

# Adatbázisrendszerek

## 3. előadás: A relációs modell

Relációséma, reláció, integritási megszorítások

Ispány Márton

2020. február 25.



**DEBRECENI  
EGYETEM**





A relációs modellt Ted Codd vezette be 1970-ben az IBM Researchnél, később munkásságáért Turing díjat kapott.

## Cikk

Codd, E., A Relational Model for Large Shared Data Banks, *Communication of the ACM* **13:6**, June 1970.

Előnyei:

- egyszerűség
- matematikai megalapozottság

Alapja a **matematikai reláció** fogalma, ami hasonló egy értékekkel kitöltött táblázathoz (mátrixhoz).

Elméleti alapjai:

- halmazelmélet
- elsőrendű predikátumkalkulus (matematikai logika)

A relációs modellben az adatbázis relációknak egy halmaza lesz. Minden reláció durván egy flat fájlként fogható fel.



Elméleti háttere az SQL relációs szabványoknak: SQL-89 (SQL1), SQL-92 (SQL2) és SQL99 (SQL3).

Első üzleti implementációk a 80-as évek elején:

- SQL/DS rendszer MVS op. rendszeren (IBM)
- Oracle DBMS

A legtöbb piaci szoftverben a relációs modellt implementálták még 10 éve, most már elterjedt a NoSQL paradigma is:

- DB2 és Informix Dynamic Server (IBM)
- Oracle és Rdb
- Sybase DBMS (Sybase)
- SQLServer és Access (MS)
- MySQL, PostgreSQL (open source)

Megelőző modellek, melyeket még „örökölt” rendszerekként használnak:

- hierarchikus
- hálós



## Reláció

A reláció értékek egy táblázata, amely sorok egy halmazából áll.

## Sor

Minden egyes sor adatelemei a modellezett kisvilág egy egyed-előfordulásáról vagy egy kapcsolat-előfordulásáról tartalmaznak tényeket (információt).

A sorokat a formális modellben elem  $n$ -eseknek vagy rekordoknak fogjuk nevezni.

## Oszlopok

Minden egyes oszlop egy oszlop fejléccel (címmel) rendelkezik, amely az illető oszlopban lévő adatok jelentéséről ad információt.



Az oszlop fejléceket a formális modellben attribútum neveknek (röviden attribútumoknak) fogjuk nevezni.

## Reláció kulcsa

Minden sor rendelkezik egy olyan adatelem értékkel (vagy azok egy halmazával), amely egyértelműen azonosítja a sort a táblázatban. Ezt **kulcsnak** nevezzük.

Például a Neptunban a Hallgatók táblának a Neptun-kód a kulcsa.

Néha a sorazonosítót vagy egy szekvenciális számsorozatot (ennek generálására minden adatbázis-kezelő rendszer ad eszközt) használunk kulcsként. Ekkor mesterséges kulcsról beszélünk.



## Definíció (atomi érték)

Atominak nevezünk egy olyan értéket, amely már nem bontható tovább a relációs modell szempontjából.

Relatív (szubjektív) fogalom: lakcím (ir. szám, település, utca, házsám) lehet atomi vagy összetett

## Definíció (tartomány)

Egy  $D$  **tartomány** atomi értékek egy halmaza.

Jellemzői:

- név
- adattípus
- formátum
- korlátozás
- további információk az értelmezéshez



## Példák

- *Mobiltelefonszámok*: 11 decimális számjegy  $+dd-dd-ddddddd$  formátumban.
- *Személyi számok*: 11 decimális számjegy  $d dddddd dddd$  formátumban.
- *Nevek*: tetszőleges hosszúságú karaktersorozat.
- *Tömegek*: nemnegatív valós szám mértékegységgel (font vagy kg).



## Definíció

**Relációséma** alatt az  $R(A_1, A_2, \dots, A_n)$  jelölést értjük, ahol  $R$  a relációséma neve,  $A_1, A_2, \dots, A_n$  pedig attribútumok. Minden  $A_i$  **attribútum** egy szerepkör neve, amelyet valamely  $D_i$  tartomány játszik.  $D_i$ -t az  $A_i$  attribútum tartományának nevezzük, és  $\text{dom}(A_i)$ -vel is jelöljük,  $n$  a reláció **foka**.

## Példák

- HALLGATÓ(Név, Személyi\_száma, Lakcím, Szak, Évfolyam, Neptun\_kód)
- TANSZÉK(Tanszék\_név, Tanszékvezető\_neve, Kar)
- AUTÓ(Márka, Típus, Gyártási\_év, Motorszám, Rendszám)





## Megjegyzés

Előfordulhat, hogy több attribútumnak is **azonos a tartománya**. Az attribútumok különböző **szerepköreit**, interpretációit jelölik ki a tartományoknak.

## Definíció

A  $R(A_1, A_2, \dots, A_n)$  relációséma egy  $r$  **relációja** – amit szokás  $r(R)$ -rel is jelölni – elem  $n$ -eseknek egy halmaza:

$$r = \{t_1, t_2, \dots, t_m\}.$$

Minden  $t_i$  **elem  $n$ -es** ( $1 \leq i \leq m$ )  $n$  darab értéknek egy rendezett listája:

$$t_i = \langle v_{i1}, v_{i2}, \dots, v_{in} \rangle,$$

ahol minden  $v_{ij}$  érték ( $1 \leq j \leq n$ ) vagy  $\text{dom}(A_j)$ -nek az eleme, vagy egy speciális NULL érték.





## Megjegyzések

- A definícióban említett elem  $n$ -eseket **rekordoknak** is nevezzük.
- Egy  $t$  rekordban szereplő  $j$ -edik értékre, amely az  $A_j$  attribútumhoz tartozik,  $t[A_j]$ -vel (vagy röviden  $t[j]$ -vel) hivatkozhatunk.

## Definíció – másképpen fogalmazva

A relációs adatmodellben egy  $r(R)$  **reláció** nem más, mint egy  $\text{dom}(A_1), \text{dom}(A_2), \dots, \text{dom}(A_n)$  tartományokon értelmezett  $n$ -ed fokú matematikai reláció, amely részhalmaza azon tartományok Descartes-szorzatának, amelyek  $R$ -et definiálják:

$$r(R) \subseteq \text{dom}(A_1) \times \text{dom}(A_2) \times \dots \times \text{dom}(A_n).$$

Ebben a definícióban a NULL értéket beleértjük  $\text{dom}(A_i)$ -ba.





## Megjegyzés

A Descartes-szorzat tartalmazza a tartományok értékeinek összes lehetséges kombinációját.  $|D|$ -vel jelölve egy  $D$  tartomány **számosságát**, a Descartes-szorzatban szereplő elem  $n$ -esek (rekordok) száma:

$$|\text{dom}(A_1)| \cdot |\text{dom}(A_2)| \cdot \dots \cdot |\text{dom}(A_n)|.$$

A megjegyzés teljes indukcióval bizonyítható.



## Értelmezés:

- $r$  az  $R$  relációséma egy konkrét tartalma (intension)
- $R$  az  $r$  reláció kiterjesztése, absztrakciója (extension)

**Heurisztika:**  $R$  egy táblázatnak az üres váza,  $r$  az adatokkal feltöltött táblázat maga.

## Rendezés:

- A rekordokat (sorokat) nem feltételezzük rendezettnek az elméleti modellben, viszont a fizikai tárolón általában rendezve helyezkednek el (indexálás).
- A relációs sémán belül és így az összes rekordban is az értékek rendezetten, az attribútumok sorrendjének megfelelően, helyezkednek el.

Attribútumok

A relációséma neve

**HALLGATÓ**

	Név	Személyi_szám	Lakcím	Szak	Évfolyam	Neptun	Telefonszám
	Brók Ernő	1 730817 2818	3756 Varbóc	PTI	1	HRB0FX	+36-20-4779833
	Hiszt Erika	2 800922 8420	9122 Felpéc	PTI	2	CC18SG	+36-30-1545493
	Könyök Ödön	1 671011 2057	7757 Babarc	MI	2	KJZ9U9	+36-20-2542208
	Szőke Barna	1 481209 0753	2668 Patvarc	PTI	1	R2D2XX	+36-70-4981212
	Virra Dóra	2 860502 3495	4967 Csaholc	MI	1	C3POO8	+36-70-3881795

Rekordok



## Megjegyzések

- Az összes lehetséges kombináció közül a reláció egy adott pillanatban csak azokat a rekordokat tartalmazza, amelyek a valós világ pillanatnyi állapotát tükrözik: ez a **reláció aktuális állapota**. Ahogyan a valós világ változik, úgy változik a reláció (állapota) is.
- A **reláció sémája** – az előzőekkel ellentétben – viszonylag statikus, **nem változik**, néhány ritka esetet leszámítva.



## 3. előadás: Relációs modell

Ispány Márton

Alapfogalmak

Relációséma

Reláció

Megszorítások

Relációs  
adatbázisséma  
és adatbázis

Feltöltés

- $R(A_1, A_2, \dots, A_n)$  – egy  $n$ -ed fokú relációséma
- $Q, R, S$  – relációsémák nevei
- $q, r, s$  – reláció(állapoto)k nevei
- $t, u, v$  – rekordok
- Egy reláció neve a **reláció aktuális állapotát** jelöli (azaz magát a relációt), míg az  $R(A_1, A_2, \dots, A_n)$  alak **kizárólag** a relációsémára hivatkozik.
- Egy  $A$  attribútum minősíthető annak a  $R$  relációsémának a nevével, amelyhez tartozik:  $R.A$ .
- $t = \langle v_1, v_2, \dots, v_n \rangle$  – egy  $t$  elem  $n$ -es egy  $r(R)$  relációban, ahol  $v_j$  az  $A_j$  attribútumnak megfelelő érték.
  - $t[A_j]$  és  $t.A_j$  – az  $A_j$  attribútumnak megfelelő  $v_j$  érték a  $t$  rekordban
  - $t[A_u, A_w, \dots, A_z]$  és  $t.(A_u, A_w, \dots, A_z)$  – a listában megadott attribútumoknak megfelelő  $\langle v_u, v_w, \dots, v_z \rangle$  értékű részrekord a  $t$  rekordból



Az adatmodell megszorításainak csoportosítása:

- Az adatmodellben benne rejlő megszorítások: **modellalapú** vagy **implicit megszorítások** (pl. az adatok egy matematikai relációba szerveződnek holott alkothatnának más struktúrát is).
- Az adatmodell sémáiban közvetlenül kifejezett megszorítások: **sémaalapú** vagy **explicit megszorítások**.
- Olyan megszorítások, amelyeket **nem lehet** közvetlenül az adatmodell sémáiban kifejezni, és ezért az alkalmazói programokkal kell kifejezni és érvényre juttatni őket: **alkalmazásalapú** vagy **szemantikus megszorítások** vagy **üzleti szabályok**.





## 3. előadás: Relációs modell

Ispány Márton

Alapfogalmak

Relációséma

Reláció

Megszorítások

Relációs  
adatbázisséma  
és adatbázis

Feltöltés

- tartománymegszorítás
- kulcsmegszorítás
- NULL értékre vonatkozó megszorítás
- egyedintegritási megszorítás
- hivatkozási integritási megszorítás



## Definíció

A **tartománymegszorítás** kimondja, hogy minden rekordban minden egyes  $A$  attribútumhoz tartozó érték a  $\text{dom}(A)$  tartományból származik vagy NULL érték, és ezen  $\text{dom}(A)$  tartományok minden elemének atomi értéknek kell lennie.

## Megjegyzés: tartományokra jellemző adattípusok

- numerikus
  - egész
  - valós
- karakter
- logikai
- sztring (fix és változó hosszúságú)
- dátum
- egyéb speciális adattípusok (idő, időbélyeg, pénz stb.)



Az SQL DDL nyelve ad eszközöket a tartománymegszorítás kikényszerítésére.

Példa: a HALLGATÓ relációt az alábbi módon hozhatjuk létre a tartománymegszorításaival együtt.

```
DROP TABLE hallgato PURGE;  
CREATE TABLE hallgato (  
    nev                VARCHAR2(50),  
    szsz               VARCHAR2(15),  
    lakcim             VARCHAR2(50),  
    szak               VARCHAR2(3),  
    evfolyam           NUMBER(1),  
    neptun             VARCHAR2(6),  
    telefon            VARCHAR2(14),  
    szul               DATE  
);
```



Definíció szerint egy relációban minden rekord különböző, azaz egy relációban nincs két olyan rekord, amelynek **minden** attribútumértéke azonos lenne.

## Definíció

Az  $R$  relációsémának létezik egy olyan **attribútumhalmaza**, amely olyan tulajdonságú, hogy tekintve  $R$  bármelyik  $r$  relációját, az adott relációban nincs két olyan rekord, amelynek az értékei azonosak lennének ezen attribútumokra vonatkozóan.

Az attribútumoknak egy ilyen részhalmazát SK-val jelölve, bármely két **különböző**  $t_1$  és  $t_2$  rekordot kiválasztva  $R$  egy  $r$  relációjából:

$$t_1[\text{SK}] \neq t_2[\text{SK}].$$

Minden ilyen SK attribútumhalmaz az  $R$  relációséma **szuperkulcsa**.

Minden relációnak van legalább egy szuperkulcsa – az összes attribútumának a halmaza.



Egy superkulcsnak lehetnek szükségtelen attribútumai, így sokkal hasznosabb fogalom a **kulcsé**, amely nem tartalmaz felesleges attribútumokat.

## Definíció

Egy  $R$  relációséma  $K$  **kulcsa**  $R$ -nek egy olyan superkulcsa, amelyből egy  $A$  attribútumot elhagyva, az így kapott  $K'$  attribútumhalmaz már nem superkulcsa  $R$ -nek.

Egy kulcs kielégíti a következő két feltételt:

- Bármilyen relációt tekintve, a reláció két különböző rekordjának nem lehetnek azonosak a kulcsban szereplő attribútumokhoz tartozó értékei.
- **Minimális superkulcs**, azaz egy olyan superkulcs, amelyből nem tudunk úgy eltávolítani egyetlen attribútumot sem, hogy az egyediségre vonatkozó feltétel továbbra is fennálljon.





Egy  $K$  kulcs **egyszerű**, ha egyetlen attribútum alkotja, egyébként **összetett**.

## Definíció

Egy relációsémának egynél több kulcsa is lehet. Ilyen esetben a kulcsok mindegyikét **kulcsjelöltnek** hívjuk.

## Definíció

A modellező feladata, hogy a kulcsjelöltek közül kiválasszon egyet a relációséma **elsődleges kulcsául**. Ez a kulcsjelölt lesz az, amelynek az értékeit a relációkban szereplő rekordok **azonosítására** fogjuk használni. A kulcsmegszorítás szerint a relációsémának **mindig rendelkeznie kell** elsődleges kulccsal.



## 3. előadás: Relációs modell

Ispány Márton

Alapfogalmak

Relációséma

Reláció

Megszorítások

Relációs  
adatbázisséma  
és adatbázis

Feltöltés

## Megjegyzés

- Egy relációséma elsődleges kulcsát alkotó attribútumokat aláhúzással szoktuk jelölni.
- Amikor egy relációsémának több kulcsjelöltje is van, tetszőlegesen lehet közülük elsődleges kulcsot választani.



Az SQL DDL nyelve ad eszközöket a kulcsmegszorítás kikényszerítésére.

Példa: a HALLGATÓ reláció számára az alábbi módon adhatunk meg elsődleges kulcsot.

```
DROP TABLE hallgato PURGE;  
CREATE TABLE hallgato (  
    nev          VARCHAR2(50),  
    szsz        VARCHAR2(15),  
    lakcim      VARCHAR2(50),  
    szak        VARCHAR2(3),  
    evfolyam    NUMBER(1),  
    neptun      VARCHAR2(6)  
    CONSTRAINT neptun_pk PRIMARY KEY,  
    telefon     VARCHAR2(14)  
);
```





## 3. előadás: Relációs modell

Ispány Márton

Alapfogalmak

Relációséma

Reláció

Megszorítások

Relációs  
adatbázisséma  
és adatbázis

Feltöltés

## Megjegyzés

A PRIMARY KEY megszorítás a NOT NULL és a UNIQUE megszorítások kombinációja.

A UNIQUE megszorítás tiltja ugyanazon értékek használatát viszont a NULL értéket nem, azaz az többször is előfordulhat.



3. előadás:  
Relációs  
modell

Ispány Márton

Alapfogalmak

Relációséma

Reláció

Megszorítások

Relációs  
adatbázisséma  
és adatbázis

Feltöltés

## Definíció

Az **egyedintegritási megszorítás** kimondja, hogy egyetlen elsődlegeskulcs-érték sem lehet NULL érték. Ha az elsődleges kulcs összetett, akkor annak egyik komponense sem lehet NULL érték.

## Megjegyzés

Megengedve a NULL értékeket az elsődleges kulcs számára, nem tudnánk egyértelműen azonosítani minden rekordot. Például ha két vagy több rekordnál NULL érték tartozna az elsődleges kulcsukhoz, akkor nem tudnánk megkülönböztetni őket, ha megpróbálnánk más relációkból hivatkozni rájuk.



Az SQL DDL nyelve további eszközöket ad megszorítások kikényszerítésére.

Példa: a HALLGATÓ reláció számára az alábbi módon adhatunk meg NULL és UNIQUE megszorításokat.

```
DROP TABLE hallgato PURGE;  
CREATE TABLE hallgato (  
    nev VARCHAR2(50) CONSTRAINT nev_nn NOT NULL,  
    szsz VARCHAR2(15) CONSTRAINT szsz_un UNIQUE,  
    lakcim VARCHAR2(50),  
    szak VARCHAR2(3),  
    evfolyam NUMBER(1),  
    neptun VARCHAR2(6) PRIMARY KEY,  
    telefon VARCHAR2(14)  
);
```

A nev attribútum nem lehet NULL érték, de két vagy több diáknak lehet ugyanaz a neve, pl. Kovács Lajos. Az szsz személyi számnak, ha ismert, egyedinek kell lenni, de lehet NULL érték is.



## 3. előadás: Relációs modell

Ispány Márton

Alapfogalmak

Relációséma

Reláció

Megszorítások

Relációs  
adatbázisséma  
és adatbázis

Feltöltés

A **hivatkozási integritási megszorítást** két reláció között értelmezzük, és a két relációban lévő rekordok közötti konzisztencia megteremtése érdekében használjuk.



## Definíció

Egy  $R_1$  relációséma FK-val jelölt attribútumhalmaza **külső (idegen) kulcsa**  $R_1$ -nek, amely hivatkozik az  $R_2$  relációsémára, ha eleget tesz a következő feltételeknek:

- Az FK-beli attribútumoknak és az  $R_2$  PK-val jelölt elsődleges kulcsattribútumainak páronként azonos a tartománya; ekkor azt mondjuk, hogy az FK attribútumok **hivatkoznak** az  $R_2$  relációsémára.
- Bármely  $r_1(R_1)$  aktuális állapotának egy  $t_1$  rekordjában egy FK-beli érték **vagy** megjelenik egy  $r_2(R_2)$  aktuális állapotának valamely  $t_2$  rekordjában PK értékeként, **vagy az értéke NULL**. Az előbbi esetben  $t_1[FK] = t_2[PK]$ , ekkor azt mondjuk, hogy a  $t_1$  rekord **hivatkozik** a  $t_2$  rekordra.

Ha e két feltétel teljesül, egy **hivatkozási integritási megszorítás** áll fenn  $R_1$ -ről  $R_2$ -re vonatkozóan.



## 3. előadás: Relációs modell

Ispány Márton

Alapfogalmak

Relációséma

Reláció

Megszorítások

Relációs  
adatbázisséma  
és adatbázis

Feltöltés

Tegyük fel, hogy egy olyan relációsémát akarunk létrehozni az egyetemi adatbázisban, amely a hallgatók által felvett kurzusokat fogja majd tartalmazni.

Ekkor szükségünk lesz arra, hogy hivatkozzunk a hallgatókat és a kurzusokat tartalmazó relációkra külső kulcsok segítségével.



```
DROP TABLE felvesz PURGE;
DROP TABLE kurzus PURGE;
CREATE TABLE kurzus (
    azon VARCHAR2(10)
    CONSTRAINT kurzus_pk PRIMARY KEY, ...
);
CREATE TABLE felvesz (
    azon          NUMBER(10),
    neptun        VARCHAR2(6),
    kurzus        VARCHAR2(10),
    datum        DATE,
    CONSTRAINT felvesz_pk PRIMARY KEY (azon),
    CONSTRAINT hallg_fk FOREIGN KEY (neptun)
        REFERENCES hallgato(neptun),
    CONSTRAINT kurzus_fk FOREIGN KEY (kurzus)
        REFERENCES kurzus(azon) );
```



## Szemantikus integritási megszorítások

### Példa

- A dolgozó fizetése nem lehet nagyobb a főnökéénél.
- Egy héten egy dolgozó maximum 56 órát dolgozhat egy projekten.

### Adatok közti függések – lásd később

- funkcionális függés
- többértékű függés

### Átmenet-megszorítások vagy dinamikus megszorítások

### Példa

- A dolgozó fizetése csak nőhet.

Az SQL-99 szabvány eszközöket ad ezen megszorítások kezelésére.





Egy relációs adatbázis rendszerint számos relációt tartalmaz, a relációkban rekordokkal, amelyek különböző módokon vannak egymással kapcsolatban.

## Definíció

Egy  $S$  **relációs adatbázisséma** az

$$S = \{R_1, R_2, \dots, R_m\}$$

relációséma-halmaz, valamint integritási megszorítások – IC-vel jelölt – halmazának az együttese.

## Definíció

$S$  egy DB **relációs adatbázis(állapot)a** olyan

$$DB = \{r_1, r_2, \dots, r_m\}$$

reláció(állapot)k halmaza, ahol minden  $r_i$  az  $R_i$  séma egy relációja, és minden  $r_i$  reláció kielégíti az IC-ben megadott integritási megszorításokat.



## DOLGOZÓ

Vnév	Knév	<u>Szsz</u>	Szdátum	Lakcím	Nem	Fizetés	Főnök_szsz	Osz
------	------	-------------	---------	--------	-----	---------	------------	-----

## OSZTÁLY

Onév	<u>Oszám</u>	Vez_szsz	Vez_kezdő_dátum
------	--------------	----------	-----------------

## OSZT\_HELYSZÍNEK

<u>Oszám</u>	<u>Ohelysín</u>
--------------	-----------------

## PROJEKT

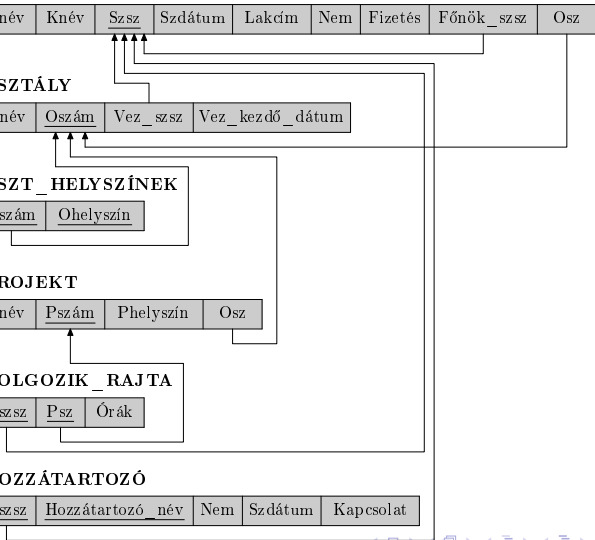
Pnév	<u>Pszám</u>	Phelysín	Osz
------	--------------	----------	-----

## DOLGOZIK\_RAJTA

<u>Dszsz</u>	<u>Psz</u>	Órák
--------------	------------	------

## HOZZÁTARTOZÓ

<u>Dszsz</u>	<u>Hozzártartozó_név</u>	Nem	Szdátum	Kapcsolat
--------------	--------------------------	-----	---------	-----------





### 3. előadás: Relációs modell

Ispány Márton

Alapfogalmak

Relációséma

Reláció

Megszorítások

Relációs  
adatbázisséma  
és adatbázis

Feltöltés

#### DOLGOZÓ

Név	Képv	Szsz	Szdátum	Lakcím	Nem	Fizetés	Főnök_szsz	Ök
Kovács	László	1 650 109 08 12	1965. január 9.	4033 Debrecen	F	390000	2 55 1208 22 19	5
Szabó	Mária	2 55 1208 22 19	1955. december 8.	1097 Budapest	N	520000	1 37 1110 45 19	5
Kiss	István	1 680 119 6749	1968. január 19.	1172 Budapest	F	325000	1 410620 4902	4
Talács	József	1 410620 4902	1941. június 20.	4027 Debrecen	F	559000	1 37 1110 45 19	4
Horváth	Erzsébet	2 620915 3134	1962. szeptember 15.	1092 Budapest	N	404000	2 55 1208 22 19	5
Tóth	János	1 720 73 1 2985	1972. július 31.	6726 Szeged	F	325000	2 55 1208 22 19	5
Fazekas	Ilona	2 690329 1099	1969. március 29.	3535 Miskolc	N	325000	1 410620 4902	4
Nagy	Zoltán	1 371110 45 19	1937. november 10.	1061 Budapest	F	715000	NULL	1

#### OSZTÁLY

Önév	Ökzám	Vez_szsz	Vez_kezdő_dátum
Kutatás	5	2 55 1208 22 19	1988. május 22.
Humán erőforrás	4	2 690329 1099	1995. január 1.
Központ	1	1 371110 45 19	1981. június 19.

#### OSZT\_HELYSZÍNEK

Ökzám	Önévszám
1	Budapest
4	Kecskemét
5	Vác
5	Tiszafüred
5	Budapest

#### DOLGOZIK\_RAJTA

Dszsz	Psz	Órák
1 650 109 08 12	1	32.5
1 650 109 08 12	2	7.5
2 620915 3134	3	40.0
1 720 73 1 2985	1	20.0
1 720 73 1 2985	2	20.0
2 55 1208 22 19	2	10.0
2 55 1208 22 19	3	10.0
2 55 1208 22 19	10	10.0
2 55 1208 22 19	20	10.0
1 680 119 6749	30	30.0
1 680 119 6749	10	10.0
2 690329 1099	10	35.0
2 690329 1099	30	5.0
1 410620 4902	30	20.0
1 410620 4902	20	15.0
1 371110 45 19	20	NULL

#### PROJEKT

Név	Pszám	Phelyszín	Ök
X termék	1	Vác	5
Y termék	2	Tiszafüred	5
Z termék	3	Budapest	5
Komputerizáció	10	Kecskemét	4
Reorganizáció	20	Budapest	1
Új fejlesztések	30	Kecskemét	4

#### HOZZÁTARTÓZÓ

Dszsz	Hozzártartózó_név	Nem	Szdátum	Kapcsolat
2 55 1208 22 19	Anna	N	1986. április 5.	lánya
2 55 1208 22 19	Bence	F	1983. október 25.	fia
2 55 1208 22 19	Máté	F	1958. május 3.	házastársa
1 410620 4902	Viktória	N	1942. február 28.	házastársa
1 650 109 08 12	Balázs	F	1988. január 4.	fia
1 650 109 08 12	Anna	N	1988. december 30.	lánya
1 650 109 08 12	Réka	N	1967. május 5.	házastársa



## 3. előadás: Relációs modell

Ispány Márton

Alapfogalmak

Relációséma

Reláció

Megszorítások

Relációs  
adatbázisséma  
és adatbázis

Feltöltés

- Minden relációnak több rekordja lehet az aktuális reláció állapotában.
- A relációs adatbázis állapota az összes reláció(állapotok) uniója.
- Amikor az adatbázis változik új állapotok jönnek létre.
- Alapvető műveletek az adatbázis megváltoztatására:
  - INSERT: egy új rekord beszúrása,
  - DELETE: egy létező rekord törlése,
  - MODIFY: egy létező rekord attribútumainak módosítása.
- Az integritási megszorítások nem sérülhetnek ezen műveletek alkalmazásakor.
- Számos adatbázis-frissítési művelet használható csoportosan.
- Számos ilyen művelet kiválthatja más hasonló művelet automatikus végrehajtását, amely azért szükséges, hogy az integritási megszorítások megőrződjenek.



## 3. előadás: Relációs modell

Ispány Márton

Alapfogalmak

Relációséma

Reláció

Megszorítások

Relációs  
adatbázisséma  
és adatbázis

Feltöltés

Az integritási megszorítások esetleges megsérülése esetén az alábbi akciók válthatnak ki:

- azon művelet törlése, amely a sérülést okozza (RESTRICT, REJECT opciók)
- a művelet végrehajtása de a felhasználó tájékoztatása a sérülésről
- további műveletek kiváltása (triggerek), amelyek helyrehozzák a sérülést (CASCADE, SET NULL opciók)
- felhasználó által írt hiba-korrigáló rutin futtatása



## INSERT művelet:

- **tartomány megszorítás:** ha az új rekord egyik attribútum értéke nem a megadott tartományba esik
- **kulcs megszorítás:** ha az új rekord kulcs attribútum értéke már létezik a reláció egy másik rekordjánál
- **hivatkozási integritás:** ha a külső kulcs érték az új rekordban egy olyan elsődleges kulcs értékre hivatkozik, amely nem létezik a hivatkozott relációban
- **egyedintegritás:** ha az elsődleges kulcs érték NULL az új rekordban

A DELETE művelet csak a hivatkozási integritási megszorításnál okozhat sérülést: olyan elsődleges kulcs értékkel bíró rekordot törölünk, amelyre más relációból hivatkozás van.

Az UPDATE művelet a tartomány megszorítás és a NULL érték megszorítást sértheti meg.