

## 29. feladat: Külső ciklusváltozó szerinti válogatás

Ellenőrizzük, hogy az adatbázisban minden Aa- és Ab-vel kezdődő rendelésszám megvan-e, azaz létezik-e Aa-0000 és Ab-9999 közé eső minden rendelésszám! Jelenítsük meg a hiányzó számlaszámokat!

Első pillanatban a feladat meglepő, tekintettel arra, hogy olyan adatoknak a kiűjtését kéri, amelyek nincsenek benne az adatbázisban. Márpedig az SQL nyelvnek a példatárban tárgyalt része meglévő táblázatok sorainak a kombinálásával és az így előállított sorokból történő válogatással foglalkozik.

Bármelyik hagyományos programozási nyelvben a feladatot egy egyszerű két-szintű ciklussal könnyen meg lehet oldani. A külső ciklus két lehetséges értéke „a” és „b”, a belső ciklust pedig 0-tól 9999-ig futtatva előállítható az összes olyan rendelésszám, amelyeknek meg kellene lenniük az adatbázisban. A feladat ennek alapján úgy fogalmazható meg, hogy generálni kell a lehetséges rendelésszámokat, majd meg kell vizsgálni, hogy az így generált rendelésszámok benne vannak-e az adatbázisban.

Az SQL lekérdezéshez kellene tehát egy olyan tábla, amelyik minden olyan rendelésszámot tartalmaz, ami a vizsgált intervallumba belesik. Egy 20.000 soros tábla konkrét előállítását nyilvánvalóan szeretnénk elkerülni.

A megoldáshoz az az ötlet visz közelebb, hogy a 00-tól 99-ig terjedő számok előállíthatók egyetlen 10 soros tábla önmagával képzett direktzorzataként.

Készítsünk el egy táblát, amelynek egyetlen egy karakter hosszúságú oszlopa van: SZÁM{számjegy}! A táblának legyen 10 sora, és a mezők értéke 0, 1 ... 9 legyen! Ebből a táblából már könnyedén előállíthatjuk a 00-tól 99-ig terjedő karaktersorozatot.

```
SELECT  első.számjegy & második.számjegy
FROM    szám első, szám második;
```

A lekérdezés eredménye 100 sor, amelyek rendre 00-tól 99-ig terjedő karaktersorozatot fognak tartalmazni.

Ha a SZÁM táblában megadtuk, hogy a *számjegy* mező legyen a kulcs, és a FROM záradékban felcseréljük a táblák sorrendjét, akkor a megjelenítésben is rendezési szempontok sorrendje fordított lesz, mert a második helyen álló tábla kulcsa szerinti rendezettségben jelennek meg a sorok.

```
SELECT  első.számjegy & második.számjegy
FROM    szám második, szám első;
```

## 29. feladat: Külső ciklusváltozó szerinti válogatás

---

Ne használjuk ki az ilyen feltételezett rendezettséget! Ha rendezett sorrendet szeretnénk, használjuk az ORDER BY záradékot!

Térjünk vissza az eredeti feladathoz! A rendelésszám felépítése: Bb-9999, ahol B és b betű, a 9 számjegy. Készítsünk el három táblázatot! Mindháromban csak egy mező legyen, amely egy hosszúságú karaktersorozatot tartalmazzon:

ELSŐ{első_betű}	tartalma:	A;
MÁSOD{második_betű}	tartalma:	a, b;
SZÁM{szám_3_6}	tartalma:	0, 1, 2 ... 9.

A következő lekérdezés egymás után generálja az Aa-val, majd Ab-vel kezdődő összes lehetséges rendelésszámot, és megvizsgálja, hogy ezek benne vannak-e a RENDELÉSFEJ tábla *rend\_szám* oszlopának a mezőiben. Ha egy generált rendelésszámot nem talál meg, kiírja azt az eredménytáblába. Végül az így kapott hiányzó rendelésszámokat növekvő sorrendbe rendezi.

```
SELECT  első_betű & második_betű & '-' & három.szám_3_6
        & négy.szám_3_6 & öt.szám_3_6 & hat.szám_3_6
        AS hiányzik
FROM    első, másod,
        szám_három, szám_négy, szám_öt, szám_hat
WHERE   első_betű & második_betű & '-' &
        három.szám_3_6 & négy.szám_3_6 &
        öt.szám_3_6 & hat.szám_3_6
        NOT IN (SELECT  rend_szám
                FROM  rendelésfej)
ORDER BY első_betű & második_betű & '-' &
        három.szám_3_6 & négy.szám_3_6 &
        öt.szám_3_6 & hat.szám_3_6;
```

Vegyük észre, hogy a keresett rendelésszámokat tartalmazó, 20.000 soros táblát nem mi készítettük el! Mivel a WHERE záradékban nem adtunk meg semmilyen összekapcsoló feltételt a FROM záradékban megadott táblákhoz, az SQL minden tábla minden elemét összepárosítja, és ezzel előállítja a szükséges táblát. A FROM záradékban előállított tábla sorainak a száma:

$$1 * 2 * 10 * 10 * 10 * 10 = 20.000$$

A három segédtábla tartalmát egy újabb lekérdezés előtt bármikor módosíthatjuk, így tetszés szerinti rendelésszám intervallum teljességét megvizsgálhatjuk. Ha szükséges, LIKE vagy BETWEEN operátorral szűkíthetjük a generált számok feldolgozását.

### Gyakorló feladatok

1. Jelenítsük meg azokat az „1”-es karakterrel kezdődő háromjegyű számokat, amelyekhez nem tartozik anyagazonosító!
2. Jelenítsük meg azokat az „A” karakterrel kezdődő lehetséges termékkódokat (2 nagybetű és 1 számjegy), amelyek nem jelölnek létező termékeket!
3. Keressük meg, hogy az elmúlt évben melyik nap nem volt semmilyen megrendelés! Hagyjuk ki az eredménytáblából a szombati és vasárnapi napokat!