

## Adabáziskezelés gyakorlat – SQL

1. Tekintsük a következő alaprelációkat (a kézenfekvő értelmezéssel):

*Kedvel* (személy, sör),  
*Kapható* (söröző, sör),  
*Látogat* (személy, söröző).

Fejezze ki SQL nyelven

- azon személyek összességét, akik járnak olyan vendéglőbe, ahol tartanak olyan sört, amit ők (a személyek) kedvelnek.
- azon sörök összességét, amelyek kaphatók egy olyan sörözőben, amelynek valamilyen látogatója kedveli őket (ti. a söröket).
- \* azon sörök összességét, melyeket minden látogató kedvel azokban a sörözőkben, ahol kaphatók.
- \* azon személyek összességét, akik minden sört kedvelnek azokban a sörözőkben, melyeket látogatnak.

Megoldás:

a)

```
SELECT DISTINCT személy
FROM Kedvel NATURAL JOIN Kapható
      NATURAL JOIN Látogat
```

b)

```
SELECT DISTINCT sör
FROM Kedvel NATURAL JOIN Kapható
      NATURAL JOIN Látogat
```

c)

```
SELECT DISTINCT sör FROM Kapható AS k1
WHERE k1.sör NOT IN (
  SELECT k2.sör
  FROM Kapható AS k2 NATURAL JOIN Látogat
  WHERE személy NOT IN (
    SELECT személy FROM Kedvel
    WHERE Kedvel.sör=k2.sör))
```

vagy

```
SELECT sör FROM Kapható
EXCEPT
(SELECT sör FROM
((SELECT személy, sör FROM Kapható NATURAL JOIN Látogat)
EXCEPT
(SELECT személy, sör FROM Kedvel))
)
```

d)

```
SELECT DISTINCT személy FROM Kedvel
EXCEPT
SELECT DISTINCT személy
FROM Kapható NATURAL JOIN Látogat
WHERE sör NOT IN (
    SELECT sör FROM Kedvel
    WHERE személy=Látogat.személy)
```

6–7. sor: megmondja, hogy egy adott személy milyen söröket kedvel.

4–7. sor: megmondja, hogy egy adott személy kaphat-e az általa látogatott sörökben olyan sört, amit nem kedvel.

3–7. sor: kiválasztja azokat a személyeket, akik nem kedvelnek minden általuk kapható sört

Egész lekérdezés: a 3–7. sornak veszi a negáltját.

*Második megoldás*

```
SELECT DISTINCT személy AS sz FROM Kedvel
WHERE NOT EXISTS (
    SELECT sör FROM Kapható NATURAL JOIN Látogat
    WHERE személy=sz
    EXCEPT
    SELECT sör FROM Kedvel WHERE személy=sz)
```

A lekérdezés 4–6. sora pontosan akkor ad nem üres eredményt, ha az *sz* személy által látogatott sörökben kapható sörök halmaza nem részhalmaza az általa kedvelt sörök halmazának, azaz *sz* vásárolhat olyan sört, amit nem szeret.

## 2. Tekintsük az alábbi Csillagflotta adatbázissémát:

*Csillaghajó* (hajónév, év, faj),  
*Dolgozó* (dolgozónév, azonosító, születés),  
*Beosztás* (azonosító, hajónév, rang).

A relációk jelentése:

*Csillaghajó*: a hajó neve, gyártási éve és az, hogy melyik faj tervei alapján készült

*Dolgozó*: neve, Csillagflotta-azonosítója, mikor született;

*Beosztás*: melyik dolgozó, melyik hajón, milyen rangban dolgozik. .

Adjunk SQL lekérdezést, mely megkeresi

a) azon a dolgozók nevét, akik klingon (faj által tervezett) hajón dolgoznak.

b) azon a dolgozók nevét, akik Catherine Janeway kapitány hajóján dolgoznak.

c) a Voyager (nevű hajó) legfiatalabb dolgozójának nevét.

Megoldás:

a)

```
SELECT dolgozónév
FROM Dolgozó NATURAL JOIN Beosztás NATURAL JOIN Csillaghajó
WHERE faj = 'klington');
```

vagy

```
SELECT dolgozónév
FROM Dolgozó NATURAL JOIN Beosztás
WHERE Beosztás.hajónév IN (
    SELECT hajónév
    FROM Csillaghajó
    WHERE faj = 'klington');
```

Az alkérdés kikeresi a klington hajókat, a fő rész meg azokat a dolgozókat adja meg, akik ezeken a hajókon vannak.

vagy

b)

```
SELECT dolgozónév
FROM Dolgozó, Beosztás
WHERE Dolgozó.azonosító = Beosztás.azonosító AND
    Beosztás.hajónév IN (
    SELECT hajónév
    FROM Beosztás, Dolgozó
    WHERE Beosztás.azonosító = Dolgozó.azonosító AND
        Dolgozónév = 'Catherine Janeway' AND
        rang = 'kapitány');
```

Az alkérdés kikeresi azokat a hajókat, ahol Catherine Janeway a kapitány, a fő rész meg azokat a dolgozókat adja meg, akik ezeken a hajókon vannak.

vagy

```
SELECT dolgozónév
FROM Dolgozó AS d1, Beosztás AS b1, Dolgozó AS d2, Beosztás AS b2
WHERE d1.azonosító = b1.azonosító AND
    d2.azonosító = b2.azonosító AND
    b1.hajónév = b2.hajónév AND
    d2.Dolgozónév = 'Catherine Janeway' AND
    b2.rang = 'kapitány';
```

c)

```
SELECT dolgozónév
FROM Dolgozó NATURAL JOIN Beosztás
WHERE hajónév = 'Voyager' AND
    születés =
    (SELECT MAX(születés)
    FROM Dolgozó NATURAL JOIN Beosztás
    WHERE hajónév = 'Voyager')
```

### 3. Tekintsük az alábbi adatbázissémát:

Járat (Járatszám, Honnan, Hova, Távolság),  
Repülőtípus (TípusAzonosító, TípusNév, RepTávolság),  
Jogosítvány (PilótaAzonosító, TípusAzonosító),  
Pilóta (PilótaAzonosító, PilótaNév, Fizetés).

#### A relációk jelentése:

Járat: *adott járatszámú járat honnan indul, hova érkezik, mennyi a két végpont távolsága (kulcs a Járatszám);*

Repülőtípus: *a típus azonosítója, neve és az, hogy mekkora maximális távolságra tud leszállás nélkül elrepülni (kulcs a RepAzonosító);*

Jogosítvány: *milyen azonosítójú pilóta milyen azonosítójú gépet tud elvezetni (itt a PilótaAzonosító és a TípusAzonosító együtt alkot kulcsot);*

Pilóta: *milyen azonosítójú pilóta, mi a neve és mennyi a fizetése (kulcs a PilótaAzonosító).*

#### Adjunk SQL kérdést, ami megkeresi

- azokat a (kiindulási város, cél város) párokat melyek közti utat minden olyan pilóta le tud repülni (valamelyik általa vezethető géppel, leszállás nélkül), aki legalább 100 000 dollárt keres. Azonos sorok ne szerepeljenek az eredményben.*
- azokhoz a pilótákhoz, akik legalább három különböző géptípust tudnak vezetni, meg akarjuk keresni azt a maximális távolságot, amelyet általuk vezethető gép leszállás nélkül repülni tud.*

#### Megoldás:

a)

```
SELECT DISTINCT Honnan, Hova
FROM Járat
WHERE Távolság <= ALL (
    SELECT m FROM (
        SELECT PilótaAzonosító, MAX(RepTávolság) AS m
        FROM Pilóta NATURAL JOIN Jogosítvány NATURAL JOIN
        Repülőtípus
        WHERE Fizetés >= 100 000
        GROUP BY PilótaAzonosító))
```

#### Magyarázat:

A legbelső SELECT megkeresi minden sokat kereső pilótára az általa repülhető maximális távolságot. Azok a várospárok lesznek jók, amiknek távolsága legfeljebb annyi, mint bármelyik így előálló maximum. Ezt csinálja a külső két SELECT.

b)

```
SELECT PilótaNév AS pn, MAX(RepTávolság)
FROM Repülőtípus NATURAL JOIN Jogosítvány
GROUP BY PilótaNév, RepTávolság
HAVING 3 <= (SELECT COUNT(*)
    FROM Jogosítvány
    WHERE PilótaNév=pn)
```

**4. Tekintsük a következő alaprelációkat:**

Úrhajós (név, bolygó),  
Él (bolygó, virág),  
Allergia (név, virág).

**A relációk jelentése:**

Úrhajós: *milyen nevű úrhajós melyik bolygóról származik, kulcs a név;*

Él: *melyik bolygón milyen virág él, a két attribútum együtt kulcs;*

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

a) *Adj SQL lekérdezést azon úrhajósok megkeresésére, akik a származási bolygójukon élő virágok közül pontosan egyre allergiásak.*

b) *\* Adj SQL lekérdezést azon úrhajósok megkeresésére, akik az összes ismert virágra (ami az Él relációban szerepel) allergiásak.*

**Megoldás:**

a)

```
SELECT Úrhajós.név
      FROM Úrhajós NATURAL JOIN
           Él NATURAL JOIN
           Allergia
      GROUP BY Úrhajós.név
      HAVING COUNT(*)=1
```

b)

```
(SELECT név FROM Úrhajós)
EXCEPT
( SELECT név FROM
  ( (SELECT név, virág FROM Úrhajós CROSS JOIN Él)
    EXCEPT
    (SELECT név,virág FROM Allergia) ) )
```

vagyis (EXCEPT nélkül)

```
SELECT u.név FROM Úrhajós AS u
WHERE u.név NOT IN (
  SELECT c.név FROM
  ( (SELECT c.név, c.virág FROM Úrhajós CROSS JOIN Él) AS c
    WHERE (c.név, c.virág) NOT IN
    (SELECT a.név,a.virág FROM Allergia AS a) ) )
```