Adatbázisok 1 – 06.06

1) Elméleti kifejtő kérdés:

a) Adja meg az SQL SELECT utasítás WHERE szűrési feltétele milyen elemi feltételekből és hogyan épül fel!

b) Ismertesse, hogy mit jelentenek, miért és hogyan használjuk SQL-ben a hiányzó értékeket (null) és az ismeretlen igazságértéket (unknown)!

c) Számolja ki az igazságértékét:

TRUE AND (FALSE OR NOT(UNKNOWN AND TRUE)) = ?

Táblák:

Tanulók(tazon, név, osztály)

Hiányzás(hazon, tazon, hiányzás_kezdete_dátum, hossza_napokban)

a) *(1. előadás, 45. oldal)*

Hasonlóan, mint a relációs algebra kiválasztás (σ) feltételében elemi feltételekből építkezünk, ahol **elemi feltételen két kifejezés =, <>, , <=, >= aritmetikai összehasonlítását, a theta műveletet értjük.**

b) *(1. előadás, 49-51. oldal)*

- Amikor egy aritmetikai műveletben az egyik tag **NULL**, akkor az eredmény is **NULL**.
- Amikor egy **NULL** értéket hasonlítunk össze bármely más értékkel (beleértve a NULL-t is) az összehasonlítási operátorok (=, <>, <=, >, >=) segítségével, akkor az eredmény **UNKNOWN** (ismeretlen).
- A szabályt könnyű megjegyezni, ha úgy tekintjük, hogy **TRUE = 1, FALSE = 0,** és **UNKNOWN = ½.**
- Ekkor **AND = MIN, OR = MAX, NOT(x) = 1- x.**

Példa:

TRUE AND (FALSE OR NOT(UNKNOWN)) =

MIN(1, MAX(0, (1 - ½))) =

MIN(1, MAX(0, ½)) =

MIN(1, ½) = ½ = UNKNOWN

c) *(1. előadás 52. oldal)*

TRUE AND (FALSE OR NOT(UNKNOWN AND TRUE)) = Unknown

1. **Unknown** (Unknown and True)
2. **Unknown** (Not Unknown)
3. **Unknown** (False or Unknown)
4. **Unknown** (True and Unknown)

2) Fejezze ki alap relációs algebrai kifejezéssel (csak az alapműveletek és a direkt szorzat használatával, vagyis ne használjunk kiterjesztett relációs algebrát, így összesítő függvényeket sem), hogy mikor volt a leghosszabb hiányzás és ez kinek a hiányzása volt? Az eredménytábla sémája legyen:

Hosszú_hiányzás(hiányzás_kezdete_dátum, tazon, név, osztály)

Táblák:

Tanulók(tazon, név, osztály)

Hiányzás(hazon, tazon, hiányzás_kezdete_dátum, hossza_napokban)

Megoldás (kód):

```
SELECT h.hiányzás_kezdete_dátum, h.tazon, t.név, t.osztály
FROM Hiányzás h, Tanulók t
WHERE h.tazon = t.tazon
MINUS

                                    VanHosszabb (ezeknél van hosszabb hiányzás)
SELECT h1.tazon
FROM Hiányzás h1, Hiányzás h2
WHERE h1.hossza_napokban  < h2.hossza_napokban
```

Megoldás (relációs algebra):

VanHosszabb (segédváltozó) :=

$\pi_{h1.tazon}\,(\sigma_{h1.hossza\_napokban < h1.hossza\_napokban}\,(\rho_{\ h1\ (Hiányzás)} \times \rho_{h2\ (Hiányzás)})\,)$

$\pi_{h.hiányzás\_kezdete\_dátum,\ h.tazon,\ t.név,\ t.osztály}\,(\rho_{\ h\ (Hiányzás)} \bowtie \rho_{t\ (Tanulók)}) - \mathbf{VanHosszabb}$

3) Fejezze ki SQL lekérdezéssel, hogy kik azok a tanulópárok, akik ugyanabba az osztályba járnak, és van olyan hiányzásuk, amely ugyanakkor kezdődött és ugyanannyi napig tartott. Az output sémája:

Egyszerre_hiányzók(név_1, név_2, osztály)

Táblák:

Tanulók(<u>tazon</u>, név, osztály)

Hiányzás(<u>hazon</u>, tazon, hiányzás_kezdete_dátum, hossza_napokban)

Megoldás (kód):

```
SELECT t1.név, t2.név, t1.osztály
FROM Tanulók t1, Tanulók t2
WHERE t1.név < t2.név
AND t1. hiányzás_kezdete_dátum = t2. hiányzás_kezdete_dátum
AND t1. hossza_napokban = t2. hossza_napokban
```

Megoldás (relációs algebra):

$$\pi_{t1.név,\ t2.név,\ t1.osztály}$$

$$(\sigma_{t1.név < t2.név \wedge t1.\ hiányzás\_kezdete\_dátum = t2.\ hiányzás\_kezdete\_dátum}$$

$$\wedge\ t1.\ hossza\_napokban = t2.\ hossza\_napokban$$

$$(\rho_{t1\ (Tanulók)} \times \rho_{t2\ (Tanulók)}))$$

4) Legyenek adva az R(A,B,C) és S(C,D) sémájú relációk. Írja át az alábbi SQL lekérdezéseket kiterjesztett relációs algebrai kifejezésre:

a)

```
SELECT A, AVG(D)
FROM R NATURAL JOIN S
WHERE B >=2
GROUP BY A;
```

$$\pi_{A,\ AVG(D)}\ (\gamma_{A}\ (\sigma_{B >= 2}\ (R \bowtie S)\ )\ )$$

b)

```
SELECT A
FROM R NATURAL JOIN S
GROUP BY A
HAVING AVG(D)>10;
```

$$\pi_{A}\ (\sigma_{AVG(D) > 10}(\gamma_{A}\ (R \bowtie S)\ )\ )$$

c)

```
SELECT DISTINCT A
FROM R, S
WHERE R.C = S.C AND B = 2
ORDER BY A;
```

$$\tau_{A}\ (\delta_{A}\ (\sigma_{R.C = S.C \wedge B = 2}\ (R \times S)\ )\ )$$

5) Ehhez a feladathoz adott az alábbi adatbázis séma (a kulcsokat aláhúzás jelöli)

OSZTALY (OAZON, ONEV, TELEPHELY)

DOLGOZO (DKOD, DNEV, FOGLALKOZAS, FONOKE, BELEPES, FIZETES, JUTALEK, OAZON)

Adja meg SQL lekérdezéssel, hogy az egyes osztályokon (ONEV) mennyi a fizetések összege, akikről nem tudjuk, hogy melyik osztályon dolgoznak, azokat az 'Ismeretlen' osztálynév alatt összesítsük, de csak azokat az osztályokat jelenítsük meg, ahol legalább három fő dolgozik! (ONEV, ÖSSZFIZ)

Megoldás (kód):

```
SELECT o.onev, SUM (NVL (d.jutalek, 0) + d.fizetes) AS "ÖSSZFIZ"
FROM Dolgozo d, Osztaly o
WHERE d.oazon = o.oazon
GROUP BY o.onev
HAVING COUNT(d.oazon) >= 3;
```

7)

a) Mit jelent a kulcs, a külső kulcs és a hivatkozási épség?

b) Adja meg hogyan definiáljuk az SQL DDL-ben (a megfelelő utasításnak csak a lényegi részét kell leírnia), hogyan hozzuk létre a Sörök, Söröző̈k és Felszolgál táblákat, mindhárom táblában definiálja a kulcsokat, továbbá a Felszolgál táblából adja meg a megfelelő hivatkozási épség megszorításokat is!

a) *(2. előadás, 23-28. oldal)*

- Az attribútumok egy halmaza egy **kulcsot** alkot egy relációra nézve, ha a reláció bármely előfordulásában **nincs két olyan sor**, amelyek a kulcs összes attribútumának értékein megegyeznének.
- **Kulcs és idegen kulcs (foreign key) hivatkozási épség megadása:**
- Az egyik tábla egyik oszlopában szereplő értékeknek szerepelnie kell egy másik tábla bizonyos attribútumának az értékei között.
- **A hivatkozott attribútumoknak** a másik táblában kulcsnak kell lennie! (PRIMARY KEY vagy UNIQUE)
- Példa: **Felszolgál(söröző, sör, ár)** táblára megszorítás, hogy a sör oszlopában szereplő értékek szerepeljenek a **Sörök(név, gyártó)** táblában a név oszlop értékei között.

b) *(2. előadás, 29. oldal és 1. előadás 24. oldal)*

```
CREATE TABLE Sörök (
    név     CHAR(20) PRIMARY KEY,
    gyártó CHAR(20) );

CREATE TABLE Söröző̈k (
    név CHAR(20),
    város VARCHAR2(40),
    tulaj CHAR(30),
    engedély DATE  DEFAULT SYSDATE
    );

CREATE TABLE Felszolgál (
    söröző CHAR(20),
    sör   CHAR(20) REFERENCES Sörök(név),
    ár    REAL );
```

8)

a) Adja meg a táblák tartalmát megváltoztató SQL DML utasításokat, az INSERT-nek több sor felvitelére alkalmas 2. alakját is adja meg!

b) A megfelelő SQL DML utasítással törölje a Sörök(név, gyártó) táblából azokat a söröket, amelyekhez létezik olyan sör, amit ugyanaz a cég gyártott.

a) *(5. előadás, 9., 10., 16. oldal)*

INSERT - sorok beszúrása

- ha egyetlen sort szúrunk be:
  INSERT <reláció> **INTO VALUES** (<konkrét értékek listája);
- ha több sort, egy lekérdezés eredményét visszük fel alkérdés segítségével:
  INSERT INTO <reláció> **(<alkérdés>)**;

DELETE – sorok törlése

- **DELETE FROM** <reláció>
  WHERE <feltétel>;

UPDATE – sorok komponensei értékeinek módosítása

- **UPDATE** <reláció>
  SET <attribútum értékadások listája>
  WHERE <sorokra vonatkozó feltétel>;

b) *(5. előadás, 18. oldal)*

- A Sörök(név, gyártó) táblából töröljük azokat a söröket, amelyekhez létezik olyan sör, amit ugyanaz a cég gyártott.

```
DELETE FROM Sörök s
WHERE EXISTS (
   SELECT név FROM Sörök
   WHERE gyártó = s.gyártó
      AND név <> s.név);
```

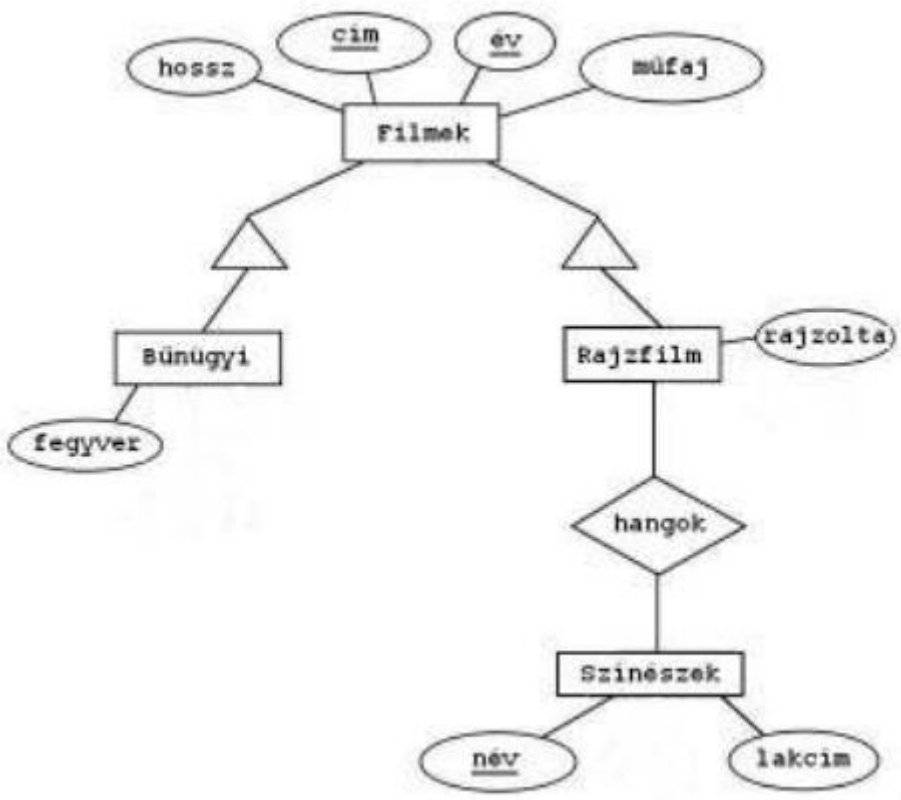
(WHERE) azok a sörök, amelyeknek ugyanaz a gyártója, mint az s éppen aktuális sorának, a nevük viszont különböző.

10)

a) Tervezés: Alakítsa át az alábbi E/K diagramot relációs adatbázis-sémává! Relációnév (attribútum-lista)

b) Ismertesse az osztályhierarchia, vagyis az egyedhalmazok öröklési „az-egy” („isa”) típusú kapcsolattal megadott hierarchia relációkká alakításának a stratégiáit, mind a háromféle átalakítási stratégiát írja le!

a) *(8. előadás, 48-49. oldal alapján)*



FILMEK (cím, év, hossz, műfaj)
BŰNÜGYI (cím, év, fegyver)
RAJZFILM (cím, év, rajzolta)
SZÍNÉSZEK (név, lakcím, cím, év)

b) *(8. előadás, 44. oldal)*

- E/R stílusban: Egy reláció minden alosztályra, de az általános osztályból csak a kulcsokat vesszük hozzá a saját attribútumokhoz.
- Objektumorientált stílusban: Egy reláció minden alosztályra, felsorolva az összes tulajdonságot, beleértve az örökölteket is.
- Nullértékek használatával: Egyetlen reláció az öröklődésben résztvevő összes osztályra. Ha egy egyed nem rendelkezik egy alosztály speciális tulajdonságával, akkor ezt az attribútumot NULL értékkel töltjük majd ki.