Adatbázisok vizsga

2005. február 1.

A feladatok különböző nehézségűek, mindegyiknél meg van adva, hogy hány pontot ér. Összesen 60 pontot lehet szerezni, a ketteshez/az aláíráshoz 20 pont kell.
INDOKLÁS NÉLKÜLI MEGOLDÁSÉRT NEM JÁR PONT!
Jó munkát!

---

1. **(10 pont)** A `Jegyek(neptun, tárgy, jegy)` reláció azt tartalmazza, hogy melyik hallgató melyik tárgyból milyen jegyet kapott. Tegyük fel, hogy egy hallgató egy tárgyból csak egy jegyet kapott (akkor is ha az 1-es), minden hallgató kapott jegyet valamilyen tárgyból és minden jegy (1,2,3,4,5) előfordul a relációban. Fejezd ki relációs algebrával azt a relációt, amely az olyan hallgatókat tartalmazza, akik nem kaptak valamilyen jegyből egyet sem! (Tehát például ha valaki kapott 2-es, 3-as, 4-es, 5-ös jegyet valamilyen tárgyakból, de semmiből sem kapott 1-est, akkor benne kell lennie a relációban. De ha 1-est is kapott valamiből, akkor nem.)

   Megoldás:

   $$\pi_{neptun}\Big(\big(\pi_{neptun}(Jegyek) \times \pi_{jegy}(Jegyek)\big) - \pi_{neptun,\ jegy}(Jegyek)\Big)$$

   $\times$ (5p), különbség (3p)
   vagy

   $$\pi_{neptun}(Jegyek) - \pi_{n1}(\sigma_{n1=n2=n3=n4=n5 \wedge j1 \neq j2 \wedge j1 \neq j3 \wedge \ldots j4 \neq j5}(Jegyek \times Jegyek \times Jegyek \times Jegyek \times Jegyek))$$

   (3p) az ötös szorzat, (1p) a nevek egyenlősége, (4p) a jól leírt feltétel: semelyik 2 jegy nem egyezik
   vagy

   $$\cup_{i=1}^{5}\left[\pi_{neptun}(Jegyek) - \pi_{neptun}\left(\sigma_{jegy='i'}(Jegyek)\right)\right]$$

2. **(8 pont)** Levezethetőek-e az alábbi szabályok az Armstrong-axiómák segítségével? Ha igen, akkor vezesd is le, ha nem akkor ezt bizonyítsd be!

   a) $X \to Y,\ Z \to W \vdash XZ \to YW$

   b) $XY \to Z,\ Z \to X \vdash Z \to Y$

   Megoldás:
   a) Igen: $X \to Y$ (adott), $XZ \to YZ$ (kiegészítés), $Z \to W$ (adott), $YZ \to YW$ (kiegészítés), $XZ \to YW$ (tranz.) (4p)
   b) Nem. Ellenpélda, két sor $r = \{(x_1, y_1, z_1), (x_1, y_2, z_1)\}$ (4p)

3. **(12 pont)** Egy állatkert nyilvántartását kell megterveznünk. Tárolnunk kell az állatok, a kifutók, a gondozók adatait és az etetések beosztását. A nyilvántartás létrehozásához SQL-ben a következő utasításokkal adtuk meg a táblákat:

```
CREATE TABLE Kifutó (
       kifutóID NUMBER(6) )
CREATE TABLE Állat (
       állatID NUMBER(6),
       kifutója NUMBER(6) NOT NULL REFERENCES Kifutó(kifutóID),
       kaja VARCHAR(20) NOT NULL,
       kajamennyiség NUMBER(5) NOT NULL)
CREATE TABLE Gondozó (
       gondozóID NUMBER(9),
       gondozónév VARCHAR(20) NOT NULL)
CREATE TABLE Etetés (
       gondozóID NUMBER(9) NOT NULL REFERENCES Gondozó(GondozóID),
       kifutóID NUMBER(6) REFERENCES Kifutó,
       műszak NUMBER(2), CHECK(műszak < 17 AND műszak >= 0))
```

Minden állatnak van egy féle étele (`kaja`), amiből egy megadott mennyiséget eszik naponta (`kajamennyiség` kg-ban). A többi attribútum értelemszerű. Egészítsük ki a fenti utasításokat úgy, hogy tükrözzék a következő feltételeket:

- Egy gondozó, egy műszakban, egy kifutóban tud etetni és egy nap legfeljebb 15 műszakba lehet beosztva.
- Egy kifutóban, egy műszakban csak egy gondozó etet.
- Reggel, az első etetéskor egy gondozó odaviszi a kifutóhoz a kifutóban levő állatok összes, egész napra szükséges kajáját. Ezért ez az összsúly nem lehet több, mint 25 kg.
- Minden táblának legyen életszerűen kulcsa.

Megoldás:

CREATE TABLE Kifutó (
kifutóID NUMBER(6) ***PRIMARY KEY*** $\Longrightarrow$ *(1p))*
CREATE TABLE Állat (
állatID NUMBER(6) ***PRIMARY KEY,*** $\Longrightarrow$ *(1p)*
kifutója NUMBER(6) NOT NULL REFERENCES Kifutó(kifutóID),
kaja VARCHAR(20) NOT NULL,
kajamennyiség NUMBER(5) NOT NULL)
CHECK(25>=
(SELECT SUM(Állat.kajamennyiség)
FROM Állat
WHERE Állat.kifutója = kifutója)) $\Longrightarrow$ *(3p)*
CREATE TABLE Gondozó (
gondozóID NUMBER(9) ***PRIMARY KEY,*** $\Longrightarrow$ *(1p)*
gondozónév VARCHAR(20) NOT NULL,
CHECK(15>=
(SELECT COUNT(*)
FROM Etetés
WHERE Etetés.gondozóID = gondozóID)) $\Longrightarrow$ *(3p)*
CREATE TABLE Etetés (
gondozóID NUMBER(9) NOT NULL REFERENCES Gondozó(GondozóID),
kifutóID NUMBER(6) REFERENCES Kifutó,
műszak NUMBER(2), CHECK(műszak < 17 AND műszak >= 0)
PRIMARY KEY(kifutóID, műszak), $\Longrightarrow$ *(1p)*
UNIQUE(gondozóID, műszak) *)* $\Longrightarrow$ *(2p)*

4. **(8 pont)** Adottak az $R(A,B), S(B,C)$ és $Q(C,D)$ relációk. A rekordok száma rendre $6\cdot10^4, 6\cdot10^5, 6\cdot10^3$, egy lapra 10 rekord fér mindegyik reláció esetén. A belső memória mérete 21 lap. R-ben kulcs B, S-ben C, Q-ban pedig D. Ki szeretnénk számítani a $R \bowtie S \bowtie Q$ relációt. Az egyik lehetőség, hogy előbb kiszámítjuk $R \bowtie S$-t, az eredményt kiírjuk a háttértárra, majd ezt illesztjük Q-val. Vegyük azonban figyelembe, hogy ebből az ideiglenes relációból egy lapra csak 6 rekord fér el. A másik lehetőség, hogy előbb $S \bowtie Q$-t számítjuk ki és azt illesztjük R-el. Itt is tudjuk, ennek az ideiglenes relációnak egy lapjára csak 6 rekord fér el. Legfeljebb hány lap**olvasás** szükséges a két esetben (külön-külön)?

   Megoldás: *Ha $(R \bowtie S) \bowtie Q$ sorrendben csináljuk: beágyazott ciklussal, először R 20 blokkját beolvasva, soronként S-et, ...:* $6\cdot10^3+6\cdot10^3\cdot6\cdot10^4/20 = 6\cdot10^3+18\cdot10^6$.*(1p) Az eredménynek legfeljebb annyi sora van mint S-nek, azaz* $6\cdot10^5$.*(1p) Utána megint beágyazott ciklus, először Q-t beolvasva, majd az előzőleg kiírtat soronként:* $6\cdot10^2+6\cdot10^2\cdot6\cdot10^5/(6\cdot20) = 6\cdot10^2+3\cdot10^6$. *(1p) Összesen* $21\cdot10^6+6\cdot10^3+6\cdot10^2 = 21006600$ *(1p)*
   Ha $R \bowtie (S \bowtie Q)$ sorrendben csináljuk: beágyazott ciklussal, először Q 20 blokkját beolvasva, soronként S-et, ... $6\cdot10^2+6\cdot10^2\cdot6\cdot10^4/20 = 18\cdot10^5+6\cdot10^2$. *(1p) Az eredménynek legfeljebb annyi sora van mint Q-nak, azaz* $6\cdot10^3$. *(1p) Utána megint beágyazott ciklus, először a kiírtat beolvasva, majd az R-et soronként:* $10^3+10^3\cdot6\cdot10^3/20 = 3\cdot10^5+10^3$. *(1p) Összesen* $21\cdot10^5+10^3+6\cdot10^2 = 2101600$ *művelet. (1p) (Tehát ezt érdemesebb csinálni.)*
   Aki a számolásnál végig $|R|+|R|\cdot|S|/M$ *helyett csak* $|R|\cdot|S|/M$*-el számol, az is megkapja a teljes pontszámot.*

5. **(12 pont)** Tegyük fel, hogy az (R,F) relációs sémának minimális fedése $G = \{X_1 \to A_1, \ldots, X_k \to A_k\}$ és Y egy kulcs a sémában. Bizonyítsd be, hogy a $\rho = (Y, X_1A_1, \ldots, X_kA_k)$ felbontás minden tagja 3NF-ben van!

   Megoldás: $R_0 = Y$ ***3NF:*** *R_0-ban nincs nemtriviális függés, mert különben X nem lenne kulcs, csak szuperkulcs* $\Longrightarrow$ *R_0 BCNF* $\Longrightarrow$ *3NF (4p)*

Többi R_i ***is 3NF:*** *tegyük fel, hogy nem az* $\Longrightarrow \exists\, U \to B$ *nemtriviális függés, hogy* U *nem szuperkulcs* R_i*-ben és* B *nem prímattribútum* R_i*-ben.*
Ha $B = A_i$*, akkor* $U \subseteq X_i$*, de* $U \neq X_i$*, hiszen akkor* U *szuperkulcs lenne* R_i*-ben.* $\Longrightarrow U \subset X_i$ $\Longrightarrow X_i \to A_i$ *baloldala csökkenthető* G*-ben* U*-ra. Ellentmondás, mert akkor* G *nem volt minimális fedés.(4p)*
Ha $B \neq A_i$ $\Longrightarrow B \in X_i$ *és* B *nem prím* R_i*-ben* $\Longrightarrow X_i$ *nem kulcs* R_i*-ben (de szuperkulcs)* $\Longrightarrow \exists\, Y \subsetneq X_i$ *kulcs* R_i*-ben* $\Longrightarrow Y \to A_i$ *fennáll* $\Longrightarrow X_i \to A_i$ *baloldala csökkenthető* G*-ben* Y*-ra, megint csak ellentmondás. (4p)*

6. **(10 pont)** Hogyan működik az időbélyeges tranzakciókezelés? Milyen ütemezéseket enged lefutni? Írd le, hogy ha egy tranzakció írni vagy olvasni akar egy adatelemet, akkor különböző esetekben mit csinál a tranzakciókezelő!

Megoldás: ***Fő elv****: minden tranzakciónak van egy időbélyege:* $t(T_i)$ *a* T_i *tranzakcióé. Az időbélyegek egyediek, növő sorrendben adja ki őket az ütemező, ahogy indulnak a tranzakciók. (1p)*
Az ütemező célja*: az időbélyegek növő sorrendjéhez tartozó soros ütemezéssel azonos hatású ütemezést enged csak lefutni, minden olyan kérést letilt (és a megfelelő tranzakciót ABORT-álja), ami ez ellen tesz. (3p)*
Megkülönböztetünk írás és olvasás műveletet, továbbá minden A *adatelemhez hozzárendelünk egy olvasási és egy írási időt* $(r(A), w(A))$*, melyek jelentése:*

- *r(A)= a legnagyobb olyan* $t(T_i)$*, amire igaz, ahogy* T_i *olvasta már A-t (0.5p)*
- *w(A)= a legnagyobb olyan* $t(T_i)$*, amire igaz, ahogy* T_i *írta már A-t (0.5p)*

(a) *Ha* T *olvasná A-t, de* $t(T) < w(A)$*, akkor ABORT* T *(0.5p)*

(b) *Ha* T *írná A-t, de* $t(T) < r(A)$*, akkor ABORT* T *(0.5p)*

(c) *Ha* T *olvasná A-t,* $t(T) \geq w(A)$*, de* $t(T) < r(A)$*, akkor* T *olvashatja A-t és* $r(A)$ *marad, ami volt és persze* $w(A)$ *is (1p)*

(d) *Ha* T *írná A-t,* $t(T) \geq r(A)$*, de* $t(T) < w(A)$*, akkor nem történik meg az írás, de nem is lesz ABORT* T *se és* $r(A)$ *és* $w(A)$ *marad, ami volt (1p)*

(e) *Ha* T *olvasná vagy írná A-t, és* $t(T) \geq w(A)$ *és* $t(T) \geq r(A)$*, akkor engedjük és* $r(A)$ *illetve* $w(A)$ *változik, attól függően, hogy írás vagy olvasás történt (1p)*