

Adatbázis kezelés I.

Adatbázis tervezés normalizálással - 1. rész

Rostagni Csaba

2025. január 6.

Ezen az órán... I

- 1 Anomáliák
- 2 Kulcsok
- 3 Normalizáció

Tartalom I

1 Anomáliák

- Beszúrási anomália
- Módosítási anomália
- Törlési anomália

Fogalmalmi különbségek

Adatbázis tervezés során

- a **tábla** helyett **reláció**
- a **mező** helyett **attribútum**

került alkalmazásra

Anomáliák

Definition (Anomália fogalma)

Az anomália az adatbázisban olyan rendellenesség, mely valamely karbantartási műveletnél plusz műveletek beiktatását igényli.

- Beszúrási anomália
- Módosítási anomália
- Törlési anomália

Redundancia

Definition (Redundancia)

Redundanciáról akkor beszélünk, ha valamely tényt vagy a többi adatból levezethető mennyiséget ismételten (többszörösen) tároljuk az adatbázisban.

- Lehet hasznos adatbiztonság szempontjából, például biztonsági mentés
- A fölösleges ismétlődést célszerű elkerülni
- Az idegen kulcs — bár redundánsan jelenik meg — a hozzá kapcsolódó adatokat takarja együttesen, így végeredményben csökkenti a redundanciát

Tartalom

1

Anomáliák

- Beszúrási anomália
- Módosítási anomália
- Törlési anomália

Beszúrási anomália

Definition (Beszúrási anomália)

Beszúrási anomáliáról beszélünk abban az esetben, amikor egy adatrekord beszúrása egy másik, hozzá logikailag nem kapcsolódó adatcsoport beszúrását kívánja meg.

Beszúrási anomália példa

`id`	`meret`	`tipus`	`ar`	`ingatlanos`	`fizetes`
1	47	lakás	86 0000	Péter	1600
2	47	ház	1 495 000	Mariann	4300
3	55	lakás	990 000	Zoli	6500
4	214	ház	1 250 000	Mariann	4300

- Az ingatlanokhoz logikailag nem kapcsolódik az ingatlanos fizetése
- Az, hogy ki árulja az ingatlant egy hasznos információ
- Mariann fizetése redundánsan szerepel a táblában

Beszúrási anomália példa

`id`	`meret`	`tipus`	`ar`	`ingatlanos`	`fizetes`
1	47	lakás	86 0000	Péter	1600
2	47	ház	1 495 000	Mariann	4300
3	55	lakás	990 000	Zoli	6500
4	214	ház	1 250 000	Mariann	4300
5	68	lakás	1 050 000	Mariann	4300

- Az ingatlanokhoz logikailag nem kapcsolódik az ingatlanos fizetése
- Az, hogy ki árulja az ingatlant egy hasznos információ
- Mariann fizetése redundánsan szerepel a táblában
- **Mariann fizetését is meg kell adni egy ingatlan új felviteléhez**

Tartalom

1 Anomáliák

- Beszúrási anomália
- **Módosítási anomália**
- Törlési anomália

Módosítási anomália

Definition (Módosítási anomália)

Abban az esetben, ha egy relációban egy adat módosítása több helyen történő módosítást igényel, akkor módosítási anomáliáról beszélünk.

Módosítási anomália példa

`id`	`meret`	`tipus`	`ar`	`ingatlanos`	`fizetes`
1	47	lakás	86 0000	Péter	1600
2	47	ház	1 495 000	Mariann	4300
3	55	lakás	990 000	Zoli	6500
4	214	ház	1 250 000	Mariann	4300
5	68	lakás	1 050 000	Mariann	4300

- A 4-es azonosítójú ingatlan eladása miatt jár 200 euró fizetésemelés az ingatlanosának

Módosítási anomália példa

`id`	`meret`	`tipus`	`ar`	`ingatlanos`	`fizetes`
1	47	lakás	86 0000	Péter	1600
2	47	ház	1 495 000	Mariann	4300
3	55	lakás	990 000	Zoli	6500
4	214	ház	1 250 000	Mariann	4500
5	68	lakás	1 050 000	Mariann	4300

- A 4-es azonosítójú ingatlan eladása miatt jár 200 euró fizetésemelés az ingatlanosának

Módosítási anomália példa

`id`	`meret`	`tipus`	`ar`	`ingatlanos`	`fizetes`
1	47	lakás	86 0000	Péter	1600
2	47	ház	1 495 000	Mariann	4300
3	55	lakás	990 000	Zoli	6500
4	214	ház	1 250 000	Mariann	4500
5	68	lakás	1 050 000	Mariann	4300

- Mivel az ingatlanos fizetése redundánsan van eltárolva, így **plusz műveleteket kell elvégezni**, a többi érték módosításához
- Ha minden "Mariann" nevű ingatlanos fizetését emeljük, akkor előfordulhatna, hogy egy **másik** személy kap emelést

Tartalom

1 Anomáliák

- Beszúrási anomália
- Módosítási anomália
- Törlési anomália

Törlési anomália

Definition (Törlési anomália)

Amennyiben egy adat törlésével másik, hozzá logikailag nem kapcsolódó adatcsoportot is elveszítünk, törlési anomáliáról beszélünk.

Törlési anomália példa

`id`	`meret`	`tipus`	`ar`	`ingatlanos`	`fizetes`
1	47	lakás	86 0000	Péter	1600
2	47	ház	1 495 000	Mariann	4300
3	55	lakás	990 000	Zoli	6500
4	214	ház	1 250 000	Mariann	4300
5	68	lakás	1 050 000	Mariann	4300

- A 1-es azonosítójú ingatlan tulajdonosával szerződést bontott az ügynökség, kerüljön törlésre

Törlési anomália példa

`id`	`meret`	`tipus`	`ar`	`ingatlanos`	`fizetes`
2	47	ház	1 495 000	Mariann	4300
3	55	lakás	990 000	Zoli	6500
4	214	ház	1 250 000	Mariann	4300
5	68	lakás	1 050 000	Mariann	4300

- A 1-es azonosítójú ingatlan tulajdonosával szerződést bontott az ügynökség, kerüljön törlésre
- Mennyi Péter fizetése?
- **Az ingatlan törlésével egy hozzá logikailag nem kapcsolódó adat (Péter fizetése) is törlésre került**

Tartalom I

2 Kulcsok

- Szuperkulcs
- Kulcs(jelölt)
- Elsődleges kulcs

Tartalom

2 Kulcsok

- Szuperkulcs
- Kulcs(jelölt)
- Elsődleges kulcs

Szuperkulcs

Definition (Szuperkulcs)

Szuperkulcsnak nevezzük azt az attribútumhalmazt, (melyek attribútumait együtt véve) egyértelműen meghatároz egy rekordot a relációban.

- Tartalmazhat olyan ("fölösleges") attribútumot, amit elhagyva is teljesíti a feltételeket

Example (*Személy* {személyi_szám, név, kor})

- Szuperkulcsok a *Személy* relációban
 - {személyi_szám} Önmagában elég a személyi_szám
 - {személyi_szám, név}
 - {személyi_szám, név, kor}
- Az alábbi attribútumhalmazok **nem** teljesítik a feltételt
 - {név}
 - {kor}
 - {név, kor}

Tartalom

2 Kulcsok

- Szuperkulcs
- Kulcs(jelölt)
- Elsődleges kulcs

Kulcs vagy Kulcsjelölt

Definition (Kulcs(jelölt))

Egy olyan (minimális) szuperkulcs, melynek bármely attribútumának eltávolítása után már nem szuperkulcs.

Example (*Autó* (rendszer, alvázszám, gyártó, típus, üzemanyag))

- A *rendszer* és az *alvázszám* önmagában is beazonosít egy autót

	Attribútumhalmaz	Szuperkulcs	Kulcs(jelölt)
1	{rendszer}	igen	igen
2	{alvázszám}	igen	igen
3	{rendszer, alvázszám}	igen	nem
4	{gyártó, üzemanyag}	nem	nem
5	{rendszer, típus, gyártó}	igen	nem
6	{alvázszám, gyártó, üzemanyag}	igen	nem

Tartalom

2 Kulcsok

- Szuperkulcs
- Kulcs(jelölt)
- Elsődleges kulcs

Elsődleges kulcs

- Az elsődleges kulcs
 - a kulcsjelöltek egyike lesz
 - az adatbázis tervezés során tetszőlegesen megválasztható
 - a relációban aláhúzással jelölendő
 - nem tartalmazhat ismétlődést
 - nem tartalmazhat NULL értéket
 - lehetőség szerint ne változzon meg, ha mégis akkor ez ne legyen gyakori
 - összetett kulcs helyett legyen egyszerű

Tartalom I

- 3 Normalizáció
 - Funkcionális függőség

Normalizáció

Definition (Normalizáció)

A normalizáció egy olyan adatbázis tervezési technika, ami csökkenti a redundanciát, elősegíti az anomáliák kiküszöbölését. A nagy táblákat több kicsire bontja és kapcsolatokat határoz meg.

Definition (Normálforma)

Adatbázis tervezés folyamatában többnyire egymásra épülő szabályok rendszerének egy eleme.

Tartalom

- 3 Normalizáció
 - Funkcionális függőség

Funkcionális függőség

$$R(A_1, A_2, A_3, \dots, B_1, B_2, \dots, B_n)$$

Definition (Funkcionális függőség)

Egy adott R relációban, egy B tartomány funkcionálisan függ az A tartománytól, ha bármely időpontban, minden egyes A értékhez egyetlen B érték tartozik az adott reláción belül.

Másképp megfogalmazva: Az A attribútumhalmaz értékei egyértelműen meghatározzák a B attribútumhalmaz értékeit.

Funkcionális függőség: Személyi szám és név kapcsolata

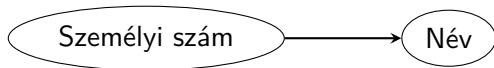
Vegyük a *Személy* (személyi_száma, név) relációt

A funkcionális függőség jelölése:

$$\{\text{személyi_szám}\} \rightarrow \{\text{név}\}$$

- Az attribútumhalmazokat kapcsolószerűjelekkel jelöljük
- A nyíl határozza meg a függés irányát, azaz a baloldali attribútum(ok) halmazától függ a jobb oldali attribútum(ok) halmaza

A funkcionális függőség vizualizálása:



- A személyi szám egyértelműen meghatározza a személy nevét
- Fordítva nem igaz, hiszen több embernek is lehet ugyanaz a neve, de a személyi számuk eltérő lesz

Funkcionális függőség: Személyi szám és név kapcsolata

A funkcionális függőségnek azért ez a neve, mert elméletben készíthetnénk olyan függvényt, ami tetszőleges, típusának megfelelő bemeneti adat esetén egyértelmű kimenetet eredményez.

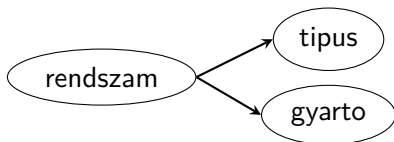
Python

```
def szemszambol_nev(szemszam):  
    if szemszam == "472278ZC":  
        return "Szentessy Péter"  
    elif szemszam == "258352DT":  
        return "Vasvári Mónika"  
    elif szemszam == "162633CX":  
        return "Vasvári Mónika"  
    ...
```


Funkcionális függőség példa: Autó

Vegyük az *Autó* (rendszer, gyártó, típus) relációt

rendszer	gyártó	típus
ABC-123	Opel	Astra
DEF-444	Vauxhal	Astra
PHP-404	VW	Jetta
ASD-365	VW	Polo

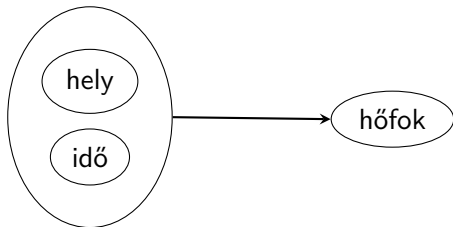


- $\{\text{rendszer}\} \rightarrow \{\text{típus}\}$
 - A rendszer meghatározza a típust
 - A típusból nem lehet a rendszert meghatározni (pl Astra)
- $\{\text{rendszer}\} \rightarrow \{\text{gyártó}\}$
 - A rendszer egyértelműen meghatározza a gyártót
 - A gyártóból nem határozható meg egyértelműen a rendszer (pl.: VW)
- $\{\text{rendszer}\} \rightarrow \{\text{típus, gyártó}\}$
 - A rendszer meghatározza a típust és a gyártót is. A fentieket magába foglalja, ez kell nekünk, ez látható az ábrán

Funkcionális függőség példa: Napi hőmérséklet

Magyarország egy nap alatt végrehajtott méréseit szeretnénk tárolni. Vegyük a *napi_hőmérséklet* (hely, időpont, hőmérséklet) relációt

hely	idő	hőfok
Győr	08:00	10
Győr	08:05	10
Pécs	08:00	10
Pécs	09:00	18



- A $\{\text{hely}, \text{időpont}\} \rightarrow \{\text{hőmérséklet}\}$ függőség teljesül
 - A hely **önmagában nem** határozza meg a hőfokot, mert ugyanazon a helyen, de másik időpontban lehet más a hőmérséklet
 - Az idő **önmagában nem** határozza meg a hőfokot, mert ugyanabban az időben lehet máshol más hőmérséklet
 - A hely és az idő **együttesen határozzák** meg a hőfokot

Triviális funkcionális függőség

Vegyük az *Autó* (rendszer, gyártó, típus) relációt

- $\{\text{típus}\} \rightarrow \{\text{típus}\}$
 - Saját magából megállapítható önmaga. Ez triviális
- $\{\text{típus, szín}\} \rightarrow \{\text{típus}\}$
 - Triviális egy funkcionális függőség, amennyiben a "bal oldali" halmaznak részhalmaza a "jobb oldali" attribútumhalmaz

Ezek a triviális függőségek a kivételektől eltekintve nem lesznek hasznunkra adatbázis tervezésnél, így a legtöbb esetben elhagyhatóak.

Funkcionális függőség: Személy

Vegyük a *Személy* (név, szül_idő, szül_hely, anyja_neve, cim, tel) relációt

név	szül_idő	szül_hely	anyja_neve	cim	tel
T. Péter	1986-01-05	Bp	K. Mária	Bp.	1234569
T. Péter	1986-01-05	Bp	M. Emese	Győr.	5525359
T. Péter	1999-10-12	Győr	K. Mária	Bp.	1122339

• Függőségek

- {név, szül_idő, szül_hely, anyja_neve, **cim**, **tel**} → {**cim**, **tel**}
- {név, szül_idő, szül_hely, anyja_neve, **cim**} → {cim, **tel**}
- {név, szül_idő, szül_hely, anyja_neve, **tel**} → {cim, **tel**}
- {név, szül_idő, szül_hely, anyja_neve} → {cim, tel}

Funkcionális függőség: Személy

$$\{\text{név, szül_idő, szül_hely, anyja_neve}\} \rightarrow \{\text{cim, tel}\}$$

- A fenti függőség az összes mezőt tartalmazza
- A függőség bal oldala minimális kulcs, más nincs
- Könnyű egy egy elemű halmazból **elsődleges kulcsot** kinevezni
- Mivel több attribútumból tevődik össze, így **összetett kulcs** lesz

Funkcionális függőség: Személy

Vegyük az alábbi relációt

Személy (személyi_száma, név, szül_idő, szül_hely, anyja_neve, cím, tel)

személyi_száma	név	szül_idő	szül_hely	anyja_neve	cím	tel
1-860105-7825	T. P.	1986-01-05	Bp	K. Mária	Bp.	1234569
1-860105-2353	T. P.	1986-01-05	Bp	M. Emese	Győr.	5525359
1-901012-4581	T. P.	1999-10-12	Győr	K. Mária	Bp.	1122339

- Nem triviális funkcionális függőségek:

- {személyi_száma} \rightarrow {szül_idő}
- {személyi_száma} \rightarrow {név, szül_idő, szül_hely, anyja_neve, cím, tel}
- {név, szül_idő, szül_hely, anyja_neve} \rightarrow {cím, tel}

Funkcionális függőség: Személy

- Vizsgáljuk meg az alábbi függőségeket
 - $\{\text{név, szül_idő, szül_hely, anyja_neve}\} \rightarrow \{\text{szemelyi_szám, cím, tel}\}$
 - $\{\text{szemelyi_szám}\} \rightarrow \{\text{név, szül_idő, szül_hely, anyja_neve, cím, tel}\}$
- Mind a két függőségre igaz, hogy a reláció összes attribútuma szerepel valamelyik oldalán
- A $\{\text{név, szül_idő, szül_hely, anyja_neve}\}$ attribútumhalmaz egy olyan **szuperkulcs**, ami minimális, így **kulcs(jelölt)** is egyben
- A $\{\text{szemelyi_szám}\}$ attribútumhalmaz egy olyan **szuperkulcs**, ami minimális, így **kulcs(jelölt)** is egyben
- Válasszuk azt a kulcsjelöltet, amelyik
 - nem, vagy csak ritkán változik
 - egyszerű
- Mivel a $\{\text{szemelyi_szám}\}$ egy attribútumból áll, így az lesz az **egyszerű kulcs** lesz

Funkcionális függőség példa: Irányítószám és település

Vegyük az *Cim* (irsz, telepules) relációt

irsz	település
1015	Budapest
1033	Budapest
1213	Budapest
6710	Szeged
6726	Szeged
9400	Sopron
9444	Fertőszentmiklós

irsz	település
8984	Gombosszeg
8984	Petrikeresztúr
8984	Iborfia

- Megállapítható -e a település nevéből az irányítószám? Nem
- Megállapítható -e az irányítószámból a település neve? Nem
- $\{\text{irsz}\} \rightarrow \{\text{település}\}$
 - Klasszikus példa, **DE Magyarországra nem érvényes**
- Melyik település irányítószáma 1015? Bécs
 - Európai adatbázisban sem állja meg a helyét