
Relációs adatmodell

Ullman-Widom: Adatbázisrendszerek Alapvetés

Második, átdolgozott kiadás, Panem, 2009

2.1. Adatmodellek áttekintése

2.2. A relációs modell alapjai

Mi is az adatmodell?

- **Az adatmodell a valóság fogalmainak, kapcsolatainak, tevékenységeinek magasabb szintű ábrázolása**
 - Hálós, hierarchikus adatmodell (apa-fiú kapcsolatok gráfja, hatékony keresés)
 - Relációs adatmodell (táblák rendszere, könnyen megfogalmazható műveletek)
 - Objektum-orientált adatmodell (az adatbázis-kezelés funkcionalitásainak biztosítása érdekében gyakran relációs adatmodellre épül)
 - Logikai adatmodell (szakértői rendszerek, tények és következtetési szabályok rendszere)
 - Félig strukturált (XML) adatmodell
-

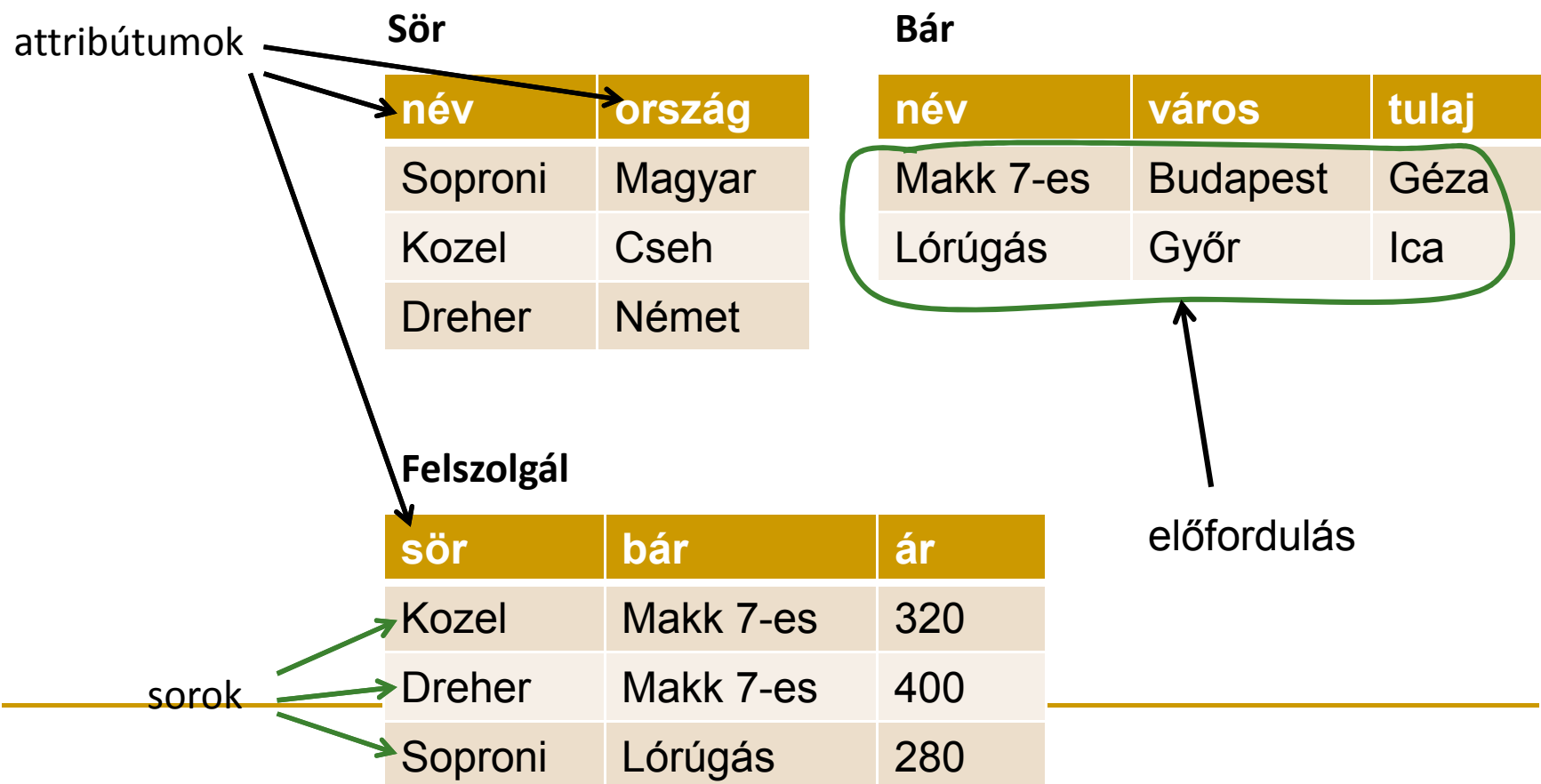
Példa féligstrukturált adatra (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<bár típus="étterem">
  <név>Makk 7-es</név>
  <város>Budapest</város>
  <tulaj>Géza</tulaj>
  <telefon>+36-70-123-2345</telefon>
  <telefon>+36-70-123-2346</telefon>
</bár>
<bár típus="kocsma">
  <név>Lórúgás</név>
  <város>Eger</város>
  <telefon>+36-30-451-1894</telefon>
</bár>
</xml>
```

Relációs adatmodell

Egy reláció sémája: **Sör (név, ország)**.

Az adatbázis sémája: **Sör (név, ország), Bár (név, város, tulaj),
Felszolgál (sör, bár, ár)**.



Relációs adatmodell

- Codd 1970-ban publikált egy cikket, amelyben azt javasolta, hogy az adatokat táblázatokban, **relációkban** tárolják. Az elméletére alapozva jött létre a relációs adatmodell, és erre épülve jöttek létre a relációs adatmodellen alapuló relációs adatbáziskezelők.
 - Relációs adatbáziskezelők például:
ORACLE , INFORMIX , SYSDATABASE , INGRES, DB2
 - A relációs modellben a koncepcionális-logikai-fizikai szint teljesen szétválik. Több absztrakt kezelő nyelv létezik.
-

Relációs adatmodell

- D tartomány : atomi értékek egy halmaza. Minden tartomány bármely értékéhez hozzátartozik egy jól definiált atomi adattípus, és egy formátum.
 - **Relációséma:** $R(A_1, A_2, \dots, A_n)$
 - R – relációnév
 - A_i – attribútum- vagy tulajdonságnevek, oszlopnevek
 - $\text{dom}(A_i)$ – lehetséges értékek halmaza, típusa
 - Egy sémán belül az attribútumok különbözőek
 - A relációs séma foka az attribútumok száma.
(A fenti példa n-edfokú.)
-

Relációs adatmodell

- **Reláció-előfordulás**: r - reláció, tábla, sorhalmaz
Egy sor egyszer szerepel. Sorok sorrendje lényegtelen
- Az $R(A_1, \dots, A_n)$ relációs séma alatt azt a $r(R)$ -rel jelölt halmazt értjük, amely halmaz elemei n -esek: $r = \{t_1, \dots, t_m\}$
- Mindegyik t_i n db értéknek egy rendezett együttese:
 $t_i = \langle v_1, \dots, v_n \rangle$ méghozzá úgy, hogy
 $v_i \in \text{dom}(A_i) \quad i=1, \dots, n$, vagy NULL érték.
- t - tuple (magyarul rekord)
- $r(R)$ - R reláció séma r relációja nem más, mint az attribútumok tartományáiból alkotott Descartes-szorzat egy részhalmaza.
- $r(R) \subseteq (\text{dom}(A_1) \times \dots \times \text{dom}(A_n))$

Relációs adatmodell

- Az attribútumok sorrendje kötött, létezik olyan megfogalmazás, amikor az oszlopok sorrendje lényegtelen (névvel hivatkozunk az oszlopokra).
 - **Másik jelöléssel**
 - $t \in r$ esetén t sor (angolul: tuple – n -es)
 - $t(A_i)$ vagy $t(i)$ – a t sor i -edik komponense
 - $t[A_{i_1}, \dots, A_{i_k}]$ – a t sor i_1, \dots, i_k -edik komponenseiből álló vektor
 - Egy t sor függvénynek is tekinthető
$$t_i : R \rightarrow D = \text{dom}(A_1) \cup \dots \cup \text{dom}(A_n)$$
ahol $t(A_i) \in \text{dom}(A_i)$, $i=1..n$
-

Relációs adatmodell

- Ez a koncepcionális szint.
 - Logikai szinten a relációk táblákban jelennek meg. A tábláknak egyedi nevű van. A relációk oszlopait az attribútumok címzik, a tábla soraiban helyezkednek el a rekordok. A tábla sorait tetszőlegesen megcserélhetjük, a sorok száma nem kötött. (Ezek a halmazszemlélet miatt vannak.)
 - Fizikai modell : a táblát a reláció valamilyen állományszerkezetben jeleníti meg. (pl. szeriális állomány.) A relációs adatbáziskezelők mindig indexelnek, a kedvenc indexelési mód a B+ fa.
-

Relációs adatmodell

- Az adatbázis tulajdonképpen relációk halmaza. A megfelelő relációsémák halmaza adja az **adatbázissémát**, a hozzá tartozó előfordulások pedig az **adatbázis-előfordulást**.
-

Mire kell odafigyelni?

Mivel attribútumok halmazáról van szó, a Példa 1 és Példa 2 relációk nevüktől eltekintve azonosak.

Példa 1

A	B	C
a	b	c
d	a	a
c	b	d

Példa 2

B	C	A
b	c	a
a	a	d
b	d	c

Mivel sorok halmazáról van szó, a Példa 1 és Példa 3 relációk nevüktől eltekintve azonosak.

Példa 3

A	B	C
c	b	d
d	a	a
a	b	c

Példa 4

A	B	C
c	b	d
c	b	d
a	b	c

Ebben a modellben Példa 4 nem reláció.

Példa megszorításra

- Az attribútumok egy halmaza egy **kulcsot** alkot egy relációra nézve, ha a reláció **bármely előfordulásában** nincs két olyan sor, amelyek a kulcs összes attribútumának értékein megegyeznének.
 - Ilyen egy attribútumú kulcs például a személyiigazolvány-szám vagy a TAJ szám.
 - **Megjegyzés:** egy kulcs nem feltétlenül egy attribútumból áll. Például a bár táblában valószínűleg az a jó, ha a név és a város együtt alkotják a kulcsot.
 - A kulcsot aláhúzás jelöli:
Bár (név, város, tulaj).
-

Vigyázat!

Ennél a konkrét előfordulásnál választhatnánk a nevet kulcsnak, sok esetben viszont ez nem megfelelő, hiszen több különböző ember is él ugyanazzal a névvel.

név	telefon
Grasshaus Ignác	20-234-4567
Menyhért Lipót	20-564-2345
Bereg Anna	20-345-1231



Filmek példa

```
Filmek(  
cím:string,  
év:integer,  
hossz:integer,  
műfaj:string,  
stúdióNév:string,  
producerAzon:integer  
)
```

```
FilmSzínész(  
név:string,  
cím:string,  
nem:char,  
születésidátum:date  
)
```

```
SzerepelBenne(  
filmCím:string,  
filmÉv:integer,  
színészNév:string  
)
```

```
GyártásIrányító(  
név:string,
```

```
cím:string,  
azon:integer,  
nettóBevétel:integer  
)
```

```
Stúdió(  
név:string,  
cím:string,  
elnökAzon:integer  
)
```