

## Ellenőrző kérdések

### 1. Kis dolgozat kérdései

#### 1. Mit hívunk statikus, mit dinamikus adatbázisnak? (1 pont)

Egy statikus adatbázis esetében ritkábban fordulnak elő módosítások, a lekérdezések gyorsasága fontosabb. Dinamikus adatbázis esetén gyakran hajtunk végre módosítást, lekérdezést ritkán végzünk.

#### 2. Fogalmazzunk meg 3 célt, amire az indexelés kiválasztásánál figyelni kell! (1 pont)

Gyors lekérdezés, gyors adatmódosítás, minél kisebb tárolási terület.

#### 3. Mit tételezünk fel, mivel arányos a beolvasás, kiírás költsége? (1 pont)

A beolvasás és kiírás költsége arányos a háttértároló és a memória között mozgatott blokkok számával.

#### 4. Adjuk meg az alábbi paraméterek jelentését! $I$ , $b$ , $B$ , $T$ , $bf$ , $M$ , $I(A)$ (7 pont)

A paramétereket az indexelés költségének méréséhez vezettük be:

$I$  – rekordméret

$b$  – blokkméret (bájtokban)

$B$  – a fájl mérete blokkokban ( $B = \lceil T/bf \rceil$ )

$T$  – (tuple) rekordok száma

$bf$  – blokkolási faktor (mennyi rekord fér el egy blokkban:  $bf = \lfloor b/I \rfloor$  (alsó egészrész))

$M$  – memória mérete blokkokban

$I(A)$  – képméret, az  $A$  oszlopban szereplő különböző értékek száma:

$$I(A) = \left| \prod_A (R) \right|$$

#### 5. Adjuk meg $R \times S$ méretét blokkokban kifejezve! (2 pont)

$$\begin{aligned} B(R \times S) &= (T(R) * T(S)) * \frac{I(R) + I(S)}{b} = \\ &= \left( T(S) * T(R) * \frac{I(R)}{b} \right) + \left( T(R) * T(S) * \frac{I(S)}{B} \right) = \\ &= T(S) * B(R) + T(R) * B(S). \end{aligned}$$

#### 6. Mit jelent az egyenletességi feltétel? (1 pont)

Fel szoktuk tenni, hogy az  $A=A$  feltételnek eleget tevő rekordokból nagyjából egyforma számú rekord szerepel. Ezt nevezzük egyenletességi feltételnek.

#### 7. Mekkora adategységet olvas az író-olvasó fej? (1 pont)

Az író-olvasó fej nagyobb adategységeket (blokkokat) olvas be.

#### 8. Mitől függhet a blokkméret? (1 pont)

A blokkméret függhet az operációs rendszertől, hardvertől, adatbázis-kezelőtől.

#### 9. Egyenletességi feltétel esetén hány blokkból áll a $\sigma_{A=a}(R)$ lekérdezés eredménye? (1 pont)

$$B(\sigma_{A=a}(R)) = B(R) / I(A)$$

**10. Soroljunk fel legalább 7 különböző fájlsszervezési módszert? (7 pont)**

- kupac (heap)
- hasító index (hash)
- rendezett állomány
- elsődleges index (ritka index)
- másodlagos index (sűrű index)
- többszintű index
- B<sup>+</sup>-fa, B<sup>\*</sup>-fa

**11. Kupac szervezés esetén mennyi a keresés költsége legrosszabb esetben? (1 pont)**

Kupac szervezés esetén legrosszabb esetben a keresési költsége B (tárméret).

**12. Kupac szervezés esetén mennyi a beszúrás költsége? (1 pont)**

- utolsó blokkba tesszük a rekordot, 1 olvasás + 1 írás
- módosítás: 1 keresés + 1 írás
- törlés: 1 keresés + 1 írás

**13. Mit mond meg a  $h(x)$  hasító függvény értéke? (1 pont)**

A  $h(x)$  hasító függvény értéke megmondja, melyik kosárba tartozik a rekord, ha  $x$  volt az indexmező értéke a rekordban.

**14. Mikor jó egy hasító függvény és ilyenkor milyen hosszúak a blokkláncok? (2 pont)**

Egy hasító függvény akkor jó, ha nagyjából egyforma hosszú blokkláncok keletkeznek, azaz egyenletesen sorolja be a rekordokat. Jó hasító függvény esetén a blokklánc  $B/K$  blokkból áll.

**15. Mennyi a  $\sigma_{A=a}(R)$  lekérdezés keresési költsége jó hasító index esetén? (1 pont)**

Legrosszabb esetben  $B/K$ .

**16. Ha túl nagynak választjuk a  $K$ -t hasításkor, akkor ez milyen problémát okozhat? (1 pont)**

Nagy  $K$  esetén sok olyan blokklánc lehet, amely egy blokkból fog állni és a blokkban is csak 1 rekord lesz. Ekkor a keresési idő: 1 blokkbeolvasás, de  $B$  helyett  $T$  számú blokkban tároljuk az adatokat.

**17. Milyen keresésre nem jó a hasító indexelés? (1 pont)**

Intervallumos ( $a < A < b$ ) típusú keresésre nem jó.

**18. Mit jelent a dinamikus hasító indexelés és milyen két fajtáját ismerjük? (3 pont)**

Olyan hasító indexelés, ahol előre nem rögzítjük a kosarak számát, a kosarak száma beszúráskor, törléskor változhat. Két fajtája van: kiterjeszthető és lineáris.

**19. Kiterjeszthető hasítás esetén a  $h(K)$  érték alapján melyik kosárba kerül a rekord? (2 pont)**

A  $h(K)$   $k$  hosszú kódnak vegyük az  $i$  hosszú elejét, és azt kosarat, amelynek kódja a  $h(K)$  kezdő szelete. Ha van hely a kosárban, tegyük bele a rekordot, ha nincs, akkor nyissunk egy új kosarat, és a következő bit alapján osszuk ketté a telített kosár rekordjait. Ha ez a bit mindegyikre megegyezik, akkor a következő bitet vesszük a szétosztáshoz, és így tovább.

**20. Milyen probléma keletkezhet kiterjeszthető hasító index esetén és mi rá a megoldás? (2 pont)**

Ha az új sorok hasító értékének eleje sok bitben megegyezik, akkor hosszú ágak keletkeznek, nem lesz kiegyensúlyozva a fa. Tegyük teljessé a bináris gráfot. A gráfot egy tömbbel ábrázoljuk. Ekkor minden kosár azonos szinten lesz, de közös blokkjai is lehetnek a kosaraknak. Túlszordulás esetén a kosarak száma duplázódik.

**21. Lineáris hasító index esetén mikor nyitunk meg új kosarat? (1 pont)**

Akkor nyitunk új kosarat, ha egy előre megadott értéket elér a kosarakra jutó átlagos rekordszám.

**22. Lineáris hasító index esetén a  $h(K)$  érték alapján melyik kosárba kerül a rekord? (2 pont)**

Ha  $n$  kosarunk van, akkor a hasító függvény értékének utolsó  $\log(n)$  bitjével megegyező sorszámú kosárba tesszük, feltéve, ha van benne hely. Ha nincs, akkor hozzáláncolunk egy új blokkot és abba tesszük. Ha nincs megfelelő sorszámú kosár, akkor abba a sorszámú kosárba tesszük, amely csak az első bitjében különbözik a keresett sorszámtól.

**23. Rendezett állomány esetén adjuk meg a bináris (logaritmikus) keresés lépéseit! (4 pont)**

- beolvassuk a középső blokkot
- ha nincs benne az  $A=a$  értékű rekord, eldöntjük, hogy a blokklánc második felében, vagy az első felében szerepelhet-e egyáltalán
- beolvassuk a felezett blokklánc középső blokkját
- addig folytatjuk, amíg megtaláljuk a rekordot, vagy a vizsgálandó maradék blokklánc már csak 1 blokkból áll.

**24. Mennyi a keresési költség rendezett mező esetében? (1 pont)**

$\log_2(B)$

**25. Mennyi a keresési költség rendezett mező esetében, ha gyűjtő blokkokat is használunk? (1 pont)**

Ha  $G$  a gyűjtő mérete, akkor az összköltség:  $\log_2(B - G) + G$ .

**26. Mennyi a keresési költség rendezett mező esetében, ha minden blokkot félig üresen hagyunk? (1 pont)**

$\log_2(2 * B) = 1 + \log_2(B)$

**27. Milyen mindig az indexrekord szerkezete? (1 pont)**

Az indexrekord szerkezete  $(a,p)$ , ahol  $a$  egy érték az indexelt oszlopban,  $p$  egy blokkmutató, arra a blokkra mutat, amelyben az  $A=a$  értékű rekordot találjuk.

**28. Adjuk meg az elsődleges index 5 jellemzőjét! (5 pont)**

- főfájl is rendezett
- csak 1 elsődleges indexet lehet megadni
- elég a főfájl minden blokkjának legkisebb rekordjához készíteni indexrekordot
- indexrekordok száma:  $T(I) = B$  (ritka index)
- indexrekordokból sokkal több fér egy blokkba, mint a főfájl rekordjaiból.  $bf(I) \gg bf$ , azaz az indexfájl sokkal kisebb rendezett fájl mint a főfájl:  $B(I) = B / bf(I) \ll B = T / bf$ .

**29. Mit hívunk fedőértéknek? (1 pont)**

Az indexfájlban nem szerepel minden érték, ezért csak fedő értéket kereshetünk, a legnagyobb olyan indexértéket, amely a keresett értéknél kisebb vagy egyenlő

**30. Mennyi a keresési költség elsődleges index esetén? (1 pont)**

$$1 + \log_2(B(I))$$

**31. Adjuk meg a másodlagos index 5 jellemzőjét! (5 pont)**

- főfájl rendezetlen (az indexfájl mindig rendezett)
- több másodlagos indexet is meg lehet adni
- a főfájl minden rekordjához kell készíteni indexrekordot
- indexrekordok száma:  $T(I) = T$  (sűrű index)
- indexrekordból sokkal több fér egy blokkba, mint a főfájl rekordjaiból:  $bf(I) \gg bf$ , azaz az indexfájl sokkal kisebb rendezett fájl mint a főfájl:  $B(I) = T / bf(I) \ll B = T / bf$ .

**32. Hogyan keresünk a másodlagos indexben és mennyi a keresés költsége? (5 pont)**

- az indexben keresés az index rendezettsége miatt bináris kereséssel történik:  $\log_2(B(I))$
- a talált indexrekordban szereplő blokkmutatónak megfelelő blokkot még be kell olvasni
- $1 + \log_2(B(I)) \ll \log_2(B)$  (rendezett eset)
- az elsődleges indexnél rosszabb a keresési idő, mert több az indexrekord.

**33. Mit hívunk klaszterszervezésű táblának? (1 pont)**

Egy tábla esetén egy A oszlopra az azonos A-értékű sorok fizikailag egymás után blokkokban helyezkednek el.

**34. Mit hívunk klaszterindexnek? (1 pont)**

Klaszterszervezésű fájl esetén index az A oszlopra.

**35. Mikor mondjuk, hogy 2 tábla klaszterszervezésű? (1 pont)**

Ha a közös oszlopokon egyező sorok egy blokkban, vagy fizikailag egymás utáni blokkban helyezkednek el.