

Használjunk Hsqldb-t!

Fred Daoud ír [blogjában](#) arról, hogy összetett és bonyolult adatszerkezetek helyett használjunk [Hsqldb](#)-t, pusztán memóriában futó SQL szerverként. Példaképpen az alábbi táblázatot adja meg, amelyben csak annyit tárolunk: ki, melyik országban, milyen kategóriában mennyit költött.

| Person | Country | Category | Amount |
|--------|-------------|----------|--------|
| Peter | Italy | Food | 200.00 |
| Peter | Greece | Food | 150.00 |
| Peter | Greece | Gifts | 50.00 |
| Lois | Italy | Clothing | 650.00 |
| Lois | Belgium | Food | 250.00 |
| Chris | Switzerland | Food | 300.00 |
| Meg | Switzerland | Clothing | 200.00 |
| Stewie | Australia | Gifts | 500.00 |

Ezt a táblázatot le tudjuk tárolni több megoldással is, például készíthetünk rá egy osztályt, mint entitást és `java.util.Vector` tárolja a sorokat. Esetleg az oszlopokra készíthetünk entitást és valamilyen `java.util.Map` származékban tároljuk le a sorokat. Ha követjük Fred eszmefuttatását, és használunk generikus deklarációt, akkor a valami ilyesmi típusban tarthatjuk a táblázat egy sorát:

Java forrás

```
Map<Person<Country<Category<Amount>>>>
```

A példa feladata egyszer: adjuk meg egy újabb táblázatban, hogy kategóriánként mennyi kiadás történt. Egyszer? Végig kell mennünk a táblázaton, az egyes kategóriákat le kell tennünk egy újabb `Map` típusba, értéként pedig a kiadás oszlop megfelelő sorát hozzá kell adnunk. Miért ne használnánk `SQL` motort erre a feladatra? A `Hsqldb` képes csak memóriában futni:

Java forrás

```
Connection c = DriverManager.getConnection("jdbc:hsqldb:mem:aname", "sa", "");
```

S az így létrehozott kapcsolaton létrehozzuk a kívánt táblát, beleszórjuk az adatokat, majd kedvünk és lehetőségeink szerint kérdezzünk le belőle. A program bezárása során a létrehozott struktúra - mivel memóriában létezett csak: elfelejtdik.