

## 12. Adatbázisbeli megszorítások ellenőrzése

### 1. Határozzuk meg, hogy a termékek hányféle anyagból épülnek fel! Írjuk ki a termék nevét, valamint azt, hogy a termék hányféle anyagból épül fel!

Első lépésben írjuk ki a termék kódját és azt, hogy a termék hányféle anyagból épül fel.

A feladat megoldásához elég a SZERKEZET tábla.

```
SELECT kód, COUNT(*)  
FROM szerkezet  
GROUP BY kód;
```

Ha a termék nevét akarjuk kiírni, szükség lesz a TERMÉK táblára is. A TERMÉK táblánál nem tettünk olyan megszorítást, hogy a név egyedi legyen. A termékkód csak egyszer szerepelhet a táblában, mivel a tábla kulcsa a kód, ami azt jelenti, hogy a kód egyértelműen meghatározza a név értékét, de általánosságban ez fordítva nem igaz. Egyáltalán nem biztos, hogy minden feladatnál igaznak kell lennie, hogy a név egyedi. Azt, hogy egy név egyedi legyen vagy sem, nem adatbázis tervezésénél kell eldönteni, hanem a valóságot kell megvizsgálni, és az igényeknek megfelelően kell az adatbázist megtervezni.

Ha a TERMÉK tábla felépítésénél a név egyediségét megkötöttük volna, akkor a GROUP BY záradékba beírhatnánk a termék nevét. Ebben az esetben nem lesz az eredmény táblának se kevesebb, se több sora, mint az előző lekérdezésben, és a COUNT(\*) értéke is ugyanaz marad. (Több semmiképpen nem lehet, mert a termékkódok száma nem lehet több mint, a terméknevek száma.)

```
SELECT név, COUNT(*)  
FROM termék, szerkezet  
WHERE termék.kód = szerkezet.kód  
GROUP BY név;
```

Példánkban, amikor a név egyediségét nem kötöttük ki, a kód szerinti csoportosítás adja meg azt, hogy minden termékről szóló információ külön sorban szerepeljen, ezért kódra kell csoportosítani. Ha a termék nevét is ki akarjuk írni, akkor a termék nevét, mint a csoport tulajdonságát kell kiírni.

```
SELECT termék.kód, MAX(név), COUNT(*)  
FROM termék, szerkezet  
WHERE termék.kód = szerkezet.kód  
GROUP BY termék.kód;
```

## 2. Keressük meg azokat a termékneveket, amelyek nem egyediek!

Határozzuk meg a TERMÉK táblában a különböző nevek számát, a különböző kódok számát és a tábla sorainak a számát.

```
SELECT COUNT(DISTINCT név)
        AS [a különböző nevek száma],
       COUNT(DISTINCT kód)
        AS [a különböző kódok száma],
       COUNT(*) AS [a sorok száma]
FROM   termék;
```

A két utolsó szám, mindig egyenlő egymással, a kód kulcs a TERMÉK táblában, és ebből következik, nem lehet két különböző sorban azonos kód. Ebben a táblában a COUNT(DISTINCT kód) és a COUNT(kód) mindig egyenlő egymással. Ha az adatbázisban minden terméknev egyed, és mindenhol ki van töltve, akkor mindhárom érték egyenlő.

A COUNT(DISTINCT mezőnév) kifejezés az ACCESS 2000-ben nem működik, bár standard SQL utasítás. ACCESS 2000-ben így írhatjuk a fenti utasítást:

```
SELECT DISTINCT név AS t_név
FROM   termék;
→ TERMÉKNÉV{t_név}
```

```
SELECT COUNT(t_név) AS [a nevek száma]
FROM   terméknév;
→ NÉVSZÁM{a nevek száma}
```

```
SELECT MIN([a nevek száma]) AS [a különböző nevek száma],
       COUNT(kód) AS [a különböző kódok száma],
       COUNT(*) AS [a sorok száma]
FROM   termék, névszám;
```

A TERMÉK és a NÉVSZÁM táblából készített direktszorzatban a nevek száma minden sorban ugyanaz az érték, de kiírni, mivel az eredménytábla csak egy sorból áll, csak egyetlen értéket lehet. Ezért írtuk a SELECT mezőlistájában a MIN([a nevek száma]) kifejezést.

Ha az első és harmadik érték nem egyenlő egymással, azaz a különböző nevek száma és a táblázatban lévő sorok száma nem egyenlő, írassuk ki azokat a neveket, amelyekhez több kód tartozik.

```
SELECT  név, COUNT(*) AS [ennyiszer fordul elő ez a név]
FROM    termék
GROUP BY név
HAVING  COUNT(*) > 1;
```

Ha ez a tábla üres, akkor minden név egyedi.

### 3. Írassuk ki TERMÉK táblából azokat a kódokat, amelyek nem találhatóak meg a SZERKEZET táblában, azaz azokat, amelyeknek nem adtuk meg a felépítését!

A TERMÉK táblában lévő kódokat vizsgáljuk meg, hogy benne vannak-e a SZERKEZET táblából kigyűjtött kódok között.

```
SELECT  kód AS [nincs szerkezete]
FROM    termék
WHERE   kód NOT IN (SELECT  kód
                    FROM    szerkezet);
```

Ha az eredménytábla üres, akkor minden terméknek megadtuk a szerkezetét is.

A TERMÉK és a SZERKEZET tábla szerepét felcserélve megkaphatjuk azokat a felesleges szerkezetleírásokat, amelyek nem tartoznak valódi termékekhez.

```
SELECT  kód AS [nincs hozzá termék]
FROM    szerkezet
WHERE   kód NOT IN (SELECT  kód
                    FROM    termék);
```

Ez a feladat leginkább akkor fordulhat elő, amikor az adatbázist külső fájlból szeretnénk feltölteni. Az adatok betöltése előtt célszerű megvizsgálni, hogy a betöltendő adatok jók-e, teljesítik-e az integritási feltételeket. Később, amikor már használjuk az adatbázist, az ilyen jellegű feltételeket megszorításként megadhatjuk, így a nem megfelelő adatok nem is kerülhetnek be az adatbázisba.

#### *Gyakorló feladatok*

1. Ellenőrizzük, hogy minden SZERKEZET táblabeli azonosító előfordul-e a TERMÉK táblában!
2. Keressük meg, hogy melyek azok a terméknevek, amelyek nem egyediek!
3. Keressük ki a RENDELÉS táblából azokat a sorokat, amelyekhez nem tartozik a RENDELÉSFEJ táblában sor. Írassuk ki az ilyen sorok minden adatát, majd készítsünk egy listát, amelyik tartalmazza minden rossz rendelésszámhoz a RENDELÉSFEJ táblából azokat a rendelésszámokat, amelyek csak az első karakterben különböznek az adott rossz rendelésszámtól! (Feltételezzük, hogy az első karaktert véletlenül elírták.)