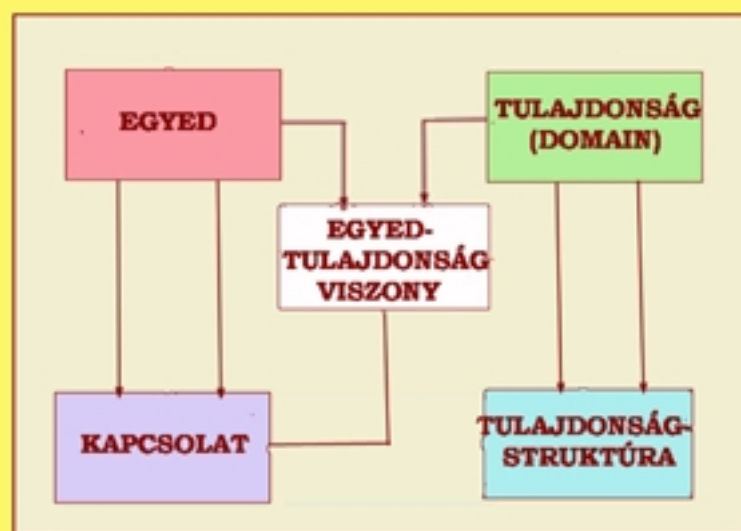


HALASSY BÉLA



ADATMODELLEZÉS FELSŐFOKON

DOKTORI DOLGOZAT (2022)

ADATMODELLEZÉS

**ADATMODELLEZÉS
FELSŐFOKON
EGYSZERŰEN ELBESZÉLVE**

HALASSY BÉLA

2025.

Apropó: Internetes segélykiáltás: „Valaki magyarázza már meg nekem, hogy mik is azok a normálformák. A tanáraimnak ez nem sikerült.”

Ajánlás: A tanulmányt azoknak a felsőfokú oktatási intézményeknek ajánlom, amelyekben az adatmodellezést ma is az ötven évvel ezelőtti anyaggal és szinten tanítják.

Mottó: Nem mondhatom el senkinek – merthogy kis hazánkban nincs informatikai szakmai publikációs fórum – elmondom hát mindenkinek.

a MEK jóvoltából.

ELŐSZÓ

Önkéntes nyugdíjba vonulásomkor (2006), amit a szakma egyre erősödő visszas jelenségei váltottak ki, azt hittem, hogy végleg felhagyok az informatikus hivatással és az ahhoz kapcsolódó publikálással.

Ámde a minap programozó unokám az egyetemen a normalizálás tárgyat tanulta. Mivel pedig én azt **már 1980-ban az IBM kutatóintézetében** is tanítottam (átlag 4,8-es hallgatói értékeléssel!), kíváncsiságból rákérdeztem a mai oktatási anyagra. Nem állhattam meg, hogy a neten alaposan is utána nézzek a tárgykörnek.

Elrémültem a sok számárság olvastán.

- **Tévesen adják meg a relációs adatbázisnak már a definícióját is:** „Reláció (azaz kapcsolati viszony)” [7]. Mármost a kapcsolat (relationship) nem relációt (relation), hanem egészen mást jelent. A kettő valódi lényegét az 1. fejezet fogja bemutatni.
- A kiadványok **teoretikus** szerzői aligha találkoztak **gyakorlati** élő adatbázissal. Erre kellő bizonyíték a B. függelékben ismertetett ún. negyedik és ötödik normálforma (4-5NF). Azokhoz hasonló badarság az életben soha se fordul elő, ellenben az integrálás jegyében erőltetni óhajtott 4-5NF bizonyítottan nem-integrált megoldásokra vezet.

Miért ne adjunk egy kicsit a szórakozásnak is? Az AI 5NF definíciója a következő:

„Az ötödik normálforma (5NF) az adatbázis-normalizálás azon szintje, mely a redundancia csökkentését célozza a **sokértékű tényeket** rögzítő relációs adatbázisokban, elkülönítve a szemantikailag összefüggő, **több kapcsolatból álló relációkat**. Egy tábla akkor van 5NF-ben, ha minden nem triviális (**azaz nem elhanyagolható**) join-függőséget a **jelölt kulcsok** rögzítenek.”

Miért kell a népeket ilyen számárságokkal traktálni?

Ámbár 20-25 igen méretes gyakorlati adatbázis tervezésében vettem részt, férfiasan be kell vallom, hogy nem tudom mi a sokértékű tény. Azt viszont igen, hogy a reláció nem kapcsolatokból áll, a triviális nem elhanyagolható jelent és a jelölt kulcs fordulat ebben a kontextusban helytelen. Az AI ifjú kora nem menti a csacsiságát: Aki nem tud arabusul, miért beszél arabusul?

- Az integrálás eszközeként a **matematikai alapú** bűvészkedést erőltetik, holott a napnál is világosabb, hogy integrált adatmodellt képtelenség **szemantikai módszerek** nélkül alkotni.

- Végül a legfurcsább az, hogy még a **matematikai módszereket is hibásan**, illetve hiányosan alkalmazzák.

Ha szabad képletesen ezt mondani: sokan tudják, hogy miként működik mechanikusan az **adatintegrálás teste** (no, azért azt se egészen!), míg fogalmuk sincs arról, hogy mi mindent rejt az **adatmodellezés lelke**.

A felsőfokú adatmodellezés nem mechanikus eljárások halmaza.

Ezt a kitételt és a könyv szemléletét egy esettel világítom meg.

A Wikipedian olvastam a példát [8], mely így szólt: „**vegyük az Étterem, Pizza fajta, Szállítási terület csupakulcs relációt**”, ami ilyen-olyan hibát rejt és ezért így-meg-úgy át kell alakítani.

Meg kell kérdeznem a kedves Olvasót, hogy szerinte ez így életszerű?

Szerintem ne csupakulcs relációt vegyünk, hanem inkább pizzát, mert az izesebb és emészthetőbb. A modellezés nem mesterkelt adatkonstrukcióknál kezdődik, hanem a valóságnál. Ott, hogy feladunk egy pizzarendelést egy adott fajtájú pizzára. Tehát kezdetben van a pizza meg annak fajtája és minden levezethető abból a „normális relációból” és kapcsolt társaiból a nyakatekert produkciók nélkül is.

„Az igazság mindig egyszerű.” (Pogány János piarista matektanár)

Gondolom már sejti az Olvasó, hogy mire számíthat tőlem.

A könyv első részében a felsőfokú **adatmodellezés elméletét** vezetem elő a mai iskolás adatintegrálási módszerek felülvizsgálatával párhuzamosan.

A második rész az **adatmodellezés gyakorlatát** ismerteti, mert a szakirodalom csak technikákkal foglalkozik, azokat összefogó eljárásokkal nem.

A tanulni vágyó fiataloknak pedig a kibővített szöveggel üzenem:

„Új seprű jól seper, de a régi tudja, hogy hol van a pizok.”
(New brooms sweep clean, but the old one knows where the dirt is.)

Halassy Béla

I. RÉSZ

AZ ADATMODELLEZÉS

ELMÉLETE

1. ADATBÁZIS ÉS ADATMODELL

Eredetileg nem szándékoztam ebben a könyvben visszatérni az alapok alapjaihoz, de az általános tájékozatlanság miatt ezt mégis meg kell tennem. Következzék tehát egy gyorstalpaló, bár csak Ábrahámától és nem Ádám-Évától kezdve. Csak a valóban nagyon járatosaknak ajánlom kihagyni.

Ábrahám a nemzetek atyja volt, megkülönböztetés nélkül. A legfontosabb mondanivalóm tehát az, hogy **az adatbázis egyszerűen csak adatbázis** hálós, relációs és minden más jelző nélkül.

Az autó fogalmát se úgy ismerjük meg, hogy van Trabant meg BMW, hiszen a gépjármű egy generikus fogalom, amit közös vonások jellemeznek. Ugyanígy beszélhetünk az adatbázisról általánosan is, mindenféle specifikus vonás nélkül. Mármint ha valaki azt se érti, hogy mi az adatbázis, az hogyan akarhat, mondjuk, a relációs adatbázisról értekezni...?

1.1 Adatállomány

Első megközelítésben maradjunk abban, hogy az adatbázis az ismeretek adatok formájában szervezett halmaza. Legelőször tehát azt kell megérteni, hogy mit jelent az ismeretek adatok szerinti szervezése.

Az ismereteket beszéd formájában, mondatokkal közöljük egymással. (Az egyéb formákat itt mellőzzük). Vegyük például a következő mondatot:

„Az FGS 802 rendszámú gépkocsi Ford Escort típusú.”

Itt egy alanyról (FGS 802 rendszámú gépkocsi) két dolgot állítunk. A két állítmány egyikét maga az alany hordozza (rendszáma FGS 802), a másika ahhoz tartozó leírás (típusa Ford Escort).

Rendezzük el ezt az ismeretet szikáran, adatszerűen:

Rendszám	Gépkocsi-típus
FGS 802	Ford Escort

Szemben a szabad, természetes szöveggel, az adat mindig egy adatnév (Rendszám) és egy adatérték (FGS 802) párosa.

Most az egyszerű mondatot bővítsük ki összetettre:

„Az FGS 802 rendszámú gépkocsi Ford típusú és 1995-ben helyezték forgalomba.”

GÉPKOCSI

Rendszám	Gépkocsi-típus	Forgalomba hely.
FGS 802	Ford Escort	1995

Az adatokat adattáblákba szervezzük kétszer két tényező szerint:

- **Egyed.** Azt a dolog-izé-valamit, amiről szó van, **egyednek** (entity) hívjuk. Ha általánosan említjük, akkor **egyed típus** (GÉPKOCSI), ha specifikusan, akkor **egyed-előfordulás** (FGS 802).
- **Tulajdonság.** Azt a dolog-izé-valamit, amivel az egyedet jellemezzük, **tulajdonságnak** (attribute) hívjuk. Ha általánosan említjük, akkor **tulajdonságtípus** (Forgalomba helyezés), ha specifikusan, úgy tulajdonság-előfordulás (1995), amit **tulajdonságértéknek** nevezünk.

Lehetetlen nem észre venni, hogy a konkrét egyedekről nem tudunk úgy beszélni, hogy ne hivatkoznánk egy – esetleg több – tulajdonságára. Ezért kell szólnunk két további fogalomról.

- **Azonosító.** Az egyed típus olyan tulajdonsága, amelyet típus szinten (Rendszám) jelölünk ki az előfordulások (FGS 802) egyedi, egyértelmű behatárolására.
- **Leíró.** Az egyed típus olyan tulajdonsága, amely általánosan is (Forgalomba helyezés), konkrétan is (1995) az azonosított egyedet jellemzi. Értéke azonos lehet több egyed-előfordulásra is.

Ha ismereteinket olyan táblázatban rögzítjük, amelynek sorai a konkrét egyedeket írják le oszlopainak a jelentése szerint, akkor a táblázatot **adat-állománynak** (file = fájl) hívjuk. A táblázat egy sorának és egy oszlopának a találkozásánál általában csak egy elemi érték fordulhat elő. Az ilyen fajlnak **síkállomány** (flat file) a neve.

1.2 Reláció (relation)

A reláció fogalmától egyesek megijednek, mások viszont elbonyolítják vagy félreértik azt. Holott az igazság nagyon egyszerű: **a reláció nem más, mint matematikai formalizmussal definiált táblázat**, ami csak az elvi kiindulását tekintve tér el a hagyományos síkfájltól.

A reláció Codd úrtól [1] átvett definíciója a következő:

Ha adottak $D_1, D_2, \dots, D_n$ egymást nem szükségszerűen kizáró halmazok, akkor R reláció az n halmazon, ha olyan $d_1, d_2, \dots, d_n$ felépítésű elemekből áll, amelyekre igaz, hogy $d_1 \subseteq D_1, d_2 \subseteq D_2, \dots, d_n \subseteq D_n$, vagyis ha egyes elemeit a leírt módon az eredeti n halmazból választjuk ki. Röviden: R a $\langle D_1^* D_2^* \dots D_n^* \rangle$ Descartes-féle (ejtsd: Dékárt) szorzat (Cartesian product) részhalmaza.

A hagyományos síkállományhoz képest annyi az eltérés, hogy míg abban a „kezdetben vala az egyed” szemlélet érvényesül, a relációsban a „kezdetben vala az egyedfüggetlen tulajdonság: az értéktartomány (domain = doméjn)” az alapelv. A különbség eddig pusztán elméleti, a gyakorlatban lényegtelen.

A reláció pontosított, szigorúbb megfogalmazása szerint

- minden adatérték elemi kell, hogy legyen és
- minden sornak kell, hogy legyen egyedi, a többiétől eltérő kulcsa.

A reláció oszlopok szerinti szemléletét intenzióknak, az oszlopok számát a reláció fokának nevezzük. A sorok (tuples) együttesét extenzióknak, a sorok számát a reláció kiterjedésének hívjuk. Az azonosító tulajdonság neve kulcs.

A **kulcs** (key) nem tartalmi-logikai, hanem hozzáférési-fizikai fogalom. Ha az adott reláció (!) sorait több tulajdonság képes egyedileg behatárolni, akkor azokat **kulcsjelölteknek** (candidate key) hívjuk.

A fogalommal a normalizálás ismertetésekor sokszor visszaélnek. A reláció olyan tulajdonságát is kulcsjelöltnek mondják, amely nem az aktuális, hanem az általa (közvetve) hivatkozott másik relációban lehet elsődleges kulcs (primary key).

Ha a sorok azonosításához egynél több tulajdonság is szükséges, akkor **összetett kulcs**ról (composite key) beszélünk. Ha több tulajdonság is alkalmas az azonosításra, azokat **alternáló kulcs**oknak (alternate key) hívjuk.

Elvileg előfordulhat, hogy egy reláció oszlopainak nincs olyan valódi részhalmaza, amely alkalmas kulcsnak. Ilyenkor minden tulajdonsága az összetett kulcs része és a táblázat ún. **csupakulcs** (all-key) reláció.

Életem során sok ezer adatállományt terveztem és/vagy elemeztem. Elárulom, hogy gyakorlatilag még fél tucatnyi csupakulcs táblázattal se találkoztam...

A hagyományos és a relációs állományok struktúrájában nincs érdemi eltérés. Ezért kijelenthetjük, hogy a sajátos szóhasználatától eltekintve

a reláció nem más, mint közönséges síkállomány.

1.3 Kapcsolat (relationship)

Minden adatszerűen tárolt, kezelt és közölt ismeret mögött síkállományok állnak, nevezzék bár azok együttesét invertált, hierarchikus, hálós, relációs vagy objektumos adatbázisnak. Viszont a kapcsolat tényezője nélkül egyik együttes se tekinthető adatbázisnak.

Most bővítsük tovább a példamondatot:

„Az FGS 802 rendszámú gépkocsi Ford típusú és 1995-ben helyezték forgalomba és a szállítható személyek száma (férőhely) 5 fő.”

GÉPKOCSI (eredeti)

Rendszám	Gépkocsi-típus	Forgalomba hely.	Férőhely
FGS 802	Ford Escort	1995	5 fő
HIS 333	Ford Escort	1998	5 fő
KJK 143	Polski Fiat	1999	4 fő

Az állományban felfedezhetünk egy ismereti redundanciát, amit piros színnel jelöltem. Ezzel összefüggésben két eset lehetséges:

- Ha a Ford Escort – 5 fő ismeretpáros más állományban is szerepel, akkor a Férőhely adat itt felesleges.
- Ha viszont nem szerepel és innen az utolsó ilyen típusú kocsit is kitörlik, akkor elveszik egy ismeret, ami fontos lehet. Ezért megfelelő helyet kell találni a számára.

Mindkét esetben az ismeretek nem jól szervezettek, tehát együttesük nem **adatbázis**, ami tudvalevőleg az ismeretek **jól szervezett** együttese.

A helyzet kulcsa az egyed fogalmi meghatározásában rejlik: **az egyed a valós világ saját ismeretekkel – elsődlegesen, közvetlenül rá jellemző tulajdonságokkal – leírható jelensége.**

Mármost eredeti táblázatunkról két dolgot mondhatunk el:

- Nem egyféle, hanem kétféle egyed ismereteit rögzíti, mivel a gépkocsi típusa valós jelenség írható le közvetlenül a férőhely jellemzővel.
- A férőhely a gépkocsinak nem saját ismerete, hanem csak közvetve, a gépkocsi típusán át az.

Ezért eredeti modellünkben hiányzik a GÉPKOCSI-TÍPUS egyed, viszont a GÉPKOCSI oda nem illő adatot tartalmaz. A kettős hibát kiküszöbölendő a kezdeti síkállományt kettőre bontjuk az alábbi módon:

GÉPKOCSI-TÍPUS

Gépkocsi-típus	Férőhely
Ford Escort	5 fő
Polski Fiat	4 fő

GÉPKOCSI (módosított)

Rendszám	Gépkocsi-típus	Forgalomba hely.
FGS 802	Ford Escort	1995
HIS 333	Ford Escort	1998
KJK 143	Polski Fiat	1999

Az adatállományokban kétféle redundancia léphet fel:

- **Felesleges redundancia.** Ilyen volt a GÉPKOCSI eredeti állományban a Férőhely adat, ami ott **leíró** tulajdonság volt. A többszörösség akkor **felesleges rossz**, ha a tétel egyik állományban se azonosító.
- **Szükséges redundancia.** Az új helyzetben a gépkocsi férőhely adatát úgy kapjuk meg, hogy a GÉPKOCSI Gépkocsi-típus leíró adatának az értékével átlépünk a GÉPKOCSI-TÍPUS-ba, ahol az adat **azonosító**. Ezért ennek az adatnak a redundanciája **szükséges jó**.

Két állomány (egyed típus) között akkor tudunk átlépni – ahogy adatbázis-berkekben mondják: **navigálni**, – ha van olyan közös adattételük (tulajdonságtípusuk), amely az egyikben azonosító, a másikban leíró **szerepű**.

Ilyenkor azt mondjuk, hogy a két egyed típus kapcsolatban áll egymással és a kérdéses adatot **kapcsolótulajdonságnak** hívjuk. Ezt dőlt szedettel jelezzük abban az egyedben, amelyben leíró szerepű.

A kapcsolatnak is két síkja van:

- A GÉPKOCSI-TÍPUS és a GÉPKOCSI kapcsolat **típusban** áll, aminek nevet adunk, például GÉPKOCSI-TÍPUSA.
- Az FGS 802 rendszámú gépkocsi kapcsolat-**előfordulást** létesít a Ford Escort gépkocsi-típus felé.

Mindez a legegyszerűbb kép. Az összetettebb eseteket a későbbiekben mutatom be. Itt csak a kapcsolat fogalmának a bevezetése, nem a kifejtése a cél.

A kapcsolat jellemzői a következők:

- **Kapcsolatfajta.** Itt csak az egyszerűt mutattam. A fajtától függenek a további jellemzők tartalmai.

- **Kapcsoló tulajdonság.** A két egyednek az az adattétele, amely az egyikben azonosító, a másikban leíró (vagy azonosítórész) szerepű.
- **Fölérendelt.** Az az egyed, amelyben a kapcsoló tulajdonság azonosító.
- **Alárendelt.** Az az egyed, amelyben a kapcsoló tulajdonság nem az.
- **Kapcsolatfok.** Azt mutatja, hogy a fölérendelt egy előfordulásához az alárendeltnek egy (1:1 fok), vagy több (1:N fok) előfordulása tartozik.

Két egyed M:N fokú viszonya NEM kapcsolat!!!

- **Kapcsolaterő.** Mutatja, hogy a fölérendelt minden előfordulásához kell-e kapcsolódnia (K-kötelező), vagy sem (O-opcionális) legalább egy alárendelt előfordulásnak és fordítva, az alárendelt felől. A jellemző értéke ezek szerint K-K, K-O, O-K és O-O lehet.

Sokan azért görcsölnek a relációs adatbázison és főleg a normalizáláson, mert **a relációs modellben egyáltalán nincs explicit kapcsolat fogalom**, ami az adatbázis lényegét megvilágítaná. A kapcsolat a relációs rendszerben csak implicit tényező, amit az is mutat, hogy fokát és erejét a föl- és alárendelt oldaláról külön-külön kell megadni, ami nagy számság, mert

a kapcsolat egyetlen megbonthatatlan fogalmi tényező.

A helyes szóhasználat érdekében fontos megjegyezni, hogy a **táblázaton belüli** viszonyok nem kapcsolatok, hanem relációk (relation), miközben a **táblázatok közötti** összefüggéseket hívjuk kapcsolatoknak (relationship).

1.4 Adatmodell és adatbázis

Az ismeretek **fogalmi** természetű dolgok, amiket a használatuk céljai és sajátosságai szerint **tartalmi-logikai** szinten rendezünk el, hogy **tárolási-fizikai** szinten kezelhessük őket.

Például a gépkocsi egy fogalom, aminek adattartalmait úgynevezett rekordkép szerint rendeztük el és sok mindentől függően így-úgy tároltuk a számítógépen.

Mármost ha nem vagyunk tisztában az ismereteinkkel mint fogalmakkal és/vagy több felhasználó esetén nem értünk egyet azokban, akkor miként hozhatnánk létre integrált adatbázist tartalmi és tárolási szinten? Mivel a reláció nem fogalmi tényező, az integrálásnak nem annál kell kezdődnie.

Itt előre bocsátok két definíciót, amiket később meg fogok ismételni.

	Az adatmodell egyed típusoknak, azok tulajdonságtípusainak (egyedek belső szerkezete) és kapcsolattípusainak (egyedek külső szerkezete), valamint az ezen tényezőkre és azok viszonyaira vonatkozó korlátoknak a szervezett együttese.
--	---

	Az adatbázis az egyed-előfordulások tulajdonság-értékeinek és kapcsolat-előfordulásainak az adatmodell szerint szervezett együttese.
--	---

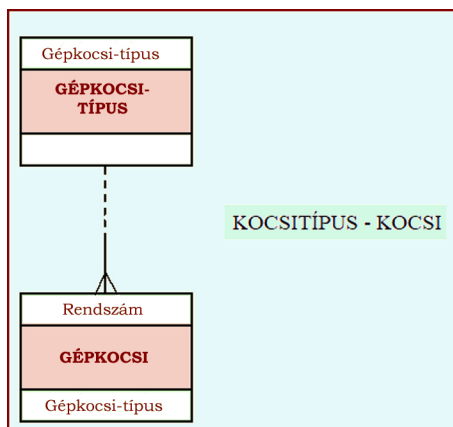
Az adatmodell nem ördögösség. Vegyük le mintapéldánkról a konkrét tartalmakat, az előfordulásokat és máris látható az adatmodell.

GÉPKOCSI-TÍPUS

Gépkocsi-típus	Férőhely
----------------	----------

GÉPKOCSI (módosított)

Rendszám	Gépkocsi-típus	Forgalomba hely.
----------	----------------	------------------



Adatmodell diagram
az én jelöléseim szerint

Megértem, ha néhányan két-három reláció normalizálásával küszködnek, hiszen kicsiben is jót kell alkotni. Viszont jelezném, hogy Kaszap István kollégám 1980 táján négyszáz-négyszázötven egyed típusból álló adatmodellt készített a Május 1. ruhagyár számára – és az közepes méretűnek számított. Csupakulcs reláció egy se volt közöttük...

A modell ábrája 10 darab A3-as lapot töltött be.

Nem melleleg akkor még nem terjedt el a relációs adatbázis fogalma és normalizálásról még hírből se hallottunk. Ennek ellenére igen jó minőségű, mondhatni „ultranormalizált” adatmodelleket terveztünk.

1.5 Az adatbázis lényege

Most ismét alakítsuk át a példamondatot kihagyva belőle a redundanciát:

„Az FGS 802 rendszámú gépkocsi Ford típusú és 1995-ben helyezték forgalomba és tulajdonosa Halassy Béla.”

Mivel a név nem azonosít, átugorva egy lépést a tulajdonos ismereteit az egyedi kódjával egészítem ki. Továbbá a gépkocsi az idők során több birtokos tulajdonában lehet, ezért itteni táblája egy adott időpont szerinti állapotot rögzít. Ennek a korlátnak a feloldása messzire vezetne, ezért most elhagyom.

GÉPKOCSI-TÍPUS

Gépkocsi-típus	Férőhely
Ford Escort	5 fő
Polski Fiat	4 fő

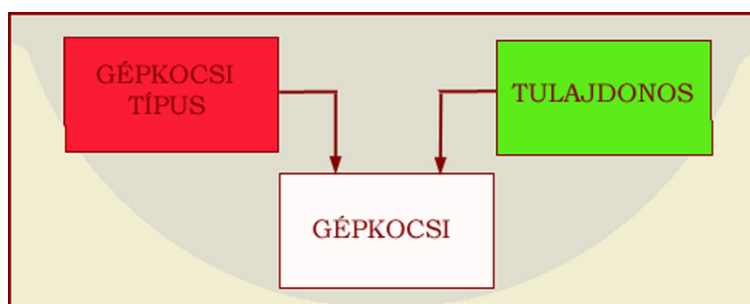
TULAJDONOS

Tulajdonos kód	Tulajdonos név
123	Halassy Béla
456	Kovács Benő
789	Szabó Márta

GÉPKOCSI (újra módosított)

Rendszám	Gépkocsi-típus	Forgalomba hely.	Tulajdonos kód
FGS 802	Ford Escort	1995	123
HIS 333	Ford Escort	1998	456
KJK 143	Polski Fiat	1999	789

A modell egyszerűsített rajzban a következő módon néz ki:



Hálós adatszerkezet

Egyesek arra kíváncsiak, hogy egy típusba milyen gépkocsik tartoznak. Mások viszont arra, hogy egy tulajdonosnak milyen kocsijai vannak.

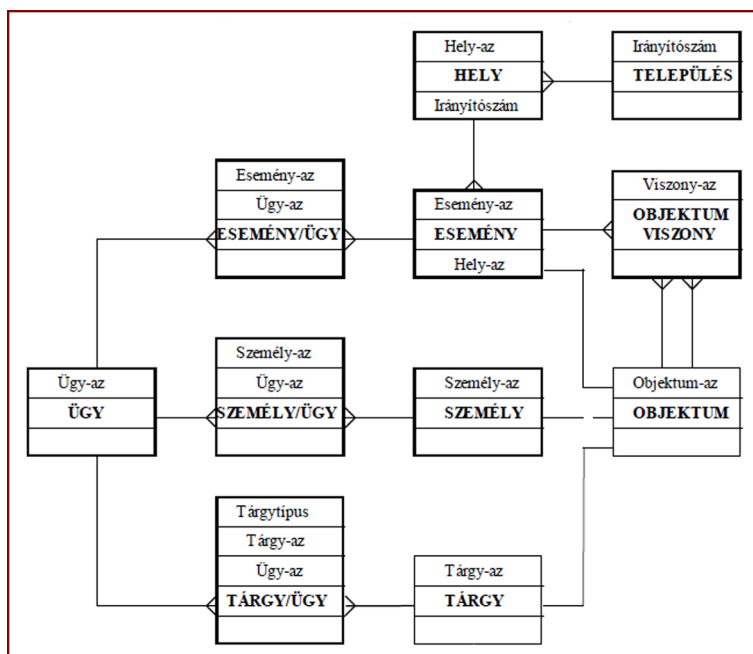
Az adatbázis célja a valós jelenségek eltérő szemléleteinek az egyesítése. Ennek megértéséhez **célszerű szem előtt tartani a Maklári-féle vájdling-reakció** üzenetét.

Amikor a réz vegyi folyamataira került sor, kémia tanárom, Maklári tanár úr kiemelte a reverzibilitást, vagyis azt, hogy miközben a réz oxidálódik, máris vissza-redukálódik. Ez a Maklári-féle vájdling reakció, aminek lényege:

**„Ha innen nézem domború, ha onnan nézem homorú,
de azért az csak EGY vájdling.”**

Ez az adatmodellezés veleje. A valós jelenségeket lehet innen nézni és onnan nézni, de informatikailag csak egy közös tükörképük (modelljük) van. Aki ezt megtanulja, az közel áll az adatbázis lényegének a megértéséhez.

Hogy az elméletet gyakorlatiasabbá fordítsam mellékelek egy valós adatmodell részletet. A minősége kicsit „viharvert”, hiszen harminc éve készült.



Zárásképpen muszáj egy kicsit zsörtölődnöm. Minden normális felfogású ember – mondjuk közgazda vagy mérnök – a jó áttekintés érdekében az ilyen és hasonló ábrákat hierarchikus szemléletben készíti a fontos jelenségekkel felülről vagy balról (ahogyan adódik, mint a jelen ábrán is) kezdve.

Nem így a relációs tervező rendszerekben. A „varjúlábak” felfelé állnak és az ábrákon felül helyezkednek el a többszörös-lényegtelen tényezők és alul az egyszeres-fontosak. No comment!

2. AZ ADATMODELLEZÉS LÉNYEGE

Mindenek előtt tisztázni kell két félreértést.

Az adatmodellezésnek nem a relációs adatbázisok a tárgyai.

Egyrészt léteznek nem-relációs adatbázisok és azokat is meg kell tervezni. Másrészt az adatbázisok – relációsak és nem-relációsak – tervezése három síkon történik: fogalmi, logikai és fizikai szinten. A tervezés legfelsőbb szintje a fogalmi szint és azt nevezzük adatmodellezésnek, márpedig a reláció nem fogalmi, hanem logikai/fizikai szintű lényeg.

Az adatintegrálás nem relációkkal való matematikai bűvészkedés.

Az integrálás elsősorban fogalmi természetű feladat. Ha ugyanis az adatbázis leendő adatai tartalmilag nem egyértelműek, akkor hiábavaló törekvés azokat szerkezetileg optimálisan összeállítani. A tartalmi egyértelműség pedig csak szemantikai eszközökkel érhető el. Sajnálatos módon ezeket a relációs adatbázisoknál egyáltalán nem alkalmazzák.

Ezért le kell szögezni a következő alaptézist:

T1	A mai adatintegrálási módszerek nem felelnek meg a felsőfokú adatmodellezés követelményeinek.
----	---

A továbbiak előtt idézzük vissza újra a két alapvető meghatározást:

D1	Az adatmodell egyed típusoknak, azok tulajdonságtípusainak (egyedek belső szerkezete) és kapcsolattípusainak (egyedek külső szerkezete), valamint az ezen tényezőkre és azok viszonyaira vonatkozó korlátoknak a szervezett együttese.
----	---

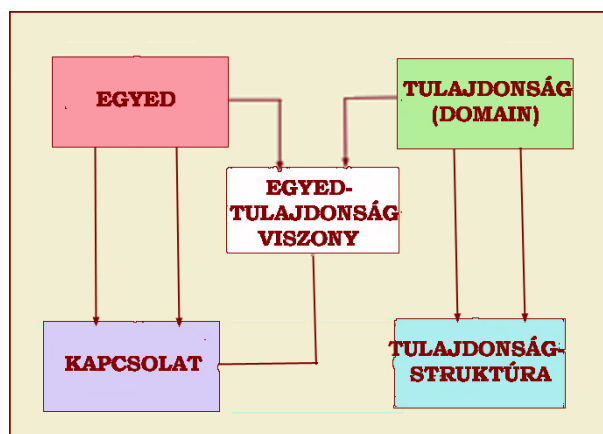
Az adatmodell tehát korlátoknak engedelmeskedő adatszerkezet.

A relációs rendszerben a kényszer (constraint) kifejezést használják, ami műveleti és nem szerkezeti tényező. Az adatrendszereknél a feltétel értelemben vett korlát (restriction) kifejezés van szokásban. Én azt fogom alkalmazni.

D2	Az adatbázis az egyed-előfordulások tulajdonság-értékeinek és kapcsolat-előfordulásainak az adatmodell szerint szervezett együttese.
----	---

Mivel a modellt típusok alkotják, a „típus” szórészt a továbbiakban el fogom hagyni kivéve ott, ahol az félreértést okozhatja.

Az adatbázisok általános strukturális képét az úgynevezett **metamodell** határozza meg. Sajnos az adatintegrálással foglalkozó hozzászólások elsőprő többségében szó sem esik róla.



Az adatbázis metamodellje

Most vessük össze ezt a struktúrát a relációs modell szerkezeti képével! Ne feledjük, hogy a reláció (relation) értékhalmozokból (domain) Descartes-féle szorzattal képezett táblázat! A táblázatok közötti viszonyokat (kapcsolat, relationship) a táblázat tulajdonságai (attribute) hordozzák, de csak implicit módon: a relációs modellben nincs explicit kapcsolattényező.

A relációs rendszerben

- az **egyed** egy fizikai adattábla neve, nincsenek fogalmi jellemzői
- az **egyed-tulajdonság-viszony** a tábla oszlópa
- a **tulajdonság** egyedtől független értéktartomány (doméjn)
- a **kapcsolat** fizikai adatokon létesült viszony, nincs fogalmi tartalma
- a **tulajdonság-struktúra** teljes egészében ismeretlen.

A relációs adatbázis fizikai voltát leginkább az mutatja, hogy a kapcsolat, ami az adatbázis „lelke”, az különbözteti meg az egyéb adathalmazoktól, az ún. „belső kulcs” alapul, ami indexszel támogatott fizikai adat.

Ez a mű nem a relációk normalizálásával foglalkozik, ami az adatmodellezésnek csak egy sokadik résztechnikája. Azonban mivel a szakirodalmat és az oktatást láthatóan az tölti ki, kénytelen vagyok rangján felül foglalkozni vele. Viszont mivel tárgyam mégis az adatmodellezés, reláció helyett az **egyed**, kulcs helyett az **azonosító** fogalmakat fogom használni.

3. ALAPVETŐ NORMÁLFORMÁK

3.1 Az alapok alapjai

Ámbár az adatmodellezés messze nem a normalizálást jelenti, ami annak csak sokadik lépése, didaktikai okokból mégis bemutatom itt az alapvető normálformákat (0NF-BCNF) azért, hogy egyes **negatív jelenségek** kifejtése közben hivatkozni lehessen rájuk.

Az alapvető normálformák lényegét leginkább egy hétköznapi fordulat segítségével lehet megjegyezni. Az angolszász bíróságokon szokott elhangozni a következő esküszöveg:

„Esküszöm, hogy az igazat mondom, csak a teljes igazat és semmi mást csak az igazat.”

Adatmodellre lefordítva:

A reláció minden tulajdonsága függ a kulcstól, csak a teljes kulcstól és semmi mástól, csak a kulcstól.

A meghatározáshoz két kiegészítést kell fűzni:

- Általában ezt a harmadik normálalak (3NF) definíciójának tartják, holott beleillik a Boyce-Codd normálforma (BCNF) is.
- Ahogyan az esküszöveg nem garantálja a teljes igazságot, a BCNF se jelent teljes normalizáltságot. (De nem a 4-5NF miatt, ld. B. függelék)!

Az esküszöveg nem tér ki arra, hogy a tanú elmondja **a teljes igazságot**; a teljesség csak a feltett kérdések válaszára vonatkozik. Node mi történik, ha egy lényegi kérdést nem tesznek fel?

Mi történik, ha a vizsgált egyedekkel kapcsolatosan nem tesszük fel a teljességre vonatkozó kérdést, hogy a látottakon kívül még milyen további, egyébként ismert attribútumok tartoznak egy már meghatározott egyedhez?

TELEPÜLÉS	Településkód , Településnév
VEVŐ	Vevőkód , Vevőnév, Vevőtípus, X

1. példa: Hiányos modell

Jelölések: A példákban az azonosító tulajdonságokat vastag,
a kapcsoló tulajdonságokat dőlt szedet jelzi.

Világosan látszik, hogy a két egyenként BCNF alakban lévő egyed együtt nem alkot teljes modellrészt, mert hiányzik a kapcsolatuk (X).

TELEPÜLÉS	Településkód, Településnév
VEVŐ	Vevőkód, Vevőnév, Vevőtípus, <i>Vevőszékhely</i>

2. példa: Nem egyértelmű modellrész

Színek: A példákban a hibákat-hiányosságokat piros, az egyértelműségi bajokat kék szín jelzi.

A példa szemantikailag hibás. A Vevőszékhelyet a településre való utalásnak szánták, de nem tudni, hogy annak melyik adatát jelenti. Ha netán a településnevet, úgy a modellrészlet kapcsolathianyos lenne, mert

két egyed nem kapcsolható össze olyan adaton, amely egyikben se azonosító.

TELEPÜLÉS	Településkód, Településnév
VEVŐ	Vevőkód, Vevőnév, Vevőtípus, <i>Településkód</i>

3. példa: Egyértelmű és teljes modellrész

Az alábbi pontok az alapvető normálformákat írják le. A tájékozottak számára ezek csak ismétlést jelentenének, mégis javasolom, hogy türelmesen olvassák el, hátha találkoznak újdonsággal is.

3.2 Funkcionális függés és függetlenség (ONF)

Az egyedek tulajdonságai *első megközelítésben* két alapvető kategóriába tartoznak, amiket *szerepeknek* hívunk.

D3	Azonosítóknak nevezzük. azokat a tulajdonságokat, amelyek az egyed minden egyes előfordulására eltérő értéket vesznek fel.
----	---

D4	Leíróknak nevezzük azokat a tulajdonságokat, amelyek az egyed több előfordulására is azonos értéket vehetnek fel.
----	--

KÁR	Kárszám, Kár dátum, Kár helyszín
-----	----------------------------------

4. példa: Tulajdonság szerepek

A KÁR egyednek a Karaszám azonosító, a másik kettő leíró tulajdonsága.

D5	Egy tulajdonság funkcionálisan függ az azonosítótól, ha annak minden külön értékéhez a tulajdonságnak csak egy értéke tartozik. Ha ez nem áll fenn, akkor az adott tulajdonság funkcionálisan független az azonosítótól.
----	--

A funkcionális függés jele „ $\rightarrow$ ”. Képlete: $A \rightarrow B$, ami úgy olvasandó, hogy A meghatározza B-t, illetve B funkcionálisan függ A-tól. Megjegyzendő, hogy a függés a később tárgyalandó kölcsönös függés kivételével ellenkező irányú függetlenséget jelent. A jelölés miatt szokás beszélni a függés baloldaláról (azonosító) és jobboldaláról (leíró).

KÁR

Kárszám	Kár dátum	Rendszám	Gépjármű-típus	Kár összeg
23456	2025.05.14.	ABC 134	Opel	n1
23456	2025.05.14.	DEF 567	Mazda	n2
56789	2025.09.01.	ABC 134	Opel	n3

5. példa: Normalizálatlan egyed

A Kár dátum funkcionálisan függ a Kárszámtól, viszont a többi leíró tulajdonság független tőle, mert egyazon Kárszám értékhez több Rendszám érték tartozhat, a Kár összeg pedig nem csak a Kárszám függvénye.

Van itt még egy általam bevezetett fogalom, ami később kihatással lesz a normalizálás menetére és eredményére. Előfordul, hogy az $A \rightarrow B$ függésben a B tulajdonság egyes sorokra **üres értéket** vesz fel. Például megeshet, hogy egy balesetben az egyik kocsit nem érte kár, tehát a Kár összeg értéke 0. Ettől még a függés fennáll, de a másokkal együtt alkotott struktúrákban – például a ciklusoknál – ez a jelenség különös fontossággal bír.

D6	Ha az A meghatározó tulajdonság minden értékéhez a B meghatározott tulajdonságnak egy valódi, nem nulla értéke tartozik, akkor a függés erős . Egyébként a függés gyenge .
----	--

3.3 Ismétlődés és 1NF

A többszörös értékeket **ismétlődő adatoknak**, illetve ha összetartoznak, **ismétlődő csoportnak** hívjuk. A Kár dátum ismétlődő adat, a Rendszám és a Gépjármű-típus párosa ismétlődő csoport (piros szedet).

D7	Egy egyed nem-normalizált , ha ismétlődő adatot/csoportot tartalmaz, amit ONF alaknak szokás nevezni.
----	--

Bizonyos anomáliák miatt az ismétlődés nemkívánatos jelenség és ki kell küszöbölni. (Az anomáliákra nézve lásd az A. függelékét.) Ez úgy történik, hogy az ismétlődő tételeket kivetítjük (projection) külön egyed(ek)be úgy, hogy az ismétlődések azonosítóját/azonosítóit az eredeti egyedben hagyjuk kapcsoló tulajdonságként.

KÁR

Kárszám	Kár dátum
23456	2025.05.14.
56789	2025.09.01.

GÉPKOCSI

Rendszám	Gépjármű-típus
ABC 134	Opel
DEF 567	Mazda

GÉPKOCSI KÁRA

Kárszám	Rendszám	Kár dátum	Kár összeg
23456	ABC 134	2025.05.14.	n1
23456	DEF 567	2025.05.14.	n2
56789	ABC 134	2025.09.01.	n3

6. példa: Modellrészlet a kivetítések után

D8	Egy egyed legalább első normálformában (1NF) van , ha minden tulajdonsága függ az (akár egyszerű, akár összetett) azonosító tulajdonságtól.
----	--

A példa mindhárom egyede legalább 1NF alakú, de a harmadik még nem tökéletes. Hibás tételét piros szín mutatja.

Mielőtt tovább haladnánk vegyük észre, hogy **második megközelítésben** az egyed tulajdonságainak két újabb szerepébe botlottunk.

- Ha az azonosító összetett (pl. Kárszám+Rendelésszám), ahol az összetételt a „+” jel mutatja, akkor alkotóelemei **azonosítórész** szerepűek. A függési képlet ilyenkor $A+B \rightarrow C$.
- Ha egy tulajdonság egy egyedben azonosító, a másikban azonosítórész vagy leíró, akkor **kapcsoló** szerepet lát el. Lásd pl. Rendszám, ami a károkból a gépkocsik felé teremt kapcsolatot.

Felhívom a figyelmet arra, hogy általában csak a leíró tulajdonságok függéseiről szokás beszélni, ami tévedés. Az azonosítótól nemcsak a leíró, hanem (triviálisan) az azonosítórész tulajdonságok is függenek. Ez nagyon fontos, ha a normalizálást automatával akarjuk majd végeztetni!

Megjegyzések:

- Egy kiinduló egyedben több ismétlődési anomália is előfordulhat és így azon több kivetítést kell végrehajtani, mint a példában is.
- A kivetített egyed adott esetben maga is lehet még normalizálatlan.
- A többszörös értéket egyes tervezetekben minősített tulajdonságnevek rejthetik. Pl. Rendszám-1, Kárösszeg-1, Rendszám-2, Kárösszeg-2 stb. Az ilyen rejtett többszörösséget emberileg kell felfedezni; erre nem képesek az automaták.
- A kivetített egyed (GÉPKOCSI KÁRA) az összetett azonosítója egyik részével N:1 fokú kapcsolattal kapcsolódik az eredeti egyedhez (KÁR).

3.4 Részleges függés – 2NF

A 6. példa egyik egyede 1NF-ban van, de további anomáliát tartalmaz.

GÉPKOCSI KÁRA

Kárszám	Rendszám	Kár dátum	Kár összeg
23456	ABC 134	2025.05.14.	n1
23456	DEF 567	2025.05.14.	n2
56789	ABC 134	2025.09.01.	n3

7. példa: Részlegesen függő adat

Az adatok többszörösségét a helytelen funkcionális függések segítségével fedezhetjük fel. Az egyedben a Kár dátum függ ugyan az azonosítótól, de egy részétől is. Ezért nem teljesül a feltétel, hogy **„csak a teljes kulcstól.”**

Ezt a problémát is kivetítéssel oldjuk fel. A részlegesen függő tulajdonságot (Kár dátum) kivetítjük a meghatározó azonosítórész (Kárszám) által azonosított egyedbe és ha már abban található, akkor egyszerűen elhagyjuk, mint jelen esetben is. (A kezdő tervezők néha elfeledkeznek arról, hogy nem mindig van szükség az adatok ide-oda helyezgetésére, gyakran egyes adatok elhagyása is megteszi.) Az eredmény a következő lesz:

GÉPKOCSI KÁRA

Kárszám	Rendszám	Kár összeg
23456	ABC 134	X
23456	DEF 567	Y
56789	ABC 134	Q

8. példa: A részleges függés elhagyása

D9	Egy egyed legalább második normálformában (2NF) van , ha minden tulajdonsága függ az azonosítótól és csak a teljes azonosítótól.
----	---

A részleges függés általános képlete a következő:

$$A+B \rightarrow C \text{ és } A \rightarrow C \text{ vagy } A+B \rightarrow C \text{ és } B \rightarrow C$$

Ha a triviális függésről se feledkezünk meg, amit zöld színnel mutatok, akkor három elemből álló függési láncot kapunk.

$$\mathbf{A+B} \rightarrow \mathbf{A} \text{ és } A \rightarrow C \text{ és } \mathbf{A+B} \rightarrow \mathbf{C} \text{ vagy } \mathbf{A+B} \rightarrow \mathbf{B} \text{ és } B \rightarrow C \text{ és } \mathbf{A+B} \rightarrow \mathbf{C}$$

3.5 Tranzitív függés – 3NF

Most alakítsuk át a GÉPKOCSI egyedet a következőre:

GÉPKOCSI

Rendszám	Gépjármű-típus	Tulajdonos kód	Tulajdonos név
ABC 134	Opel	123	xyz
DEF 567	Mazda	456	vwq
GHI 134	Fiat	123	xyz

9. példa: Tranzitívan függő adat

Ha a tulajdonos nevét máshol is tároljuk, akkor az a baj. Ha nem, akkor az itteni többszörösség a probléma. Nem teljesül ugyanis az a részfeltétel, hogy a tulajdonság **csak a kulcstól függ és semmi mástól**. Adott időben (!) a Tulajdonos név függ a Rendszámtól, de függ a Tulajdonos kódtól is. (N.B. Az időfaktorral ebben a műben nem foglalkozok.)

Ezt a problémát is kivetítéssel oldjuk fel. A tranzitívan függő tulajdonságot (Tulajdonos név) kivetítjük a meghatározó azonosítórész (Tulajdonos kód) által azonosított egyedbe, vagy ha már ott található, akkor egyszerűen elhagyjuk. Az eredmény a következő lesz:

TULAJDONOS

Tulajdonos kód	Tulajdonos név
123	xyz
456	vwq

GÉPKOCSI

Rendszám	Gépjármű-típus	Tulajdonos kód
ABC 134	Opel	123
DEF 567	Mazda	456
GHI 134	Fiat	123

10. példa: A tranzitív függés feloldása

D10	Egy egyed legalább harmadik normálformában (3NF) van , ha minden tulajdonsága függ az azonosítótól, a teljes azonosítótól és egyéb leíró tulajdonságtól nem.
-----	---

A tranzitív függés általános képlete a következő:

$$A \rightarrow B \text{ és } B \rightarrow C \text{ és } \mathbf{A \rightarrow C}$$

Kevés szakírónak tűnik fel, hogy a **részleges függés is tranzitív függés**. Azért nem, mert nem veszik tekintetbe a triviális függéseket.

S11 A részleges függés a tranzitív függés speciális esete.

$A+B \rightarrow B$ és $B \rightarrow C$ és $A+B \rightarrow C$
 $X \rightarrow B$ és $B \rightarrow C$ és $X \rightarrow C$

Ha felvesszünk egy $X=A+B$ **adatcsoportot**, akkor rögtön világossá válik az S11 segédétel igazsága.

A három tagú függési láncnak mindig a harmadik tagja tranzitív és az jelzi a hibát (piros). Eddig a tranzitivitás mögött **logikai redundancia** állt, ami haszontalan **fizikai redundanciát** okozott és a harmadik függést tartalmazó egyedből ki kell emelni. A következő esetben ez változik meg.

3.6 Külső kulcstörő függés – BCNF

Ismeretes, hogy minden gépkocsinak kell, hogy legyen egy és egyidőben csakis egy kötelező felelősségbiztosítása. Ennek szerepelnie kell **valahol** a károkkal kapcsolatos egyedekben, de nem mindegy, hogy melyikben.

Az idők során a biztosítóváltás miatt egy kocsinak több biztosítása is lehet, ezért a Rendszám nem határozza meg a Biztosításszámot. Ezért ez az egyed harmadik normálformában van:

GÉPKOCSI KÁRA

Kárszám	Rendszám	Biztosításszám	Kár összeg
23456	ABC 134	XXX	n1
23456	DEF 567	YYY	n2
56789	ABC 134	ZZZ	n3

11. példa: Külső kulcstörést okozó helyzet

Viszont egy KGFB minden időben csak egy gépkocsira vonatkozhat ezért fennáll a Biztosításszám $\rightarrow$ Rendszám függés. Így nem teljesül az a feltétel, hogy a tulajdonság **csak a kulcstól függ és semmi mástól**. A Rendszám ugyanis az azonosító részeként függ a teljes azonosítótól is, meg az ahhoz képest külső Biztosításszám tulajdonságtól is.

D11 **Külső kulcstörésről beszélünk**, ha egy egyed valamelyik leíró tulajdonsága függ az összetett azonosító egészétől, de ugyanakkor meghatározza annak egy részét.

A külső kulcstörő függés általános képlete a következő. A triviális függést zöld színnel mutatom. A második sorban az X az A+B csoportot jelöli.

$A+B \rightarrow C$ és $C \rightarrow B$ és $A+B \rightarrow B$
 $X \rightarrow C$ és $C \rightarrow B$ és $X \rightarrow B$

Kevés szakírónak tűnik fel, hogy a **belső kulcstörő függés is tranzitív függés**. Azért nem, mert nem veszik tekintetbe a triviális függéseket.

S12 A külső kulcstörő függés a tranzitív függés speciális esete.

Ámde ebben az esetben egy különös probléma adódik. A rossz függést nem lehet a függési lánc harmadik tagjának a felszámolásával megoldani, mert az triviális! Itt a függési láncban lévő ellentmondás **nem redundanciát** jelez, hanem azt, hogy **a modellrészlet nem valósághű**.

Ugyanis a biztosító nem a gépjárműre, hanem annak biztosítására fizeti ki a kár összegét! Ezek szerint rossz a kiinduló egyed azonosítója, amit nem lehet a szokásos kivetítéssel megoldani. A modellrészletet át kell tervezni úgy, hogy a régi rossz azonosítót lecseréljük az új megfelelőre. (A szakirodalomban mutatott mindenféle bűvészkedés helyett...)

BIZTOSÍTÁS

Biztosításszám	Rendszám
xyz	123
vwq	456
stu	789

GÉPKOCSI KÁRA

Kárszám	Biztosításszám	Kár összeg
23456	XXX	n1
23456	YYY	n2
56789	ZZZ	n3

12. példa: A külső kulcstörés feloldása

D12 Egy egyed **BCNF** alakban van, ha harmadik normálformájú és nem tartalmaz külső kulcstörő függést.

A „hivatalos” definíció mindenféle kulcsokról és kulcsjelöltekről szól a kulcsot ugyancsak zavarosan hol azonosítónak, hol kapcsoló tulajdonságnak tekintve.

Megjegyzések:

- A függések meghatározásánál mindig a teljes három tagú függési láncot kell figyelembe venni, beleértve a triviális függéseket is.

- Mindig a függési lánc harmadik tagja, a látszólagos tranzitivitás jelzi, hogy valamilyen problémát meg kell oldani, de azt nem, hogy milyen és milyen módon.
 - o Ha a harmadik függés nem-triviális, akkor a probléma **redundanciát** takar, ami kivetítéssel (2-3NF) oldható meg.
 - o Ha a harmadik függés triviális, akkor **rosszul választották meg az azonosítót** és azt a megfelelőre kell lecserélni (BCNF).

Ezen kívül még a következőket kell tekintetbe venni:

- A kulcstörő függés **külső** (azonosítón kívüli) jelzője sejteti, hogy van **belső** (azonosítón belüli) kulcstörő függés is.
- Van, amikor a leíró tulajdonság és van, amikor az azonosítórész okoz problémát. Ebből következik, hogy a **függés jellege** egy fontos ismerv, amit az alapvető normálformák nem vesznek tekintetbe.

Mindkét kérdéskörre az 5. fejezetben térek ki. Előbb azonban ideje, hogy ismertessem az adatmodell iránti követelményeket.

4. AZ ADATMODELL KÖVETELMÉNYEI

A szakirodalom csak a redundanciát tartja az integritást sértő hibának. Az adatbázist alkotó relációk legyenek redundancia-mentesek, amit azok megfelelő normálalakra hozásával lehet elérni. Az adatmodell integritásának öt kritériuma van, ezért

T2	a csak normalizálásra épülő adatmodellezés korlátos és nem lehet általa optimális adatmodellt készíteni.
----	--

4.1 Valósághűség

A modellt alkotó fogalmak feleljenek meg a nevüknek.

CIKK	Cikkszám, Egységár
RENDELÉS-1	Rendelésszám, Cikkszám , Mennyiség
RENDELÉS-2	Rendelésszám + Cikkszám , Mennyiség, Egységár

13. példa: A valósághűség megsértése

Hiba lenne **látszólagos redundanciája** alapján kivenni a RENDELÉS-2 egyedből az Egységárat. Azzal hiányossá válna a modell, mert az valójában Kialkudott árat jelent, tehát csak a nevet kell valósra változtatni így:

CIKK	Cikkszám, Egységár
RENDELÉS-2	Rendelésszám + Cikkszám , Mennyiség, Kialkudott ár

14. példa: A valósághűség biztosítása

4.2 Érthetőség

A modellt alkotó fogalmak legyenek beszédesek, de ne szószátyárok.

RENDELÉS-1	Rendelésszám, Cikkszám
RENDELÉS-2	Rendelésszám + Cikkszám , Kialkudott ár

15. példa: Az érthetőség megsértése

Itt csak a nevek üressége jelent problémát, ami egyszerűen orvosolható.

A neveket beszélőkre, a valóságot tükrözőkre kell változtatni.

VEVŐ-RENDELÉS	Vevő-Rendelésszám, Cikkszám
SZÁLLÍTÓ-RENDELÉS	Szállító-Rendelésszám+Cikkszám, Kialkudott ár

16. példa: Az érthetőség biztosítása

Egyes relációs adatbázisokban minden reláció azonosítójának Belső kulcs a neve. Terveik nem tekinthetők fogalmi szintűeknek, sőt még logikaiaknak sem.

4.3 Egyértelműség

A modellt alkotó fogalmak legyenek mentesek a szinonimáktól és homonimáktól, amik gyakran párosulnak a valósághűség és/vagy az érthetőség megsértésével.

A 13. példában az Egységár **homoníma**, ami fogalmi redundanciában nyilvánult meg, valójában a valósághűség megsértését takarta. A hasonló homonimák kizárására kerülni kell az olyan egyszerű adatmegnevezéseket, mint a Név, Dátum, Típus, Mennyiség stb., mert ezek többnyire homonimák.

TELEPÜLÉS	Településkód, Településnév
VEVŐ	Vevőkód, Vevőnév, Vevőtípus, Vevőszékhely

17. példa: Nem egyértelmű modellrész (megismételt 2. példa)

A példában a Vevőszékhely a településkód **szinonímája** volt. Ennek felfedezése nehezebb, mint a homonimáé, mert csak fizikai redundancia áll mögötte. Adott konstelláció esetén egy automata rámutathat arra, hogy a két egyed valamelyike nem rendelkezik kapcsolattal, de ennek kicsi az esélye. A szinonimák kibogarászása nagyrészt emberi feladat marad.

4.4 Minimalitás

Az adatok többszörösségének három válfaját kell megkülönböztetni:

- Semleges **fizikai redundancia**. Az eltérő jelentésű tulajdonságok fizikai érték-átfedése érdektelen a számunkra. Ilyen lehet a különböző numerikus adatok, a dátumok, egyes nevek stb. érték-azonossága.
- Kapcsolathordozó **logikai redundancia**. Evidens, hogy a Rendelés és a Rendelés-tétel egyedben egyaránt szerepel a Rendelésszám, ui. ez

hordozza a két egyed közötti kapcsolatot. Három módon teheti: úgy, hogy az egyikben azonosító, a másikban azonosítórész, leíró, vagy azonosító szerepnév. (Az utóbbit lásd majd az egyed-altípusnál.)

- Haszontalan **logikai redundancia**. Ha egy tulajdonság több egyedben is előfordul úgy, hogy egyikben se azonosító, akkor nem hordoz kapcsolatot és a megfelelő egyed(ek)ből eltávolítandó. Viszont vigyázni kell vele, mert ha homoníma, akkor a neveket kell pontosítani.

A logikai redundancia legegyszerűbb esete a sima tranzitívitas, amit a 3. példa kibővítése szemléltet:

VEVŐ	Vevőkód, Vevőnév, Vevőtípus, <i>Településkód</i> , <i>Településnév</i>
TELEPÜLÉS	<i>Településkód</i> , <i>Településnév</i>

18. példa: Haszontalan logikai redundancia

4.5 Teljesség

Az adatmodellben előfordulhatnak hiányosságok abban az értelemben, hogy nem veszünk észre, vagy nem tartunk fontosnak bizonyos tulajdonságokat. A teljesség kritériuma azt jelenti, hogy a feltárt és fontos modell-tényezőket és azok összefüggéseit maradéktalanul meghatározzuk.

Szinte lehetetlen ellenőrizni, hogy egy egyedből elhagytunk-e odaillő tulajdonságo(ka)t (**egyedek belső szerkezete**). Ugyanakkor az egyedek összefüggéseire vonatkozó szabályok megkönnyítik a kapcsolatok teljességének az ellenőrzését (**egyedek külső szerkezete**).

Mindennek van egy axiomatikus közös alapja:

Minden egyednek kell, hogy legyen egy és csakis egy azonosító tulajdonsága és minden azonosító tulajdonság csak egy egyedben lehet ilyen szerepű.

4.6 A kritériumok alkalmazása

A szakirodalom csak a normalizálás, a minimalitás **matematikai** alapú eszközeivel foglalkozik. A fentiek bizonyították, hogy pusztán ezekkel nem lehet optimális adatmodellt alkotni; ahhoz **szemantikai** alapú eszközök is szükségesek. A kétféle eszköztár együtt úgy működik, hogy a matematika tárja fel a mennyiségi természetű problémákat, amelyek jellegétől függően szemantikai és/vagy matematikai beavatkozásra lesz szükség.

5. A NORMALIZÁLÁS HIÁNYOSSÁGAI

5.1 A normálformák (1NF-5NF) nem teljesek

Az alapvető normálformák (1NF-BCNF) még az egyébként is ellentmondásos magasabb normálformákkal (4NF-5NF) együtt se képesek biztosítani a teljes adatintegritást, vagyis az optimális adatmodell megalkotását.

T3 A normálformák mai rendszere hiányos és ezért nem alkalmas egyes tervezési anomáliák felszámolására.

Ennek az az oka, hogy rosszul határozták meg a normálformák alapját.

S31 A normálformák mai alapja inkonzisztens.

Minden osztályozás akkor konzisztens, ha az osztályba tartozó értékek **homogén halmazt** alkotnak. Vegyük ellenpéldaként azokat a statisztikákat, amelyekben a személy neme tulajdonság férfi, nő és **gyerek** (?) értékű lehet. Ezekben két ismerv értékei – a személy valódi **neme** és a személy **életkora** – keverednek egymással. Ez karbantartási anomáliával jár, mert ha a gyerek átlépi a korhatárt, akkor módosítani kell a megfelelő adatbázis-részletet.

Könnyű belátni a fenti segédétel igazságát. A normálformák két eltérő ismérven alapulnak:

- Az egyik ismerv az, hogy **létezik-e** a funkcionális függés.
- A másik az, hogy ha igen, **milyen alapú**: normál vagy kulcsrész.

	1	2	3	4	5	6	7	8
$A \rightarrow B$	K	K	K	K	N	N	N	N
$B \rightarrow C$	K	K	N	N	K	K	N	N
$A \rightarrow C$	K	N	K	N	K	N	K	N
Triviális függés	X							
Lehetetlen eset		X						
Részleges függés				X				
Tranzitív függés								X
Külső kulcsrész							X	
Belső kulcsrész			X					
Metszefüggés					X			
Csoportfüggés						X		

A függések döntési táblázata

A döntési táblázatban három tulajdonság (A, B, C) függéseit vizsgáljuk úgy, hogy azokat normál (N) és kulcsrész (K) típusokba soroljuk. Ekkor a három ismert normálalakkal szemben nyolc esettel kell számolnunk.

Ha a két ismérvet következetesen alkalmazzuk, akkor kiderül, hogy egyes BCNF alakú táblázatok még nem optimálisak!!!

A táblázat oszlopai, ún. döntési szabályai három csoportba tartoznak:

- **Kézenfekvő esetek:**

Ha B része A-nak és C része B-nek, akkor triviális, hogy C része A-nak is (1. oszlop) és az ellenkezője nem fordulhat elő (2. oszlop).
Képlettel:

$$A+B+C \rightarrow A+B \rightarrow A \text{ és } A+B+C \rightarrow A+B \rightarrow B$$

- **Ismert esetek:**

A 4. oszlop a részleges függés esete:

$$A+B \rightarrow B \text{ és } B \rightarrow C \text{ és } A+B \rightarrow C$$

A 7. oszlop a külső kulcstörő függése (BCNF):

$$A+B \rightarrow C \text{ és } C \rightarrow B \text{ és } A+B \rightarrow B$$

A 8. oszlop a tranzitív függést mutatja:

$$A \rightarrow B \text{ és } B \rightarrow C \text{ és } A \rightarrow C$$

Megjegyzendő, hogy A, B és C mindegyike lehet összetett is!

- A BCNF által **le nem fedett esetek**: 3,5,6 szabályok.

Az adatbázis-tervezés során ezekkel az esetekkel nem törődnek. A következő alponthan be fogom mutatni, hogy ez milyen anomáliákra vezet.

S32	A BCNF alakú modell még lehet szuboptimális.
-----	--

A BCNF alapuló adatbázis-tervezés az inhomogén alapismérvek miatt csak korlátozottan elemezhető, aminek legfőbb oka a csoportok hiánya.

5.2 A csoportszerkezet hiánya

A döntési táblázat mások által elhanyagolt szabályai (3,5,6 oszlopok) mindig csoportot involváltnak.

T4	A csoport konstrukciója nélkül a modellben hamis, a valóságban nem létező összefüggéseket lehet létrehozni.
----	---

A segédítél igazolásához vegyük alapul a következő egyedeket:

VEVŐ-RENDELÉSTÉTEL	Rendelésszám+Cikkszám, ...
SZÁLLÍTÓ-RENDELÉSTÉTEL	Szállítástétel-azonosító, <i>Rendelésszám, Cikkszám</i>

19. példa: Csoporthiányos modellrészlet

S41	Az A+B csoport nem azonos az A, B tulajdonság-párossal.
-----	---

Semmi se akadályozza meg, hogy a második egyedbe beillesszünk olyan sorokat, amelyekben a Rendelésszám **meg** a Cikkszám **külön-külön** létező rendelésre meg cikke utalnak, de a Rendelésszám **és** a Cikkszám **együtt** nem hivatkozik létező rendeléstételre!

A relációs rendszerek egyik furcsa sajátossága, hogy míg az $X \rightarrow Y$ függés **baloldalán** megengednek összetett tulajdonságot ($A+B \rightarrow C$), vagyis az azonosító lehet csoport is, addig a **jobboldalán** ezt kizárják ($A \rightarrow B+C$), tehát a leíró nem lehet csoport.

Ez még nem lenne baj a megvalósult adatbázisban, de érthetetlen, hogy miért nem engedik meg a használatát az elemzés és az ellenőrzés során. Azzal ugyanis rengeteg anomáliára deríthetnének fényt.

Az alábbiakban a döntési táblázat különleges eseteit mutatom be.

5.3 Belső kulcstörő függés – HBNF

A T4 tézis igazolásához vegyük elő újra a 13. példát kicsit módosítva:

CIKK	Cikkszám, <i>Egységár</i>
RENDELÉS-1	<i>Rendelésszám, Cikkszám</i> , Rendelt mennyiség
RENDELÉS-2	<i>Rendelésszám+Cikkszám</i> , Szállított mennyiség, <i>Egységár</i>

20. példa: A valóságghűség megsértése

D13	Belső kulcstörő függésről akkor beszélünk, ha egy tulajdonság függ egy összetett másiktól, de ugyanakkor meg is határozza annak egy részét. Lásd a táblázat 3. oszlopát!
-----	---

A helyzet annyiban hasonló a BCNF-hez, hogy az azonosító egy részét nemcsak a teljes azonosító határozza meg (triviálisan), hanem egy másik tulajdonság is. Annyiban tér el attól, hogy ott a másik szerepe leíró, itt pedig azonosítórész. A belső kulcstörő függés képlete:

$A+B \rightarrow A$ és $A \rightarrow B$ és $A+B \rightarrow B$

Megjegyzendő, hogy csoportok hiányában a BCNF alakot se tudjuk használni, ha az azonosító többszörösen összetett ($A+B+C$). Ha egy külső leíró D

tulajdonság annak egy részlegesen összetett részét (B+C) határozza meg így:
 $D \rightarrow B+C$, akkor a BCNF csődöt mond!

A példában a Cikkszám kulcstörő függésben áll a Rendelésszámmal:

Rendelésszám+Cikkszám $\rightarrow$ Cikkszám és Rendelésszám $\rightarrow$ Cikkszám

A BCNF-hez hasonlóan ez a helyzet se oldható fel kivétítéssel. A baj az, hogy a példa nem valósághűen tükrözi a tényeket.

A RENDELÉS-1 egyed ugyanis azt feltételezi, hogy a rendelések és a cikkek **1:N fokú viszonyban** állnak egymással, miközben a RENDELÉS-2 szerint a **viszony M:N fokú**. Ez a kétféle viszony nem lehet egyszerre igaz!

A belső kulcstörő függést feloldó normálalakot a BMW AG.-nél rólam nevezték el Halassy Béla normálformának (HBNF). A problémát nem lehet kivétítéssel megoldani: csak a valósághűség helyreállítása, vagyis a modell szemantikai átalakítása hozhat eredményt.

CIKK	Cikkszám, Egységár
VEVŐ-RENDELÉS	Vevő-Rendelésszám, Cikkszám, Rendelt mennyiség
SZÁLLÍTÓ-RENDELÉS	Szállító-Rendelésszám+Cikkszám, Szállított mennyiség, Tételár

21. példa: A HBNF helyreállítása

Vegyük észre, hogy a valósághűség mindig egyértelműsítéssel, a korábbi homonimák kiküszöbölésével jár (lásd rendelésszámok minősítése).

Vannak egytétéles vevőrendelések, amelyeknél egységárat alkalmaznak és vannak többtétéles szállítórendelések, amelyeknél tételárat használnak. A tapasztalatlan tervező az eredeti egyedekben részleges függést fedezne fel, a RENDELÉS-2 egyedből kivenné az egységárat – és azzal ismereteket veszítené!

5.4 Metszet-függés

Egy vállalatban a többtétéles rendeléseket nem egyben, hanem az adott mennyiséget részletenként diszponálva szállítják ki.

RENDELÉSTÉTEL	Rendelésszám+Cikkszám , Rendelt Mennyiség
DISZPOTÉTEL	Diszposzám+Cikkszám, Rendelésszám , Diszponált mennyiség

22. példa: Metszet-függés

Feltételek: Természetesen a diszpozíciónak mindig arra a rendelésre kell utalnia, amire maga a rendeléstétel is hivatkozik. Ezért szerepel a DISZPOTÉTEL-ben a Rendelésszám. Ezzel szemben egy diszpozíció – nem a tétel,

hanem maga a teljes szállítás – több rendelés tételeit is tartalmazhatja. Így nem áll fenn a Diszpozíciószám → Rendelésszám függés és a fordítottja sem: a rendelések és a diszpozíciók M:N fokú viszonyban állnak.

A két egyed látszólag minden igényt kielégítően 5NF alakú. Valójában egy anomáliát rejtene: nem fejezik ki az alapvető tényt, hogy minden egyes diszpozíciónak egy rendeltételre kell mutatnia. Mint fentebb kifejtettem, a **Rendelésszám, Cikkszám nem azonos a Rendelésszám+Cikkszám**mal. A második egyedbe bevihető olyan sor, amely érvényes rendelésre és cikkekre mutat, de azok érvényes párosára nem.

D14	Metszet-függésről beszélünk. ha egy egyed összetett azonosítója meghatározza egy másik összetett azonosítóját meg annak a részét is. Lásd a döntési tábla 5. oszlopát!
-----	---

A metszet-függés általános képlete a következő:

A+B → B+C és B+C → B és A+B → B

Mivel a másik két függés triviális, nem redundancia a probléma és a baj nem küszöbölhető ki hagyományos normalizálással. Csak a szemantikai megoldás, az egyedek egyértelműsítése segít.

RENDELÉSTÉTEL	Rendelésszám+Cikkszám , Rendelt mennyiség
DISZPOTÉTEL	Diszposzám+Cikkszám , Rendelésszám+Cikkszám , Diszponált mennyiség

23. példa: A metszet-függés feloldása

Tévedés lenne azt hinni, hogy a Cikkszámot kétszer kell definiálni és annak megfelelően a DISZPOTÉTEL egyedben tárolni! A csoportok mindig **virtuális konstrukciók**, amiket az elemzésben és kezelésben használunk, a tárolásban viszont nem (mint egy reláció összetett kulcsánál is tesszük). Az elemzés alapjául szolgáló metamodellben a példa az alábbi módon néz ki:

RENDELÉSTÉTEL	Rendeltétel azonosító , ...
DISZPOTÉTEL	Diszpozíció azonosító , Rendeltétel azonosító ,...

CSOPORTOK

Rendeltétel-azonosító	Rendelésszám+Cikkszám
Diszpozíció-azonosító	Diszposzám+Cikkszám

24. példa: Korrekt modellrészlet

Mint látjuk az élethű metamodell nemcsak egyedeket, egyed-tulajdonság-viszonyokat, hanem tulajdonság-struktúrákat (itt: csoportokat) is tartalmaz. Ezek a hagyományos normálforma elemzésekből kimaradnak és ezért azok nem mindig alkalmasak az optimális adatmodell megtervezésére.

5.5 Csoport-függés

Most az előbbivel szemben azt feltételezzük, hogy minden rendeléstételt külön diszponálnak és a Diszposzám egyszerű sorszám.

DISZPOZÍCIÓ	Diszposzám, Rendeléstétel azonosító, Cikkszám
CSOPORTOK	
Rendeléstétel-azonosító	Rendelésszám+Cikkszám

25. példa: Csoport-függés

A példa rejtett redundanciát tartalmaz, mert kétszer szerepel benne a Cikkszám: egyszer expliciten, egyszer pedig impliciten az Rendeléstétel-azonosító részeként. Azonban nem a többszörösség az egyetlen lehetséges baj. Előfordulhat, hogy a két Cikkszám értéke nem ugyanaz! Ekkor a normalizálással való eltávolítása ismeretvesztéssel járna.

D15	Csoport-függésről beszélünk, ha egy egyed egy összetett tulajdonság részeként is és egyszerűként is tartalmaz egy tulajdonságot. Lásd a döntési tábla 6. oszlopát!
-----	---

A csoport-függés általános képlete a következő:

$A \rightarrow B+C$ és $B+C \rightarrow B$ és **A $\rightarrow$ B**

Jelen esetben is, mint minden csoportot involváló példában, a három függés **matematikai képlete** mutat rá a hibára, de az eset **szemantikai megoldást** igényel a szokásos matematikai – kivetítés – helyett.

DISZPOZÍCIÓ	Diszpozíciószám, Rendeléstétel azonosító, Helyettesítő-Cikkszám
CSOPORTOK	
Rendeléstétel-azonosító	Rendelésszám+Cikkszám

26. példa: A csoport-függés feloldása

Magyarázat: A Cikkszám nem ugyanazt jelentette, mint az Rendeléstétel azonosítóban lévő implicit Cikkszám. Gyakran előfordul, hogy nem tudják a kért cikket szállítani és a vevő tudtával valamilyen helyettesítő árut adnak a megrendelőnek. Ebben az esetben világos, hogy a kétféle tétel homonima.

Ha a két Cikkszám értéke mindig azonos, úgy az elemi tételt egyszerűen el kell hagyni az egyedből. Ha nem, akkor a jelentésének megfelelően kell minősíteni. A minősítésre és a vele kapcsolatos szerepnév fogalomra a következő fejezetben térek ki.

6. MELLŐZÖTT ESETEK

Egyesek annyira a szokványos normálformák bűvöletében élnek, vagy éppen azok meghaladását célzó új normálformák kiagyalásán ötölnek, hogy a közismert esetek összetettebb változataira nem is gondolnak.

6.1 Pszeudo-tranzitívitás

Egy sajátos függési helyzet is igazolja a csoportok nélkülözhetetlen voltát. Tegyük fel, hogy a Diszposzám meghatározza a Rendelésszámot!

RENDELÉSTÉTEL	<i>Rendelésszám+Cikkszám, Rendelt mennyiség</i>
DISZPOTÉTEL	<i>Diszposzám+Cikkszám, Rendelt-mennyiség, Diszponált-mennyiség</i>

27. példa: Pszeudo-tranzitívitás

D16	Egy egyed pszeudo-tranzitív függést tartalmaz, ha a D tulajdonsága függ a saját B+C azonosítója mellett egy másik egyed A+C azonosítójától is úgy, hogy az összetett azonosítók nem-közös részei között ($A \rightarrow B$ vagy $B \rightarrow A$) függés áll fenn.
-----	--

Ez a tranzitívitás alapképlete úgy, hogy a második és harmadik tagban van egy kiegészítő közös elem is (C).

$A \rightarrow B$ és $B+C \rightarrow D$ és **$A+C \rightarrow D$**

A hagyományos normalizálás szerint itt semmilyen probléma sem lépett fel, hiszen **látszólag** nem sérült a harmadik normálforma. A Diszposzám $\rightarrow$ Rendelésszám függést elfedi, hogy az azonosítók olyan csoportok, amikben szerepel egy közös, a függéseket tekintve semleges („konstans”) tag is.

A pszeudo-tranzitivitást ugyanúgy kell feloldani, mint az egyszerűt: a redundáns Rendelt-mennyiséget el kell távolítani az alárendelt egyedből.

6.2 Közvetett tranzitívitás

A szokásos normalizálási eljárás során csak a tulajdonságok **egyedeken belüli** függési viszonyaival foglalkoznak, az **egyedek közöttiekkel** nem.

VEVŐ	Vevőkód, Vevő telephely
SZERZŐDÉS	Szerződésszám, Vevőkód
RENDELÉS	Rendelésszám, Szerződésszám
SZÁLLÍTÁS	Szállításkód, Rendelésszám, Vevőnév

28. példa: Közvetett tranzitívitas

Az egyedek célszerű normalizálási sorrendjét pedig hírből sem ismerik. Persze most, hogy együtt látjuk az egyedeket azonnal feltűnik a kettős baj. Az, hogy a Vevőnév a SZÁLLÍTÁS egyedben tranzitív függésű, a VEVŐ-ből meg hiányzik. Azonban a példa túl egyszerű és a valóságban nem mindig látjuk együtt az egyedeket. Mi történik, ha a tételt elhagyják a SZÁLLÍTÁS egyedből, de nem helyezik a VEVŐ-be, amit normalizálnak tartanak?

A normalizálás a közvetlen meghatározó egyedébe való kivetítés, vagyis a Vevőnevet először áttolják a RENDELÉS-be, onnan a SZERZŐDÉS-be és csak a harmadik lépésben kerül a korrekt helyére, a VEVŐ-be.

Nyilvánvaló, hogy az egyedek egyenként történő ad hoc sorrendű normalizálása nemcsak időigényes, hanem hibáknak is kitett. Én egy másféle normalizálási módszert használok, lásd majd a *függési fonal* fogalmát.

D17	Egy adatmodell közvetett tranzitívitas tartalmaz, ha a függő tulajdonságot olyan másik határozza meg, amely nem az adott egyed közvetlen fölérendeltjének az azonosítója.
-----	--

A közvetett tranzitívitas képlete:

$A \rightarrow B, F$ és $B \rightarrow \dots \rightarrow E$ és **$E \rightarrow F$**

Ez a probléma nem is olyan egyszerű, mint amilyennek tűnik. Lássunk csak egy összetett képletet, amelyben F sehol se azonosító. Honnan kell kivetíteni és hol fog megmaradni?

$E1 (A, \dots)$ és $E2 (B, A, \mathbf{F})$ és $E3 (C, A, \mathbf{F}) \dots E_x (D, B, C, \mathbf{F})$

6.3 Egyszerű és összetett ciklus

A valós életben sokszor találkozunk az alábbihoz hasonló helyzettel, amikor az eltérő felhasználók látásmódjai sajátosak.

GÉPJÁRMŰ	Rendszám, Casco-azonosító, Kocsitípus
CASCO	Casco-azonosító, Rendszám, Kocsitípus

29. példa: Egyszerű ciklus-1

GÉPJÁRMŰ	<i>Rendszám, KGFB-azonosító, Kocsitípus</i>
KGFB	<i>KGFB-azonosító, Rendszám, Kocsitípus</i>

30. példa: Egyszerű ciklus-2

D18	Az adatmodell egyszerű ciklust tartalmaz, ha két egyed kölcsönösen tartalmazza egymás azonosítóját.
-----	--

Az egyszerű ciklus képlete: $A \leftrightarrow B$.

Ezekben az esetekben a hagyományos normalizálás, vagyis a tranzitív Kocsitípus kivetítése nem történhet a hagyományos módon mert minden a Rendszámtól függő tulajdonság **formálisan** függ a Casco-azonosítótól és a KGFB-azonosítótól is.

A megoldáshoz a függéserőt és az időfaktort kell vizsgálni.

- Minden gépjárműnek van casco-ja? Mivel nem midegyiknek van, a Rendszám $\leftrightarrow$ Casco-azonosító kölcsönös függés a baloldaltól **gyenge**. Ezért a GÉPJÁRMŰ-ből elhagyjuk a Casco-azonosítót, a CASCO-ból pedig az immár világosan tranzitív Kocsitípust.

GÉPJÁRMŰ	<i>Rendszám, Kocsitípus</i>
CASCO	<i>Casco-azonosító, Rendszám</i>

31. példa: Egyszerű ciklus-1 feloldása

- Időben változatlan a gépjármű kötelező biztosítása? Mivel változik, a Rendszám $\leftrightarrow$ KGFB-szám függés a baloldaltól **látszólagos**. Ezért itt is az előzőhöz hasonló módon járunk el.

GÉPJÁRMŰ	<i>Rendszám, Kocsitípus</i>
KGFB	<i>KGFB-azonosító, Rendszám</i>

32. példa: Egyszerű ciklus-2 feloldása

Itt kell megemlékezni az ún. alternáló azonosítóról.

D19	Egy B tulajdonság az A azonosító tulajdonság alternáló azonosítója , ha az A minden egyes értékéhez minden időben a B egy adott (nem-nulla) értéke tartozik és fordítva.
-----	---

Az alternáló és az alternatív nem ugyanaz! Az utóbbi mindig pont két dolog között áll fenn: *vagy az egyik, vagy a másik*. Az alternáló dolog lehet többszörös is.

Az alternáló azonosító igen ritka jelenség. Sokféle szervezet ad saját, belső használatú azonosítót a személyeknek. Azonban ezek értéke általában nem örök és/vagy nem stabil. Például a SZIG-szám nem a Személyi-szám

alternáló azonosítója, mert csak adott korban felül létezik és időben változhat. Általánosan ha a kölcsönös függés az egyik oldalról gyenge, akkor nem lehet szó alternáló azonosítóról.

A fenti két egyszerű esetről sokkal bonyolultabbakkal is találkozhatunk. Például valaki vezetni akarja a megyéket, a járásokat és a városokat, illetve a megye és a járás központját jelentő településeket.

MEGYE	Megyekód, <i>Városkód</i>
JÁRÁS	Járáskód, <i>Megyekód</i> , <i>Városkód</i>
VÁROS	Városkód, <i>Járáskód</i>

33. példa: Összetett ciklus

D20	Az adatmodell összetett ciklust tartalmaz, ha két egyed azonosítója más egyedeken át közvetve meghatározza egymást.
-----	--

Képlete: $A \rightarrow B$ és $B \rightarrow \dots \rightarrow E$ és $E \rightarrow A$.

Esetünkben: Városkód $\rightarrow$ Járáskód $\rightarrow$ Megyekód $\rightarrow$ Városkód

Az összetett ciklus körkörös **beillesztési anomáliát** okoz. Csak akkor tudunk az adatbázisba vinni egy várost, ha már abban van az általa hivatkozott járás, abban az általa hivatkozott megye, ami visszamutat a városra.

Eddigi ismereteink alapján ezt a problémát nem lehet megoldani. Ezért a megoldást elnapolom a 7.2 Feltételes függés pont végére. A Városkód lila színe mutatja, hogy itt a baj kettős természetű: nemcsak redundanciát, hanem egyértelműségi hiányt is jelez. Ugyanis arra a városra akarna utalni, amely a megye székhelye, de ez nem tűnik ki a modellrészletből.

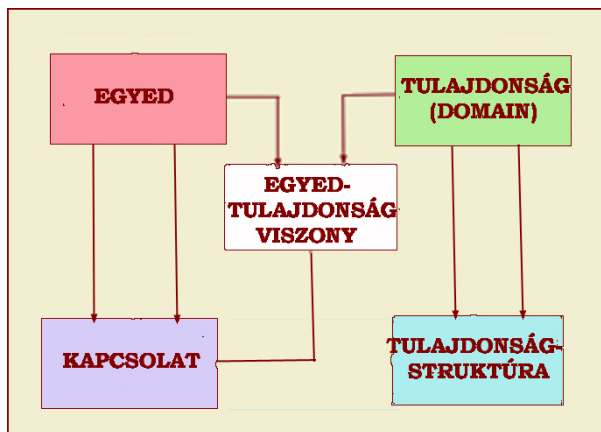
7. SZEREPNEVEK

Elérkeztünk oda, hogy egy új adatmodellezési konstrukciót kell bevezetnünk, amit a hagyományos normalizálásban ugyanúgy nem használnak, mint a csoportot – amivel ráadásul kombináltan is fellép – és emiatt nem is tudnak meghatározni egyes hasznos, sőt nélkülözhetetlen szerkezeteket.

Gyakran előfordul, hogy egy adott egyed típus bizonyos előfordulásait az alaptulajdonságokon kívül speciálisakkal is le akarjuk írni, illetve az alaptulajdonság speciális értékeit külön tulajdonságként akarjuk meghatározni. Az alábbiakban három ilyen esetet fogok bemutatni. Előtte azonban tisztázni kell, hogy mi a szerepnév lényege.

7.1 Tulajdonság, doméjn, attribútum

A tulajdonságtípus tényezőjét az adatmodellben kétféle értelemben és kétféle konstrukcióban használjuk. A kényelem kedvéért megismétlem itt az adatbázis metamodelljét:



Az adatbázis metamodellje

D21	A domain (helyes kiejtése doméjn!) az egyedektől független tulajdonság értékhalmaza. Úgy is mondhatjuk, hogy abszolút tulajdonság.
D22	Az attribútum egyed típusba leképezett doméjn, egyed-tulajdonság-viszony, másképpen (egyed-kötött) relatív tulajdonság.

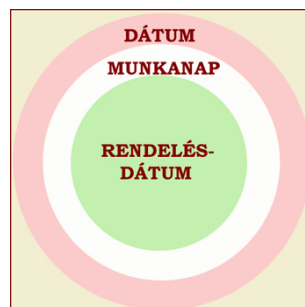
A doméjnek viszonyait **tulajdonság-struktúráknak** hívjuk. Ezek nélkülözhetetlenek a tökéletes adatmodell megalkotásához, bár a hagyományos normalizálásban szó sem esik róluk.

A struktúrák a következők lehetnek:

- Az 5.2 pontban szóltam a **csoport-szerkezetek**ről.
- Nem fogom részletezni a **számítási-szerkezeteket**. A számított tulajdonságok értelemszerűen redundánsak, de a számítási algoritmusok tényezőit és magát az algoritmust célszerű korlátként megadni.
- A továbbiakban a **szerepnév-szerkezetek**ről lesz szó.

D23	Egy tulajdonság egy másik szerepnév , ha értékei az utóbbi értékalmazának a (valódi vagy nem-valódi) részhalmazát alkotják.
-----	--

Vegyünk például a Rendelés-dátum tulajdonságot és doméjnt! Tegyük fel, hogy rendelést csak munkanapon lehet feladni! Ekkor három értéktartománynak a következő összefüggéseit fedezhetjük fel:



Értéktartományok

Képlettel: $\text{Dátum} \supset \text{Munkanap} \supseteq \text{Rendelés-dátum}$.

Magyarázat:

Minden munkanap dátum, de nem minden dátum munkanap. A munkanap a dátum **valódi részhalmaza**.

Minden rendelés-dátum munkanap és minden munkanap **elvileg** lehet rendelés-dátum (bár **gyakorlatilag** ez nem valószínű). Ettől függetlenül a rendelés-dátum a munkanapnak **nem-valódi részhalmaza**.

A szerepnévnek két válfaját kell megkülönböztetni:

- **Korlátozást** szolgáló szerepnév. Ez esetben a részhalmaz-függés a **kölcsönös függés** olyan speciális esete, amely az egyik irányban gyenge. Példa: Minden Export-rendelésszám is egy Rendelés-szám, de

nem minden Rendelésszám Export-rendelésszám. Ezt a helyzetet **részhalmaz**-függésnek nevezem és a következő jelölést használom rá:

$$A' \Rightarrow A$$

- **Minősítést** szolgáló szerepnév. Ebben az esetben a részhalmaz nem valódi: megegyezik a teljes értékhalommal és arra való, hogy annak eltérő jelentésű értékeit megkülönböztesse egymástól. Példa: Minden Születésnap és Névnapi egyben Naptári nap is, aminek minden értékét felvehetik. Ennek a függésnek **osztályozó**-függés a neve. Képlete:

$$A' \Leftrightarrow A$$

Ez ritka eset. Általában akkor fordul elő, ha egy egyedben két azonos értelmű, de eltérő értékű tulajdonságot veszünk fel.

SZEMÉLY	Személy-azonosító, Születésnap, Névnapi
----------------	--

34. példa: Kettős szerepnév

Evidens, hogy a Születésnap is és a Névnapi is egy dátum, tehát mind a kettő a Dátum doméjn szerepneve. Mivel a relációs rendszerek nem ismerik a többszörös kapcsolat tényezőjét, ezzel a helyzettel nem tudnak mit kezdeni és a Személy relációt elkezdik lebontogatni.

7.2 Feltételes függés

A relációs rendszerekben alkalmazzák a **nulla érték** fogalmát. Önmagában is ellentmondásos, mert a ki nem töltött nem-numerikus adatokat is így nevezik. Ennél azonban nagyobb baj, hogy nem különböztetik meg a jelenséget okozó különböző helyzeteket.

A lényeg, hogy a SZEMÉLY egyed hipotetikus Szülések-száma tulajdonságával szemléltetem. Négy eset lehetséges:

- „A” személy nő, aki N-szer szült. A tulajdonság értéke **N**.
- „B” személy nő, aki egyszer sem szült. Az érték **valódi 0**.
- „C” személy nő, akiről nem tudjuk, hogy hányszor szült. Az érték ekkor **ismeretlen 0**.
- „D” személy férfi vagy kicsi lány. Az érték **nem-értelmezhető (NA)**.

A megkülönböztetés szemantikai és gyakorlati (pl. statisztikai) értelmét nem akarom most kifejtteni. Arra viszont rá kell mutatnom, hogy a „nem-értelmezhető” érték is valódi, a nullától eltérő tartalom.

Jelöljük ezt az értéket az „NA” jellel, ami nem-alkalmazhatót (angolul: not applicable) jelent. A továbbiakban a nem-értelmezhető tartalmat **NA-érték**-nek fogom nevezni. (Figyelem: az NA nem tévesztendő össze a statisztikában használt n.a. „nincs adat” megjelöléssel, amely összekeveri a nem-ismert, inszignifikáns, ismeretlen és nem-értelmezhető lényegeket!)

Az NA-értékre épül saját találmányom, az ún. **feltételes függés**.

D24	Egy E egyed A azonosítója akkor és csak akkor határozza meg feltételes függéssel az egyed B tulajdonságát, ha létezik olyan C tulajdonság, amelynek adott tartalma esetében a B mindig NA-értéket vesz fel
-----	---

SZEMÉLY	Személy-azonosító , Foglalkozás, Orvoscód, Rendelő-körzet
----------------	--

35. példa: Feltételes függés

A feltételes függés jele $C(i) \rightarrow B$, ahol C a feltételtényező és i a konkrét feltétel. Mármint ha a függés Foglalkozás (orvos) $\rightarrow$ Orvoscód, akkor valódi értéke kell, hogy legyen az Orvoscódnak és a hozzá kapcsolódó adatoknak. Ellenben ha a függés Foglalkozás (nem-orvos) $\rightarrow$ Orvoscód, akkor az utóbbi és a hozzá kötődő összes tulajdonság NA-értékű kell, hogy legyen.

A feltételes függés az opcionalitás egyik sajátos esete.

A Szülések-száma erősen függ a Személy-azonosítótól, ha minden személynél meg kell adni egy konkrét értéket, ami lehet NA is! Egyébként a függés gyenge.

A relációs rendszerekben nincs eszköz, ami szabályozná, hogy mikor kell és mikor nem szabad kitölteni az Orvoscód és Rendelő-körzet adatot.

Az intelligens modellezésben a feltételes függés egy jelzés, ami alapján két teendő várhat a tervezőre:

- Mérlegelheti, hogy érdemes-e specializálni a feltételesen függő tételt egy egyedaltípusba. Lásd a következő pontot. Ekkor voltaképpen a megkötést egy szerkezeti tényező fogja érvényesíteni.
- Ha nem alkot altípust, akkor más módon kell ezt tennie. A SZEMÉLY egyednél korlátként kell kifejeznie, hogy mikor adható meg és mikor nem valós Orvoscód érték a hozzá kötődő adatokkal együtt.

Az eddigi normalizálási eljárások **intenzionálisak** voltak. A tulajdonságok viszonyait elemeztük és a táblát „horizontálisan” bontottuk meg.

Azonban vannak esetek, amikor a relációt **extenzionálisan** kell megbontani, ami a tábla sorai mentén, „vertikálisan” történik. Adott feltételek esetén az egyed-előfordulásokat új egyedtipusokban osztjuk szét.

ERŐFORRÁS	Erőforrás-azonosító, Személy-adatok, Anyag-adatok, Eszköz-adatok
SZEMÉLY	Személy-azonosító, Személy-adatok
ANYAG	Anyag-azonosító, Anyag-adatok
ESZKÖZ	Eszköz-azonosító, Eszköz-adatok

36. példa: Alapegyed és extenzionális megbontása

Ha az alapegyedben nem szerepelnek az eltérő erőforrásfélék azonosítói, akkor az egyed „tökéletes” 5NF alakban van, mivel a tulajdonságai egymástól függetlenek és intenzionálisan nem normalizálható. Mégis tudjuk, hogy az ERŐFORRÁS rosszul tervezett egyed. Mert ha az erőforrás személy, akkor az anyag és az eszköz adatai NA-értékűek. Ha anyag, akkor a személy és az eszköz adatai nem értelmezhetőek stb.

A tábla **alternáló NA-értékei** jelzik, hogy a jelenségeket nem modelleztük helyesen. Ezért a táblát a sorai mentén extenzionálisan meg kell bontani a mutatott módon.

Amint látjuk, vannak esetek, amikor a hagyományos normalizálás mint matematikai eljárás önmagában nem vezet integrált adatmodellhez és ha tetszik, ha nem szemantikai (itt: osztályozási) eljárást kell alkalmazni.

Végül térjünk vissza a 33. példa ciklusára.

MEGYE	Megyekód, <i>Városkód</i>
JÁRÁS	Járáskód, <i>Megyekód</i>
VÁROS	Városkód, <i>Járáskód</i>

MEGYE	Megyekód
JÁRÁS	Járáskód, <i>Megyekód</i>
VÁROS	Városkód, <i>Járáskód, Megyeszékhely, Járásszékhely, Székhely-mutató</i>

SZEREPENEVEK

Megyekód	Megyeszékhely
Járáskód	Járásszékhely

37. példa: Az összetett ciklus feloldása és szemantikai tisztázása

Itt összetett feltételes függéssel találkozunk. A Székhely-mutató három értéket vehet fel és azoktól függően lesz a három kapcsoló adat NA vagy konkrét értékű. Mivel se a megyében, se a járásban nincs utalás városra, a szokásos hierarchikus bevitelnek (és más műveleteknek) nincs akadálya.

Az már nem tervezési, hanem kezelési-szabályozási kérdés, hogy a városoknál betartják-e a megfelelő kapcsolati korlátot.

7.3 Explicit egyedaltípus – Specializáció

Gyakran előfordul, hogy egy egyedtípusnak vannak sajátos tulajdonságai, amelyek bizonyos egyed-előfordulásokra értelmetlen (NA) értéket vesznek fel. Tipikus példa az alábbi:

RENDELÉS

Rendelésszám	Rendelésirány	Devizanem	Országkód
1	hazai	-	-
2	export	dollár	X
3	export	euró	Y
4	hazai	-	-

RENDELÉS	Rendelésszám
EXPORT-RENDELÉS	Export-rendelésszám, Devizanem, Országkód

SZEREPENEVEK

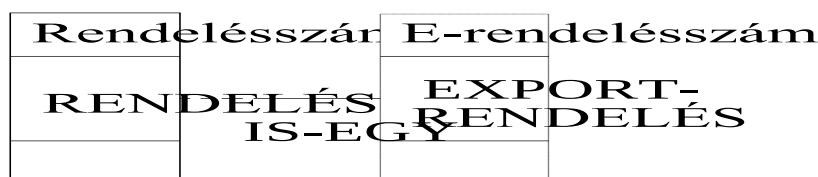
Rendelésszám	Export-rendelésszám
--------------	---------------------

38. példa: Egyed típus és lebontása altípussal

Az alapegyed egyik tulajdonsága (Rendelésirány) adott értékétől (export) feltételesen függenek más tulajdonságainak (Devizanem, Országkód) értékei. Ezeket a tulajdonságokat külön egyedbe választjuk le, amelynek azonosítója (Export-Rendelésszám) az eredeti azonosító (Rendelésszám) szerepneve. Az így keletkezett egyed az eredeti egyed **altípusa**.

D25 Specializációnak hívjuk egy egyed feltételesen függő tulajdonságainak az egyed-altípusba történő leválasztását. (A fordított összevonási művelet a **generalizáció**.)

Minden Export-rendelésszám is egy Rendelésszám, de nem minden Rendelésszám egyben Export-rendelésszám is. Így a közöttük fennálló kölcsönös függés a Rendelésszám felől gyenge. Ez a részhalmaz-függés tipikus esete.



Az egyed és annak altípusa 1:1 fokú felülről opcionális, alulról kötelező úgynevezett **IS-EGY** kapcsolatban áll egymással.

Az egyedtypusnak és altípusának értelemszerűen nem lehetnek közös tulajdonságai, mert azok egytől-egyig tranzitív függésűek lennének.

Nincs egzakt szabály arra nézve, hogy mikor **kell** egyedaltípust alkotni. Általában akkor **ajánlatos**, ha az egyednek vannak összetartozó gyengén függő leírói és/vagy egyes előfordulásai különös kapcsolatokat hordoznak. A példában feltételezhető a DEVIZA és ORSZÁG egyedekre történő kimutatás. (A kettő nem tartozik egybe: európai országban is fizethetünk dollárral.)

7.4 Implicit egyedaltípus – Visszamutatás

Olykor előfordul, hogy különböző okokból egy egyedtypus előforduláshalmazában egyesek kitüntetett jelentőségűek. Ilyenek szoktak lenni az ún. „**leg**ek”, a legelső, a legnagyobb, a legfontosabb stb.

Egy lehetséges példa az alábbi:

SZEMÉLY

Személy-azonosító	Főnök-azonosító
A	C
B	C
C	D

SZEMÉLY	Személy-azonosító, Főnök-azonosító
---------	------------------------------------

SZEREPEK

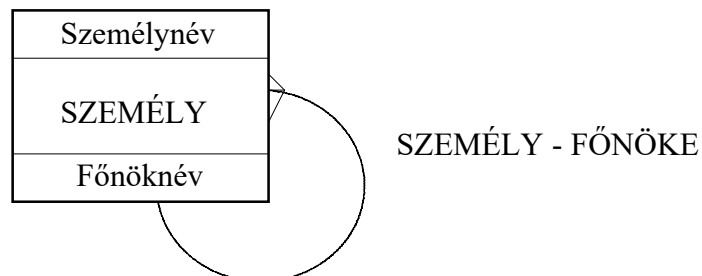
Személy-azonosító	Főnök-azonosító
-------------------	-----------------

39. példa: Személyek és főnökeik

A főnök (C) is egy személy, akinek a saját tulajdonságai egy saját sorban jelennek meg.

A SZEMÉLY egyed normalizált. Viszont anomáliákat rejt. Ha kitöröljük a harmadik sort, akkor ki kellene törölnünk az első kettőt is. Ha módosítjuk a főnök, mint személy azonosítóját akkor a megfelelő sorban is azt kell tenni. Azonban honnan tudjuk ezt? Sehonnan, hacsak nem alkalmazunk rá egy sajátos konstrukciót, aminek **visszamutató kapcsolat** a neve.

D26	Visszamutató kapcsolatról beszélünk, ha egy egyedtypus előfordulásai ugyanannak az egyedtypusnak más előfordulásaival állnak összefüggésben.
-----	--



A visszamutató kapcsolat két oldalról opcionális N:1 fokú viszony. Nem minden személy főnök és nincs minden személynek főnöke.

7.5 Családfa egyed és többszörös kapcsolat

A valós életben gyakran előfordul, hogy egy egyedtípus több előfordulását kell egymáshoz kapcsolni. Ilyen az emberek családfája, az alkatrész-tételek darabjegyzéke/beépülési jegyzéke stb.

TÉTEL		CSALÁDFA			
Tételkód	Tételnév		<i>Tételkód-F</i>	<i>Tételkód-A</i>	Darab
T1	A		T1	T2	2
T2	B		T1	T3	2
T3	C		T2	T4	12
T4	D		T3	T4	8

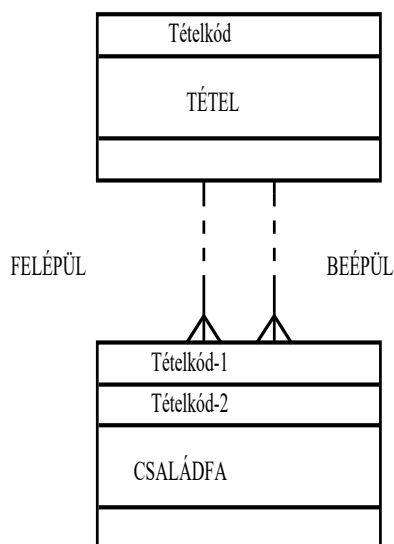
TÉTEL	Tételkód, Tételnév
CSALÁDFA	<i>Tételkód-1, Tételkód-2</i> , Beépülő darab

SZEREPENEVEK

Tételkód	Tételkód-1 (főlérendelt)
Tételkód	Tételkód-2 (alárendelt)

40. példa: Családfa szerkezet

D27	Családfa kapcsolat ról beszélünk akkor, ha egy egyedtípus előfordulásai ugyanannak az egyedtípusnak más előfordulásaival állnak többszörös viszonyban.
-----	---



A családfa kapcsolat kétszeres felülről opcionális, alulról kötelező 1:N fokú viszony, amiben a két azonosítórész értéke nem lehet azonos és azok több előforduláson át sem alkothatnak ciklust.

A családfa kapcsolat sajátos válfaja lenne a **házasság-típusú** kapcsolat, ami az előzőtől abban tér el, hogy a viszony pontosan 1:2 fokú. Azonban ennek a használata az időfaktor miatt (vö. válaszok) nem igazán célszerű. Viszont pont az időfaktor miatt a családfa egyed azonosítójában harmadik tényezőként gyakran megjelenik a dátum.

SZEMÉLY	Személy azonosító, Személy név, Személy nem
HÁZASSÁG	Férj azonosító, Feleség azonosító, Házasság dátum

41. példa: Időfaktoros családfa

Itt korlátként kell meghatározni, hogy a családfa azonosítójának az első tagja csak férfi, második tagja csak nő lehet (vagy fordítva).

Megjegyzés: Célszerű a neveket ragozatlan formában megadni. A Házasság dátuma és a Házassági dátum, a Személyi azonosító és a Személy azonosítója mint ún. **technikai szinonimák nem fognak egymásra találni a modellező automatában.**

A családfa egyed többszörösen kapcsolódik az alapegyedéhez. Láttunk másik esetet is a kétszeres kötődésre, vö. Cikkszám és Helyettesítő cikkszám (26. példa). Mivel a relációs rendszerekben az alárendeltek csak egyszeresen kapcsolhatók a fölérendelthez, mert a kapcsolatokat nem lehet minősíteni, kijelenthetjük, hogy

a relációs rendszer képtelen egyes természetes helyzetek modellezésére.

Ennek megfelelően a fenti példa alábbi változatát se képesek befogadni, jóllehet az tökéletes normálformákat mutat feltéve, hogy a Férj azonosítót és a Feleség azonosítót szerepnévként határozzák meg.

SZEMÉLY	Személy azonosító, Személy név, Személy nem
HÁZASSÁG	Házasság azonosító, <i>Férj azonosító</i>, <i>Feleség azonosító</i>, Házasság dátum

SZEREPEK

Személy azonosító	Férj azonosító
Személy azonosító	Feleség azonosító

42. példa: Időfaktoros családfa változata

8. AZ EGYEDEK KAPCSOLATAI

8.1 A relációs rendszerek hibás kapcsolatfelfogása

A kapcsolat az adatmodellnek az egyeddel és a tulajdonsággal azonos fontosságú, egyenrangú tényezője. Sőt, mint tudjuk, a sima adatállományok rendszertelen halmazából éppen a kapcsolatokat által lesz adatbázis.

Ehhez képest

a relációs modell a kapcsolatot harmadrangú tényezőnek tekinti.

Ezt a következő tények bizonyítják:

- A kapcsolat **nem fogalmi szintű** tényező. Ezt mutatja, hogy nem lehet nevet adni a kapcsolatoknak. Megsértve az ún. fizikai adatfüggetlenség fennmarasztalt elvét, a kapcsolatokat belső kulcsok és indexeik hordozzák.
- Adatokat az ún. join, azaz összekapcsolás műveletével **egymással össze nem függő** egyedekből is össze lehet válogatni úgy, hogy az M:N fokú viszonyban álló egyedekből összevegyített adatok téves ismereteket is tartalmazhatnak.
- Ennek ellentétéképpen az egymással elvileg kapcsolatban álló egyedek között adott esetekben nem lehet gyakorlati kapcsolatot létesíteni. Ennek a relációs rendszerek **konstrukciós szegénysége**, a csoportok és szerepnevek hiánya az oka.

A relációs rendszerben nem adható meg többszörös, köztük családka kapcsolat, visszamutató és leíró vagy azonosítórész csoportokra épülő kapcsolat.

Az adatbázis tervezése során a kapcsolatokat ábrán kell megadni úgy, hogy az egyedek „dobozai” közé vonalakat húzgálnak. Ez nem éppen fogalmi modellezésre valló módszer.

Egy valódi fogalmi adatmodellben, szemben a relációssal, a következő esetekben van kapcsolat az egyedek között:

- A fölérendelt A egyed azonosítója az alárendelt B egyed leíró tulajdonsága vagy összetett azonosítójának a része ($B \rightarrow A$ vagy $A+B \rightarrow A$).

- Az alárendelt A' egyed azonosítója a fölérendelt A egyed azonosítójának a szerepneve (egyedaltípus, $A' \Rightarrow A$).
- Az A egyed leíró tulajdonságának a szerepneve az egyed azonosítójával egyezik (visszamuszáttás, $A \Rightarrow A'$).

Ezek az esetek akkor is fennállnak, ha az ún. kapcsoló tulajdonságok összetettek, vagyis csoportok, vagy ha a kapcsoló tulajdonság szerepnév.

Például az A azonosította egyedben lévő $B+C$ leíró csoport kapcsolatot teremt a $B+C$ azonosította egyed felé, amint hogy az A azonosította egyedben lévő B' szerepnév kapcsolatot teremt a B azonosította egyed felé.

Amennyiben a fölé- vagy alárendelt tulajdonság szerepnév vagy csoport, azt a definíció kiegészítéseként expliciten meg kell adni.

Végeredményben a fogalmi adatmodellben sokkal többféle kapcsolatot lehet definiálni, mint a relációs rendszerekben. Ugyanakkor csak 1:M és 1:1 fokú kapcsolatok adhatók meg: az M:N fokú viszonyok nem kapcsolatok!

A kapcsolat metat tulajdonságai a következők:

A – B kapcsolat **Kapcsolatnév**, Kapcsolat fok, Kapcsolat erő, Fölérendelt, Alárendelt
43. példa: Kapcsolat definíció

A relációs rendszer hibás kapcsolat-fogalmára jellemző, hogy a kapcsolat fokát és erejét (opcionális) kettősen, a két kapcsolt egyed irányából külön-külön kell megadni. Ez sületlenség, mert a kapcsolat nem két, hanem egyetlen fogalmi tényező.

8.2 Szemantikai szegénység

Valamikor 1975 táján egy levelezésben felhívtam Codd úr figyelmét arra, hogy az ún. relációs modell szemantikailag rendkívül szegényes. Nem enged meg olyan struktúrákat és korlátokat, amik a valóságban léteznek és amik a hasznára válnának. Némileg át is dolgozta eredeti koncepcióját, de új írása a címe dacára [2] alig tartalmaz újdonságot, szemantikait meg egyet sem.

Egy idevágó példa. A relációs modell már megengedi az ún. birtoklási („has a”) 1:N fokú kapcsolatok mellett az ún. is-egy („is a”) 1:1 fokúakat is. Ennek értékeléséhez nem árt tudni, hogy a szemantika – ha jól tudom – tizenhét kapcsolatfajtát különböztet meg. A mondottak mellett olyanokat, mint a hasonlóság, a leszármazás, a hozzátartozás, az elsődlegesség (általában sorrendiség), az általam alkalmazott családfa, visszamuszáttás és többszőrösség, valamint az itt ki nem fejtendő egész-rész viszony.

9. A NORMALIZÁLÁS ÖSSZEFOGLALÁSA

A ma tanított és alkalmazott normalizálási módszerekről a következőket lehet elmondani:

- A tulajdonságok közötti **függési helyzeteket nem fedik le**. A döntési táblázat által mutatott nyolc közül hármat (3., 5. és 6. szabályok) még hírből se ismernek.
- Ugyanakkor sokat babrálnak a **haszontalan 4NF és 5NF** alakokkal, amelyekről bebizonyítottam, hogy a többértelműség elősegítésével többet ártanak, mint használnak.
- Még az egyszerűbb függési helyzeteket is rosszul kezelik, amennyiben **nem ismerik a csoport fogalmát**. Mint láttuk, az téves kapcsolatú alárendeltek bevitelére vezethet.
- Nem ismerik a normalizálás részlegességét, vagy éppenséggel végzetes **hibáját eredményező függési helyzeteket**, mint a pszeudo-tranzitivitás, összetett tranzitivitás, valamint az egyszerű és összetett ciklus.
- Mivel nem ismerik a **szerepneveket**, képtelenek kifejezni és megfelelő korlátokhoz kötni az olyan elkerülhetetlen vagy hasznos konstrukciókat, mint az egyedaltípus, visszamutató kapcsolat és családfa egyed.
- Egyes konstrukciók korlátként definiálandó feltételekhez kötöttek. A mai **normalizálási eljárások nem ismernek ilyen korlátokat**.
- Maga a normalizálási eljárás az egyedek egyenkénti normalizálásának **a minden rend nélküli sora, darab-darab alapon**. Ezért előfordul, hogy egy már normalizált egyedbe új tulajdonság kerül és azt vagy újra normalizálják, vagy sem.
- Ha a már meglevő adatmodell valamelyik pontján megváltozik, akkor **nem tudható, hogy a változás milyen tényezőket érint**, emiatt újra kell normalizálni a teljes szerkezetet.

Mindezen hiányosságoknak a töredéke is elég ahhoz, hogy igazoltnak tekintsük a T1 alaptézist, amely szerint

a mai integrálási módszerek elemi iskolás fokúak és valójában nem is érdemlik meg a nevüket.

II. RÉSZ

AZ ADATMODELLEZÉS

GYAKORLATA

10. NORMALIZÁLÁSI KONCEPCIÓK

10.1 Két hagyományos eljárás

Az 1NF-3NF alakokat eredetileg a relációk egyenkénti vizsgálatával és a helytelen függésű tulajdonságok kivetítésével (projection) kívánták elérni. Az eljárás a relációk belső szerkezetét vizsgálja és **attribútum-függéseken** alapul. Az eljárás közismert neve: **adatmodell-analízis** vagy **-dekompozíció**.

Idővel rájöttek arra is, amit én már 1980 óta feszegettem [3], hogy ezzel a módszerrel nem lehet felfedezni bizonyos relációk **közötti** összefüggéseket. Ezért vezették be az **univerzális reláció** fogalmát, amely az összes tulajdonságot tartalmazza és azon vizsgálták a tételek **doméjn-függéseit**. Az eljárás közismert neve: **adatmodell-szintézis** vagy **-integráció**.

A kombinatorikával tisztában lévők számára világos, hogy ez ugyancsak nehézkes eljárás, mert

- N darab doméjn esetén $N*(N-1)/2$ függést kellene vizsgálni
- és akkor is csak az elemi függések kerülnének sorra.

Mindkét eljárás hibája, hogy nem az egyedek egymásra épülő belső és külső struktúráján alapul és ezért nem vezethetnek optimális eredményre.

Az adatmodell „csontváza”, makroszerkezete az egyed típusok összetett hálójá, amit az azonosítók függései határoznak meg. Ennek összeállításával és elemzésével meghatározható az optimális adatmodell.

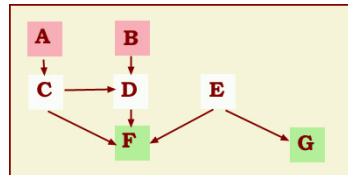
10.2 Innovatív adatintegrálás

Már a SYDES [4], majd az AMOR [5] rendszeremnél a fentieknél egy jóval praktikusabb módszert alkalmaztam. Ennek megértéséhez három alapvető fogalom bevezetése szükséges.

D28	Az adatmodell hálójának azt az egyedét, amelynek nincs fölérendeltje, az adatmodell végpontjának , amelynek nincs alárendeltje, az adatmodell kezdőpontjának nevezem.
-----	---

D29	Az adatmodell hálójának a kezdőpontjától a végpontjáig vezető utak a fonalak (thread).
-----	---

Az alábbi ábrában az adatmodell kezdőpontjai az F és G egyedek, végpontjai az A és B egyedek. A struktúrát több fonál alkothatja úgy, hogy azoknak vannak közös tényezői. Például az alábbi ábrán fonál az $F \rightarrow C \rightarrow A$, $F \rightarrow C \rightarrow D \rightarrow B$, $F \rightarrow D \rightarrow C \rightarrow A$, $F \rightarrow D \rightarrow B$ függéssorozat.



Az adatmodell fonalai

A fonál nem más, mint a modell egy lineáris szerkezeti egysége. Ha tehát a modellt a fonalaira bontjuk, akkor valójában „szétbogozzuk” az egymással összeszövődött struktúrákat.

Az AMOR fonalakat normalizál, nem egyedeket és nem univerzális relációt.

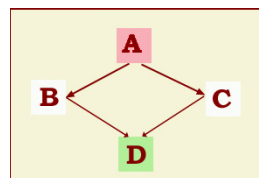
Evidens, hogy problémát okozó függések csak az egymással hierarchikus viszonyban álló egyedek között, tehát a fonalakon léphetnek fel. Példánkban a G egyedre épülő részstruktúrák nem okozhatnak galibát az F-re épülőkkel együtt, ezért azok együttes vizsgálatára nincs szükség.

Viszont ugyanakkor elegánsan megoldódik két probléma:

- A vizsgálat elején ki lehet zárni a ciklusokat, vagyis az önmagukba visszatérő fonalakat.
- Nincs szükség a kivetített tulajdonságok vándoroltatására.

Ha például az F egyedben és az A egyedben szerepelne a közös L leíró tulajdonság, a hagyományos eljárással azt előbb a C egyedbe kellene kivetíteni és csak onnan az A-ba. A fonál szerinti normalizálásnál viszont tudjuk, hogy mindig a fonál legfelső pontján van a redundáns tétel megfelelő helye:

$F, L \rightarrow C \rightarrow A, L$ esetén az $F \rightarrow C \rightarrow A, L$ megoldás a helyes.



„Gyémánt” szerkezet

Az adatmodellben gyakran előfordul az ún. **gyémánt-szerkezet**, aminél két egyednek (B és C) közös alá- (D) és fölérendeltje (A) van. Ilyenkor a két mellérendeltnek több közös redundáns tulajdonsága is lehet, amik azonban nem okoznak kedvezőtlen függéseket. Viszont gyakran homonimák.

Összefoglalóan: A normalizálás elején összeállítjuk a függési fonalakat. Ha valamelyik önmagába visszatér, akkor ciklust fedeztünk tel. Egyébként vesszük egyenként azokat és megállapítjuk, hogy a fonal mentén vannak-e redundánsnak mutató tulajdonságok, amelyek nem homonimák. Ezeket kiküszöböljük a fonalnak a kezdőponthoz közelebb eső („alsó”) egyedeiből.

Ha az adatmodell fonalaiban nincs (elfogadhatatlan) redundancia, akkor a makroszerkezete optimálisnak tekinthető.

11. INTELLIGENS ADATMODELL

11.1 Az egyedek külső szerkezete

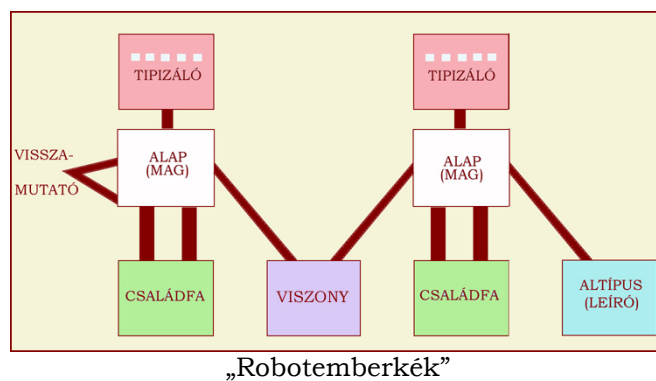
Itt megismétlem az adatmodell meghatározását:

D30	Az adatmodell egyed típusoknak, azok tulajdonságtípusainak (egyedek belső szerkezete) és kapcsolattípusainak (egyedek külső szerkezete), valamint az ezen tényezőkre és azok viszonyaira vonatkozó korlátoknak a szervezett együttese.
-----	---

Az adatintegrálás messze nem csak normalizálásból áll. A valósághűség, érthetőség és egyértelműség kritériumainak a teljesítése sokféle összefüggés ellenőrzését követeli meg.

A relációs modellezésben a relációkat darab-darab minősítés nélküli módon határozzák meg, elemzik és kezelik. Ezzel szemben **az intelligens adatmodellezésben** az egyedeket minősítjük, kategóriák szerint határozzuk meg és az azoknak megfelelő korlátok szerint elemezzük és kezeljük őket.

Az intelligens adatmodell tényezőit az ún. robotemberkék szemléltetik, amelyek az alapvető egyed- és kapcsolatfajtákat mutatják be.



11.2 Az egyedtípusok fajtái

Az AMOR adatmodellezési rendszerben [5] az egyedtípusoknak a robotemberkék által mutatott fajtáit különböztetem meg.

1. **Alapegyed.** Például a Személy és a Lakhely. Nincs speciális korlátja.
2. **Viszonyegyed.** Két alapegyed kapcsolata. Például Személy lakhelye. A viszonyegyedben szerepelnie kell két alapegyed azonosítójának vagy az összetett azonosító részeként, vagy egyszerű leíróként.

$A+B \rightarrow A$ és $A+B \rightarrow B$ vagy $C \rightarrow A, B$
 Személy azonosító+Lakhelykód $\rightarrow$ Személy azonosító
 Személy azonosító+Lakhelykód $\rightarrow$ Lakhelykód
 vagy
 Személy lakhelye azonosító $\rightarrow$ Személy azonosító, Lakhelykód

3. **Családfaegyed.** Egy alapegyed előfordulásainak a kapcsolata. Pl. Személyi családfa. A családfaegyed kulcsa a fölérendelt kulcsának a szerepneveiből összetett.

$A_1 \Rightarrow A$ és $A_2 \Rightarrow A$, ahol $C=(A_1 + A_2)$ a családfa azonosító
 ahol
 Szülő-azonosító $\Rightarrow$ Személy-azonosító
 Gyerek-azonosító $\Rightarrow$ Személy-azonosító

A családfaegyedben olyan minősítő ismérveket használunk, amelyek az alapegyed előfordulásait feltételeken osztályozzák. Korlátok: A_1 nem lehet azonos A_2 -vel és a kettő nem állhat ciklusos viszonyban.

Gyakran előfordul, hogy az időfaktor az összetett azonosító része:

$$C=(A_1+A_2+T)$$

4. **Altípusegyed.** Egy alapegyed kiegészítő tulajdonságait modellezi. Az altípusegyed azonosítója az alapegyed azonosítójának a szerepneve, vagy az altípusegyed leíróként tartalmazza az alapegyed azonosítóját.

$A' \Rightarrow A$
 Orvos-személy-azonosító $\Rightarrow$ Személy-azonosító

5. **Tipizálóegyed.** Osztályozásra szolgál a régi kódtáblák mintájára, de nálam nem kód-megnevezés páros, hanem csak megnevezés. Voltaképpen kötött értékhalmoz, ami által a tartalom egyértelműbb és a kezelés egyszerűbb, mint „kódegyedek” esetén. Az ábrán mutatottal szemben tipizáló egyede nem csak alapegyednek lehet.

6. **Visszamatató kapcsolat.** Ritkán használt, de nem kizárt tényező.

$$A \Rightarrow A'$$

$$\text{Személy-azonosító} \Rightarrow \text{Főnök-azonosító}$$

Az idők során egy személynek több főnöke is lehet, ezért az ilyen egyed mindig csak az aktuális vagy kiemelt összefüggést mutatja. A teljes személy-főnök kapcsolat családfa jellegű, így:

$$\text{Személy-azonosító} \rightarrow \text{Személynév}$$

$$\text{Főnök-azonosító} + \text{Beosztott-azonosító, Dátumtól, Dátumig}$$

ahol

$$\text{Főnök-azonosító} \Rightarrow \text{Személy-azonosító}$$

$$\text{Beosztott-azonosító} \Rightarrow \text{Személy-azonosító}$$

Az AMOR adatmodelljében **minden tényező minősített és a minősítés szerinti korlátok ellenőrzéseinek alávetett**. Még az alapegyed is az, de erre nem térek ki.

11.3 Az azonosító mátrix

A hagyományos normalizálásban a tulajdonságoknak csak egyféle, az aktuális egyedben betöltött szerepével foglalkoznak.

Az ezzel kapcsolatos kitételek eléggé zavarosak. Például gyakran használják az „ami a relációban kulcs/nem-kulcs” kitélt minősítés (elsődleges, alternáló, kapcsoló) nélkül, vagy a kulcsjelölt kifejezést, ami félreérthető.

Ezzel szemben az AMOR rendszerben kétféle szerepet különböztetnek meg:

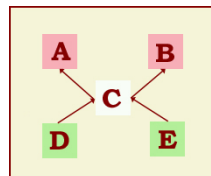
D31	A tulajdonságnak az egyeden belüli feladatát relatív szerepnek, a legfontosabb relatív szerepét abszolút szerepnek nevezzük.
-----	--

A relatív szerep az egyed-tulajdonság-viszony, az abszolút szerep viszont a doméjn strukturális jellemzője. Ha egy tulajdonság valamelyik egyedben azonosító, akkor abszolút szerepe is az. Ellenkező esetben ha van egyed, amiben azonosítórész, akkor abszolút szerepe is az, egyébként leíró.

A leíró abszolút szerepű tulajdonságok kevésbé fontosak a normalizálás szempontjából, mint az azonosítók. Például amíg az azonosítók viszonya ciklikus vagy más módon súlyosan hibás, addig nincs mit kezdeni a leírókkal.

A normalizálás alapjául szolgáló függési hálót is az azonosítók függései szerint építjük fel. Ezek alapján állapítható meg az is, hogy a hálóban nincs-e kapcsolathiányt mutató szakadás.

A továbbiak szemléltetésére vegyünk egy egyszerű képletes adatmodellt:



Mintamodell

Tegyük fel, hogy a D és E egyedek mindegyike tartalmazza az X leíró tulajdonságot. Ha a két egyed egyike se tartalmazza a másik azonosítóját, akkor ez csak egy megengedhető logikai redundancia. Ha viszont az egyik **elvileg** tartalmazná a másik azonosítóját, de ténylegesen nem teszi, akkor az egyedháló kapcsolathiányos, ami miatt nem fedezhető fel az X tranzitivitása.

Az AMOR módszerben ennek és a hasonló problémáknak az elkerülésére vezettem be a **kulcsmátrix** tényezőjét.

	D	E	C	A	B
D	-	?	*	*	*
E		-	*	*	*
C			-	*	*
A				-	?
B					-

Kulcsmátrix

A kulcsmátrix egy függési mátrix, de csakis az azonosító tulajdonságok függéseit tartalmazza. Így mérete (több) nagyságrenddel kisebb, mint a teljes függési mátrixé lenne, tehát kezelhető. Olyan trianguláris táblázat, amelybe nemcsak a közvetlen, hanem a közvetett kulcsfüggéseket is bejegyezzük. A 2. táblázat az 5. ábrára vonatkozó mátrixot mutatja.

Mint látjuk, az ábrának megfelelően csak két helyzet kérdéses. Van-e az A és B, illetve a D és E egyedek között kapcsolat? Ha netán van, akkor az elemzéssel elejét vettük a problémának.

A kulcsmátrix elemzésével párosuló fonálnormalizálás nemcsak sokkal hatékonyabb eljárás, mint a dekompozíció, az univerzális reláció vagy a szintézis normalizálási algoritmus, hanem garantáltan az optimumot jobban megközelítő végeredményre is vezet.

12. AZ ADATMODELLEZÉS LÉPÉSEI

Az AMOR először formailag elemzi az adatmodellezés alapját (inputját) az előírt szabványok szerint. Az alaptényezőkből (egyed, doméjn és tulajdonság) összeállítja a szintaktikailag helyes modellt. A kiinduló modell-adatbázist áttölti az elemző-adatbázisba az eredeti példányt érintetlenül hagyva.

Ne felejtjük el, hogy a **szinonimák** kiszűrése érdekében csak annyit tehetünk, hogy elkerüljük a ragozott alakokat és betartjuk a szabványokat (kis- és nagybetű, kötőjel vagy üres stb.). A **homonimák** redundanciaként fognak jelentkezni. Ezek száma csökkenthető, ha minősítést alkalmazunk a dátumok, nevek, sorszámok, mennyiségek stb. jellegű tulajdonságokra.

A struktúraelemzés főprogramja előkészíti az elemző-adatbázist, majd egy összetett ciklusban sorra meghívja az elemző-programokat. Szükség esetén (adott ponttól) iterál. Az elemzés lépései a következők

1. **Alapelemzés.** Az elemi strukturális hibákat szűri ki. Ilyenek csak akkor fordulhatnak elő, ha az alapokat (részben) importálják. Ha a modell teljesen az AMOR-ral készül, akkor nem léphetnek fel.

Azonosító-elemzés

Van-e azonosító nélküli egyed? Az ugyanis nem normalizálható.

Van-e több azonosítójú egyed? Egy egyednek csak egy azonosítója lehet, az alternáló azonosítókat akként kell megadni és azok csak egy egyedben szerepelhetnek leíróként.

Van-e többszörösen azonosító tulajdonság? Egy tulajdonság csak egy egyednek lehet az azonosítója.

Struktúra azonosság

Vannak-e (közel) azonos felépítésű egyedek? Azok ui. teljesen normalizálatlanok. Kimutatja az egyedek átfedésének a mértékét.

Elsődleges csoportelemzés

Van-e azonos egyedben együtt egy csoport és expliciten a tagja is?

Elsődleges szerepnévelemzés

Az egyedben csak akkor szerepelhet együtt egy szerepnév és annak elsődlegese, ha az elsődleges az azonosító. Minden más esetben az egyed normalizálatlan és normálformája nem is vizsgálható.

2. **Csoportelemzés**

Van-e egytagú csoport? Van-e rossz csoport/szerepnév viszony?

Vannak-e azonos felépítésű csoportok? Van-e ciklus a csoportokban?

3. **Szerepnévelemzés**

Van-e több főlérendeltű szerepnév? Van-e rossz kulcs(rész) szerepnév?

Van-e csoport szerepneveként csoport? Van-e ciklus a szerepnevekben?

4. **Hálóelemzés.** Elkezd felépíteni az egyedek **szuperhálóját**. Ha ciklust talál, azonnal leáll. Ha nem, akkor megvizsgálja, hogy a háló teljesen konnektív-e. Ha a háló nagymértékben inkonnektív, akkor az elemzés nem folytatható.
5. **Azonosító normalizálás.** Ha súlyos hiba lép fel, akkor abortál. Egyébként meghívja a 7. programrészt.
6. **Leíró normalizálás.** Hiba esetén abortálható. Egyébként meghívja a 7. programrészt.
7. **Összetett függések elemzése.** Az 5-7. elemzések a következőkre irányulnak:
 - Belső kulcstörő függés
 - Részleges függés
 - Metszet függés
 - Csoport függés
 - Külső kulcstörő függés
 - Normál tranzitivitás
 - Pszeudo-tranzitivitás
8. **Teljesség- és minimalitáselemzés**
 - Vannak-e maradék redundáns (nem rossz függésű) tulajdonságok?
 - Maradtak-e egyed nélküli tulajdonságok? Maradtak-e tulajdonság nélküli egyedek?
9. **Kapcsolatelemzés**
 - Jól megadott kapcsolatok, amiket az összefüggések igazolnak.
 - Rosszul megadott kapcsolatok, amiket a viszonyok nem igazolnak.
 - Meg nem adott kapcsolatok, amiknek létezniük kellene.

Mindez a valóságban a leírtaknál sokkal bonyolultabb, mélyebb intelligenciával működő programrendszer által valósul meg. Ha valaki nem az itt leírt logika szerint normalizál, akkor vagy elvesz a modellből szükséges eleme(ke)t, vagy hozzáad szükségtelen(e)ket.

III. RÉSZ

AZ ADATMODELLEZÉS

RENDSZERE

13. AZ INTEGRÁLÁS TÉNYEZŐI

A mai hazai adatbázis-menedzselési módszerek 50 évvel vannak elmaradva a világszínvonaltól. Svédországban már 1975-ben magas szinten gyakorolták az adatadminisztrátori – nem adatbázis-adminisztrátori! – funkciót. Ugyancsak Svédországban az információs rendszerek tervei (állami) szabványoknak kellett, hogy megfeleljenek. A New York-i Chase Manhattan Bankban 1975-ben jómagam választottam ki az adatnyilvántartás megfelelő eszközét (Data Catalogue).

13.1 Az adatmodell mint rendszer

A normálformák bővöleteivel telített szakirodalom az adatintegrálás igen összetett feladatát relációs táblázatok ad hoc átalakítására egyszerűsíti le. Sokakban fel se ötlük, hogy

az adatmodell is egy rendszer,

aminek meg kellene teremtenie az alapjait és amit fel kellene építeni.



Az adatmodell rendszere

Az adatintegrálásnak nemcsak az adatok, hanem mindezen tényezők a tárgyai kellene, hogy legyenek. Világos, hogy integrált adatmodell nem állítható elő megfelelő szabványok, eljárások és eszközök nélkül.

Művemnek nem feladata, hogy ezt a rendszert részletesen ismertesse. Ezért csak ízelítőt fog adni abból, hogy mi mindenre lenne szükség ahhoz, hogy az adatintegrálást valóban professzionális módon végezzék.

13.2 Szabványok és szabványos eljárások

A részfeladatokkal megbízott fejlesztők elfeledkeznek arról az axiomatikus tényről, miszerint

AZ ADATBÁZIS – EGY!

Természetesen minden szervezetben sok adott **logikai** tartalommal és **fizikai** elrendezésben megvalósított adatbázis létezhet. A kitétel arra hívja fel a figyelmet, hogy **fogalmi** szinten a tartalmaknak egyezniük kell, különben az integrálás meseszerű ábránd marad.

Ha a gépkocsi típus és modell ugyanazt jelenti egy vállalat egyes részlegeiben, míg eltérő a jelentésük másokban, akkor az adatbázis nem lesz integrált. Ez történt a BMW AG.-ben, ahol a gyártás és a MEO adatbázisai ellentmondtak egymásnak.

A tényezők neveire, fajtáira és ezek értékeinek az osztályozásaira szabványokat **kell** bevezetni. Ha egyik részlegünkben X néven vesz részt a normalizálásban az a tényező, amely a másikban az Y nevet kapta, az adatmodell sohase lesz integrált, mert a két része ellent fog mondani egymásnak.

Ha a modellelemzés eltérő módszereit – más lépések és-vagy más sorrend – alkalmazzuk a teljes modell részhalmazaira, akkor erőfeszítésünk kárba vész. Az adatmodellezés szabványos eljárások szerint **kell**, hogy történjen.

Legalább 20-25 igen nagy adatmodell megalkotásában vettem részt. Közülük csak az a kettő vallott kudarcot, amelyeknél nem alkalmaztak szabványokat.

13.3 Felhasználók: az adatadminisztrátor

A szabványos eljárások egyik aspektusa, hogy ki (hol, mikor) és milyen eszközön végezheti el az adatintegrálás bizonyos eljárásait.

Az adat(bázis)-integrálás nem egyszerű feladat. A szabványok és eljárások nem könnyen követhetők. Ezért szükség van egy olyan informatikai funkciókörre, amely tájékoztat, irányít és ellenőriz a metamodell körüli dolgokban.

A funkció neve adatadminisztrátor. Nem adatbázis-adminisztrátor és nem rendszergazda. Ez utóbbi kettő egy-egy **lokális** adatbázisért felel tartalmi (logikai) és tárolási (fizikai) szinten. Viszont mint mondtam, fogalmi szinten a **globális** adatbázis egy, amiről valakinek gondoskodnia kell.

A főkönyvelő törődik a pénzügyi ismeretek pontos és szabványos vezetésével. Az adatadminisztrátor ehhez hasonló feladatot lát el **szervezeti szinten** úgy, hogy az ő főkönyve a metaadatbázis.

13.4 Eszközök: a metamodellel kezelése

A szabványok és szabványos eljárások (standards & procedures) csak úgy érvényesíthetők (néha még úgy se), ha valamilyen eszköz kikényszeríti őket. Az adatintegrálás aligha lehet sikeres, ha az adatmodellt nem a 2. fejezetben mutatott metamodellel szerint építjük fel. Azonban önmagában véve még az se elegendő. Tényezőire nézve ki kell dolgoznunk a magunk részletes előírásait.

A metamodellel maga is adatbázis. Azonban az integrálásnak csak segédeszköze, aminek a helyes használatán legalább annyi múlik, mint a metamodellel korrekt felépítésén. Ezért rendelkezünk kell egy olyan metakezelő-rendszerrel, amelynek az integritást szigorúan betartató programjai célszerű sorrendet követnek. Erre mutat példát a 12. fejezetet.

14. AZ INTEGRÁLÁS MAKROMODELLJE

Nem egyszer tapasztaltam, hogy a szervezeti egységek az integrálást úgy fogták fel, hogy a **saját berkeiken belül** kell optimalizálniuk az adatbázis szerkezetét a többiekre való tekintet nélkül. Ebből aztán szép kalamajkák fakadtak a **szervezet egészében**. Végeredményben az előző fejezetben leírt tényezőkből egyik se valósult meg, még részlegesen sem.

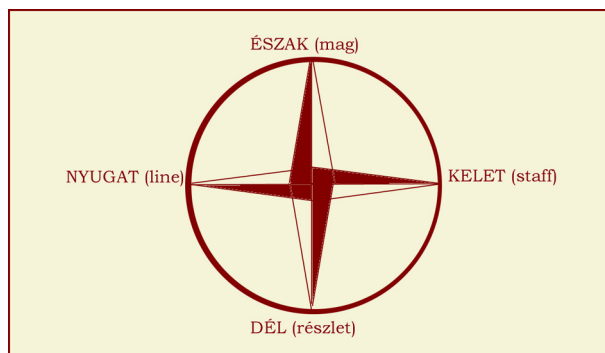
Mint azt több helyen is kifejtettem, a következő axiómát kellene követni:

AZ ADATMODELL – EGY

Egy szervezeten belül létezhetnek szervezeti egységenként, azok sajátos feladataiból adódó speciális adathalmazok, amelyek másokéival csak lazán kapcsolódnak. Viszont vannak olyanok is, amelyek bár ha nem is a **konkrét tartalmukban**, de az **általános szerkezetükben** erősen egymásra ütnek.

Például ugyancsak valószínű, hogy az X és az Y részlegek titkárságának a munkája és annak megfelelően adatigénye sok rokon vonást mutat. A költői kérdés az, hogy vajon célszerű-e azok adatbázisait egymástól teljesen eltérő módon megtervezni, avagy intelligensebb megoldás, ha prototípusokra törekvően generikus modellekben gondolkozunk?

A négy alapvető szervezeti szféra a következő:



Az adatmodell makromodellje

Magyarázat:

- **Funkciók (nyugat, line).** Minden szervezet egy vagy több alaptevékenység ellátását végzi. Ezeket a funkciókat **vonalak**nak hívjuk, amelyek adatigényei akár jelentősen is eltérhetnek egymástól.

Például egy elektronikai cég foglalkozhat egyféle készülékek gyártásával és másfélék szervizelésével. A gyártás és a szervizelés csak részben azonos adatigényű, viszont az se valószínű, hogy mindenben eltérő.

- **Erőforrások (kelet, staff).** A funkciók gyakorlásához erőforrásokra – anyag, eszköz, energia, munkaerő – van szükség. Ezeket az ellátások biztosítják, amelyek meglehetősen hasonló logikával és adatigénnyel működnek bár nem a tartalmuk, hanem a természetük szerint.

A logisztikainak nevezett rendszerek azért vehetők egy kalap alá, mert erőforrásokkal dolgoznak, amelyeknek vannak igényei, készletei, szállítói stb.

- **Adminisztráció (dél, részlet).** A technikai kiszolgálás mindenütt nagyjából ugyanazokból a feladatokból áll: kapcsolattartás, dokumentumkészítés, -elosztás, -tárolás stb.
- **Vezetés (észak, mag).** A többivel szemben ez két eltérő természetű részből áll. Bár a felsővezetők gondolkodásában sok közös vonás van, ezen a téren nyilván akadnak nem általánosítható helyi sajátosságok is. A másik oldalon viszont ebbe a körbe tartoznak az általánosítható osztályozások.

Például nem lenne tanácsos, ha minden részleg saját devizanem, ország, fizetési mód stb. kötött értékcszleteket (régi fogalommal: kódokat) alkalmazna.

Mindenben nincs semmi különös. Mivel az adatintegrálásban az adott szervezet egészét kell szem előtt tartani, a szervezet teljes adaterőforrását egyetlen rendszerként kell felfogni. Olyannak, amiben vannak egymástól elkülönült, de egymással mégis kapcsolatban álló részrendszerek.

A. FÜGGELÉK: ANOMÁLIÁK

A.1 Az anomáliák háttere

Az adatkezelésben fellépő nehézségeket-bajokat hívjuk anomáliáknak. Az adatkezelési eljárásoknak két nagy csoportja van.

- Az egyik műveletei a lekérdezés, másolás és rendezés nem változtatják meg az adatbázis tartalmát.
- A másik tevékenységei az adattartalom megváltoztatását célozzák és ezeket karbantartási (update) műveleteknek hívjuk. Ide tartozik az adatbevitel, adatmódosítás és adattörlés.

Sokan abban a tévhitben élnek, hogy a visszasságok csak a második csoport műveleteit érintik. Azonban az anomáliáknak van egy olyan válfaja is, ami a lekérdezéssel és az egyértelműség hiányával kapcsolatos.

*Egy közlekedési hatóságnál a vezérigazgató jelentést kért adott jelenségekben szereplő gépjárművekről a típusaik szerint. Majd dührohamot kapott a riport láttán. Kiderült, hogy a főkönyvelő a típust **modellnek**, a főmérnök **márkának** értelmezte. A nagyfőnök az előbbiről érdeklődött és az utóbbi szerinti kimutatást kapott.*

A homonímák és szinonímák téves ismeretközlésre vezethetnek.

A karbantartási anomáliák a redundanciával kapcsolatosak, amiknek a válfajait a 3.4 pontban mutattam be. Ott utaltam rá, hogy vannak jó és rossz redundanciák. Arra viszont csak közvetve tértem ki, hogy sokszor a jó logikai **redundancia hiánya** okoz problémát.

Ennek igazolására vegyük elő a 4.2 pontban szereplő 15. példát.

VEVŐ-RENDELÉSTÉTEL
SZÁLLÍTÓ-RENDELÉSTÉTEL

Rendelésszám+Cikkszám, ...
Szállítástétel-azonosító, **Rendelésszám, Cikkszám**, ...

Jóllehet az 5NF normalizálási szabályai szerint ez egy hibátlan modell-részlet, a két egyed mégsem kapcsolható egymáshoz anélkül, hogy ne lépne fel karbantartási anomália. A második egyedbe olyan előfordulás is bevihető, amely nem mutat az első egyed érvényes előfordulására! Ennek az az oka, hogy a második egyedben nem a Rendelésszám+Cikkszám együttese fordul elő csoportként, hanem csak a Rendelésszám, Cikkszám részei kapcsolásra csak külön-külön alkalmas tartalommal.

A.2 Karbantartási problémák

A problémákat az alábbi, egyszerű tranzitivitásokat mutató, a harmadik normálformát sértő példán át mutatom be. A megállapítások érvényesek minden olyan esetre, amelyben a logikai redundancia nemkívánatos.

GÉPKOCSI

Rendszám	Tulajdonos azonosító	Tulajdonosnév	Telephely	Gépkocsi típus	Gépkocsi típusadatok
ABC 134	T1	Kovács Rózsa	Budapest	Lada	XXX
BCD 265	T2	AB kft	Szeged	BMW	YYY
DEF 896	T3	Kovács Tibor	Pécs	Polski	ZZZ
FGH 333	T4	XY vállalat	Budapest	Lada	XXX
KLM 344	T1	Kovács Rózsa	Budapest	Polski	ZZZ

- **Beviteli anomália.** Egy rendelkezésre álló ismeretet nem lehet az adatbázisa vinni, amíg a kapcsolt ismereteket meg nem adják. Például nem lehet megadni a Mazda gépkocsi-típus adatait, amíg nem tárolnak egy konkrét Mazda gépkocsit.

Az érintett adatokat csak részben mutatja a példa. A kocsiknak nemcsak típusa és férőhelye, hanem márkanéve, lóereje, köbcentíje, súlya stb. is van.

- **Törlési anomália.** Ez mindig a bevitelivel párosul, mivel mindkettő közös gyökere az, hogy hiányzik a fölérendelt egyedtípus, esetünkben a GÉPKOCSI TÍPUS. Ha kitöröljük a BCD 265 rendszámú gépkocsi adatait és az az utolsó BMW, akkor elvesznek a BMW típus adatai.
- **Módosítási anomália.** Ha Kovács Rózsa férjhez megy és megváltoztatja a nevét, azt több előforduláson kell végrehajtani.

További problémát jelent, ha a GÉPKOCSI egyed mellett létezik külön GÉPKOCSI TÍPUS és TULAJDONOS egyed is. Ugyanis ekkor többszörösen kellene végrehajtani a karbantartásokat a megfelelő egyedeken, különben az adatbázis rövid idő alatt inkonzisztenssé válik.

B. FÜGGELÉK: ÓRÜLT NORMÁLFORMÁK

Avagy hogyan oldjunk meg álproblémákat valódiakat kreálva a helyükbe. A jelzőt nem én találtam ki: maga a kitalálójuk ragasztotta rájuk.

A szakirodalomban nem egyszer előfordul, hogy valaki előáll egy teljesen **mesterséges**, a valós életben soha elő nem forduló problémával, majd azt „bravúrosan” megoldva nevet ad az általa felfedezett új konstrukciónak.

B.1 Negyedik normálforma (4NF)

Ezen „nagy találmányok” sorába tartozik a **többértékű függés** (MVD) és az azon alapuló **negyedik normálforma**. Ezeket nem részletezem; kéretik utánuk nézni. Inkább azonnal a dolgok közepébe vágok.

Előbb az esetet szemléltetem egy Wikipediából vett példával [8], majd értékelem a megoldást. A példa tartalma egyébként közömbös, lehetne A,B,C attribútum is, csak a szerkezetére kell ügyelni.

PIZZA SZÁLLÍTÁSI ENGEDÉLYEK		
Étterem	Pizza fajta	Szállítási terület
A1 Pizza	Vastag tésztájú	Springfield
A1 Pizza	Vastag tésztájú	Shelbyville
A1 Pizza	Vastag tésztájú	Főváros
A1 Pizza	Töltött kéreg	Springfield
A1 Pizza	Töltött kéreg	Shelbyville
A1 Pizza	Töltött kéreg	Főváros
Elite Pizza	Vékony tésztájú	Főváros
Elite Pizza	Töltött kéreg	Főváros
Vincenzo pizzája	Vastag tésztájú	Springfield
Vincenzo pizzája	Vastag tésztájú	Shelbyville
Vincenzo pizzája	Vékony tésztájú	Springfield
Vincenzo pizzája	Vékony tésztájú	Shelbyville

ÉTTEREM PIZZAFAJTÁI		ÉTTEREM SZÁLLÍTÁSI TERÜLETEI	
Étterem	Pizza fajta	Étterem	Szállítási terület
A1 Pizza	Vastag tésztájú	A1 Pizza	Springfield
A1 Pizza	Töltött kéreg	A1 Pizza	Shelbyville
Elite Pizza	Vékony tésztájú	A1 Pizza	Főváros
Elite Pizza	Töltött kéreg	Elite Pizza	Főváros
Vincenzo pizzája	Vastag tésztájú	Vincenzo pizzája	Springfield
Vincenzo pizzája	Vékony tésztájú	Vincenzo pizzája	Shelbyville

Az eredetileg háromszlopos csupakulcs relációt itt nem részletezett karbantartási anomália miatt megbontják és helyébe két kétszlopos szintén csupakulcs relációt kreálnak, ami állítólag kiküszöböli az anomáliát.

Mármost szögezzük le az integrált adatbázisstervezés két általánosan figyelembe veendő szabályát.

1. Sohasem **előfordulásokon**, mindig **típusokon** normalizálunk. Az előfordulások ui. jönnek-mennek és csak a helyes típusszabályok tudják meggátolni, hogy ezt ne az integritás sérelmére tegyék.
2. Az adatmodellben nem lehetnek M:N fokú egyedek (relációk) közötti kapcsolatok. (N.B.: A relációs modellben az ún. **join** művelettel nagy galádságokat lehet elkövetni).

Az étterem új pizzafajtajával állhat elő, amit nem szállít mindenhová, illetve új szállítási területet célozhat meg, de nem minden pizzafajtajával.

Ezért messze nem igaz, hogy az eredmény relációkból egyértelműen előállítható a kiinduló reláció!

Az adatbázisstervezés egyébként se úgy indul, hogy megállapítjuk az adott relációról, hogy az netán csupakulcs jellegű. Az adatbázisstervezés úgy indul, hogy vizsgáljuk, hogy **mi lehetne** a reláció kulcsa. Látjuk, hogy az Étterem nem lehet az, mert attól **független a további két attribútum**, amelyek ún. ismétlődő adatok. Mit teszünk ilyenkor? Az ismétlődéseket külön-külön relációkba vágjuk le a feltételezett kulccsal. És mi lesz az eredmény?

ÉTTEREM PIZZAJAJTÁI		ÉTTEREM SZÁLLÍTÁSI TERÜLETEI	
Étterem	Pizza fajta	Étterem	Szállítási terület
A1 Pizza	Vastag tésztájú	A1 Pizza	Springfield
A1 Pizza	Töltött kéreg	A1 Pizza	Shelbyville
Elite Pizza	Vékony tésztájú	A1 Pizza	Főváros
Elite Pizza	Töltött kéreg	Elite Pizza	Főváros
Vincenzo pizzája	Vastag tésztájú	Vincenzo pizzája	Springfield
Vincenzo pizzája	Vékony tésztájú	Vincenzo pizzája	Shelbyville

Az eredmény ugyanaz, mint az MVD lebontásnál azzal a **jelentős különbséggel**, hogy a két eredményezett reláció egymáshoz nem kapcsolható, csak külön-külön az itt nem mutatott ÉTTEREM relációhoz annak kulcsa szerint.

Összefoglalás: **A 4NF szerinti lebontás nem jelent újat az 1NF normalizálásához képest. Egyébként pedig a többértékű függés nem más, mint fordított irányú összetett funkcionális függés.**

Pizza fajta+Szállítási terület → Étterem

B.2 Ötödik normálforma (5NF)

Itt ismét a Wikipedia „csodáját” fogom elővenni, amire jellemző, hogy az alapfogalmakat is tévesen használja [9]. A normálformát ui. „projekt-csatlakozás” normálformának (PJNF) nevezi, holott a „join” nem csatlakozást, hanem (össze)kapcsolást jelent, a P pedig a „projection” (kivetítés) helyett áll. A példában még az adatneveket se tudták konzekvensen használni!

Utazó eladók terméktípusai márka szerint		
Utazó eladó	Márka	Terméktípus
Jack Schneider	Acme	Porszívó
Jack Schneider	Acme	Kenyértartó
Mary Jones	Robusto	Metszőolló
Mary Jones	Robusto	Porszívó
Mary Jones	Robusto	Kenyértartó
Mary Jones	Robusto	Esernyőtartó
Louis Ferguson	Robusto	Porszívó
Louis Ferguson	Robusto	Távcső
Louis Ferguson	Acme	Porszívó
Louis Ferguson	Acme	Állólámpa
Louis Ferguson	Nimbus	Nyakkendőtartó

A megbontás utáni relációk:

Terméktípusok / Utazó eladók		Terméktípusok / Márka	
Utazó eladó	Terméktípus	Márka	Terméktípus
Jack Schneider	Porszívó	Acme	Porszívó
Jack Schneider	Kenyértartó	Acme	Kenyértartó
Mary Jones	Metszőolló	Acme	Állólámpa
Mary Jones	Porszívó	Robusto	Metszőolló
Mary Jones	Kenyértartó	Robusto	Porszívó
Mary Jones	Esernyőtartó	Robusto	Kenyértartó
Louis Ferguson	Porszívó	Robusto	Esernyőtartó
Louis Ferguson	Távcső	Robusto	Távcső
Louis Ferguson	Állólámpa	Nimbus	Nyakkendőtartó
Louis Ferguson	Nyakkendőtartó		

Márkák / Utazó eladók	
Utazó eladó	Márka
Jack Schneider	Acme
Mary Jones	Robusto
Louis Ferguson	Robusto
Louis Ferguson	Acme
Louis Ferguson	Nimbus

A képtelen feltételezések szerint ha egy eladó egy márkának egy termék-típusát forgalmazza, akkor a márkának minden terméktípusával is azt teszi. És ugyanezt tessék körkörösén a márkáról és a terméktípusról is feltételezni.

Állítólag az egész a redundancia megszüntetése érdekében történik és az eredményrelációk összekapcsolásával az eredeti reláció visszaállítható.

Mármost ezek teljesen hamis megállapítások, mert:

- A megbontással nem sok redundancia szűnt meg, viszont sokkal több adatot kell tárolni és kezelni.
- Az összekapcsolás időigényes művelet, hátha még háromszorosan körkörösén szükséges.
- **Elvileg** nem is állítja helyre az eredeti relációt, mert **gyakorlatilag** a három reláció bármelyikébe beilleszthető olyan új sor, amely nem felel meg a feltételeknek.
- Ugyanis ezek ellenőrzése még macerásabb, mint az eredeti meg nem bontott relációnál.

A **kapcsolásfüggés** (join-dependency, JD) nemcsak képtelenség, hanem egyben káros konstrukció is. **Káros**, mert megengedi az egymással M:N fokú viszonyban álló egyedek összekapcsolását, ami hamis ismeretekre vezethet. **Képtelenség**, mert hiszen a join művelettel egymáshoz kapcsolhatnánk pl. minden olyan relációt, amelyben van sorszám jellegű adat. Hol rögzítjük azt, hogy mely kapcsolások jogosak és melyek nem...?

Megjegyzendő, hogy a „join” helyes fordítása kapcsolás és nem csatolás. A csatolás hozzáadást, a kapcsolás összevegyítést jelent. A join voltaképpen nem más, mint a hagyományos fájlstruktúrából ismert „merge” funkció.

Összefoglalás: **Mivel a kapcsolásfüggés nem más, mint az ismétlődő csoport kegyes leplezése, az 5NF szerinti lebontás nem hoz érdemi újat az 1NF normalizálásához képest, viszont rossz megoldásra vezet, amiből nem tűnnek el a karbantartási anomáliák, sőt.**

A 4NF és 5 NF bővebb kifejtését és értékelését lásd a [6] forrásban (15.3 és 15. 4 pontok.)

Nem árt megjegyezni, hogy néha egy korlátnak a **struktúrával** való érvényesítése sokkal kacifántosabb és drágább, mint a **procedurális** validálás...

IDEVÁGÓ IRÁSAIM ÉS HIVATKOZÁSOK

- [1] Codd, E.F.: A Relational Model of Data for Large Shared Data Banks. CACM, Vol. 13, No. 6, 1970.
- [2] Codd, E.F.: Extending the Database Relational Model to Capture More Meaning. ACM TODS, Vol. 4, No. 4, 1979.
- [3] SZIAM: A Data Model Design Aid. In: Specification and Design of Software Systems. Conference on Operating Systems. Szerk.: E. Knuth & E.J. Neuhold. Springer Verlag. Visegrad 1982.
- [4] SYDES: A Method and Tool for Information Systems Analysis and Design. Comparative Review of Information Systems Design Methods. London 1988.
- [5] AMOR - AdatMOdellező Rendszer 2004.
- [6] Az adatbázistervezés alapjai és titkai. IDG lapkiadó. 1994.
- [7] Relációs adatbázis
https://hu.wikipedia.org/wiki/Rel%C3%A1ci%C3%B3s_adatb%C3%A1zis
- [8] Negyedik normálforma – borzalmas!
https://hu.wikipedia.org/wiki/Negyedik_norm%C3%A1lforma
- [9] Ötödik normálforma – borzalmas! Még a nevét sem ismerik...
https://hu.wikipedia.org/wiki/%C3%96t%C3%B6dik_norm%C3%A1lforma

TARTALOMJEGYZÉK

ELŐSZÓ.....	4
1. ADATBÁZIS ÉS ADATMODELL.....	8
1.1 Adatállomány.....	8
1.2 Reláció (relation).....	9
1.3 Kapcsolat (relationship).....	11
1.4 Adatmodell és adatbázis.....	13
1.5 Az adatbázis lényege.....	15
2. AZ ADATMODELLEZÉS LÉNYEGE.....	17
3. ALAPVETŐ NORMÁLFORMÁK.....	19
3.1 Az alapok alapjai.....	19
3.2 Funkcionális függés és függetlenség (ONF).....	20
3.3 Ismétlődés és 1NF.....	21
3.4 Részleges függés – 2NF.....	23
3.5 Transzitiv függés – 3NF.....	24
3.6 Külső kulcstörő függés – BCNF.....	25
4. AZ ADATMODELL KÖVETELMÉNYEI.....	28
4.1 Valóságűség.....	28
4.2 Érthetőség.....	28
4.3 Egyértelműség.....	29
4.4 Minimalitás.....	29
4.5 Teljesség.....	30
4.6 A kritériumok alkalmazása.....	30
5. A NORMALIZÁLÁS HIÁNYOSSÁGAI.....	31
5.1 A normálformák (1NF-5NF) nem teljeseek.....	31
5.2 A csoportszerkezet hiánya.....	32
5.3 Belső kulcstörő függés – HBNF.....	33
5.4 Metszet-függés.....	34
5.5 Csoport-függés.....	36
6. MELLŐZÖTT ESETEK.....	37
6.1 Pszeudo-tranzitivitás.....	37
6.2 Közvetett tranzitivitás.....	37
6.3 Egyszerű és összetett ciklus.....	38
7. SZEREPNEVEK.....	41
7.1 Tulajdonság, doméjn, attribútum.....	41
7.2 Feltételes függés.....	43
7.3 Explicit egyedaltípus – Specializáció.....	46
7.4 Implicit egyedaltípus – Visszamutatás.....	47

7.5	Családfa egyed és többszörös kapcsolat.....	48
8.	AZ EGYEDEK KAPCSOLATAI.....	51
8.1	A relációs rendszerek hibás kapcsolatfelfogása.....	51
8.2	Szemantikai szegénység.....	52
9.	A NORMALIZÁLÁS ÖSSZEFOGLALÁSA.....	53
10.	NORMALIZÁLÁSI KONCEPCIÓK.....	56
10.1	Két hagyományos eljárás.....	56
10.2	Innovatív adatintegrálás.....	56
11.	INTELLIGENS ADATMODELL.....	59
11.1	Az egyedek külső szerkezete.....	59
11.2	Az egyedtipusok fajtái.....	59
11.3	Az azonosító mátrix.....	61
12.	AZ ADATMODELLEZÉS LÉPÉSEI.....	63
13.	AZ INTEGRÁLÁS TÉNYEZŐI.....	68
13.1	Az adatmodell mint rendszer.....	68
13.2	Szabványok és szabványos eljárások.....	69
13.3	Felhasználók: az adatadminisztrátor.....	69
13.4	Eszközök: a metamodell kezelése.....	70
14.	AZ INTEGRÁLÁS MAKROMODELLJE.....	71
A.	FÜGGELÉK: ANOMÁLIÁK.....	73
A.1	Az anomáliák háttere.....	73
A.2	Karbantartási problémák.....	74
B.	FÜGGELÉK: ŐRÜLT NORMÁLFORMÁK.....	75
B.1	Negyedik normálforma (4NF).....	75
B.2	Ötödik normálforma (5NF).....	77
	IDEVÁGÓ IRÁSAIM ÉS HIVATKOZÁSOK.....	79
	TARTALOMJEGYZÉK.....	80