**2.7. Adatbázis értelmezése**

iDevice ikon

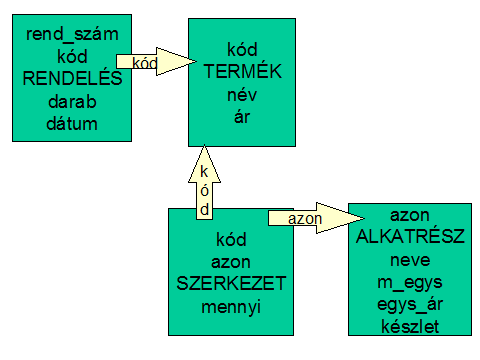
Ebben a fejezetben egy kész, normalizált adatbázist értelmezünk, majd bizonyos függőségek érvényességéről fogunk dönteni. Készítünk mintasorokat az adatbázisra, majd egy kimutatáshoz gyűjtjük össze az adatokat. Végül olyan sorokkal bővítjük az adatbázis tábláit, melyek egy előírt eseményt tükröznek.

**2.7.1. Gyártás**

iDevice ikon

Egy cég rendeléseket vesz fel dátum időpontra a raktárán lévő alkatrészekből összeszerelhető termékeire.

A vevő nem regisztrált.



10. ábra

A rövid leírás és a kapcsolati ábra szerint a következőket lehet tudni:

Minden Terméknek van kódja, neve és aktuális ára; minden Alkatrésznek van azonosítója, neve, mértékegysége, aktuális egységára és pillanatnyi készlete. A Termék törzs kulcsa (a kód) külső kulcsként megtalálható a Szerkezetben, illetve az Alkatrész kulcsa (az azon) is külső kulcs a Szerkezetben. A 2 törzs közti viszonyt a Szerkezet kapcsolótábla oldja meg, amiben tárolásra kerül, hogy bármelyik egységnyi termékhez melyik alkatrészből mennyi szükséges annak mértékegysége szerint. A Rendelés valójában rendeléstétel-tábla, mert a kulcsa nemcsak a rendszám (és benne nem a bizonylat fej- és lábléc-adatai vannak), de mivel semmilyen rendelésfejbe illő adatot nem tárolunk, Rendelés a neve. Ebbe a rendelés alkalmával generált rendszám mellé különféle termékkódokat vihetünk fel, amelyekről azt tároljuk, hogy hány egységet és mikorra kérik.

A kapcsolatban álló táblák között nyíl látható, mely mindig a sokból mutat az egybe (rajta pedig a kapcsoló tulajdonság helyezkedik el).

A függőségi család pedig ugyancsak kiolvasható az adatbázis tábláiból, miután minden táblán belül a kulcs meghatározza a többi tulajdonságot:

{kód} → {név, ár}  
{azon} → {neve, m\_egys, egys\_ár, készlet}  
{kód, azon} → {mennyi}  
{rend\_szám, kód} → {darab, dátum}  
Láttuk az előzőekben, hogy meglevő függőségekből újakat lehet származtatni, és egy függőségről el lehet dönteni, hogy következik-e a meglevőkből.

Fennállnak-e az alábbi egyértelmű meghatározások?

* {rend\_szám, kód} → {név, ár}  
  igen; a bal oldal meghatározza önmaga részeit, tehát a kódot is, ami pedig meghatározza a nevet és árat
* {rend\_szám} → {darab}  
  nem; a bal oldal semmit sem határoz meg egyértelműen (a darabhoz szükséges a rend\_számon túl a kód is)
* {azon, kód} → {név, neve, mennyi}  
  igen; a bal oldal egyik tagja az alkatrész nevét, másik tagja a termék nevét, a két tag együtt pedig a szerkezet mennyiségét határozza meg
* {dátum, kód} → {rend\_szám}  
  nem; egy napra ugyanabból a termékből többen is rendelhettek, vagyis a rend\_szám nem egyértelmű (csak a rend\_szám és kód ismeretében biztos a dátum)
* {rend\_szám, kód} → {készlet}  
  nem; a készlethez ismerni kell az azonosítót, ami nincs a bal oldalon, és a bal oldal semmilyen része sem határozza meg az azonosítót (rend\_számtól függetlenül bármelyik terméknek több alkotója lehet a szerkezet alapján, tehát nem egyértelmű az alkotó aktuális készlete)

**Készítsünk mintasorokat az adott adatbázisra!**

A legfontosabb megszorítás az, hogy az egyes táblákban a kulcstartalom csak egyszer fordulhat elő (összetett kulcs esetén az összefűzött jelsorozat egyedi), majd az adatbázis épsége miatt a külső kulcsok tartalma csak olyan érték lehet, ami már létezik abban a táblában kulcsként, ahova mutat.

**TERMÉK**

|  |  |  |
| --- | --- | --- |
| **kód** | **név** | **ár** |
| 111 | termék1 | 2000 |
| 123 | termék2 | 2000 |
| 134 | termék3 | 15000 |
| 141 | termék4 | 45000 |
| 156 | termék5 | 1890 |
| 176 | termék7 | 5600 |

**RENDELÉS**

|  |  |  |  |
| --- | --- | --- | --- |
| **rendszám** | **kód** | **darab** | **dátum** |
| 977889 | 123 | 3 | 97.02.12 |
| 977889 | 134 | 1 | 97.02.12 |
| 977890 | 156 | 2 | 97.02.23 |
| 977891 | 134 | 1 | 97.03.11 |
| 977891 | 141 | 5 | 97.02.23 |
| 977891 | 176 | 3 | 97.02.23 |

**ALKATRÉSZ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **azon** | **neve** | **me** | **eár** | **készl** |
| 12333 | alkatrész12 | kg | 1200 | 30 |
| 23230 | alkatrész23 | l | 90 | 12 |
| 34555 | alkatrész34 | db | 450 | 21 |
| 44555 | alkatrész44 | m | 5600 | 19 |
| 56789 | alkatrész56 | kg | 380 | 120 |
| 87900 | alkatrész87 | g | 12 | 51 |

**SZERKEZET**

|  |  |  |
| --- | --- | --- |
| **kód** | **azon** | **mennyi** |
| 123 | 56789 | 2.50 |
| 123 | 34555 | 0.25 |
| 134 | 12333 | 1.30 |
| 134 | 56789 | 3.00 |
| 134 | 44555 | 0.75 |
| 134 | 23230 | 0.10 |

A mintasorokból kiolvasható, hogy pillanatnyilag:

* 7-féle terméket és 6-féle alkatrészt tartunk nyilván
* a 123-as termék 2, a 134-es pedig 4 alkotóból áll
* az 56789-es alkatrész 2 termékbe is beépül
* valójában 3 rendelés tételeit látjuk: a 977889-es rendelésszámon 2 terméket, a 977890-esen 1-et és a 977891-esen 3 tételt kértek
* a 134-es termékről 2 megrendelés is szól; az egyik 97.02.12-re 1 db, a másik 97.03.11-re 1db
* minden alkatrészből nullánál nagyobb készlet van a megfelelő mértékegység szerint
* valójában a termék- és alkatrésznevek semmitmondóak, de szintén egyediek

Lehetséges kimenet a mintasorok alapján:

|  |
| --- |
|  |
|  | |  | | --- | | A '97.02.12 - re rendelt termékek alkatrészigénye :    1.3 kg alkatrész12  0.1 l alkatrész23  0.75 db alkatrész34  0.75 m alkatrész44  10.5 kg alkatrész56    Ennek önköltsége : 10096.50 Ft  Várható árbevétel : 21000.00 Ft | |

Az adatbázisból (mintasoraitól függetlenül) sokféle lekérdezést készíthetünk, egyelőre saját algoritmusunkat követve, mely inkább a hagyományos fájlkezelést imitálná, mint az SQL nyelvet. Pl.

* melyik termékből rendelték eddig a legkevesebbet
* adott napra miből mennyit kell elkészíteni
* melyik alkatrész a leggyakoribb az alkotók között
* adott termékből maximum hány készíthető el a pillanatnyi készletek alapján
* a holnapra elkészítendő megrendelésekhez mennyi hiányzik a szükséges alkotókból
* az egyes termékek által hozott haszon (árbevétel-önköltség)
* a legdrágább termék
* a legköltségesebb termék
* minden termék legdrágább alkotója
* a legforgalmasabb nap (amikorra a legtöbb megrendelés szólt)
* az a nap, amikorra csak egyféle terméket kellett készíteni
* mihez kell egy adott nevű alkatrész
* mi a szerkezete egy adott nevű terméknek

(A lekérdezéseket a következő részben SQL-ben meg fogjuk tudni oldani.)

Bővítsük a fenti adatbázist (legyen CUKRÁSZDA) azokkal a mintasorokkal, amelyek a következő eseményt tükrözik:  
**„A mai napon két vevő is rendelt holnapra 1 Dobostortát, de az egyik még 2 adag Sarokházat is kért."**

A feladat 2 termékről és 2 rendelésről szól, melyből az egyik 1, a másik 2 tételből áll. A 2 termék receptje nem került említésre, ezért tetszőlegesen készítjük azt el. Az az információ, hogy a mai napon vesszük fel a rendelést, nem tárolódik, hiszen fizikai sorrendben kerülnek fel a rendeléstételek, melyek elkészülésének csak a dátuma lényeges az ügyvitelben.)

A termékek (cukrászkészítmények) kódját mi generáljuk, az árat tetszőlegesen választjuk:

|  |  |  |
| --- | --- | --- |
| **kód** | **név** | **ár** |
| 101 | Dobostorta | 2200 |
| 106 | Sarokház | 550 |

A recept leírásához előbb az alkotókat kell felvenni, melyek leíróinak értéke ugyancsak tetszőleges:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **azon** | **neve** | **me** | **eár** | **készl** |
| 12345 | cukor | kg | 180 | 30 |
| 23456 | liszt | kg | 90 | 42 |
| 34567 | tejszín | dl | 110 | 20 |
| 45678 | csokoládé | dkg | 5 | 100 |

A recept legyen (természetesen már létező termékkódokkal illetve alkatrész-azonosítókkal) pl.:

|  |  |  |
| --- | --- | --- |
| **kód** | **azon** | **mennyi** |
| 101 | 12345 | 1.00 |
| 101 | 23456 | 1.00 |
| 101 | 45678 | 5.50 |
| 106 | 34567 | 5.00 |
| 106 | 45678 | 15.00 |

A rendeléstételek (ha pl. a mai dátum 2001.12.09.):

|  |  |  |  |
| --- | --- | --- | --- |
| **rendszám** | **kód** | **darab** | **dátum** |
| 018821 | 101 | 1 | 01.12.10 |
| 018821 | 106 | 2 | 01.12.10 |
| 018822 | 101 | 1 | 01.12.10 |