

Adatbázisok vizsga

2005. január 17.

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!

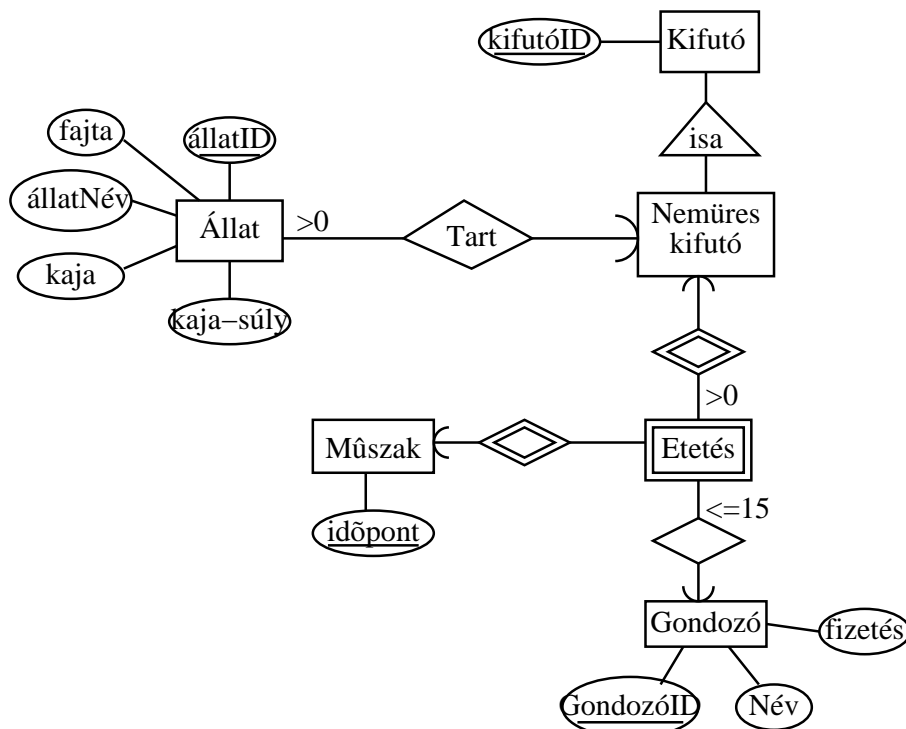
Jó munkát!

1. (10 pont) Egy állatkert nyilvántartását kell megterveznünk. Tárolnunk kell az állatok, a kifutók, a gondozók adatait és az etetések beosztását, a következő feltételek mellett:

- Egy kifutóban több (akár különböző fajtájú) állat is lehet, de lehetnek üresek is. A kifutóknak van azonosítójuk.
- Minden állatnak van azonosítója, neve, fajtája és egy féle étele, amiből egy megadott mennyiséget eszik naponta.
- A gondozóknak nyilvántartjuk az azonosítóját, nevét és fizetését.
- A munkanap reggel 8-tól délután 5-ig tart és 18 félórás műszakra van osztva.
- Minden nem üres kifutóhoz hozzá van rendelve legalább egy műszak, amikor az állatok enni kapnak.
- Egy gondozó, egy műszakban, egy kifutóban tud etetni és egy nap legfeljebb 15 műszakba lehet beosztva.
- Egy kifutóban, egy műszakban csak egy gondozó etet.

Tervezz E/K diagrammot, amiben az összes feltételt feltüntetjük!

Megoldás:



2. (12 pont) Tekintsük a következő alaprelációkat:

Űrhajós(név, bolygó, űrhajónév), Űrhajó(űrhajónév, kapitánynev), Allergia(név, virág)

A relációk jelentése:

Űrhajós: milyen nevű űrhajós melyik bolygóról származik és melyik hajón szolgál (mindenki csak egy hajón szolgál), kulcs a név;

Űrhajó: melyik űrhajón ki a kapitány (aki maga is űrhajós), kulcs az űrhajónév;

Allergia: milyen nevű űrhajós milyen virágra allergiás, a két attribútum együtt kulcs.

Add meg sorkalkulussal azon űrhajósok nevének halmazát, akik minden olyan virágra allergiásak, amelyekre a hajójuk kapitánya is allergiás.

Megoldás:

$$\{r^{(1)} \mid \exists t^{(3)}(\check{U}rhaj\acute{o}s(t) \wedge r[1] = t[1] \wedge \exists s^{(2)}(\check{U}rhaj\acute{o}s(s) \wedge t[3] = s[1] \wedge \forall u^{(2)}(\neg(Allergia(u) \wedge u[1] = s[2]) \vee (\exists v^{(2)}(Allergia(v) \wedge t[1] = v[1] \wedge u[2] = v[2])))\})\}$$

3. (12 pont) Adott a következő két reláció, melyben különféle adománygyűjtő kampányok adatait tartjuk nyilván:

Adomány(Személy, Dátum, Összeg, KampányID), Kampány(KampányID, KezdDátum, VezetőNév)

A kulcsokat aláhúzással jelöltük. Az Adomány relációban a KampányID lehet NULL is, ha valaki spontán adományozott.

Írj egy olyan SQL utasítást, amely egy olyan VIEW-t ad meg, ami kibővíti a Kampány relációt még egy attribútummal, ami azt tartalmazza, hogy az adott kampányban mennyi pénzt gyűjtöttek 2004.01.01. óta.

Megoldás:

```
CREATE VIEW
```

```
KampányB(KampányId, KezdDátum, VezetőNév, Szumma) AS
```

```
SELECT * FROM Kampány NATURAL JOIN
```

```
( SELECT KampányID, SUM(Összeg)
```

```
FROM Kampány, Adomány
```

```
WHERE Adomány.KampányID = Kampány.KampányID AND Adomány.Dátum >= '2004.01.01'
```

```
GROUP BY Kampány.KampányID)
```

```
vagy CREATE VIEW
```

```
KampányB(KampányId, KezdDátum, VezetőNév, Szumma) AS
```

```
(SELECT KampányID, KezdDátum, VezetőNév, SUM(Összeg)
```

```
FROM Kampány NATURAL JOIN Adomány
```

```
WHERE Adomány.Dátum >= '2004.01.01'
```

```
GROUP BY Kampány.KampányID, Kampány.KezdDátum, Kampány.VezetőNév)
```

4. (10 pont) Legyen F egy függéshalmaz, X, Y pedig attribútumhalmazok. Bizonyítsuk be a következő állítást: $F \vdash X \rightarrow Y$ akkor és csak akkor igaz, ha $Y \subseteq X^+(F)$

Megoldás: \Rightarrow : Tegyük fel, hogy $F \vdash X \rightarrow Y$ és legyen $A \in Y$.

$F \vdash X \rightarrow A$, hiszen vegyük $X \rightarrow Y$ levezetését és alkalmazzuk a felbontási szabályt a végén.

Definíció szerint ekkor $A \in X^+(F)$. Ez minden $A \in Y$ -ra igaz.

\Leftarrow : Legyen $Y = A_1 \dots A_k \subseteq X^+(F)$.

Így definíció szerint $\forall A_i \in Y$ -ra $F \vdash X \rightarrow A_i$.

Ekkor $X \rightarrow Y$ levezetése: vesszük az A_i -k levezetését és a végén alkalmazzuk az unió szabályt $k-1$ -szer.

5. (8 pont) Mi a ritka index? Mikor lehet használni? Hogyan kell végrehajtani egy ritka indexszel ellátott állományban keresést, beszúrást, törlést?

6. (8 pont) Tegyük fel, hogy az ütemezőbe következő sorrendben érkeznek az írási és olvasási kérések:

$$r_1(x), r_2(z), w_3(x), w_1(x), w_2(z), r_1(z), r_3(z)$$

Az ütemező egyszerű zárolási modellben zárkérésekkel biztosítja a konzisztenciát. Ha egy zárkérést nem tud azonnal teljesíteni, akkor azt a tranzakciót várakoztatja, amíg felszabadul a kívánt adatelem, ekkor viszont azonnal folytatódik az a tranzakció. A zárat csak a tranzakció befejezése után engedi fel, de akkor azonnal.

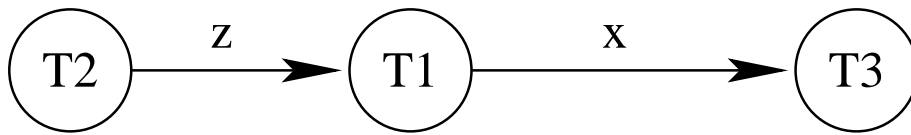
Milyen zárolási ütemezést eredményez ez? Sorosítható-e az így kapott ütemezés? Ha igen, adj egy ekvivalens soros ütemezést, ha nem bizonyítsd be.

Megoldás:

A zárolási ütemezés:

$$l_1(x), l_2(z), u_2(z), l_1(z), u_1(x), u_1(z), l_3(x), l_3(z), u_3(x), u_3(z)$$

A sorosítási gráf:



Mivel ez DAG, sorosítható. Ekvivalens soros ütemezés: t_2, t_1, t_3