MTA Számitástechnikai és Automatizálási Kutató Intézete, **Közlemények** 20/1978.

RELÁCIÓS ADATBÁZIS MODELL

Demetrovics János

1. Bevezetés

Az adatbázis a valóságról adatokat tartalmaz. Az adatok a valóság objektumainak tulajdonságaira vonatkozhatnak, s ezek közül a tulajdonságok közül a kapcsolatot jelentő tulajdonság az adatbázisban sok szemléleti problémát okoz, egészen az "adatok" és "kapcsolatok" megkülönböztetéséig. A probléma kialakulása a fejlődés következménye: a homogén rekordokból álló file-ok a valóság objektumait a rekordoknak feleltették meg, kapcsolat ábrázolására minimális információt tartalmaznak. A közvetlen hozzáférésű tárak adták meg a valóság objektumai közötti kapcsolatok ábrázolásának lehetőségeit, rekordok közötti kapcsolatok formájában. Igy alakultak ki a kapcsoló adatok, s a valóság objektumai közötti kapcsolat igy vált rekordok közötti kapcsolattá. A következőkben ez a kettősség nem lesz zavaró, ha leszögezzük, hogy a kapcsolat ábrázolására az adatbázisban több lehetőségünk is lehet, s az adatbázison belüli kapcsolaton mindig az adatbázis objektumai közötti kapcsolatot értünk.

Az adatbáziskezelő rendszereket többféleképpen szoktuk osztályozni, de a leginkább használatos osztályozás éppen a szerint történik, hogy a felhasználó szempontjából miként valósul meg a kapcsolatokkal való manipulálás. E szerint három adatbázis rendszer tipust különböztetünk meg: hierarchikust, hálózatost és relációst.

A relációs megközelitésben a kapcsolatokat ugyanugy ábrázoljuk, mint a valós világ többi adatát, azaz n-esek segitségével. Ez azt jelenti, hogy a relációs adatmodell a valóságot és a kapcsolatot ugyanazon tipusu objektumnak tekinti. A relációs adatbázis modell nem a gépi oldal lehetőségeit igyekszik kihasználni, illetve bőviteni, hanem a felhasználó szempontjából indul ki és olyan eszközöket igyekszik kialakitani, amelyek segitségével ugy kerülünk a számitógéphez közelebb, hogy nincs szükség ujabb, magasabb szintü ismeretek elsajátitására. A hagyományos adatbázis modell a gépi hatékonyságot tekinti elsődlegesnek, mig a relációs adatmodell az adatfüggetlenséget még inkább kihangsulyozza és fontosnak tartja.

A hagyományos adatbázis modell a gépi hatékonyság lehetőségeit általában ugy tudja növelni, hogy az emberi oldalon valamilyen módon növeli a bonyolultságot. Ez álatában a felhasználók körének szükülését vonja maga után.

A relációs adatmodell a gép hatékonyságát másodlagosnak tekinti és ezért az adatokat a lehető legtermészetesebb formában, mátrix alakban szemléli.

Dolgozatunkban a relációs adatmodellről kivánunk egy áttekintést nyujtani.

2. Relációs adatbázis modell

A relációs adatbázis modell elmélete időben legkésőbb fejlődött ki. 1968-ban Childs [1] vetette fel a relációs elmélet adatbázis modellezésben történő felhasználását, amelyet

E.F. Codd [2,3,4] a későbbi években jelentős részben meg is valósitott.

A relációs adatbázis modell a relációk matematikai elméletén alapszik. Ez komoly elméleti alapot ad az adatbázis modell számára és a relációs elmélet számos eredményét alkalmazni lehet olyan fontos problémák megoldására, mint például az adatbázis kezelő /al/nyelv [5,6,7] megkonstruálása. Másrészről azonban a felhasználónak szembe kell nézni azzal a ténnyel, hogy uj kifejezéseket is meg kell tanulni a már eddig is használt hasonló fogalmakra.

A relációs adatbázis azon az alapvető törekvésen alapszik, hogy a felhasználó számára az adatokat a lehető legszemléletesebb módon tárgyalja. Nyilvánvaló, hogy a legszemléletesebb adatszerkezet a táblázat, vagyis egy olyan mátrix amelynek m sora és n oszlopa van (m x n). Lényegében a relációs adatbázis modell t darab két dimenziós mátrixból áll: m_i x n_i , $1 \leqslant i \leqslant t$.

Matematikailag a relációt a következőképpen értelmezzük:

Definició. Legyenek adva $D_1, D_2, \ldots, D_n$ halmazok, amelyek nem feltétlenül különbözőek. Azt mondjuk, hogy R egy reláció ezen n halmaz D_1 x D_2 x . . . x D_n direkt szorzatán, ha létezik, olyan $g(x_1, x_2, \ldots, x_n)$ predikát, melynek segitségével minden szám n-esről $(a_1, a_2, \ldots, a_n)$ el tudjuk dönteni, hogy $(a_1, a_2, \ldots, a_n) \epsilon R$ vagy pedig $(a_1, a_2, \ldots, a_n) \bar{\epsilon} R$, ahol $a_i \epsilon D_i$, $1 \leqslant i \leqslant n$.

Más szóval az $R = \{ (a_1, a_2, \ldots, a_n) : a_i \epsilon D_i \quad (i = 1,2, \ldots, n)$ és $g(a_1, a_2, \ldots, a_n) = 1$ halmazt relációnak nevezzük. D_i a reláció i-edik értelmezési tartománya, és az értékkészlet nem más mint a D_i halmazban lévő elemek. n-t az R reláció fokának, m-t pedig az R reláció számosságának nevezzük, ahol $m = |R|$ és m a relációban lévő n-esek számát jelenti.

Az igy definiált R relációt részletesebben jelöljük a következőképpen $R(D_1, D_2, \ldots, D_n$ Vagyis R a reláció neve, a zárójelben pedig az értelmezési tartományok nevei vannak feltüntetve. Ha az értelmezési tartományok közül bizonyosakat ki kivánunk tüntetni, akkor azokat aláhuzzuk.

DIÁK (<u>AZONOSITÓ</u>, NÉV OSZTÁLY)

AZONOSITÓ	NÉV	OSZTÁLY
D1	FEKETE	3
D2	KISS	3
D3	KISS	4
D4	KESERÜ	5

1. ábra

Az 1. ábra a DIÁK relációt mutatja, amely az AZONOSITÓ, NÉV, és az OSZTÁLY tartományokon van értelmezve. Az OSZTÁLY tartomány például a 3,4 és 5 értéket tartalmazza vagyis azt, hogy a diákok melyik osztályba járnak. A DIÁK reláció számossága 4, foka pedig 3.

Amint az 1. ábra is mutatja, a relációt legegyszerübben táblázatok segitségével tudjuk ábrázolni, ahol mindegyik sor n adatot tartalmaz. Vegyük észre, hogy az adatsor illetve adatoszlop is táblázat.

A táblázat oszlopai az adott egyed tipusra jellemző tulajdonságokat tartalmazzák, a sorai pedig konkrét előfordulásokat határoznak meg, az érték készlet konkrét értékeivel. Vagyis a két dimenziós mátrix esetén egy sor és egy oszlop találkozásában egyetlen érték található, amely egy konkrét tulajdonság konkrét értékét adja meg. Meg kell jegyezni, hogy ez az érték nulla vagy üres elem is lehet.

Definició. Az olyan értelmezési tartományt vagy értelmezési tartományok halmazát, melyek értékei egyértelmüen azonositják a relációt, a reláció elsődleges kulcsának nevezzük.

Az 1. ábrán bemutatott DIÁK relációból észrevehetjük a relációk főbb tulajdonságait.

A táblázat tanulmányozásánál vegyük észre a következő tulajdonságokat:

1.) Nincs két azonos sor.
Ez a tulajdonság azt a természetes követelményt takarja, hogy két különböző egyed előfordulása legalább egy adott tulajdonságban el kell, hogy térjen egymástól.

2.) A sorok sorrendje lényegtelen.
A sorokra egy vagy több azonos értelmezési tartomány konkrét értékével hivatkozunk ami valójában nem más mint az elsődleges kulcs. Az 1.) tulajdonságból következik, hogy elsődleges kulcs mindig létezik, legrosszabb esetben az egész sor az elsődleges kulcs.

A továbbiakban az elsődleges kulcs értelmezési tartományait alá fogják huzni.

Az 1. ábrán lévő DIÁK relációban minden diák kap egy azonositót, amely egyértelmüen azonositja a diákot. Vegyük észre, hogy a NÉV és OSZTÁLY értelmezési tartományok nem feltétlenül azonositják a DIÁK reláció konkrét sorait. Ugyanez az eset áll fenn a 2. ábrán megadott TANÁR reláció esetében is. A TANÁR-DIÁK reláció esetén (3.ábra) két értelmezési tartomány van az AZONOSITÓ-T és az AZONOSITÓ-D.

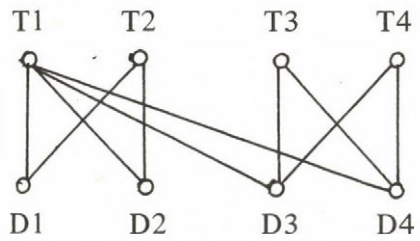
TANÁR (AZONOSITÓ, NÉV, TANTÁRGY)

AZONOSITÓ	NÉV	TANTÁRGY
T1	SOLYOM	ANGOL
T2	NAGY	FIZIKA
T3	FEHÉR	FIZIKA
T4	NAGY	MAGYAR

2. ábra

TANÁR–DIÁK (AZONOSITÓ-T, AZONOSITÓ-D)

AZONOSITÓ-T	AZONOSITÓ-D
T1	D1
T1	D2
T1	D3
T1	D4
T2	D1
T2	D2
T3	D3
T3	D4
T4	D3
T4	D4



3. ábra

Ezek a tartományok a TANÁR, illetve a DIÁK relációban valóban azonositottak egy-egy tanárt, illetve diákot, azonban a TANÁR-DIÁK relációban csak az együttes értékük (vagyis a teljes sor) azonosit egy-egy sort, amely értelemszerüen nem más, mint az, hogy ki kit tanit. Vegyük észre, hogy a TANÁR-DIÁK reláció segitségével valósitjuk meg azt az $M : N$ kapcsolatot, ami a tanárok és diákok között van. Vagyis azt, hogy egy tanár sok diákot tanit és egy diák több tanártól tanul.

Definició. Ha az elsődleges kulcs több értelmezési tartományból áll, akkor összetett kulcsról beszélünk.

3. Az oszlopok sorrendje lényegtelen

Az oszlopokra mindig az értelmezési tartomány nevével hivatkozunk és sohasem az oszlop poziciójával. Minden sorban az azonos tulajdonságok azonos helyen szerepelnek.

Nehézség merül fel abban az esetben, ha egy R relációban ugyanaz az értelmezési tartomány többször szerepel. Ilyenkor ezeknek az értelmezési tartományoknak a neveit megkülönböztetjük valami jellel. Például a TANÁR-DIÁK relációban a TANÁR és a DIÁK relációból kiindulva értelemszerüen kétszer kellene, hogy az AZONOSITÓ értelmezési tartomány szerepeljen. A bonyodalmak elkerülése végett a TANÁR reláció AZONOSITÓ-ját a TANÁR-DIÁK relációban AZONOSITÓ-T-nek hivjuk, mig DIÁK reláció AZONOSITÓ-ját a TANÁR-DIÁK relációban AZONOSITÓ-D-nek. Vegyük észre, hogy az AZONOSITÓ-D-t jelölhettük volna egyszerüen AZONOSITÓ névvel is.

Megjegyzés. Codd müveiben, s több más dolgozatban is az oszlopok sorrendje lényeges. Ezekben a dolgozatokban kihasználják a reláció oszlopainak a pozicióját.

3. Funkcionális függőség

Amikor egy R relációt jelentés szerint vizsgálunk, akkor bizonyos értelmezési tartományok közt függőséget tapasztalhatunk.

Ezek közül hármat kivánunk részletesebben megvizsgálni a funkcionális függőségét, a teljes funkcionális függőséget és a tranzitiv függőségét.

Definició. Egy adott R reláció Y értelmezési tartománya funkcionálisan függ az X értelmezési tartománytól, akkor és csak akkor, ha minden R-beli Y értéket meg határoz. Ezt lehet formálisan definiálni;

Ha $R = R(\ldots, X, \ldots, Y, \ldots)$ és $\tilde{a}_1 \in R$, $\tilde{a}_2 \in R$, ahol $\tilde{a}_1 = (a_{11}, a_{12}, \ldots, a_{1n})$ $\tilde{a}_2 = (a_{21}, a_{22}, \ldots, a_{2n})$, akkor létezik olyan $Y = f(X)$ függvény, hogy $f(a_{ix}) = a_{iY}$ $(i = 1,2)$.

Másszóval ez azt jelenti, hogy az Y tulajdonság értékét az X tulajdonság értéke adja meg.

Megjegyzés. A funkcionális függőség vonatkozhat összetett tartományra is, azaz $X = (X_1, X_2, \ldots, X_m)$, ahol $R = R(\ldots, X_1, \ldots, X_m, \ldots, Y, \ldots)$.

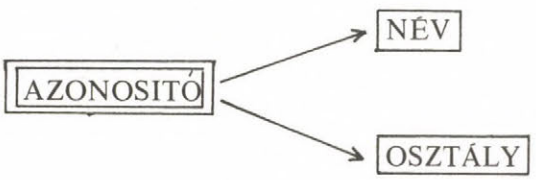
Definició. Legyen X összetett tartomány. Teljes függőségéről beszélünk, ha az Y értelmezési tartomány funkcionálisan függ az X tartománytól, de nem függ az X semmilyen valódi részhalmazától sem. Továbbiakban funkcionális függőség alatt mindig funkcionális teljes függőséget értünk, hacsak nem hangsulyozzuk ki, hogy funkcionális nem teljes függőségre gondoltunk.

Megjegyzés. Nyilvánvaló, hogy az elsődleges kulcstól minden más értelmezési tartomány funkcionálisan – esetleg teljes módon – függ. Amennyiben a kulcs nem összetett úgy a funkcionális függőség mindig funkcionális teljes függőséget jelent. Bizonyos relációkban léteznek olyan tartományok, amelyek az adott relációban nem elsődleges kulcsok, de egy másik relációban ezek a tartományok egyértelmüen azonosithatnak bizonyos tartományokat. Ezeket a tulajdonságokat reprezentáló értékeket idegen kulcsoknak nevezzük.

Definició. Az R reláció Z értelmezési tartománya tranzitive függ az X-től, ha Z funkcionálisan függ az X elsődleges kulcstól, és függ az Y idegen kulcstól is, ami természetszerüen függ az X elsődleges kulcstól.

Illusztráljuk példánkon keresztül ezeket a függőségeket.

A DIÁK relációban a NÉV és OSZTÁLY tartományok funkciónálisan függnek a DIÁK reláció AZONOSITÓ tartományától.



4. ábra

Ezeket a funkcionális függőségeket diagrammal ábrázolhatjuk, amint azt a 4. ábra mutatja.

A funkcionális függések felismerése az egyik legfontosabb és legkényesebb része az adatok szemantikájának. Csak a függések teljes feltárása után mondhatjuk azt, hogy az adatokat teljesen megértettük.

4. A relációk normalizálása

A relációk normalizálását illetve a normálformáinak fogalmát Codd vezette be, ő határozta meg az 1., 2. és a 3. normálforma tulajdonságait is [2, 5, 6].

Nézzük az MTA INTÉZET relációt (5. ábra)

MTA INTÉZET (M#, NÉV, TELEPHELY, CIM)

M	NÉV	TELEPHELY	CIM
M1	MTA SzTAKI	FŐ ÉPÜLET VÁR SZÁMITÓKÖZPONT	KENDE URI VICTOR HUGO
M2	MTA KFKI	FŐ ÉPÜLET CSILLAGVIZSGÁLÓ	KONKOLY KONKOLY

5. ábra

Rendes mátrix formában ezeket az adatokat nehezen tudjuk ábrázolni. A nehézség abból az egyszerü tényből adódik, hogy az értelmezési tartományban lévő értékek nem elemiek, hanem tovább lehet bontani kisebb elemekre.

Definició. Az R relációt 1 normálformájunak nevezzük, ha a relációban szereplő minden érték elemi.

Könnyü belátni, hogy minden relációt 1 normálformára lehet hozni. Ennek érdekében minden ismétlődő elemnél meg kell ismételnünk az egyeszeres elemeket.

Az MTA INTÉZET reláció 1 normálformáju alakját a 6. ábra mutatja:

M#	NÉV	TELEPHELY	CIM
M1	MTA SzTAKI	FŐ ÉPÜLET	KENDE U
M1	MTA SzTAKI	SZÁMITÓKÖZPONT	VICTOR H
M1	MTA SzTAKI	VÁR	URI U.
M2	MTA KFKI	FŐ ÉPÜLET	KONKOLY
M2	MTA KFKI	CSILLAGVIZSGÁLÓ	KONKOLY

M#
TELEPHELY → NÉV
CIM

6. ábra

A továbbiakban az 1 normálformáju relációt egyszerüen normalizáltnak nevezzük.

Megjegyzés. A relációs adatbázis modell csak a normalizált (1NF) relációkat tartalmazhat. Ellenkező esetben felvetődik az összes olyan probléma, ami a hierarchikus adatbázis modellal kapcsolatos, mivel egy normalizálatlan reláció analóg a hierarchikus rekordok file-jával.

Most már pontosabban meg tudjuk határozni a relációs adatmodellt. Felhasználói szempontból nézve ez nem más mint az osztályozott foku (n) normalizált relációk (1NF) időben változó halmaza. Az idő-variáns fogalmat azért kell bevezetnünk, mert igy valósitható meg a szám n-esek módositása, törlése, beszurása stb. A szám n-es valójában egy rekodnak felel meg.

Könnyü belátni, hogy a hagyományos terminológia szerint a reláció megfelel egy (homogén) file-nak, a rekordnak egy szám n-es stb. Vagyis a hagyományos logikai adatállományoknál létező fogalmak megfelelői (rekordtipus, rekordelőfordulás, tulajdonság, konkrét érték, azonositó stb.) a relációs adatmodellnél is megtalálhatók. Ezek a megfeleltetések csak közelitések.

Csak néhány példát emelünk ki az eltérések közül:
A file-ok rendszerint szekvenciálisak;
előfordulhat, hogy több rekordnak is ugyanaz az azonositója; szerepelhetnek adatcsoportok; stb.

Definició. Az 1 normálformájú *R* reláció 2 normálformáju akkor és csak akkor, ha minden olyan értelmezési tartomány, amely nem elsődleges kulcs vagy annak a része, funkcionálisan függ az elsődleges kulcstól.

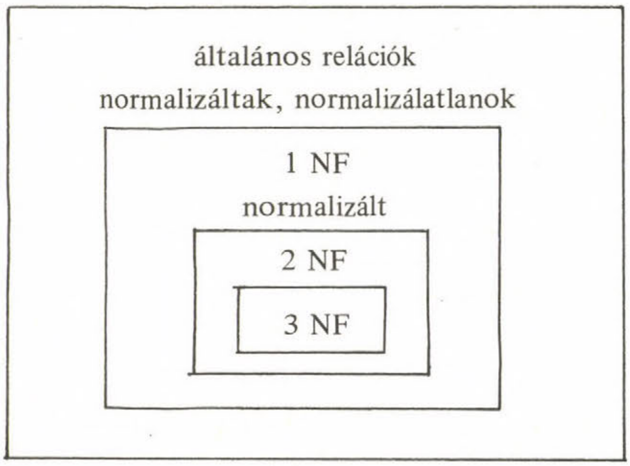
Megjegyzés. Könnyü belátni, hogy ha az elsődleges kulcsot egyetlen egy értelmezési tartomány határozza meg, azaz maga a kulcs egyszerü, akkor minden 1 normálformáju reláció 2. normálformáju is. Ez következik abból az egyszerü tényből, hogy ilyenkor a függőség mindig funkcionális teljes függőség is.

Ez a megjegyzés értelmében a DIÁK és TANÁR relációk második normálalakuak. Könnyü belátni, hogy bár a TANÁR-DIÁK reláció kulcsa összetett, a reláció mégis második normálformában van.

Definició. Második normálformában lévő *R* reláció harmadik normálformában van, akkor és csak akkor, ha nem tartalmaz tranzitiv függőséget.

5. **A normálformák előnyei**

A Codd által definiált normálformák hierarchiáját a 7. ábra mutatja be.



7. ábra

Az eddig elmondottakat egy példán keresztül illusztráljuk, s közben rámutatunk a normálfotmák jelentőségére, illetve előnyeire.

Egy boltban minden nap zárás után a következő adatokat jegyzik fel (NAPI HELYZET): mennnyi volt az aznapi bevétel (ÖSSZEG), s ezt egy azonositóval (DÁTUM)-al is ellátják. Milyen árukból (ÁRUNÉV) mennyit adatak el (DARAB). Az áru ÁRUKOD-al el van látva, s fel van tüntetve az ára (EGYSÉGÁR). Továbbá regisztrálják azt, hogy történt-e pénz szállitás vagy nem. Amennyiben a napi bevétel (ÖSSZEG) 10.000,-Ft alatt van, akkor nem történik pénzszállitás a bankba, ellenkező esetben azonban igen.

NAPI HELYZET (DÁTUM, ÁRUKOD,

ÁRUNÉV, EGYSÉGÁR, ÖSSZEG, SZÁLLITÁS, DARAB)

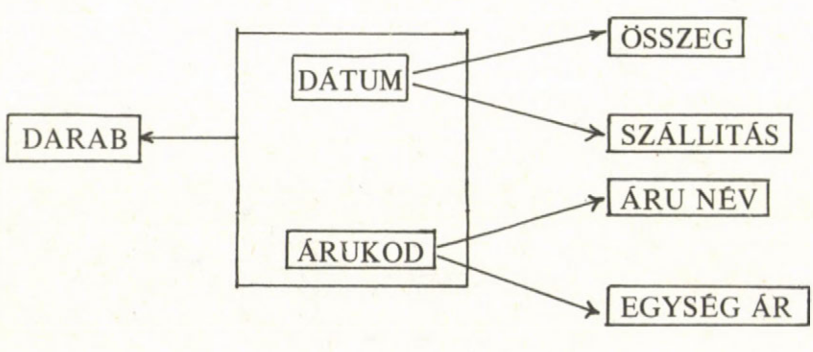
Könnyü belátni, hogy a NAPI HELYZET reláció sorait sem a DÁTUM, sem pedig az ÁRUKOD konkrét értékei nem azonositják egyértelmüen, mivel naponta több árut adhatnak el.

Ebből következik az is, hogy a relációk normalizátlan. (Az ÁRUKOD, ÁRUNÉV, EGYSÉGÁR, DARAB értelmezési tartományokban nem biztos, hogy elemi érték szerepel.) Ha ezekben a nem elemi értékeket tartalmazó értelmezési tartományokban az értékeket egymás alá irjuk és a DÁTUM, ÖSSZEG, SZÁLLITÁS megfelelő értékeit értelemszerüen duplikáljuk., akkor a NAPI HELYZET reláció első normálformájú lesz, amelynek összetett kulcsa lesz (DÁTUM, ÁRUKOD).

DÁTUM	ÁRUKOD	ÁRUNÉV	EGYSÉGÁR	ÖSSZEG	SZÁLLITÁS	DARAB
19770915	A1	RÁDIÓ	1000	10000	IGEN	1
19770915	A3	TELEVIZIÓ	4000	10000	IGEN	2
19770915	A6	KERÉKPÁR	1000	10000	IGEN	1
19770916	A2	MOSÓGÉP	3000	6000	NEM	2
19770917	A1	RÁDIÓ	1000	15000	IGEN	3
19770917	A4	MAGNETOFON	2000	15000	IGEN	5
19770917	A9	FÉNYKÉPEZŐ	2000	15000	IGEN	1

8. ábra

A 9. ábrán a NAPI HELYZET reláció funkcionális függőségeit ábrázoljuk.



9. ábra

Megjegyzés. Vegyük észre, hogy a funkcionális nem teljes függőség ellenőrzése nem más, mint annak a ténynek a megvizsgálása, hogy a feltételezett kulcs valóban elsődleges kulcs-e.

A NAPI HELYZET reláció nem második normálformáju, mivel bizonyos értelmezési tartományok függnek az összetett kulcs egyes részeitől is.

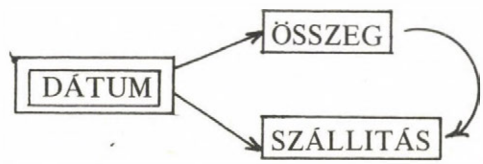
Álatlában nem állunk meg az 1. normálformánál, hanem a relációkat tovább bontjuk, s ezért van, mert az 1. normálalak több "rendellenességet" tartalmaz. Ezeket a kapcsolatokat szüri ki a 2. illetve 3. normálforma. Ezeknek a bevezetése "megtisztitja" az 1. normálformát a három legjelentősebb nehézkeségtől: a törlési-, a hozzáadási- és a felujitási adatfüggőségtől.

Ezekre az adatfüggőségekre még visszatérünk a 2. illetve 3. normálformára bontás után.

Könnyü belátni, hogy a NAPI HELYZET relációt 3 darab 2. normálformáju relációra bonthatjuk, amelyeket a 10. ábra illusztrál.

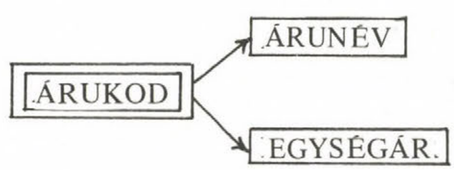
BEVÉTEL (DÁTUM, ÖSSZEG, SZÁLLITÁS)

DÁTUM	ÖSSZEG	SZÁLLITÁS
19770915	10000	IGEN
19770916	6000	NEM
19770917	15000	IGEN



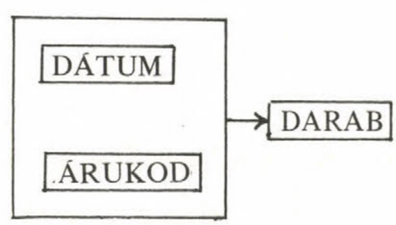
ÁRU (ÁRU-KOD, ÁRU-NÉV, EGYSÉGÁR)

ÁRU-KOD	ÁRU-NÉV	EGYSÉGÁR
A1	RÁDIO	1000
A2	MOSOGÉP	3000
A3	TELEVIZIO	4000
A4	MAGNETOFON	2000
A6	KERÉKPÁR	1000
A9	FÉNYKÉPEZŐGÉP	2000



MENNYISÉG (DÁTUM, ÁRU-KOD, DARAB)

DÁTUM	ÁRUKOD	DARAB
19770915	A1	1
19770915	A3	2
19770915	A6	1
19770916	A2	2
19770917	A1	3
19770917	A4	5
19770917	A9	1



10. ábra

Az ÁRU és MENNYISÉG reláció 3. normál formáju is, de a BEVÉTEL reláció a tranzitiv függőség miatt még nem az. Ha meg akarjuk szüntetni, akkor 2 darab relációra kell bontani (11. ábra), amelyek már 3. normál formában lesznek.

NAPI BEVÉTEL (DÁTUM, ÖSSZEG)

DÁTUM	ÖSSZEG
19770915	10000
19770916	6000
19770917	15000

$\boxed{\text{DÁTUM}} \rightarrow \boxed{\text{ÖSSZEG}}$

SZÁLLITÁS (ÖSSZEG, SZÁLLITÁS)

ÖSSZEG	SZÁLLITÁS
10000 <	NEM
10000 ⩾	IGEN

$\boxed{\text{ÖSSZEG}} \rightarrow \boxed{\text{SZÁLLITÁS}}$

11. ábra

Végül megállapithatjuk, hogy a NAPI HELYZET relációt 4 darab 3, normálformáju relációra tudtuk bontani (ÁRU, MENNYISÉG, NAPI BEVÉTEL, SZÁLLITÁS), amelyek tartalmazzák mindazt az információt melyet a NAPI HELYZET reláció tartalmazott.

A relációs adatbázis legfontosabb pontja a normalizálás. Legjobban ezt a tényt az bizonyitja, hogyha megengednénk normalizálatlan relációk használatát, akkor megnyitnánk a kaput az összes olyan probléma előtt, ami a hierarchikus közelitéssel kapcsolatos, mivel a normalizálatlan reláció analóg a hierarchikus rekordok több vagy kevesebb fileval.

Könnyen belátható, hogy a funkciónális függetlenséget nem tartalmazó második normálforma lényeges előnyökkel rendelkezik az első normálformávall szemben.

Nézzük meg ezek közül elsősorban azokat amelyek kapcsolatban vannak az adatfüggőséggel.

A felfrissitési adatfüggőség problémáját az alábbi példa szemlélteti. Tételezzük fel, hogy hónap közben visszamenőleg modositani kell a NAPI HELYZET relációban az ÁRUKOD tartomány konkrét értékeit. Ez a feladat jelentős feldolgozási igényt jelent, mivel a NAPI HELYZET reláció minden egyes sorát el kell olvasni, s meg kell vizsgálni, hogy az ÁRUKOD aktuális értéket kell-e módositani. Nyilvánvaló, hogy ez a megoldás lényegesen több munkát és időt igényel, mint a második normál formában levő ÁRU relációban történő változtatások átvezetése.

Vegyük észre, azt, hogy ha a modositandó tulajdonság idegen kulcs, akkor az adatkapcsolatok helyessége is veszélyben van, mivel ez az érték az adatbázis más relációjában is szerepel.

Tehát megállapithatjuk, hogy a második normálformában lévő relációkban minden egyes módositás lényegesen könnyebben és gyorsabban hajtható végre, mint első normál formában.

A törlési adatfüggőséggel kapcsolatos problémákat azonnal láthatjuk, hogyha a NAPI HELYZET relációból az 19770916A2 kulcsu sort. Ez ugyanis azt jelenti, hogy a mosógépre vonatkozó alapvető információinkat (ára, kódja) is elveszitettük, pedig a mosógépre vonatkozó adatokat még tárolunk. Ezzel szemben az ugyanilyen azonositóju sor törlése a MENNYISÉG relációból nem jelenti a mosógépre vonatkozó alapvető információk elvesztését, mivel az az ÁRU relációban került tárolásra.

Megállapitjuk tehát, hogy a második normálforma használatánál – szemben az első normálformával – törlés esetén nem vész el lényeges információ, amely az első normálforma esetén esetleg elveszett volna. A hozzáadási adatfüggőség a szemléltetésére vegyük azt a példát, hogy a NAPI HELYZET relációhoz hozzá kell adni egy új információt a lemezjátszóról (ÁRU-KOD: A8, EGYSÉG: 2000) amit az árusitók már tudnak mivel árulják a lemezjátszót, de a NAPI HELYZET relációba nem tudjuk beirni, mert még senki nem vett egyetlenegy lemezjátszót sem, s igy nincs elsődleges kulcsunk erre a szám-7-es-re.

A második normálformában levő ÁRU relációban minden további nélkül beilleszthető a lemezjátszóval kapcsolatos új információink.

Megállapithatjuk, hogy a második normálforma estén a relációk konzisztensen bővithetők anélkül, hogy az adatbázisba hamis adatokat vinnénk be.

Vegyük észre, hogy a két utóbbi adatfüggőséggel kapcsolatos problémáink szorosan összefüggnek a relációink terjedelmével.

A második normálforma a felsorolt adatfüggőségi előnyein túl számos más előnnyel is rendelkezik, ezek közül pl. jelentős az, hogy egyszeresen szerepelnek azok az adatok, amelyek az első normálforma esetén feleslegesen ismétlődnek (ÁRUNÉV, EGYSÉGÁR). Mivel a második normálforma bevezetésével megszünt az ilyen eredetü redundancia, ezért a relációink együttes mérete csökkent (általában!), az eredeti relációval szemben. Igy kisebb a veszély az inkonzisztens adatok tekintetében.

Könnyü belátni, hogy a harmadik normálforma előnyei a második normálformával szemben hasonlóak az előbb elmondotakhoz. Ez azzal a ténnyel magyarázható, hogy a harmadik normálforma megoldja a második normálformában levő tranzitiv függőségek eliminilását is.

Irodalom

[1] D.L. Childs: Description of a Set-Theoretic Data Structure. Proc. FJCC, Vol. 33, Part 1, December 1968. (557-564).

[2] E.F. Codd: A relational model of date for large shared data banks, CACM 13 (1970) 377-387.

[3] E.F. Codd: Normalized data base structure: A brief tutorial, Proc. 1971. ACM-SIGFIDET Workshop on Data Description, Access and Control.

[4] E.F. Codd: Further normalization of the data base relational model, Courant Computer Science Symposia 6 "Data Base System", (Prentice Hall, Englewood Cliffs, N.J., 1971) 33-64.

[5] C.J. Date: An introduction to database systems, Addison-Wesley, Reading, Mass. 1975.

[6] Halassy Béla: Adatbázisok számitógépes kezelésének alapvatő kérdései. SZÁMOK 1978.

[7] Gyürki József: Modern számitógépes adatbázis kezelő software rendszerek és nyelvek. BME Jegyzet 1976.

Summary

The relational data base model

János Demetrovics

In the present paper the author gives a survey on relational data base structures which is mainly base on Codd's approach.

РЕЗЮМЕ

Релационная модель баз данных

Янош Деметрович

В настоящей работе лается некоторый обзор баз данных о нормалформах по работам E.F. Codd.