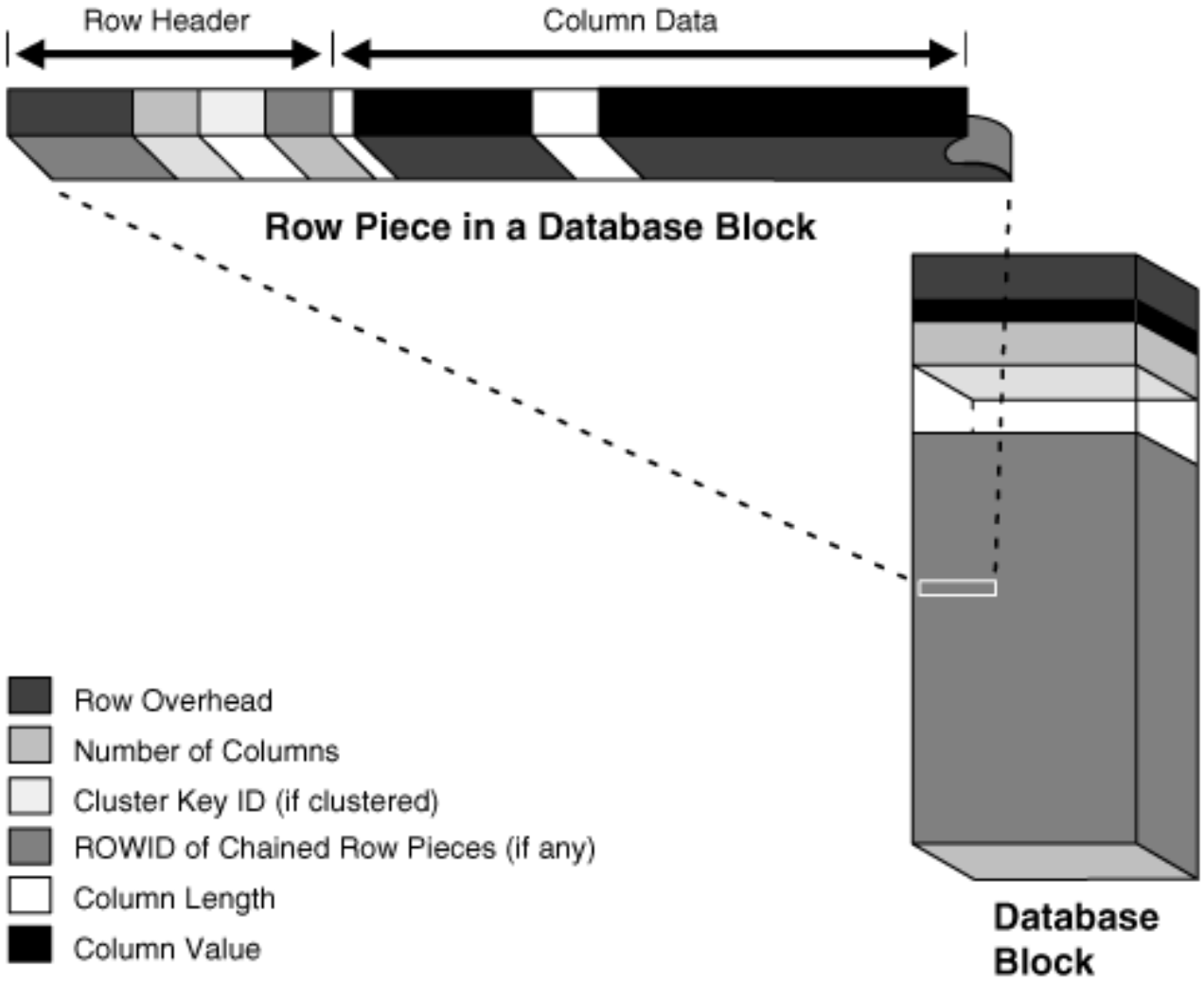


3 After the amount of used space falls below 40%, new rows can again be inserted into this block.



4 Rows are inserted up to 80% only, because PCTFREE specifies that 20% of the block must remain open for updates of existing rows. This cycle continues . . .

Sorok formátuma és mérete (Oracle) I.



Sorok formátuma és mérete II.

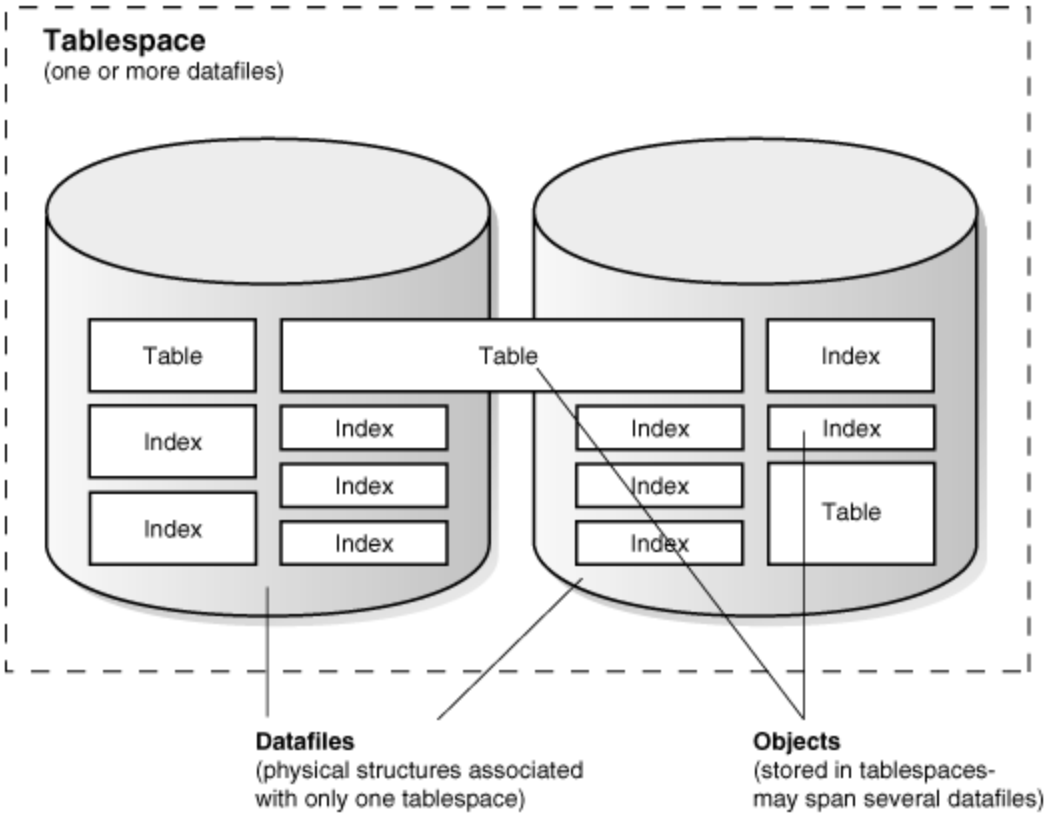
- Egy rekord legfeljebb 255 mezőt tartalmazhat. Ha ennél több mező szerepel, a rendszer a maradékot igyekszik ugyanabban a blokkban láncolva elhelyezni (infra-block chaining).
- Egy-egy rekord fejléce a sorok oszlopairól, az egymáshoz láncolt rekordtöredékekről (row pieces), klaszter esetén a klaszter kulcsáról tárol információkat.
- Ha a teljes sort tartalmazza a blokk, a fejléc legalább 3 byte hosszú.
- Ezek után minden egyes oszlopnál tárolódik az oszlop hossza (ha 255 byte-nál rövidebb 1 byte-on, ha hosszabb 3 byte-on) és a mező értéke.
- Ha változó hosszúságú adatról van szó, az adat által elfoglalt hely a változtatásoknak megfelelően módosul.
- NULL érték esetén az Oracle csak az oszlop hosszát tárolja (zero).
- A sor végén szereplő NULL értékek esetén még az oszlop hossza sem tárolódik. Ezt érdemes figyelembe venni a táblák attribútumainak felsorolásakor a CREATE utasításban.

ROWID

- A ROWID a sorokat az elhelyezkedése (index-szervezett táblák) vagy címe alapján azonosítja.
- Egy sor mindaddig megőrzi azonosítóját, míg nem törlődik, ezért hasznos lehet SELECT, UPDATE, DELETE utasításokban. A ROWID-n történő hivatkozás a sorok elérésének leggyorsabb módja.
- Az indexekben a kulcsérték(ek) mellett szintén a ROWID tárolódik.
- A kiterjesztett ROWID (extended ROWID) formátuma:

  OOOOOOFFFBBBBBBRRR, itt:

  - OOOOOO: az adatobjektum száma, ami azonosítja a megfelelő szegmenst. Azoknak objektumoknak, amelyek ugyanahhoz a szegmenshez tartoznak, ugyanaz az azonosítója.
  - FFF: a (ORACLE) az *adatállomány* relatív száma táblaterületen belül.
  - BBBBBB: az adatblokk azonosítója az *adatállományon* belül.
  - RRR: a blokkon belüli sor.


Tablespace
(one or more datafiles)
Table
Index
Index
Table
Index
Index
Index
Index
Index
Index
Index
Table
Datafiles
(physical structures associated
with only one tablespace)
Objects
(stored in tablespaces-
may span several datafiles)

Tablespaces, Datafiles,

- Databases, tablespaces, and datafiles are closely related, but they have important differences:
- An Oracle database consists of one or more logical storage units called tablespaces, which collectively store all of the database's data.
- Each tablespace in an Oracle database consists of one or more files called datafiles, which are physical structures that conform to the operating system in which Oracle is running.
- A database's data is collectively stored in the datafiles that constitute each tablespace of the database. For example, the simplest Oracle database would have one tablespace and one datafile. Another database can have three tablespaces, each consisting of two datafiles (for a total of six datafiles).

- Az adatbázis, a táblaterület és az adatállomány (file) fogalma nagyon közel áll egymáshoz;
- Egy ORACLE adatbázis egy vagy néhány logikai tároló egységből áll, amelyeket táblaterületnek hívnak, és amelyek összessége tárolja el a teljes adatbázist.
- Mindegyik táblaterület egy vagy néhány adatállományból áll, amelyeket fizikailag úgy szerveznek, hogy illeszkedjenek ahhoz az operációs rendszerhez, amelyen az ORACLE adatbázis fut.
- Az adatbázis összes adatát az olyan adatállományok tárolják, amelyek egy-egy táblaterületbe tartozva annak alkotórészei. A legegyszerűbb ORACLE adatbázis egy táblaterületből és egy adatállományból állna.
- Egy másik példában, azaz adatbázis állhatna három táblaterületből, amelyek mindegyike
- két adatállományból. (azaz összesen hat adatállományból).