9. Relációs adatbázisok logikai tervezése

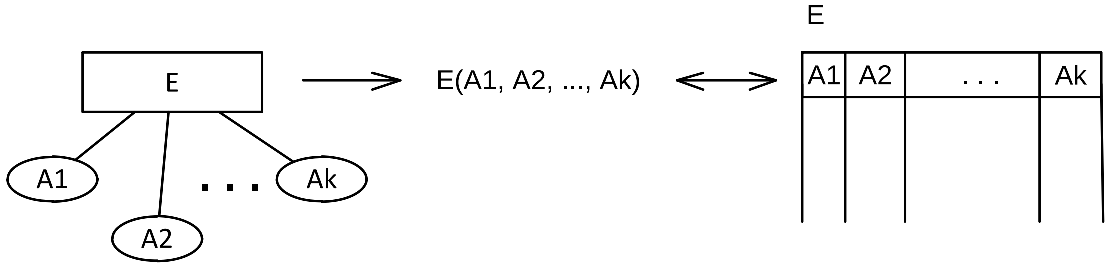
Bár a relációs adatmodell nem is az egyedüli, nem is a legjobb minden szempontból, mégis, a legelterjedtebb adatbázis-kezelő rendszerek még ma (2012.) is a relációs adatmodellen alapulnak. Emiatt kitüntetett jelentősége van a relációs adatbázisok tervezésének. Jelen jegyzetben ezért szántunk a problémának külön fejezetet.
A tervezés első lépésben az adatbázis logikai tervezésére terjed ki (csak ez után szokás a fizika tervezést végezni, tehát pl. a segédstruktúrákat kialakítani, tárolási paramétereket meghatározni), hiszen konkrét feladat megoldásához általában valamely megvásárolható relációs adatbázis-kezelő rendszert használnak fel. Ilyenkor nincsen sem mód sem szükség az adatbázis-kezelő működésének számos részletét megváltoztatni. A tervezés ekkor tehát arra korlátozódik, hogy meg kell határozni az adatbázis logikai szerkezetét (a relációs sémákat, definiálni kell az egyes adattípusokat), fizikai szerkezetének a paramétereit, a célszerű/szükséges segédstruktúrákat, majd meg kell írni magát az adatokat manipuláló programot/programokat, az *adatbázis alkalmazás*okat. Az alkalmazásokban az adatbázishoz való hozzáférés alapulhat pl. a szabványosított SQL nyelven, amelyet beágyazhatnak valamely magas szintű procedurális nyelvbe, de jellegzetesen valamilyen API-n keresztül hívják meg. Az alkalmazásfejlesztésnek számos, különböző szinten automatizált fejlettebb módszere is létezik.
Bárhogyan hozzuk is létre az alkalmazásainkat, az első lépés mindig az adatbázis logikai (koncepcionális) megtervezése. A tervezésnek két karakterisztikusan különböző módszerét ismertetjük a továbbiakban, amelyek azonban kombinálhatók is, és adott esetben jól kiegészítik egymást.

9.1. Tervezés E-R diagramból

A **Error! Reference source not found.**. szakaszban megismerkedtünk az E-R diagrammal, amely szemléletes ábrázolásmódja következtében hatékonyan támogatja a valóság modellezésének folyamatát. Az **Error! Reference source not found.**. fejezetben megismerkedtünk a relációs adatmodellel. Természetes az az igény, hogy E-R diagramokat relációs sémákká kíséreljünk meg átalakítani, így alapozva meg a valóságot jól modellező relációs sémák kialakítását. Az átalakítás teljes, ha megmondjuk, hogy az E-R diagram elemeit hogyan kell a relációs modellben megengedett adatstruktúrába (értsd: relációs sémá(k)ba) transzformálni.

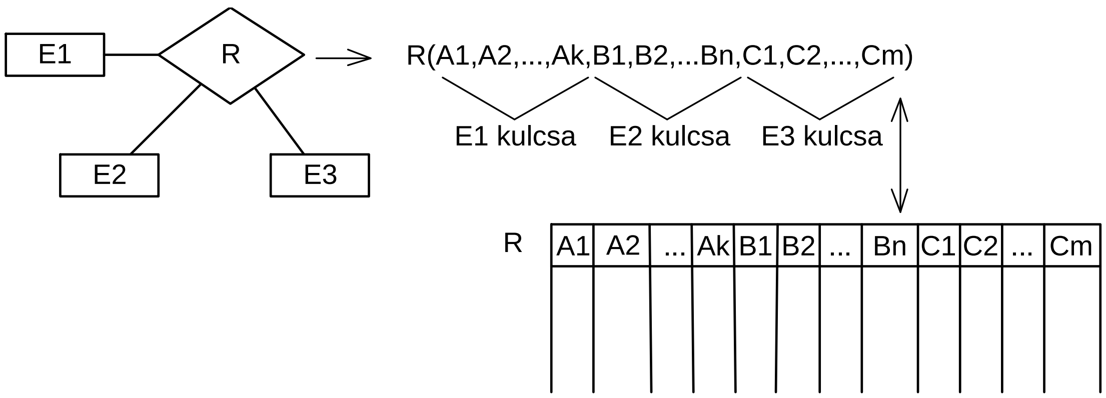
1. Az entitáshalmazokat olyan relációs sémával ábrázoljuk, amely tartalmazza az entitáshalmaz összes attribútumát. A reláció minden egyes n-ese az entitáshalmaznak pontosan egy példányát fogja azonosítani (ld. 9.1.*a)* ábra).

   Ha az entitáshalmazok között olyan is van, amelynek egyes attribútumait egy (általánosabb) entitáshalmaz egy „isa” kapcsolaton keresztül meghatározza, akkor a specializált entitáshalmazhoz rendelt relációs sémába az általánosabb entitáshalmaz attribútumait is fel kell venni.



9.1.*a)* ábra: Entitáshalmaz transzformációja relációs sémába

2. A kapcsolatokat olyan relációs sémákká alakítjuk, amelyek attribútumai között szerepel a kapcsolatban résztvevő összes entitáshalmaz kulcsa is (ld. 9.1.*b)* ábra). Feltételezzük, hogy két entitáshalmaz valamely kulcsattribútuma nem azonos nevű még akkor sem, ha az entitáshalmazok megegyeznek (mint pl. a HÁZASSÁG: EMBER, EMBER kapcsolatban). Névkonfliktus esetén az attribútumokat átnevezéssel kell megkülönböztetni. Az így kapott relációban minden egyes n-es olyan entitáspéldányokat rendel egymáshoz, amelyek a szóbanforgó kapcsolatban vannak egymással.



9.1.*b)* ábra: Kapcsolat transzformációja relációs sémába

A kapcsolatok relációs sémákba átalakítására a kapcsolat funkcionalitása és egyéb tulajdonságai (pl. specializáció kifejezése esetén) függvényében számos más lehetőség is van, amelyek adott esetben jobbak is lehetnek a bemutatott, egészen általános módszertől.
Ha pl. a **Error! Reference source not found.**.*a)* ábrán a DOLGOZIK: EMBER, OSZTÁLY kapcsolattípust akarjuk relációs sémákba transzformálni, akkor kihasználhatjuk a kapcsolat N:1 jellegét (függvényszerűségét), és az általános módszerből adódó három relációs séma helyett már kettővel is kifejezhetjük a kapcsolatot:

OSZTÁLY(OSZT_ID, OSZT_NÉV, ÉPÜLET, EMELET)
EMBER(SZEM_SZÁM, NÉV, ANYJA_NEVE, SZÜL_DÁTUM, OSZT_ID)[1]

Vegyük észre, hogy így ráadásul a relációs sémák struktúrájába sikerült beleépítenünk a kapcsolat függvényszerűségét biztosító kényszert is, amit elveszítettünk volna akkor, ha az általános módszer szerint transzformáltuk volna a kapcsolatot.

[1] Azokat az attribútumokat, amelyek egy adott relációs sémában nem, de egy másikban kulcsattribútumok, szokás szerint kétszeres aláhúzással jelöljük. Ennek az elnevezése: *idegen (vagy külső) kulcs* (ld. még a 9.2.3.1.3. szakaszt).

Megjegyzés: egy több-egy kapcsolat két relációs sémába történő leképzésekor a „több” oldalon álló egyedhalmaznak megfelelő relációban az idegen kulcsnak megfelelő attribútum(ok) NULL értéket vehetnek fel (pl. a fenti példában ha egy ember nem dolgozik egyik osztályon sem, akkor az OSZT_ID attribútuma NULL lesz). Azzal, hogy két relációs sémát hozunk létre, az eredeti információ kinyeréséhez eggyel kevesebb természetes illesztés művelet szükséges.
Vegyük észre továbbá azt is, hogy az E-R diagram relációs sémákba alakításával elveszítettük az egyedek és kapcsolatok formális megkülönböztethetőségét.

Példa: Transzformáljuk relációs sémákká a **Error! Reference source not found.**.*b)* ábra E-R diagramját!
Relációs sémák az entitáshalmazokból:

KIRENDELTSÉG(*KKÓD*, *HELY*)
ALKALMAZOTT(*AKÓD*, *NÉV*, *BEOSZTÁS*, *FIZETÉS*)

Relációs séma az egyetlen kapcsolatból:

DOLGOZIK(*KKÓD*, *AKÓD*, *DÁTUM*)

Strukturálisan ezek a relációs sémák pontosan ugyanúgy néznek ki, mint a

K(*KK*, *H*)
A(*A*, *N*, *B*, *F*)
D(*KK*, *A*, *D*)

sémák, azonban legfeljebb a kényszerek (kulcsok, idegen kulcsok) segíthetnek abban, hogy a sémák eredetére következtetni lehessen.

9.2. Tervezés sémadekompozícióval

Ha az E-R modellezés segítségével jutunk el a relációs sémáinkhoz, akkor a sémákban található attribútumokat és a relációk (valamint a teljes, az adatokon műveleteket végző adatbázis alkalmazás) további tulajdonságait az fogja meghatározni, hogy milyen „ügyesek” voltunk az E-R diagram megalkotása során. A relációk azonban tetszőleges számú attribútumot tartalmazhatnak. Így akár a rendszerben található valamennyi adatot beépíthetjük egyetlen sémába (ún. *univerzális sémába*), és ekkor egyetlen tábla írja le az egész rendszert. Ez a felhasználó számára roppant kényelmes, hiszen nem kell tudnia, hogy melyik adatot melyik reláció tartalmazza, így bizonyos lekérdezésekhez nem kell a relációk összekapcsolásával sem vesződnie, csupán ki kell válogatnia a számára érdekes adatokat.
Másrészről – pl. tárolási és adatmanipulációs hatékonyság szempontjából – ez a megközelítés nem előnyös, mert általában sok „felesleges” adatot is tartalmaz. Ezek kezelése lassítja a rendszer működését, a háttértárat és a memóriát feleslegesen foglalja, az adatbázist következetlenné/ellentmondásossá teheti.

A többször tárolt adatok egy adatbázisban, ill. relációban feleslegesek lehetnek. Nem minden többször tárolt adat felesleges. Ha egy reláció azt tartalmazza, hogy kinek milyen színű a szeme, akkor a „barna” (vagy annak a kódja) igen sokszor is előfordulhat ugyanabban a relációban, mégsem érezzük ezt feleslegesnek. Ugyanakkor feleslegesnek érezzük, ha *ugyanannak* a személynek a szeme színe fordul elő többször is a relációban. Ezt a hétköznapi nyelvben is redundanciának nevezzük, mert máshonnan is tudhatjuk, hogy az illető szeme színe barna. Általánosabban:

Definíció: ha egy relációban valamely attribútum értékét a relációban található más attribútum(ok) értékéből ki tudjuk következtetni valamely ismert következtetési szabály segítségével, akkor a **relációt redundáns**nak nevezzük.
A kikövetkeztethető adatokra gyakran azt mondjuk, hogy *származtatott* adatok.
Például:
Az alábbi reláció szállítók adatait tartalmazza, ki milyen árut (*TÉTEL*) mennyiért (*ÁR*) szállít és a szállító (*NÉV*) hol lakik (*CÍM*)[2].

NÉV	*CÍM*	*TÉTEL*	*ÁR*
Tóth István	Bp. Fa u. 5.	tégla	30
Tóth István	Bp. Fa u. 5.	vas	200
Kis János	Baja Ó u. 9.	tégla	40
Kis János	Baja Ó u. 9.	pala	20
Nagy Géza	Ózd Petőfi u. 11.	cement	350

Ha feltételezzük, hogy egy szállítónak csak egyetlen címe lehet, akkor ebben a relációban a címek többszörös tárolása teljesen felesleges, redundanciát okoz.
Vegyük észre, hogy az egyes relációk redundanciáját csökkenthetjük, ha az adatbázist alkalmas módon több, egyenként kevesebb attribútumot tartalmazó relációból alakítjuk ki.

9.2.1. Anomáliák

A redundáns relációknak megfelelő adattárolással kapcsolatban egy sor kellemetlen jelenség fordulhat elő. Ezeket hagyományosan *anomáliáknak* nevezik.

9.2.1.1. Módosítási anomália (update anomaly)

Tételezzük fel, hogy Tóth István címe megváltozik. A változást elvileg annyi helyen kell átvezetni, ahány helyen Tóth István címe szerepel. Ha csak egy helyen is ezt elmulasztjuk (pl. rendszerösszeomlás miatt), később különböző helyekről többféle címet is kiolvashatunk. Az ilyen szituáció tehát nemcsak *többletmunkát* jelent, hanem a logikai ellentmondások, ill. az információ elvesztésének lehetőségét is magában hordozza.

9.2.1.2. Beszúrási anomália (insertion anomaly)

Ennek lényege, hogy nem tudunk tetszőleges adatokat nyilvántartásba venni, ha nem ismert egy másik adat, amivel a tárolandó adat meghatározott kapcsolatban áll. Más szavakkal: nem tudunk a relációba olyan elemet felvenni, amelynek olyan mezője kitöltetlen (NULL), amely a reláció definíciója miatt nem lehet kitöltetlen. Ez a helyzet egy reláció kulcsmezőivel. Pl.: egy új szállítót nem tudunk felvenni, ha még nem szállított semmit.
Másrészt, ha a fenti példában Tóth István elkezd egy harmadik tételt is szállítani, de ennek az adatbázisba írásánál a lakcímére már nem a Bp. Fa u. 5.-öt adjuk meg, hanem pl. az új lakcímét, Bp. Rönk u. 17.-et, akkor elveszítjük Tóth István lakcímét, hiszen ez után már nem tudhatjuk, hogy mi a lakcíme valójában.

[2] Vegyük észre, hogy az áruszállítás ténye, ill. az, hogy egy adott szállítónak mi a lakcíme, valamint, hogy egy adott árucikknek mi az ára, mind azáltal van kifejezve, hogy a megfelelő adatértékeket egyazon sorba írva összerendeltük őket, ugyanakkor az összerendelés pontos szemantikájáról az adatstruktúra elemei – szigorúan véve – semmit nem mondanak.

9.2.1.3. Törlési anomália (deletetion anomaly)

Ha csak egy attribútum értékét szeretnénk törölni, akkor előfordulhat, hogy ez valamiért nem lehetséges (pl. mivel része valamely kulcsnak). A kérdéses attribútumértéktől ilyenkor úgy szabadulhatunk meg, ha az egész sort töröljük, de ilyenkor elveszíthetünk olyan adatokat/információkat is, amelyekre még szükségünk lehet.
Ha pl. a cement tételt akarjuk törölni, akkor az egész sort kell, de ekkor elveszítjük Nagy Géza címét is.

A fenti problémákat megoldhatja a relációk függőleges felbontása (*vertikális dekompozíció*ja). Az építési anyag szállítók relációját például két részre célszerű felbontani:

szállító

NÉV	*CÍM*
Tóth István	Bp. Fa u. 5.
Kis János	Baja Ó u. 9.
Nagy Géza	Ózd Petőfi u. 11.

építőanyagok

NÉV	*TÉTEL*	*ÁR*
Tóth István	tégla	30
Tóth István	vas	200
Kis János	tégla	40
Kis János	pala	20
Nagy Géza	cement	350

A felbontás módja most nyilvánvalónak tűnik, de a valóságban ez ritkán van így. Ezen kívül számos probléma is felmerülhet. Nem világos pl., hogyan lehet biztosítani, hogy az eredeti reláció mindig helyreállítható legyen. Továbbá: hogyan lehet jó felbontásokat készíteni? Milyen értelemben jobb az egyik felbontás a másiknál?
Ahhoz, hogy ezekre a kérdésekre válaszolni tudjunk, a függőségek, kulcsok és normálformák mélyebb ismerete szükséges.

9.2.2. Adatbázis kényszerek

Adatbázis kényszerek alatt azokat a szabályokat értik, amelyek segítségével az adatbázisunk tartalmát olyan módon lehet jellemezni/korlátozni, hogy az valamely adatértelmezési elvárásnak, ill. elképzelt/elvárt feltételeknek megfeleljen. A leggyakrabban használt kényszerek az alábbiak (vö: **Error! Reference source not found.**):

- Értékfüggő kényszerek (pl. 0 < *TESTMAGASSÁG* < 300)
- Értékfüggetlen kényszerek
  - Tartalmazási függőség (pl. az idegen kulcsok értékeinek halmaza részhalmaza a neki megfeleltethető kulcsértékek halmazának)
  - Funkcionális függőség (ld. 9.2.3.)
  - Többértékű függőség (ld. **Error! Reference source not found.**.)

9.2.3. Funkcionális függőségek

Láttuk, hogy a relációs adatbázisok hatékony működtetésének egyik központi kérdése a relációkon belüli redundancia csökkentése. Hogyan is keletkezett a redundancia? Legegyszerűbb formájára a *szállító* relációban láttunk példát. A redundancia ott két okból keletkezett:

1. a szállító nevét több sorban is fel kell használnunk az általa szállított *különböző* tételek azonosításához.

2. ha feltételezzük, hogy a valóság úgy „működik”, hogy nincs két azonos nevű, különböző lakhelyű szállító, akkor minden olyan sorban, ahol megjelent egy szállító neve, *törvényszerűen* megjelent *ugyanaz* a lakcím is, ami nyilván felesleges ahhoz, hogy tudjuk, hol lakik a szállító.

Ez utóbbi tényt úgy is kifejezhetjük, hogy azt mondjuk: a *NÉV* attribútum értéke egyértelműen meghatározza a *CÍM* attribútum értékét, tehát minden olyan két sorban, amelyekben a *NÉV* értéke megegyezik, megegyezik a *CÍM* értéke is. A jelenségnek megfelelő matematikai konstrukciót *funkcionális (függvényszerű) függőség*nek (vagy függésnek) nevezik, melynek pontos definíciója a következő:

Definíció: Legyen adott az $R(A_1, A_2, \ldots, A_n)$ relációs séma, ahol A_i-k, $i = 1, 2, \ldots, n$ attribútumok. Legyen X és Y a reláció attribútumainak két részhalmaza: $X \subseteq R$ és $Y \subseteq R$. Ha az X értékei (mindig) *meghatározzák* a reláció által hozzárendelt Y értékeit, akkor azt mondjuk, hogy az Y attribútumok *funkcionálisan függenek* az X attribútumoktól. Mindezt így jelöljük: $X \to Y$.

Formálisan a „meghatározottságot” úgy fejezhetjük ki, ill. úgy értelmezzük, hogy ha *bármely*, az R sémára illeszkedő r reláció *bármely* két t, $t' \in r(R)$ sorára fennáll az, hogy ha $t[X] = t'[X]$, akkor $t[Y] = t'[Y]$ (ahol $t[Z]$ jelenti: $\pi_Z(t)$-t, azaz a t n-es Z attribútumhalmazra eső vetületét).

A definíció szoros kapcsolatban van a jegyzetben már korábban is többször felbukkant implikáció műveletével.

Az implikáció műveletéről

Vizsgáljuk meg a következő állítást: „*ha esik a hó, akkor nulla foknál hidegebb van*”.

Ez az állítás nyilván igaz, ha minden hóeséskor nulla foknál hidegebb van. Nem állít ugyanakkor semmit a hőmérsékletről, amikor nem esik a hó, sem általában a hőmérsékletről a hóeséstől függetlenül. Az állítást egyedül úgy cáfolhatjuk meg, ha mutatunk egy olyan esetet, amikor esik a hó, de nincs nulla foknál hidegebb.

A vizsgált állítást pl. az alábbi módon formalizálhatjuk: „*esik a hó*” $\to$ „*nulla foknál hidegebb van*”. Jelöljük az „*esik a hó”* állítást A-val, a „*nulla foknál hidegebb van”* állítást B-vel. Ekkor az állítás az $A \to B$ alakban írható fel.

Az implikáció formálisan a logikai „és”, „vagy” és „kizáró vagy” mellett a kétváltozós logikai műveletek egyike. Az implikáció műveletét általában a $\to$ szimbólummal jelölik. Magát a műveletet az alábbi igazságtáblázat definiálja (az 1 az igaz, a 0 a hamis értéket jelöli):

A	B	$A \to B$
0	0	1
0	1	1
1	0	0
1	1	1

A táblázat alapján látható, hogy $A \to B$ igazságértéke $\neg A \vee B$ igazságértékével ekvivalens. Tehát $A \to B$ akkor hamis, ha $\neg(\neg A \vee B)$ igaz. Az egyik De Morgan-azonosság felhasználásával látható, hogy $\neg(\neg A \vee B) = A \wedge \neg B$. Ennek alapján a bevezető példa esetén az implikáció akkor hamis, ha az „*esik a hó*” $\wedge$ $\neg$„*nulla foknál hidegebb van*” kiértékelése igaz értéket ad. Vegyük észre, hogy ez megfelel az állítás bevezetőben említett egyetlen lehetséges szemantikus cáfolatának („esik a hó, és nincs nulla foknál hidegebb”). Kijelenthető tehát, hogy az implikáció műveletével logikailag helyesen formalizálhatunk „ha... akkor...” típusú állításokat.

Az implikáció nem kommutatív és nem asszociatív művelet. Az $A \to B$ implikációban az A kifejezést *premisszának,* a B kifejezést *konklúziónak* nevezzük.

A matematikában számos tétel is egy implikációnak felelnek meg: ilyenkor az előtagot (A) feltételnek, az utótagot (B) állításnak nevezzük. Az ilyen tételek megfordítását kapjuk, ha felcseréljük a feltételt és az állítást (ami – mint tudjuk – vagy igaz lesz, vagy nem, *A* és *B* konkrét tartalmától függően). Egy ilyen (ha… akkor…) típusú tétel bizonyításához gyakran az indirekt módszert használjuk, amikor az állítást (*B*) tagadjuk, és ekkor ellentmondásra kell jutnunk a feltétellel (*A*). Formálisan ezt $\neg B \rightarrow \neg A$ alakban írhatjuk, amit a formális logikában *kontrapozíciónak* neveznek. (A 9. és a **Error! Reference source not found.**. fejezetekben található indirekt bizonyításokon is megfigyelhető a kontrapozíció működése.)

Egyes szerzők az implikációt a $\Rightarrow$ vagy a $\supset$ szimbólumokkal jelölik. Az implikáció fontos szerepet játszik a mesterséges intelligenciában, a formális nyelvek elméletében, az érveléstechnikában, a deklaratív programozásban és sok más tudományterületen, köztük az adatbázis-kezelés elméletében is.

Legyen most az *A* állítás az, hogy „az *R* sémára illeszkedő *bármely r* reláció *bármely* két $t, t' \in r(R)$ sorára $t[X] = t'[X]$”, a *B* állítás pedig legyen az, hogy „$t[Y] = t'[Y]$”. Ekkor az $A \rightarrow B$ implikáció teljesülése esetén definíció szerint funkcionális függésről beszélünk az *X* és az *Y* attribútum(halmaz)ok között. Formálisan erre vezettük be az $X \rightarrow Y$ jelölést.

Megjegyzések:

1. A funkcionális függőség definíció szerinti teljesüléséhez nem követelmény, hogy egy *R* sémára illeszkedő bármely *r* reláción valóban legyen két olyan $t, t' \in r(R)$ sor, amelyre fennáll, hogy $t[X] = t'[X]$ (azaz nem követeli meg, hogy az *A* állítás igaz legyen). Ha tehát egy konkrét *R* sémára illeszkedő *r* relációban nincsen ilyen két t, t' sor, az $X \rightarrow Y$ függőség akkor is fennállhat. Ekkor nyilván nem lehet ellenőrizni, hogy $t[Y] = t'[Y]$ igaz-e (azaz a *B* állítás igaz-e), miközben a funkcionális függés fennáll (ez a helyzet, ha pl. egy relációnak csak egyetlen eleme van). Ez összhangban van az implikáció definíciójával: az implikáció értéke akkor is lehet igaz, ha az *A* állítás nem igaz, azaz nincsenek az *X* attribútum(halmaz)on megegyező sorok.
2. Érdemes külön kitérni a definíciónak arra a feltételére is, hogy bármely *r*(*R*)-re (amit a gyakorlatban úgy is lefordíthatunk, hogy „bármely időpillanatban”). Gyakori az, hogy *egy adott r*(*R*) relációban (tehát pl. egy kiválasztott időpillanatban) minden olyan *t*, *t'* sorra, melyre *t*[*X*] = *t'*[*X*] fennáll *t*[*Y*] = *t'*[*Y*] is. Ilyenkor *eseti* funkcionális függőségről beszélünk. Megkülönböztetésül ezért néha az eredeti (fenti) definícióval kapcsolatban *érdemi* funkcionális függőséget emlegetünk. Amikor a valóságos viszonyokat kívánjuk modellezni (pl. egy információs rendszerben, a rendszer *tervezésének* fázisában), az természetesen az érdemi függőségek segítségével történhet. Ugyanakkor egy működő rendszer *használata során*, az adatainak elemzéséhez az eseti függőségeknek lehet nagyobb szerepe. Az eseti függőségek összessége csak bővebb lehet, mint az érdemi függőségeké. *A továbbiakban, ha nem hangsúlyozzuk külön, akkor* – a jelen jegyzet céljaival összhangban – *csak érdemi függőségekről lesz szó.*
3. Az előzőekből az is következik, hogy a funkcionális függőségek meghatározása egyfajta modellezési kérdés. Ez azt is jelenti, hogy egy reláció egyetlen pillanatnyi állapotából sohasem dönthető el egy (érdemi) függőség fennállása, legfeljebb arra következtethetünk, hogy mely függőségek nem állnak fenn.
4. Láthatóan a funkcionális függések egy relációban csak akkor okoznak redundanciát, amennyiben valamely $X \rightarrow Y$ funkcionális függés mellett a relációnak valóban van legalább két olyan eleme, amelyek *X*-ben azonosak. Ha viszont az *X* (szuper)kulcs, akkor ez a feltétel garantáltan soha nem fog teljesülni (ld. Boyce-Codd normálforma, 9.2.4.5.), ill. természetesen az is

előfordulhat, hogy az R sémára („véletlenül”) csak olyan elemeket illesztünk, amelyek mellett az X értékek különbözőek.

Példa.: Az S(*NÉV*, *CÍM*, *VÁROS*, *IRÁNYÍTÓSZÁM*, *TELEFON*) relációs sémában a valóságot „elég jól” modellező funkcionális függőségek pl. az alábbiak:

CÍM,VÁROS $\rightarrow$ *IRÁNYÍTÓSZÁM*	(ha ismerjük a címet és a város nevét, akkor ehhez egyértelműen tartozik egy irányítószám)
IRÁNYÍTÓSZÁM $\rightarrow$ *VÁROS*	(egy irányítószámhoz csak egy város tartozik)
NÉV $\rightarrow$ *CÍM*	
NÉV $\rightarrow$ *VÁROS*	
NÉV $\rightarrow$ *IRÁNYÍTÓSZÁM*	
NÉV $\rightarrow$ *TELEFON*	(ha nincsenek azonos nevek, akkor egy névhez egyértelműen tartozik egy lakcím, városnév, irányítószám és telefonszám)

Adott R sémán értelmezett (megadott, ismert) funkcionális függőségeket gyakran egyetlen halmazba gyűjtjük össze: ezt a halmazt jelöljük pl. F_R-rel.

Megadott funkcionális függőségekből kiindulva a továbbiakban számos új fogalmat definiálunk, melyekre a későbbiekben hivatkozni fogunk.

Definíció: Ha $X,Y \subset R$ és $X \rightarrow Y$ de Y nem függ funkcionálisan X egyetlen valódi részhalmazától sem, akkor X-et Y *determinánsának* nevezzük. Azt is mondjuk, hogy Y *teljesen függ* (funkcionálisan) X-től. Ha van $X' \subset X$, hogy $X' \rightarrow Y$, akkor Y *részlegesen függ* X-től.

9.2.3.1. Relációs sémák kulcsai

Az E-R modellezésnél már használtuk a kulcs fogalmát. A fizikai szervezésnél is előkerült a (keresési) kulcs fogalma. Ott azt mondtuk, kulcs minden, ami szerint keresni akarunk.

Most megadjuk egy relációs sémán értelmezett kulcs matematikai definícióját is. Az előző szakaszban bevezetett jelöléseket alkalmazzuk.

Definíció: X-et pontosan akkor nevezzük *kulcsnak* az R relációs sémán, ha

1. $X \rightarrow R$, és
2. X-nek nincs olyan X' valódi részhalmaza, hogy $X' \rightarrow R$.

9.2.3.1.1. Szuperkulcs, kulcs

X-et *szuperkulcs*nak nevezzük, ha igaz, hogy $X \rightarrow R$. Más szavakkal akkor, ha X tartalmaz kulcsot.

Néha hangsúlyozandó egy kulcs minimális voltát *minimális kulcs*ról beszélünk.

Ha egy kulcs csak egy attribútumból áll, akkor *egyszerű kulcs*, egyébként *összetett kulcs* a neve.

Pl.:

Az építőanyagok (9.2. szakasz) relációs sémájának a {*NÉV*,*TÉTEL*} attribútumhalmaz (összetett) kulcsa, ha az ottani funkcionális függőségeket értelmezzük a relációs sémán.

A jelen szakaszban definiált S relációs sémának {*NÉV*} (egyszerű) kulcsa. Egy szuperkulcsa lehet pl. a {*NÉV*,*TELEFON*} attribútumhalmaz.

A kulcsok néhány fontos tulajdonsága:

1. a kulcs a relációnak egy és csakis egy elemét („sorát”) határozza meg,

2. egy kulcs attribútumai nem lehetnek NULL-értékűek (azaz értékük meghatározatlan).

Tétel: Minden relációs sémának van kulcsa.

Válasszuk ugyanis az attribútumok teljes halmazát. Ez a kulcsokra vonatkozó első feltételnek eleget tesz, hiszen nincs olyan attribútum, amit ne vettünk volna figyelembe. Tehát meghatározza a relációs séma minden attribútumának értékét. Ha a második feltétel is teljesül, akkor kulcs, ha pedig nem, akkor szuperkulcs, tehát tartalmaz kulcsot.

9.2.3.1.2. Elsődleges kulcs

Ha X és Z az R relációs sémának egyaránt kulcsai, miközben $X \neq Z$, akkor az R relációs sémának több kulcsa is van.
Ezek közül kiválasztunk egyet, amelyet *elsődleges* kulcsnak (primary key) nevezünk. A többi kulcsot *kulcsjelöltnek* (candidate key) hívjuk.

9.2.3.1.3. Idegen (külső) kulcs

Adott egy R és egy R' relációs séma. Tételezzük fel, hogy $R' \neq R$. Ha $\exists D \subseteq (R \cap R')$, hogy $D \rightarrow R'$ és D minimális – azaz R' kulcsa –, akkor D neve az R sémával kapcsolatban *idegen kulcs*. Más szavakkal: egy sémában lehetnek olyan attribútumok, amelyek egy másik sémára illeszkedő relációban a sorokat egyértelműen azonosítják, tehát ott kulcsok. Ezeket idegen kulcsoknak nevezzük.

9.2.3.2. Funkcionális függőségek további tulajdonságai

A korábbiak alapján már sejthetjük, hogy a megadottakon kívül más funkcionális függőségek is igazak lehetnek. Egy jó példa erre a *triviális függőség,* amit ugyan ritkán hangsúlyozunk, amikor az adatok egy valóságos helyzetbeli jelentését írjuk le funkcionális függőségekkel, de attól még fennáll. Ha egy R séma tartalmazza az A és B attribútumokat, akkor mindig fennállnak ezen a sémán az $A \cup B \rightarrow B$ vagy az $A \cup B \rightarrow A$ függőségek is. Hiszen minden R sémára illeszkedő r relációban $\forall t$, $t' \in r(R)$ elemekre, ha $t[A \cup B] = t'[A \cup B]$ akkor $t[B] = t'[B]$ valamint $t[A] = t'[A]$ egyaránt törvényszerűen fennáll.

Jelölés: a továbbiakban az $A \cup B$ helyett egyszerűen AB-t írunk.

Ha azonban több függőség is ismert, vagy a sémának számos attribútuma van, akkor már nem nyilvánvaló annak a megválaszolása, hogy mi az adott F_R függőségek mellett még fennálló függőségek teljes rendszere. Mivel ennek a kérdésnek pl. a relációs sématervezésnél nagy jelentősége lesz, ezért a cél az, hogy az összes függőség előállítására a gyakorlatban is jól használható „módszert” adjunk. A „módszer” következtetési szabályok, ún. axiómák alkalmazása lesz. Ezektől az axiómáktól a következő két alapvető tulajdonságot várjuk el:

- csak olyan függőségeket lehessen velük előállítani, amelyek „igazak”, ugyan-akkor
- „meg lehessen kapni” minden „igaz” (funkcionális) függőséget, amelyek egy adott függőséghalmazban található függőségekkel nincsenek ellentmondásban.

Az „igaz” és a „meg lehet kapni” kifejezések alatt pontosan az alábbiakat értjük:

Definíció: (*igaz függés adott függéshalmaz mellett*) egy adott R sémán az attribútumain értelmezett F_R függéshalmaz mellett egy $X \to Y$ függőség pontosan akkor igaz, ha minden olyan $r(R)$ reláción fennáll, amelyeken F_R összes függősége is fennáll. Jelölése: $F_R \models X \to Y$.

Definíció: („meg lehet kapni", azaz *levezethető*) egy $W \to Z$ funkcionális függőség pontosan akkor vezethető le adott F_R függőségekből, ha az axiómák ismételt alkalmazásával F_R-ből kiindulva megkaphatjuk $W \to Z$-t. Jelölése: $F_R \vdash W \to Z$.

Az imént definiált tulajdonságokkal bíró – valójában következtetési – szabályok *Armstrong „axiómái"* néven váltak ismertté.

9.2.3.3. Armstrong axiómái a funkcionális függőségekről

Adottak az R sémán az X, Y, Z attribútumhalmazok.

a) Ha $X \subseteq Y$, akkor $Y \to X$ (reflexivitás vagy triviális függőség).
b) Ha $X \to Y$ és $Y \to Z$, akkor $X \to Z$ (tranzitivitás).
c) Ha $X \to Y$, akkor $XZ \to YZ$ (bővíthetőség).

Tétel: (*igazság tétel*[3]) Az Armstrong axiómák igazak (helyesek), alkalmazásukkal csak igaz függőségek állíthatók elő adott függéshalmazból.

Formálisan: $F_R \vdash X \to Y \Rightarrow F_R \models X \to Y$

Bizonyítás: Lássuk be egyenként, hogy az axiómák igazak, akkor nyilván igaz lesz minden olyan függőség is, amelyet az axiómák (véges számban) ismételt alkalmazásával kapunk (azaz le tudunk vezetni). Lássuk be pl., hogy *c)* igaz! Ehhez tételezzük fel, hogy *c)* nem igaz, bár $X \to Y$ fennáll. Ez azt jelenti, hogy ha $\exists t, t'$ n-esek valamely $r(R)$ relációban, hogy $t[XZ] = t'[XZ]$, akkor $t[YZ] \neq t'[YZ]$. A Z-hez tartozó attribútumok nyilván megegyeznek a t és t' sorokban, hiszen különben $t[XZ]$ és $t'[XZ]$ nem lehetne azonos. Tehát az Y-beli attribútumok értékében különbözik $t[YZ]$ és $t'[YZ]$, azaz $t[Y] \neq t'[Y]$. De ez nem lehetséges, mert a kiindulási $X \to Y$ függőség miatt ha $t[X] = t'[X]$, akkor $t[Y] = t'[Y]$. Tehát ellentmondásra jutottunk *c)* tagadásával, így *c)* igaz kell, hogy legyen.

Az axiómák tömören:

a) $X \subseteq Y \models Y \to X$
b) $X \to Y$ és $Y \to Z \models X \to Z$
c) $X \to Y \models XZ \to YZ$

9.2.3.4. Az axiómák következményei

d) $X \to Y$ és $X \to Z \models X \to YZ$ (egyesítési szabály).
e) $X \to Y$ és $WY \to Z \models XW \to Z$ (pszeudotranzitivitás).
f) $X \to Y$ és $Z \subseteq Y \models X \to Z$ (dekompozíciós/felbontási szabály).

Fentiek most már az Armstrong axiómák felhasználásával is bizonyíthatóak, példaképpen nézzük meg *d)* bizonyítását!

$X \to Y \models X \to XY$ (*c)* miatt)
$X \to Z \models XY \to ZY$ (*c)* miatt)
$X \to XY$ és $XY \to ZY \models X \to ZY$ (*b)* miatt), ami éppen a bizonyítandó állítás.

[3] Eredetileg: soundness theorem. Emiatt talán jobb lenne helyesség tételnek nevezni, de nem ez honosodott meg.

Példa: Vegyük elő újra az $R(NÉV, TÉTEL, CÍM, ÁR)$ sémát a rajta értelmezett $NÉV, TÉTEL \rightarrow ÁR$ és $NÉV \rightarrow CÍM$ függőségekkel! Keressük meg R kulcsát!
$NÉV, TÉTEL \rightarrow ÁR \vDash NÉV, TÉTEL \rightarrow ÁR, NÉV, TÉTEL$
$NÉV \rightarrow CÍM \vDash NÉV, TÉTEL \rightarrow CÍM, TÉTEL$.
Az egyesítési szabályt alkalmazva adódik, hogy $NÉV, TÉTEL \rightarrow ÁR, NÉV, TÉTEL, CÍM$. Mivel $NÉV, TÉTEL$ együttesen meghatározza R mindegyik attribútumát, ezért $NÉV, TÉTEL$ együtt (szuper)kulcsot alkotnak. Láthatóan bármelyik attribútumot is hagyjuk el, már nem fogja R mindegyik attribútumát meghatározni, tehát a $\{NÉV, TÉTEL\}$ attribútumhalmaz minimális is, azaz kulcs.

9.2.3.5. Attribútumhalmaz lezárása

Gyakran felmerülő kérdés, hogy bizonyos attribútumok értékeinek ismeretében esetleg milyen más attribútumok értékeit tekinthetjük még ismertnek azt kihasználva, hogy a (funkcionális) függőségek rendszerén keresztül egyes attribútumok meghatározzák más attribútumok értékeit. Lényegében ezt fejezi ki az alábbi definíció:

Definíció: Az X *attribútumhalmaz lezárása* adott F függéshalmaz mellett az a legbővebb $W \subseteq R$ halmaz, amelyre az $X \rightarrow W$ függőség az adott F függéshalmaz mellett fennáll. Jelölése: $X^+(F)$.

Formálisan: $X^+(F) = \{A \mid A \in R \text{ és } F \vDash X \rightarrow A\}$

Tehát $\forall Y$-ra, amelyre $X \rightarrow Y$ igaz, hogy $Y \subseteq X^+$, és megfordítva: $\forall Y$-ra, amelyre $Y \subseteq X^+$ igaz, hogy $X \rightarrow Y$.

Algoritmus: Egy attribútumhalmaz lezárása csaknem lineáris időben meghatározható az alábbi algoritmus segítségével:

$X^{(0)} = X$
:
$X^{(i+1)} = X^{(i)} \cup \{A \mid \exists V \subseteq X^{(i)}, V \rightarrow U \in F \text{ és } A \in U\}$

Példa: Adott az $R(A, B, C, D)$ séma és a funkcionális függőségek F halmaza: $F = \{A \rightarrow B, B \rightarrow D\}$. Mi az $X = AC$ attribútumhalmaz lezárása, X^+?

$X^{(0)} = AC$
$X^{(1)} = AC \cup \{B\}$ ($A \rightarrow B$ miatt)
$X^{(2)} = ACB \cup \{D\}$ ($B \rightarrow D$ miatt)

Mivel $X^{(2)} = ABCD$, vagyis az R séma attribútumainak teljes halmaza, így az eljárás véget ért, $X^+ = ABCD$, amiből az is következik, hogy AC szuperkulcsa a relációnak. (Mivel AC minimális is, ezért valójában kulcs.)

Tétel: (*teljesség tétel*) Az Armstrong axiómák teljesek, azaz belőlük minden igaz függőség levezethető. (Nem bizonyítjuk.)

Mostantól kezdve a $\vDash$ és a $\vdash$ jeleket egyenértékűeknek, felcserélhetőknek tekinthetjük.

9.2.3.6. Függéshalmaz lezárása

Láttuk, hogy (funkcionális) függőségeket definiálva az Armstrong axiómák segítségével továbbiakat is meg tudunk határozni, sőt mindegyiket, amelyik logikailag a megadottakból következik. Gyakran hasznos, ha ezekre a függőségekre közösen tudunk hivatkozni.

Definíció: Az *F függéshalmaz lezárása* mindazon függőségek halmaza, amelyek az *F* függéshalmaz elemeiből az Armstrong axiómák alapján következnek.

Formálisan: $F^+ = \{X \to Y \mid F \models X \to Y\}$.

Még ha F viszonylag kicsi is, F^+ igen nagy lehet. Ha pl. $F = \{A \to B, B \to C\}$, akkor F^+ elemei: $F^+ = \{A \to B,\ B \to C,\ A \to AB,\ AB \to B,\ B \to BC,\ BC \to C,\ A \to C,\ AB \to C, A \to \emptyset, A \to A, B \to \emptyset, B \to B, \ldots\}$ ($\emptyset$ az üres halmaz jele)
Tanulság: könnyen belátható, hogy – az adott függőségek szerkezetétől függően – a lezárt halmaznak 2^n számú eleme is lehet (sőt, néha még több is), így a lezárt meghatározása esetenként igen költséges művelet. Ez az oka annak, hogy a függéshalmaz lezártjára elméleti megfontolásokban gyakran fogunk hivatkozni, gyakorlatban használható algoritmusaink ellenben lehetőleg nem támaszkodnak a meghatározására.

Gyakori az a feladat, hogy el kell dönteni egy függőségről – legyen ez $A \to B$ –, hogy következik-e adott *F* függőséghalmazból. A feladat direkt megoldása lehetne, hogy kiszámítjuk F^+-et és megvizsgáljuk, hogy elemei között van-e $A \to B$. Függéshalmaz lezártjának költséges számítása helyett gyorsabban eredményre jutunk, ha (lineáris időben!) A^+-t határozzuk meg a megismert algoritmussal. Amennyiben $B \in A^+$, akkor $A \to B \in F^+$. Ez az attribútumhalmaz lezártja definíciójának közvetlen következménye.
Ha függőségeknek egy bonyolult rendszere adott, gyakran szeretnénk könnyebben áttekintető, egyszerűbb formába alakítani – nyilván úgy, hogy közben az új alak ugyanazt az „információt” hordozza, mint az eredeti. Ez alatt azt értjük, hogy a módosított függéshalmaz segítségével pontosan ugyanazokat a függőségeket lehessen előállítani. A függéshalmaz lezárása segítségével lehetőségünk van arra, hogy egyszerű definíciót adjunk két függéshalmaz egyenlőségére.

Definíció: Két függéshalmaz pontosan akkor *ekvivalens*, ha lezártjaik megegyeznek.

Ezt így jelöljük: $F = G \Leftrightarrow F^+ = G^+$ és azt is mondjuk, hogy *F* lefedi *G*-t, ill. *G* lefedi *F*-et.
Láttuk, hogy az ekvivalencia eldöntése a definíció alapján igen költséges lehet, így a gyakorlatban használhatóbb az alábbi algoritmus:

Vizsgáljuk meg, hogy $F \subseteq G^+$ és $G \subseteq F^+$ egyaránt teljesül-e. Ha igen, akkor ekvivalensek.

Először az $F \subseteq G^+$ teljesülését vizsgáljuk. Minden $X \to \mathrm{Y} \in \mathrm{F}$ függőségre a tartalmazás hatékonyan eldönthető $X^+(G)$ kiszámításával: ha $Y \subseteq X^+(G)$, akkor $X \to Y \in G^+$.

$G \subseteq F^+$ eldöntése nyilván azonos elven történhet.

Definíció: F *minimális függéshalmaz* (beszélünk *F minimális fedés*éről is) akkor, ha
1. a függőségek jobb oldalán csak egyetlen attribútum van,
2. a függőségek bal oldaláról nem hagyható el attribútum,
3. nincs olyan függőség, amely elhagyható.

Tétel: Adott függéshalmazzal ekvivalens minimális függéshalmaz mindig előállítható.

Bizonyítás: konstruktív.
Adott egy F függéshalmaz.

1. A felbontási szabály alapján minden $X \to Y \in F$ függőség helyettesíthető $X \to A_1$, $X \to A_2, \ldots,$ $X \to A_n$ függőségekkel (ahol $Y = \{A_1, A_2, \ldots, A_n\}$), így egy F' függéshalmazt kapunk. Nyilvánvaló, hogy F és F' ekvivalensek.
2. Minimalizáljuk a függőségek bal oldalát. Ehhez vizsgáljuk meg $\forall S \to B \in F'$ függőségre, ahol $S = \{D_1, D_2, \ldots, D_n\}$, hogy elhagyható-e valamely D_i attribútum. Definíció szerint ehhez az kell, hogy $(F'')^+ = (F')^+$ fennálljon (ahol $F'' = F' \setminus \{S \to B\} \cup \{\{D_1, D_2, \ldots, D_{i-1}, D_{i+1}, \ldots, D_n\} \to B\}$), ami ekvivalens az $F' \subseteq (F'')^+$ és $F'' \subseteq (F')^+$ egyidejű fennállásával. A függéshalmazok csak egyetlen függésben különböznek:
$F' = \{X_1 \to A_1, X_2 \to A_2, \ldots, \{D_1, D_2, \ldots, D_{i-1}, D_i, D_{i+1}, \ldots, D_n\} \to B\}$
$F'' = \{X_1 \to A_1, X_2 \to A_2, \ldots, \{D_1, D_2, \ldots, D_{i-1}, D_{i+1} \ldots, D_n\} \to B\}$.
$F' \subseteq (F'')^+$ ezért ekvivalens $B \in S^+(F'')$, vagyis
$B \in S^+(F' \setminus \{S \to B\} \cup \{D_1, D_2, \ldots, D_{i-1}, D_{i+1}, \ldots, D_n \to B\})$ vizsgálatával,
$F'' \subseteq (F')^+$ pedig $B \in (D_1, D_2, \ldots, D_{i-1}, D_{i+1}, \ldots, D_n)^+(F')$ vizsgálatával.
Mivel $B \in S^+(F' \setminus \{S \to B\} \cup \{D_1, D_2, \ldots, D_{i-1}, D_{i+1}, \ldots, D_n \to B\})$ triviálisan teljesül $S = \{D_1, D_2, \ldots, D_n\}$ miatt, elegendő csupán a másik irányt megvizsgálni. Eredményül egy F'' függéshalmazt kapunk, amely továbbra is ekvivalens F-fel.
3. Vizsgáljuk meg $\forall T \to C \in F''$ függésre, hogy elhagyható-e. Definíció szerint ehhez az kell, hogy $(F'' \setminus \{T \to C\})^+ = (F'')^+$ fennálljon, aminek az eldöntése költséges. Célszerűbb ezért azt vizsgálni, hogy $C \in T^+(F'' \setminus \{T \to C\})$ vajon teljesül-e – amint már korábban beláttuk, ez ekvivalens, ugyanakkor hatékonyabb. Végül egy olyan F^* függéshalmazt kapunk, amely továbbra is ekvivalens F-fel, de a minimális függéshalmaz mindegyik tulajdonságának megfelel.

Megjegyzés: Vegyük észre, hogy a 2. és 3. lépésekben az egyes attribútumok, ill. függőségek elhagyásának sorrendje tetszőleges. Ennek következtében a végeredményül kapott minimális függéshalmazok különbözőek is lehetnek. Következmény: egy adott függéshalmazzal ekvivalens minimális függéshalmaz nem feltétlenül egyértelmű!

Példa:
Adott: $F = \{AB \to CD, AC \to BD, C \to AB\}$, mi egy minimális fedése?

1. $F' = \{AB \to C, AB \to D, AC \to B, AC \to D, C \to A, C \to B\}$
2. $C \to A$ miatt az $AC \to B$ függőség $C \to B$-re, $AC \to D$ $C \to D$-re redukálható: $F'' = \{AB \to C, AB \to D, C \to B, C \to D, C \to A\}$. (A függőségek száma azért csökkent, mert $C \to B$ szerepelt az eredeti függéshalmazban is.)
3. $AB \to C$ és $C \to D$ miatt (a tranzitivitási axióma következtében) $AB \to D$ elhagyható. A kapott függéshalmaz minimális:
$F_1^* = \{AB \to C, C \to D, C \to A, C \to B\}$.
A minimális fedés nem egyértelmű, mert F''-ből $C \to A$, $C \to B$ és

$AB \to D$ miatt $C \to D$ elhagyható, ekkor
$F_2^* = \{AB \to C, AB \to D, C \to A, C \to B\}$.

9.2.4. Relációs sémák normálformái

Ahhoz, hogy a 9.2.1. szakaszban ismertetett anomáliákat elkerülhessük, a relációink sémái meghatározott feltételeket kell, hogy teljesítsenek. Ezeket a feltételeket *normálformák*nak nevezik (Codd, 1970). A normálformák tehát megszorítások a relációs séma tulajdonságaira vonatkozóan annak érdekében, hogy a sémákra illeszkedő relációkkal végzett műveletek során egyes nemkívánatos jelenségeket elkerülhessünk.

9.2.4.1. A nulladik normálforma (0NF)

Ilyen alakúnak tekintünk minden olyan relációs sémát, amelyben legalább egy attribútum nem atomi abban az értelemben, hogy az attribútum értéke nem tekinthető egyetlen egységnek, azaz egyes részeihez külön is hozzá akarunk férni.
0NF-ben van az a relációs séma, amelyik pl. ismétlődő csoportot tartalmaz az attribútumai között.
Példa 0NF alakra:

EGYETEM_NÉV
REKTOR
    KAR
    DÉKÁN
        TANSZÉK
        VEZETŐ

UNI(*EGYETEM_NÉV*, *REKTOR*(*KAR*, *DÉKÁN*(*TANSZÉK*, *VEZETŐ*)*)*)
A * jellel az ismétlődő csoportokat jelöltük.

9.2.4.2. Az első normálforma (1NF)

Definíció: Egy relációs séma 1NF alakú (vagy más szóval *normalizált*), ha csak atomi attribútum-értékek szerepelnek benne.

Megj.: Az 1NF alak definíciója nyilvánvalóan nem a redundancia csökkentést szolgálja. Egyszerűen csak kiindulási alapot teremt a további normalizálás számára. Hacsak nem hangsúlyozzuk külön az ellenkezőjét, a továbbiakban mindig fel fogjuk tételezni, hogy a sémáink normalizáltak a fenti értelemben.

9.2.4.3. A második normálforma (2NF)

Definíció: Egy R relációs séma $A \in R$ attribútuma *elsődleges*, ha A eleme a séma valamely K kulcsának. Egyébként A *másodlagos attribútum*.

Ezek szerint egy relációs séma kulcsai az attribútumokat két diszjunkt halmazba sorolják:
Ha R kulcsainak halmaza $\{K_1, K_2, \ldots, K_n\}$, ahol $K_i \subseteq R$, akkor $\cup K_i$ $\forall i$-re az elsődleges attribútumok, $R \setminus \cup K_i$ pedig a másodlagos attribútumok halmaza.

Definíció: Egy 1NF relációs séma 2NF alakú, ha benne minden másodlagos attribútum a séma bármely kulcsától teljesen függ.
Más szavakkal: másodlagos attribútum nem függ egyetlen kulcs egyetlen valódi részhalmazától sem.

A 2NF definíciójának célja már nyilvánvalóan a redundanciacsökkentés. Ha ugyanis egy attribútumhalmaznak (jelen esetben kulcsnak) már egy része is meghatároz egy másodlagos attribútumot, akkor annak a másodlagos attribútumnak a meghatározásához elegendő csupán a kulcs adott része attribútumértékeit tárolni.
Ha megsértjük a 2NF definícióját, és lehetővé tesszük másodlagos attribútumnak részkulcstól való függését is, akkor ebben a másodlagos attribútumban az attribútumértékek redundáns tárolása megvalósulhat, amint azt az alábbi példa szemlélteti.

Példa:
Az $R(ABCD)$ atomi attribútumokat tartalmazó séma az $F = \{AB \to C,\ B \to D\}$ függéshalmaz mellett 1NF-ben van, ugyanis a kulcsa (AB), elsődleges attribútumai A és B, másodlagos attribútumai C és D. A D attribútum a $B \to D$ függés miatt az AB kulcs részétől is függ, tehát nem lehet 2NF-ben. Az (R, F) sémára illeszkedik pl. a

A	*B*	*C*	*D*
a1	b1	c1	d1
a2	b1	c2	d1

reláció, amely a D attribútumban redundáns tárolást valósít meg a $B \to D$ függés következtében, hiszen a B értékeinek ismétlődését semmi nem zárja ki, ilyen esetekben viszont a D értékei már törvényszerűen ismétlődni fognak a $B \to D$ függés (kulcs valódi részhalmaza határoz meg egy másodlagos attribútumot) miatt. Ennek elkerülésére zárja ki a 2NF definíciója a másodlagos attribútumok részkulcstól való függését.

A definíció közvetlen következményei:

- Ha minden kulcs egyszerű, akkor 1NF $\Rightarrow$ 2NF.
- Ha nincsenek másodlagos attribútumok, akkor 1NF $\Rightarrow$ 2NF.

Tétel: Minden 1NF relációs séma felbontható 2NF sémákba úgy, hogy azok felhasználásával az 1NF sémára illesztett „eredeti” relációk helyreállíthatók (torzulás nélkül, ld. a veszteségmentes dekompozícióról szóló 9.2.5. szakaszt).

Példa:
Az **Error! Reference source not found.**. szakaszban ismertetett feladathoz a relációs sémák megtervezéséhez a *NAPI_HELYZET* sémából indulunk ki, amely minden attribútumot tartalmaz:
NAPI_HELYZET(*DÁTUM, ÖSSZEG, ÁRUNÉV, DB, ÁRUKÓD, EGYSÁR, BEFIZ*)
Ez a séma 1NF alakú, mivel benne minden attribútum atomi.
A sémán az alábbi funkcionális függőségeket definiáljuk (amelyek a valóságos viszonyokkal is jó összhangban vannak):

$ÁRUKÓD \to ÁRUNÉV, EGYSÁR$	(az áru kódja meghatározza az áru nevét és egységárát)
$DÁTUM, ÁRUKÓD \to DB$	(minden nap minden árucikkből meghatározott darabszámút adtak el)
$DÁTUM \to ÖSSZEG, BEFIZ$	(minden nap egy jól meghatározott összeget visznek a bankba)
$ÖSSZEG \to BEFIZ$	(a $BEFIZ = ÖSSZEG - 4000$ törvényszerűség összekapcsolja *BEFIZ* és *ÖSSZEG* egy-egy értékét)
$BEFIZ \to ÖSSZEG$	

Más funkcionális függőséget nem definiálunk.

Láthatóan a *NAPI_HELYZET* egy kulcsa a {*DÁTUM*, *ÁRUKÓD*} attribútumhalmaz, amely összetett kulcs. Nem nyilvánvaló ugyan, de a relációnak ez az egy kulcsa van.
Elsődleges attribútumok: *DÁTUM*, *ÁRUKÓD*
Másodlagos attribútumok: *DB*, *ÖSSZEG*, *BEFIZ*, *ÁRUNÉV*, *EGYSÁR*
A *NAPI_HELYZET* reláció nincsen 2NF-ben, mert pl. a *DÁTUM* $\rightarrow$ *ÖSSZEG*, *BEFIZ* függőség miatt van olyan attribútum (pl. *ÖSSZEG*), amelyet már a kulcs egy része is meghatároz (*DÁTUM*).
Bontsuk fel ezért a *NAPI_HELYZET*-et több relációra az alábbiak szerint:
ÁRU(*ÁRUKÓD*, *ÁRUNÉV*, *EGYSÁR*)
MENNYISÉG(*DÁTUM*, *ÁRUKÓD*, *DB*)
BEVÉTEL(*DÁTUM*, *ÖSSZEG*, *BEFIZ*)
Ahhoz, hogy ezen sémák normálformáiról mondhassunk valamit, szükségünk van a sémákon értelmezett függőségekre. (Ehhez részletesen ld. a vetített függőségekről szóló szakaszt). Itt csak a végeredményt közöljük:
$F_{ÁRU}$ = {*ÁRUKÓD* $\rightarrow$ *ÁRUNÉV*, *EGYSÁR*}
$F_{MENNYISÉG}$ = {*DÁTUM*, *ÁRUKÓD* $\rightarrow$ *DB*}
$F_{BEVÉTEL}$ = {*DÁTUM* $\rightarrow$ *ÖSSZEG*, *BEFIZ*; *ÖSSZEG* $\rightarrow$ *BEFIZ*; *BEFIZ* $\rightarrow$ *ÖSSZEG*}
Könnyen belátható, hogy most már mindhárom reláció 2NF-ben van: az első és utolsó azért, mert kulcsuk egyszerű kulcs, *MENNYISÉG* pedig azért, mert *DB*-t az összetett kulcs egyik része sem határozza meg.

9.2.4.4. A harmadik normálforma (3NF)

A definícióhoz szükségünk lesz néhány új fogalomra.

Definíció: (triviális függőség) Ha az *X*, *Y* attribútumhalmazokra igaz, hogy $Y \subseteq X$, akkor az $X \rightarrow Y$ függőséget *triviális függőségnek* nevezzük, egyébként a függőség *nemtriviális*.

Definíció: (tranzitív függés) Adott egy *R* séma, a sémán értelmezett funkcionális függőségek *F* halmaza, $X \subseteq R$, $A \in R$. *A tranzitívan függ X*-től, ha $\exists Y \subset R$, hogy $X \rightarrow Y$, $Y \not\rightarrow X$, $Y \rightarrow A$ és $A \notin Y$.

Pl.: adott *R*(*Járat*, *Dátum*, *Pilótakód*, *Név*) = *R*(*J*, *D*, *P*, *N*),
$F = \{JD \rightarrow P, P \rightarrow N, N \rightarrow P\}$.
Az *N* attribútum tranzitívan függ (J, D)-től, mert $(J, D) \rightarrow P$, $P \not\rightarrow JD$, $P \rightarrow N$ és nyilván $N \notin P$ az egyszerű attribútumok miatt.

Definíció 1: (3NF) Egy 1NF *R* séma 3NF, ha $\forall A \in R$ másodlagos attribútum és $\forall X \subseteq R$ kulcs esetén $\nexists Y$, hogy $X \rightarrow Y$, $Y \not\rightarrow X$, $Y \rightarrow A$ és $A \notin Y$.

Szavakkal: ha egyetlen másodlagos attribútuma sem függ tranzitívan egyetlen kulcstól sem.

Definíció 2: (3NF) Egy 1NF *R* séma 3NF, ha $\forall X \rightarrow A$, $X \subseteq R$, $A \in R$ nemtriviális függőség esetén

- *X* szuperkulcs vagy
- *A* elsődleges attribútum.

A 3NF első definíciója szemléletesebb, ha a redundanciacsökkentést tartjuk szem előtt. Felesleges ugyanis egyazon relációban tárolni az *X*, *Y*, *A* attribútumokat. Hiszen minden olyan sorban, amelyben *X* és *Y* értékeit rendeljük egymáshoz, meg kell adnunk A értékét is, amelyek azonosak kell, hogy legyenek minden olyan sorban,

amelyben Y értéke azonos. Márpedig Y értéke azonos lehet különböző X értékekre is. Ilyenkor az $Y \rightarrow A$ függőségnek megfelelő (y, a) értékeket többszörösen tároljuk, nyilvánvaló redundanciát „építve bele" a relációba.
Egyes szerzők jobban kedvelik a 3NF második definícióját, talán azért, mert gyakran könnyebben ellenőrizhető ebben a formában, hogy egy séma teljesíti-e a 3NF kritériumait. A definíció szemléletes tartalma ugyanakkor itt már nem nyilvánvaló.

Tétel: Def. 1. $\Leftrightarrow$ Def. 2.

Bizonyítás:
Előre: Def. 1. $\Rightarrow$ Def. 2.
Indirekt: Def. 1. feltételei mellett t. f. h. $\exists Z \rightarrow B$ nemtriviális függőség, ahol Z nem szuperkulcs, és B nem elsődleges attribútum.
Viszont minden relációnak létezik kulcsa, legyen ez X. Igaz tehát, hogy $X \rightarrow Z$, $Z \not\rightarrow X$ (mert akkor Z szuperkulcs lenne), $Z \rightarrow B$, $B \notin Z$ (mert különben $Z \rightarrow B$ triviális függőség lenne). Ez pedig éppen egy másodlagos attribútum kulcstól való tranzitív függése, ellentmondásban Def. 1. feltételeivel. Tehát Def. 1. $\Rightarrow$ Def. 2.
Visszafelé: Def. 2. $\Rightarrow$ Def. 1.
Indirekt: Def. 2. feltételei mellett t. f. h. $\exists Y \subset R$, $\exists X$ kulcs és $\exists A$ másodlagos attribútum, hogy $X \rightarrow Y$, $Y \not\rightarrow X$, $Y \rightarrow A$ és $A \notin Y$.
$X \rightarrow Y$: mivel X kulcs, ezért nincs ellentmondásban Def. 2-vel,
$Y \not\rightarrow X$, tehát Y nem lehet szuperkulcs (ellenkező esetben a séma összes attribútumát meghatározná, tehát az X kulcsot is),
$Y \rightarrow A$ és $A \notin Y$ miatt tehát létezik egy nemtriviális függőség, melyben Y nem szuperkulcs, A nem elsődleges attribútum, ellentmondásban Def. 2. feltételeivel. Tehát Def. 2. $\Rightarrow$ Def. 1.
Megjegyzés: a definíció szerint minden F^+-beli függőségeken ellenőrizni kellene a feltételt.

Tétel: ahhoz, hogy egy (R, F) sémáról (Def. 2. alkalmazásával) eldöntsük, hogy 3NF-e, elég az F-beli funkcionális függőségek vizsgálata. (Nem bizonyítjuk.)

Tétel: Minden legalább 1NF relációs séma felbontható 3NF sémákba úgy, hogy azokból az eredeti reláció információveszteség nélkül helyreállítható.

Bizonyítás: ld. 9.2.7. szakasz: veszteségmentes dekompozíció 3NF-be.

Példa:
Az előző szakasz *ÁRU*, *MENNYISÉG* és *BEVÉTEL* relációs sémáit vizsgáljuk meg, hogy teljesítik-e a 3NF kritériumait:
Az *ÁRU*(*ÁRUKÓD*, *ÁRUNÉV*, *EGYSÁR*) séma nemtriviális függősége csupán *ÁRUKÓD* → *ÁRUNÉV*, *EGYSÁR*, tehát *ÁRUKÓD* kulcs, így *ÁRU* 3NF alakú.
A *MENNYISÉG*(*DÁTUM*, *ÁRUKÓD*, *DB*) séma egyetlen nemtriviális függősége *DÁTUM*, *ÁRUKÓD* → *DB*, tehát (*DÁTUM*, *ÁRUKÓD*) kulcs, így *MENNYISÉG* is 3NF alakú.
A *BEVÉTEL*(*DÁTUM*, *ÖSSZEG*, *BEFIZ*) séma nemtriviális függőségei:

DÁTUM → *ÖSSZEG*, *BEFIZ*
ÖSSZEG → *BEFIZ*
BEFIZ → *ÖSSZEG*

A *BEVÉTEL* egyetlen kulcsa tehát *DÁTUM*, másodlagos attribútumai *ÖSSZEG* és *BEFIZ*. Az utóbbi két függőségben a determináns nem szuperkulcs, és az sem teljesül,

hogy ilyenkor a függőségek jobb oldalán elsődleges attribútum áll. *BEVÉTEL* így *nem* 3NF alakú. Bontsuk fel ezért az alábbi formában:
BEVÉT(*DÁTUM*, *ÖSSZEG*)
BEFIZ(*DÁTUM*, *BEFIZETÉS*)
Könnyen belátható, hogy mindkettő 3NF alakú.

Tétel: Ha egy séma 3NF alakú, akkor 2NF is egyben. (A 9.2.3 szakaszban definiált implikáció műveletével ezt így is megfogalmazhatjuk: 3NF(R)→2NF(R))

Bizonyítás: Indirekt. T. f. h. az *R* séma nem 2NF, ekkor ellentmondásra kell jutnunk a 3NF definíciójával (azaz $\neg 2NF(R) \to \neg 3NF(R)$). Ezzel ekvivalens, ha belátjuk azt, hogy másodlagos attribútum részkulcstól való függéséből következik kulcstól való tranzitív függése.

Legyen $A \in R$ másodlagos attribútum, *K* egy kulcs, amelyekre $K \to A$, továbbá $\exists K' \subset K$, hogy $K' \to A$ is igaz. $K' \not\to K$, hiszen ekkor *K* nem lehetne minimális, tehát nem lehetne kulcs. Továbbá $A \notin K$, mert *A* másodlagos attribútum, tehát egyetlen kulcsnak sem lehet eleme. Tehát: $K \to K'$, $K' \not\to K$, $K' \to A$, $A \notin K'$, ami éppen azt jelenti, hogy az *A* attribútum tranzitívan függ a *K* kulcstól, ellentmondásban a 3NF definíciójával.

A bizonyításból az is következik, hogy minden másodlagos attribútum részleges függése kulcstól egyúttal a másodlagos attribútum tranzitív függését is jelenti.

9.2.4.5. A Boyce-Codd normálforma (BCNF)

Vegyük észre, hogy a 3NF definíciója csak a másodlagos attribútumok kulcstól való tranzitív függését zárja ki. Lehetséges tehát elsődleges attribútum kulcstól való tranzitív függése 3NF sémákban. Ez viszont azt jelenti, hogy – a már ismert okfejtést követve – a 3NF relációk is tartalmazhatnak még redundanciát. Indokolt tehát további normálforma, a *Boyce-Codd normálforma (BCNF)* bevezetése. Be fogjuk látni, hogy a BCNF sémákra illeszkedő relációk már redundanciamentesek.

Definícó 1: (BCNF) Egy 1NF *R* séma BCNF, ha $\forall A \in R$ attribútum és $\forall X \subseteq R$ kulcs esetén $\nexists Y$, hogy $X \to Y$, $Y \not\to X$, $Y \to A$ és $A \notin Y$.

Szavakkal: egyáltalán nincs tranzitív függőség kulcstól.

Definíció 2: (BCNF) Egy 1NF *R* séma BCNF, ha $\forall X \to A$, $X \subseteq R$, $A \in R$ nemtriviális függőség esetén *X* szuperkulcs.

Vegyük észre, hogy minden séma, amely legfeljebb két attribútumot tartalmaz, törvényszerűen BCNF.

Tétel: Def. 1. $\Leftrightarrow$ Def. 2.

Bizonyítás: Hasonló a 3NF definícióinál leírtakhoz
Előre: Def. 1. $\Rightarrow$ Def. 2.
Indirekt: Def. 1. feltételei mellett t. f. h. $\exists Z \to B$ nemtriviális függőség, hogy *Z* nem szuperkulcs.
Viszont minden relációnak létezik kulcsa, legyen ez *X*. Igaz tehát, hogy $X \to Z$, $Z \not\to X$, $Z \to B$, $B \notin Z$. Ez pedig éppen a *B* attribútum *X* kulcstól való tranzitív függése, ellentmondásban Def. 1. feltételeivel. Tehát Def. 1. $\Rightarrow$ Def. 2.
Visszafelé: Def. 2. $\Rightarrow$ Def. 1.

Indirekt: Def. 2. feltételei mellett t. f. h. $\exists Y \subset R$, $\exists X$ kulcs és $\exists A$ attribútum, hogy $X \rightarrow Y$, $Y \not\rightarrow X$, $Y \rightarrow A$ és $A \notin Y$.

$X \rightarrow Y$: mivel X kulcs, ezért nincs ellentmondásban Def. 2-vel,

$Y \not\rightarrow X$, tehát Y nem lehet szuperkulcs,

$Y \rightarrow A$ és $A \notin Y$ miatt tehát létezik egy nemtriviális függőség, melyben Y nem szuperkulcs, ellentmondásban Def. 2. feltételeivel. Tehát Def. 2. $\Rightarrow$ Def. 1.

Tétel: ahhoz, hogy egy (R, F) sémáról (Def. 2. alkalmazásával) eldöntsük, hogy BCNF-e, elég az F-beli funkcionális függőségek vizsgálata. (Nem bizonyítjuk.)

Példa: Az előző szakasz *ÁRU*, *MENNYISÉG*, *BEVÉT*, *BEFIZ* sémái mind BCNF alakúak, ami a definíció alapján könnyen ellenőrizhető.

Tétel: Ha egy séma BCNF alakú, akkor 3NF is.

Bizonyítás: A két definíció közvetlen következménye.

Tétel: A BCNF sémákra illeszkedő relációk nem tartalmaznak redundanciát (legalábbis funkcionális függőségek következtében).

Következmény: Emiatt egyetlen attribútum értékét sem lehet kikövetkeztetni más attribútumok értékeinek ismeretében, ismert funkcionális függőség alapján.

Bizonyítás: Indirekt. T. f. h. a séma BCNF, de mégis van egy rá illeszkedő relációban redundancia. Ez azt jelenti, hogy van benne olyan két sor, t és t', hogy egy A attribútum értékét a t sor értékei és a sémán értelmezett funkcionális függőségek alapján a t' sorban nem írhatjuk be tetszőlegesen, és Y ráadásul nem üres. Vizsgáljuk meg az alábbi relációt:

	X	Y	A
t	x	$y1$	a
t'	x	$y2$	?

Az X és az Y attribútumhalmazokat jelölje ki az, hogy t és t'-ben az X értékei mind azonosak, míg léteznek olyan attribútumok is, amelyek t-n és t'-n különböznek. Ez utóbbiak az Y attribútumok, tehát $y1 \neq y2$. T. f. h. az attribútumok között definiált függőségi kapcsolatok miatt a ? helyére kötelezően a-t kell írnunk. Ez azt jelenti, hogy léteznie kell egy $Z \rightarrow A$ függőségnek, ahol nyilván $Z \subseteq X$. Viszont Z nem lehet szuperkulcs, mert akkor a t és t' soroknak azonosaknak kellene lenniük, ami ellentmond annak a feltételezésnek, hogy $y1 \neq y2$. Ha pedig Z nem szuperkulcs, akkor a $Z \rightarrow A$ függőség ellentmond a BCNF definíciójának.

A normálforma definíciókból és a viszonyukból az következik, hogy egy reláció redundanciáját (funkcionális függések következtében) az okozhatja, ha az attribútumai tranzitívan függenek a kulcs(ok)tól. A 3NF a másodlagos attribútumok tranzitív (és egyúttal részleges) függését zárja ki, a BCNF pedig az elsődleges attribútumokét is.

Ez és az előző tétel tehát azt sugallja, hogy a relációs adatbázisok tervezése során célszerű BCNF sémákat kialakítani. Hiszen ekkor – ha az attribútumaink között csak funkcionális függőségekkel leírható függőségi kapcsolatok vannak – a relációink redundanciamentesek lesznek, ami lényegesen megkönnyíti az egyes relációk (táblák) tartalmának módosítását végző alkalmazások megírását, ill. a hatékonyság biztosítását. Tegyük fel ugyanis, hogy nem bízhatunk a redundanciamentességben: ekkor minden egyes érték bevitele előtt ellenőrizni kell(ene), hogy a relációban már meglévő elemek és az új elem együttesen nincs-e ellentmondásban valamely ismert kényszerrel (jelen esetben funkcionális függőséggel). Az ellenőrzés ugyan elvégeztethető egy adatbázis alkalmazással, kliens oldalon vagy magával az adatbáziskezelő-rendszerrel is, de bármelyik is igen költséges lehet, különösen, ahogy egyre nagyobb mennyiségű adatot szeretnénk kezelni. Ezzel szemben BCNF sémák esetén elég csak a kulcsattribútumok értékeinek egyediségét biztosítani (ami pl. indexeléssel hatékonyan támogatható is), a többi attribútum értékét már tetszőlegesen vihetjük be a relációba. A redundancia minél alacsonyabb szinten tartása tehát kritikus az ún. tranzakciókezelő (manapság leginkább On-Line Transaction Processing, OLTP) rendszereknél.[4] Ennek tipikus példái a repülőtéri helyfoglaló rendszer, banki átutalásokat végző rendszerek vagy éppen egy hallgatói tanulmányi rendszer (pl. Neptun).

A valóság ezzel szemben az, hogy még további szempontok is léteznek a relációs sématervezésnél és a relációs adatbázisok üzemeltetésénél, amelyeket – bár még nem ismerünk – majd figyelembe kell vennünk. Emiatt nem lesz mód arra, hogy mindig BCNF sémákat alakítsunk ki.

A gyakorlatban ritkán dolgozunk egyetlen sémával, így esetenként egész sor, egy adott adatbázishoz tartozó sémáról kell megállapítani, hogy milyen normálformában található. Nem meglepő módon:

Definíció: Egy adatbázis BCNF (3NF, 2NF, 1NF) alakú, ha a benne található összes relációs séma rendre legalább BCNF (3NF, 2NF, 1NF).

9.2.5. Veszteségmentes felbontás (lossless decomposition)

Definíció: Egy R relációs sémának $\rho = \{R_1, R_2, \ldots, R_k\}$ egy *felbontás*a, ha $R_1 \cup R_2 \cup \ldots \cup R_k = R$.

A felbontás célja általában az, hogy ezáltal a részsémák valamely magasabb normálformába kerüljenek. Egy séma felbontásával nyilván a sémára illeszkedő relációk is felbomlanak. Itt azonban körültekintően kell eljárnunk, mert egy reláció ötletszerű felbontásakor információt is veszíthetünk. Ez abban nyilvánul meg, hogy később nem tudjuk a részrelációkból (azaz a részsémákra vetített relációkból) az eredeti relációt – és ezzel együtt annak eredeti információtartalmát – helyreállítani. A helyreállítás itt a részrelációk természetes illesztését jelenti, ha ez nem adja vissza az eredetit, akkor más mód nincsen rá.

Példa: Legyen az $R(A, B, C)$ sémán egyetlen funkcionális függőség értelmezve: $C \to A$. Vizsgáljuk meg a $\rho_1 = (AB, BC)$ és a $\rho_2 = (AC, BC)$ felbontásokat az alábbi reláción:

[4] Az OLTP rendszerekkel szemben az ún. döntéstámogató rendszerekre viszont a ritkán módosuló adatok, nagy tömegű, bonyolult és változatos lekérdezések jellemzőek. Ilyen esetekben a lekérdezés sebességére kell optimalizálni és emiatt kifejezetten ellenjavallt lehet a normalizált sémák alkalmazása. Ld. analitikus alkalmazások, dimenziós modellezés, adattárházak.

$R(A,B,C)$

A	B	C
a	c	e
a	d	f
b	c	g
b	d	h

$R_1'(A,B)$

A	B
a	c
a	d
b	c
b	d

$R_2'(B,C)$

B	C
c	e
d	f
c	g
d	h

$R'(A,B,C)$ó $= R_1' \bowtie R_2'$

A	B	C
a	c	e
a	c	g
a	d	f
a	d	h
b	c	e
b	c	g
b	d	f
b	d	h

$R(A,B,C)$

A	B	C
a	c	e
a	d	f
b	c	g
b	d	h

$R_1''(A,C)$

A	C
a	e
a	f
b	g
b	h

$R_2''(B,C)$

B	C
c	e
d	f
c	g
d	h

$R''(A,B,C_2) = R_1'' \bowtie R''$

A	B	C
a	c	e
a	d	f
b	c	g
b	d	h

Láthatóan $R' \neq R$, miközben $R'' = R$. Tehát az R'-höz tartozó ρ_1 felbontásnak gyakorlati haszna aligha van, hiszen a felbontás következtében az eredeti relációt többé nem tudjuk helyreállítani. Mivel ezzel információt veszítettünk, ezért ezt úgy fejezzük ki, hogy ρ_1 *veszteséges*. A ρ_2 felbontás viszont használható lehet, ha bebizonyosodik, hogy minden $r(R, F)$ esetén hasonlóan viselkedik. Utóbbi esetben ρ_2-re azt mondjuk, hogy *veszteségmentes*.

Tanulság: amikor relációs adatbázis sémákat tervezünk, nem elegendő csupán a minél magasabb normálformákra, azaz minél kevesebb redundanciára törekedni. Egyidejűleg a veszteségmentesség biztosítása feltétlen követelmény, hiszen semmit sem ér egy kevés redundanciát tartalmazó adatbázis, amelyből a modellezett valósággal nem összhangban levő, ill. annak ellentmondó adatokat/információt is ki lehet olvasni!

Definíció: Az R relációs séma egy $\rho(R_1, R_2, \ldots, R_n)$ *felbontását veszteségmentesnek* mondjuk, ha $\forall r(R)$ $\pi_{R_1}(r) \bowtie \pi_{R_2}(r) \bowtie \ldots \bowtie \pi_{R_n}(r) = r$.

Legyen $m_\rho(r) \equiv \pi_{R_1}(r) \bowtie \pi_{R_2}(r) \bowtie \ldots \bowtie \pi_{R_n}(r)$ (*project-join mapping*).

Tétel: Adott egy R séma és egy $\rho(R_1, R_2, \ldots, R_n)$ felbontása. $\forall r(R)$ $r \subseteq m_\rho(r)$.

Bizonyítás: Vegyünk egy tetszőleges $t \in r$ sort. Képezzük t vetületeit az R_i részsémákra, legyen ez $t[R_i]$, amely nyilván eleme az i-edik részrelációnak. Ez a vetület nem változik $m_\rho(r)$-ben sem, és a természetes illesztés tulajdonságai miatt $m_\rho(r)$ valamelyik sorában mindegyik $t[R_i]$, $i=1,\ldots n$ meg fog jelenni. Így $t \in m_\rho(r)$.

A tétel állítása más szavakkal: tetszőleges (tehát nemcsak veszteségmentes!) felbontás után a természetes illesztés eredményeképpen sorok nem tűnhetnek el, csak újak keletkezhetnek.

Tétel: Adott egy R séma és R-nek egy $\rho(R_1, R_2, \ldots, R_n)$ veszteségmentes felbontása. Legyen $\sigma(S_1, S_2, \ldots, S_m)$ valamely $R_i \in \rho$ részsémának szintén veszteségmentes felbontása. Ekkor a $\tau(R_1, R_2, \ldots, R_{i-1}, R_{i+1}, \ldots, R_n, S_1, S_2, \ldots, S_m)$ R-nek szintén veszteségmentes felbontása.

Bizonyítás: A join asszociativitását kihasználva triviális.

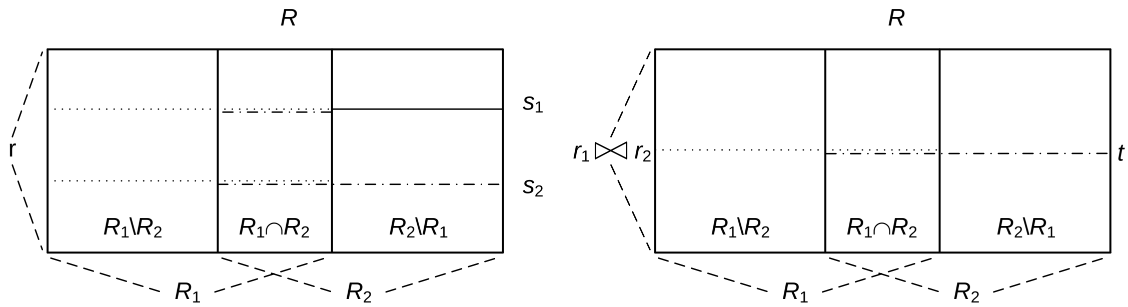
Tétel: Adott egy R séma és R-nek egy $\rho(R_1, R_2, \ldots, R_n)$ veszteségmentes felbontása. Tetszőleges $\tau \supseteq \rho$ esetén τ is veszteségmentes.

Bizonyítás: Ismét a join asszociativitását használjuk fel.
τ veszteségmentes, ha $\forall r(R)$ relációra $r = m_\tau(r)$.
$m_\tau(r) = m_\rho(r) \bowtie \pi_{R_k}(r) \bowtie \pi_{R_l}(r) \bowtie \ldots \bowtie \pi_{R_m}(r)$, ahol $R_k, R_l, \ldots, R_m \in \tau$, de $\notin \rho$.
Készítsük el ezért $m_\rho(r)$-et, melyre nyilván $m_\rho(r) = r$. Vegyünk ezután egy olyan R_i sémát, melyre $R_i \in \tau$, de $R_i \notin \rho$. Képezzük $m_\rho(r) \bowtie \pi_{R_i}(r)$-t. Mivel $m_\rho(r) = r$, ezért $m_\rho(r) \bowtie \pi_{R_i}(r) \equiv r \bowtie \pi_{R_i}(r) \equiv r$.
E miatt $m_\rho(r) \bowtie \pi_{R_k}(r) \bowtie \pi_{R_l}(r) \bowtie \ldots \bowtie \pi_{R_m}(r) = r$, tehát $m_\tau(r) = r$, azaz τ veszteségmentes.

Tudjuk most már, hogy vannak „jó” és „rossz” sémafelbontások, de egyáltalán nem világos, hogy ennek az oka a reláció elemeiben vagy a séma szerkezetében (értsd: a sémán értelmezett függőségek rendszerében) keresendő. Konkrét esetben természetesen mindig kipróbálható, hogy egy felbontás melyik kategóriába esik – mint a fenti példában –, sématervezéskor azonban ez a megközelítés nyilván használhatatlan, hiszen tervezési időben nem ismerhetjük a sémára a jövőben illeszkedő példányokat.
Szerencsére erre nincs szükség, megmutatható, hogy egy felbontás veszteségmentessége vagy veszteségessége kizárólag a relációs sémán és a sémán értelmezett függőségeken múlik. Természetesen ez csak akkor igaz, ha a reláció elemei nincsenek ellentmondásban a sémán értelmezett függőségekkel! Ebben az esetben a séma vizsgálata választ ad egy felbontás veszteségességére. Erre szolgál a következő tétel.

Tétel: Adott az R séma, a séma attribútumain értelmezett F függőséghalmaz és egy $\rho = (R_1, R_2)$ felbontás. ρ veszteségmentes $\Leftrightarrow$ $(R_1 \cap R_2) \to (R_1 \setminus R_2) \in F^+$ vagy $(R_1 \cap R_2) \to (R_2 \setminus R_1) \in F^+$.

Bizonyítás:
Visszafelé: Azt akarjuk belátni $(R_1 \cap R_2) \to (R_1 \setminus R_2) \in F^+$ vagy $(R_1 \cap R_2) \to (R_2 \setminus R_1) \in F^+$ alapján, hogy tetszőleges $t \in r_1 \bowtie r_2$ egyúttal r-nek is eleme.



Biztos, hogy $\exists s_1 \in r$, hogy $s_1[R_1] = t[R_1]$[5] (pontozás), továbbá

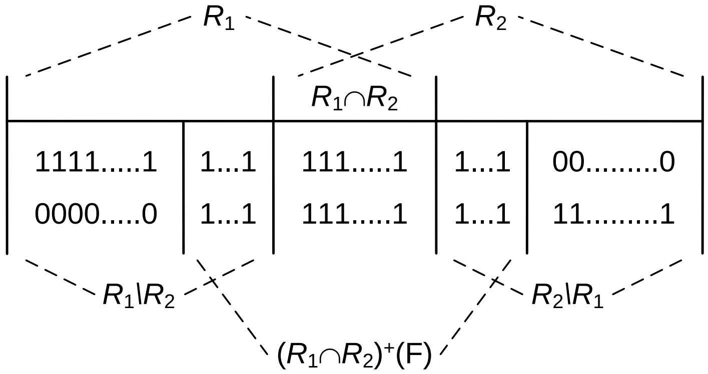
$\exists s_2 \in r$, hogy $s_2[R_2] = t[R_2]$ (pont-vonal).

Viszont $t[R_1 \cap R_2] = s_1[R_1 \cap R_2] = s_2[R_1 \cap R_2]$.

T. f. h. a kétféle lehetőség közül $(R_1 \cap R_2) \to (R_1 \setminus R_2)$ áll fenn, amely szerint, ha $s_1[R_1 \cap R_2] = s_2[R_1 \cap R_2]$, akkor kötelezően $s_1[R_1 \setminus R_2] = s_2[R_1 \setminus R_2]$. Emiatt – mint a fenti ábrán is látható – $s_2 = t$, azaz megtaláltuk azt az r-beli sort, amelyik egy tetszőleges $t \in r_1 \bowtie r_2$-vel megegyezik.

Előre: Indirekt módon bizonyítjuk. T. f. h. sem $(R_1 \cap R_2) \to (R_1 \setminus R_2) \in F^+$ sem $(R_1 \cap R_2) \to (R_2 \setminus R_1) \in F^+$ nem áll fenn. Ekkor ellentmondásra kell jutnunk a feltétellel. Ehhez elegendő egyetlen olyan, az adott sémára és függőségekre illeszkedő relációt találni, amely veszteségesen bontható fel.

Az alábbi ábrán látható reláció ilyen tulajdonságú.



Látható, hogy sem $(R_1 \cap R_2) \to (R_1 \setminus R_2)$ sem $(R_1 \cap R_2) \to (R_2 \setminus R_1)$ nem áll fenn. Azt kell még belátni, hogy kielégíti F minden függőségét. Valamely $V \to W \in F$-re:

- ha V „kilóg” $(R_1 \cap R_2)^+$-ből, akkor nincs két azonos sor, tehát $V \to W$ teljesül bármely W-re.
- ha $V \subseteq (R_1 \cap R_2)^+$, akkor $(R_1 \cap R_2) \to V$, $V \to W$ miatt $(R_1 \cap R_2) \to W$, melynek feltétele, hogy $W \subseteq (R_1 \cap R_2)^+$. Így W értékei is azonosak, tehát $V \to W$ ekkor is teljesül.

Utolsó lépésként ellenőrizhető, hogy $r_1 \bowtie r_2 \neq r$, tehát a felbontás valóban veszteséges.

Példa: Tekintsük újra a 9.2.5. szakaszban szereplő $R(A, B, C)$ sémát, amelynek F függéshalmaza csak a $C \to A$ függőséget tartalmazza. Vizsgáljuk meg rendre a $\rho_1 = (AB, BC)$ és a $\rho_2 = (AC, BC)$ felbontásokat a fenti tétel alapján!

ρ_1:

$AB \cap BC = B, AB \setminus BC = A, B \to A \notin F^+$, hiszen nyilván nem következik $C \to A$-ból.

$AB \cap BC = B, BC \setminus AB = C, B \to C \notin F^+$, hiszen nyilván nem következik $C \to A$-ból.

Tehát ρ_1 veszteséges.

ρ_2:

$AC \cap BC = C, AC \setminus BC = A, C \to A \in F^+$ (triviális), tehát ρ_2 veszteségmentes.

$(AC \cap BC = C, BC \setminus AC = B, C \to B \notin F^+)$

[5] A természetes illesztés definíciója miatt.

Megjegyzés: A tételt arra is felhasználhatjuk, hogy segítségével veszteségmentes felbontásokat készítsünk $\rho = (R_1, R_2)$ formába. Ehhez bármely $X \rightarrow Y \in F$ nemtriviális (vagy akár $X \rightarrow Y \in F^+$) függés alapján, ahol X és Y diszjunktak, legyen

- $R_1 = XY$, ill.
- $R_2 = R \setminus Y$.

Könnyen ellenőrizhető, hogy ekkor az $(R_1 \cap R_2) \rightarrow (R_1 \setminus R_2)$ pontosan $X \rightarrow Y$ formában fog teljesülni, azaz ρ veszteségmentes lesz.

Tétel: Adott $R(XYZ)$ és funkcionális függőségek F halmaza esetén, ahol X, Y és Z (páronként diszjunkt) attribútumhalmazok is lehetnek $\rho = (XY, XZ)$ pontosan akkor veszteségmentes, ha $X \rightarrow Y \in F^+$ vagy $X \rightarrow Z \in \mathrm{F}^+$.

Bizonyítás: az előző tétel közvetlen következménye.

Az előző tételek hatékony analízis és tervezési módszert adtak arra, hogyan lehet egy sémát veszteségmentesen két részre bontani. Az alábbi módszer tetszőleges számú részre bontott részsémáról lehetővé teszi a veszteségmentesség eldöntését.
Adott egy $R(A_1, A_2, \ldots, A_k)$ séma, a sémán értelmezett F függőségek és R-nek egy $\rho(R_1, R_2, \ldots, R_n)$ felbontása. Táblázatot készítünk n sorral és k oszloppal. Az oszlopokat a séma attribútumainak, a sorait a részsémáknak feleltetjük meg. Kiindulási állapotként úgy töltjük ki a táblázatot, hogy az i-edik sor j-edik oszlopába

- a-t írunk, ha $A_j \in R_i$,
- b_i-t írunk, ha $A_j \notin R_i$.

Ezután mindaddig módosítjuk a táblázat elemeit az F függőségeinek figyelembe vételével, az alábbiak szerint, ameddig a táblázat változik:
Vegyük egy tetszőleges $X \rightarrow Y \in F$ függést. Ha létezik két olyan sor a táblázatban, amely X-en azonos, akkor Y-on is egyenlővé tesszük őket, mégpedig

- ha valahol a-t találunk, akkor a másik sor azonos oszlopának eleme is legyen a,
- ha nem a egyik sem, akkor b_i-t és b_l-t tegyük egyenlővé tetszőlegesen.

Példa: Legyen $R(S, A, I, P)$, $F = \{S \rightarrow A, SI \rightarrow P, P \rightarrow S\}$, $\rho(SA, SI, IP, PS)$.
Kezdőállapot:

	S	A	I	P
SA	a	a	b_1	b_1
SI	a	b_2	a	b_2
IP	b_3	b_3	a	a
PS	a	b_4	b_4	a

Végállapot:

	S	A	I	P
SA	a	a	b_1	b_1
SI	a	a	a	a
IP	a	a	a	a
PS	a	a	b_4	a

Tétel: a ρ felbontás veszteségmentes $\Leftrightarrow$ van csupa a-ból álló sor.

Bizonyítás előre:
A táblázatot tekintsük olyan $r(R)$ relációnak, ahol a-kat és b_i-ket $\mathrm{DOM}(A_j)$-ből választottuk. A végállapotban r kielégíti F összes függőségét, hiszen az algoritmus pontosan az F-beli függőségek sértéseit korrigálja. Bontsuk fel r-et is ρ-nak megfelelően, ekkor a táblázat kezdőállapotának konstrukciója miatt igaz, hogy

$\pi_{R_i}(r)$-ben lesz olyan sor, amely csupa a-ból áll, minden i-re. Tekintve, hogy $m_\rho(r) \equiv \pi_{R_1}(r) \bowtie \pi_{R_2}(r) \bowtie \ldots \bowtie \pi_{R_n}(r)$, így $m_\rho(r)$ -ben biztosan keletkezik olyan sor, amely csak a-kat tartalmaz. Ha ρ veszteségmentes, akkor $m_\rho(r) = r$, minden r-re, tehát a végállapotbeli táblázat $\equiv m_\rho(r)$, azaz tartalmazza a csupa a-ból álló sort.

Bizonyítás visszafelé: nem bizonyítjuk.

9.2.6. Függőségőrző felbontások

Vizsgáljuk meg a következő példát! Adott az R(*VÁROS*, *ÚT*, *IR_SZÁM*) = (V, U, I) séma, amelyen – a valósággal jó összhangban – az $F = \{VU \to I,\ I \to V\}$ függőségeket értelmeztük. Készítsük el a séma $\rho = (UI, VI)$ felbontását! (Könnyen ellenőrizhető az előző tétel segítségével, hogy ρ veszteségmentes, tehát tetszőleges, (R, F)-re illeszkedő r reláció esetén r helyreállítható természetes illesztéssel a részsémákra vonatkozó vetületeiből.)

Egy adatrögzítő a részsémákba az alábbi adatokat vitte be:

r_1

ÚT	*IR_SZÁM*
Kossuth	2142
Kossuth	2144

r_2

VÁROS	*IR_SZÁM*
Baja	2142
Baja	2144

Helyreállítva az eredeti relációt az alábbi sorokat kapjuk:

ÚT	*IR_SZÁM*	*VÁROS*
Kossuth	2142	Baja
Kossuth	2144	Baja

Az eredménnyel az a probléma, hogy nyilvánvalóan nincs összhangban a feltételezett $VU \to I$ függőséggel, amely szerint a valóságunk „működik”, tehát azzal, hogy egy városban egy utcának csak egyetlen irányítószáma lehet. A részrelációk természetes illesztése „hamis” eredményre vezetett. A hiba azonban nem az illesztésnél van, hiszen a felbontás veszteségmentes, tehát az „eredeti” relációt kell visszakapnunk. A valódi ok az, hogy *nem is létezett* „eredeti” reláció, azaz nem tudunk olyan r relációt konstruálni, amely illeszkedik (R, F)-re, és ugyanakkor a fenti r_1 és r_2 vetületei lennének.

Egy „jó” felbontás esetén biztosak akarunk lenni abban, hogy ha a részrelációkba csak szemantikailag helyes (adott esetben tehát a felvett funkcionális függéseknek megfelelő) adatok kerülnek bele, akkor a relációk valamely illesztése eredményeként sem fogunk majd az adatbázisból olyan sorokat kiolvasni, amelyek a felvett – és az adatok jelentését valamilyen szinten leíró – funkcionális függéseknek ellentmondanak.

Másrészt mindaddig, amíg a séma egyben van, minden egyes sor bevitele előtt elvileg lehetőség van széleskörűen ellenőrizni, hogy a bevitt adatok bizonyos *kényszerfeltételeknek* (integrity constraints) megfelelnek-e. Ilyenek lehetnek pl. meghatározott függőségi feltételek is. Sajnos, amint az R sémát felbontottuk, már nem feltétlenül tudjuk az eredeti függőségeket alkalmazni a részrelációkban annak elkerülésére, hogy „hamis” adatok kerüljenek be az adatbázisba, legfeljebb a függőségeknek a részsémákra való „vetületeit”. Ez egyes esetekben elegendő lehet (ezt részesítjük előnyben), más esetekben pedig nem.

Definíció: Adott az R séma attribútumain értelmezett függőségek F halmaza. A *függőségeknek egy $Z \subset R$ attribútumhalmazra való vetítése* az a $\pi_Z(F)$ függőséghalmaz, amelyre $\pi_Z(F) = \{X \rightarrow Y \mid X \rightarrow Y \in F^+$ és $XY \subseteq Z\}$.

Fontos, hogy a vetített függőséghalmazban benne legyen minden olyan függés, amely F-ből következik, ill. levezethető. Pl. egy $R(ABC)$ sémán értelmezett $F = \{A \rightarrow B, B \rightarrow C\}$ függőséghalmaz esetén $\pi_{AC}(F) = \{A \rightarrow C\}$, mert a tranzitivitási axióma miatt $A \rightarrow C \in F^+$.

Ezek a vetített függőségek bizonyos esetekben elegendőek lehetnek a fenti céljainkra. Ehhez az kell, hogy az R_i részsémákra vetített függőségek ugyanazt az információt hordozzák, mint az eredeti F függéshalmaz.

Definíció: Adott egy R relációs séma és egy, a sémán értelmezett F függéshalmaz. A séma $\rho = \{R_1, R_2, \ldots, R_k\}$ felbontása *függőségőrző*, ha $\{\pi_{R_i}(F)\} \models F$.

Egy relációs séma veszteségmentes, de nem függőségőrző felbontása eredményezheti tehát, hogy a részsémák összességén sem tudjuk többé az eredeti sémában érvényes mindegyik függőséget alkalmazni. Ennek következtében a részrelációink illesztésével kapott lekérdezési eredményekbe nem megengedett adatok is bekerülhetnek. Ez ellen csak úgy védekezhetnénk, ha minden egyes új sor bevitele előtt előállítanánk az eredeti relációt, és ellenőriznénk azon a függőségi viszonyok fennállását. Ez érezhetően nagyon költségessé tenné sok sort tartalmazó relációk esetén az új sorok bevitelét. Ezt elkerülendő célszerű tehát olyan sémafelbontásokat készíteni, amelyek függőségőrzők. Ebben az esetben ugyanis ha a részrelációkba bevitt adatok valóban megfelelnek a tervezett funkcionális függéseknek (pl. nincs téves adatbevitel), akkor garantálható, hogy a lekérdezések eredményei sem fognak a függéseknek (tehát az adatszemantikának) ellentmondani.

A definíciókon alapuló kézenfekvő, de nehézkesen alkalmazható lehetőséget nyújt egy sémafelbontás függőségőrző tulajdonságának tesztelésére az alábbi algoritmus:

- elkészítjük F^+-t,
- vetítjük F^+ mindegyik függőségét minden részsémára,
- meghatározzuk a vetített függőségek lezárását,
- ha ez azonos F^+-tal, akkor a felbontás függőségőrző, ellenkező esetben biztosan nem az.

Megj.: Egy felbontás lehet

- veszteségmentes és függőségőrző,
- veszteségmentes és nem függőségőrző,
- veszteséges és függőségőrző,
- veszteséges és nem függőségőrző.

9.2.7. Sémadekompozíció adott normálformába

Az előzőek alapján belátható, hogy gyakorlati szempontból az olyan sémafelbontások bírnak jelentőséggel, amelyeknél biztosítható, hogy az eljárás eredménye meghatározott tulajdonságokat mutató sémák halmaza lesz. A legfontosabb tulajdonságok, amelyeket figyelembe kell venni (nem feltétlenül fontossági sorrendben):

- adott normálforma elérése,
- veszteségmentesség,
- függőségőrző tulajdonság.

Adott tulajdonságú felbontások létezését állítják a következő tételek:

Tétel: Minden R relációs séma és a sémán értelmezett F függéshalmaz esetén $\exists\rho$ sémafelbontás, amely veszteségmentes és függőségőrző, továbbá $\forall R_i \in \rho$-ra R_i 3NF tulajdonságú.

Bizonyítás: konstruktív. Képezzük az adott függéshalmaz egy minimális fedését, legyen ez G. Ha $G = \{X_1 \to A_1,\ X_2 \to A_2,\ \ldots,\ X_n \to A_n\}$ alakú, akkor a felbontás legyen $\rho = \{X_1A_1, X_2A_2, \ldots, X_nA_n\} \cup \{K\}$, ahol K az R séma egy kulcsa.
A ρ felbontás természetesen függőségőrző, mert a részsémákra vetített függőségek közül egy pontosan megegyezik az egyik G-beli függőséggel, így a vetített függőségek lezártja nyilván megegyezik G lezártjával.
A 3NF tulajdonságot indirekt módon bizonyítjuk. T. f. h. $\exists Y \to B \in G^+$, $YB \subseteq X_iA_i$, valamely i-re, amely az i-edik séma 3NF tulajdonságát sérti: azaz $B \notin Y$, Y nem szuperkulcsa R_i-nek és B nem elsődleges attribútum.

- Ha $B = A_i$, akkor $Y \subseteq X_i$ és mivel Y nem szuperkulcsa X_iA_i-nak, így $Y \subset X_i$. De ekkor a feltételezett $Y \to B$ miatt $Y \to A_i$ ugyanazt fejezi ki, mint $X_i \to A_i$, így G-ben $X_i \to A_i$ helyett $Y \to A_i$,-nek kellett volna szerepelnie.
- Most t. f. h. $B \neq A_i$. Mivel X_i szuperkulcs X_iA_i-ben az ismert $X_i \to A_i$ függőség miatt, ezért $\exists Z \subseteq X_i$, amely kulcs. $B \neq A_i$ miatt $B \in X_i$, de $B \notin Z$, mert feltételeztük, hogy B nem elsődleges attribútum. Így X_i feltétlenül bővebb Z-nél legalább B-vel. Ekkor viszont az $X_i \to A_i$ függőség helyettesíthető G-ben $Z \to A_i$-vel.

Meg kell még vizsgálnunk, hogy a K kulcsként (esetleg) megjelenő n + 1-edik séma is 3NF-e. Ha $Y \to B$ nemtriviális függés és $YB \subseteq K$, akkor K nem lehet minimális, hiszen belőle B elhagyható.
Láthatóan mindhárom esetben ellentmondásra jutottunk a feltétellel, az első kettőben azzal, hogy G minimális függéshalmaz, a harmadikban a K kulcs minimális tulajdonságával. Tehát mindegyik részséma 3NF.
Lássuk be most ρ veszteségmentességét. Ehhez a táblázatos tesztet fogjuk használni. Megmutatjuk, hogy a táblázat végállapotában a K sora csupa a-t tartalmaz.
Képezzük először K^+-t a tanult algoritmussal. Ennek során rendre a $B_1, B_2, \ldots, B_k$ attribútumokkal bővítjük $K^{(0)} = K$-t, ebben a sorrendben. Nyilván igaz, hogy $K \cup \{B_1, B_2, \ldots, B_k\} = R$. Megmutatható, hogy a táblázat módosítása során B_i-k sorrendjében lehetőségünk van K sorába a-kat írni. Ezt i-re vonatkozó teljes indukcióval láthatjuk be.
$i = 0$ megfelel annak, hogy a K sorában a K attribútumainak oszlopaiban a-k vannak.
T. f. h. $i-1$-re még működik, majd $K^{(i)}$ előállítása során adjuk hozzá $K^{(i-1)}$-hez B_i-t valamely $Y \to B_i \in G$ függőség miatt. K^+ konstrukciója miatt biztos, hogy $Y \subseteq K^{(i-1)}$, ahol $K^{(i-1)} = K \cup \{B_1, B_2, \ldots, B_{i-1}\}$. Ekkor a táblázatban a K és az YB_i sémák sorainak attribútumértékei az Y oszlopaiban mind megegyeznek, mégpedig mind a, mert

- K sorában K attribútumainak oszlopai a táblázat konstrukciója miatt mindenkor a-k, $B_1, B_2, \ldots, B_{i-1}$ oszlopai pedig már a-k az indukciós feltétel miatt,

- YB_i sorában Y attribútumainak oszlopaiban a táblázat konstrukciója miatt mindenkor a-k állnak.

Mivel YB_i sorában B_i oszlopában a áll, ezért jogosan írhatunk K sorába B_i oszlopába is a-t.

Megjegyzések:

1. Vegyük észre, hogy a K kulcs csak a veszteségmentes felbontáshoz kell. Ha a sémafelbontásból kihagyjuk, csak a függőségőrző és 3NF tulajdonság garantálható.
2. ρ konstrukciója során kiderülhet, hogy valamely R_i részséma tartalmazza R valamely kulcsát. Ekkor felesleges egy $i+1$-edik sémát külön definiálni egy kulcs számára, hiszen a veszteségmentességet biztosító minden mechanizmus ekkor is működik.

Példa („mini Neptun rendszer”):

Egyetemi kurzusok adatait szeretnénk tárolni, a séma $R(K, T, I, H, D, J)$ séma. A függőségek:

- $K \to T$: minden kurzushoz csak egy tanár tartozhat.
- $IH \to K$: egy időpontban egy helyszínen legfeljebb egy kurzus lehet.
- $IT \to H$: egy időpontban egy tanár csak egy helyszínen lehet.
- $KD \to J$: egy kurzusban egy diák csak egy jegyet kaphat.
- $ID \to H$: egy időpontban egy diák csak egy helyszínen lehet.

A $\rho = (KT, IHK, ITH, KDJ, IDH)$ sémafelbontás az 1. megjegyzés miatt 3NF, függőségőrző felbontás. Ha azt szeretnénk, hogy a felbontás veszteségmentes is legyen, akkor keressük meg R (egyik) kulcsát! Azt gondoljuk, hogy az ID attribútumhalmaz kulcs. Határozzuk meg ezért a lezárását:
$ID^{(0)} = ID$, $ID^{(1)} = IDH$, $ID^{(2)} = IDHK$, $ID^{(3)} = IDHKJ$, $ID^{(4)} = IDHKJT$, tehát $ID^+ = IDHKJT$, az attribútumok teljes halmaza, így ID legalábbis szuperkulcs. Ugyanakkor minimális is, mert F-ben sem I, sem D egyedül semmilyen más attribútumot nem határoz meg, $I^+ = I, D^+ = D$, vagyis az DI valóban kulcs.
Ha hozzávesszük a ρ felbontáshoz a DI sémát a 2. megjegyzés szerint, akkor az eredmény ρ-val azonos lesz, hiszen ρ utolsó sémája tartalmazza a DI attribútumokat. Tehát valójában ρ veszteségmentes is volt.

Tétel: Minden, legalább 1NF R sémának létezik veszteségmentes felbontása BCNF sémákba.

Bizonyítás: Iteratívan készítjük el a felbontást. Az iteráció minden fázisában igaz lesz, hogy a pillanatnyi felbontás veszteségmentes.

1. Ha R az adott F függőségek mellett BCNF, akkor nincs tennivaló, készen vagyunk.
2. Ha R nem BCNF, akkor $\exists X \to A \in F^+$, ami megsérti a BCNF tulajdonságokat, azaz $A \not\in X$ és X nem szuperkulcsa R-nek. Legyen ekkor a felbontás $\rho_1(R_1, R_2)$, $R_1 = XA$, $R_2 = R \setminus A$, melyről belátható, hogy
   - veszteségmentes felbontás és
   - R_1 és R_2 is kevesebb attribútumot tartalmaz, mint R. R_2 nyilván kisebb R-nél, hiszen az A attribútumokat nem tartalmazza. R_1 pedig azért kisebb, mert különben $X \to A$ nem sérthetné a BCNF tulajdonságot.

3. Vizsgáljuk meg, hogy R_1, ill. R_2 BCNF-e. Ehhez meg kell határoznunk a részsémákra vetített függőségeket. Ha mindkettő BCNF, akkor készen vagyunk.
4. Ha R_1, R_2 között van, amelyik nem BCNF, akkor arra a sémára ismételjük meg az eljárást a 2. ponttól. Mivel egy veszteségmentes felbontás veszteségmentes felbontása is veszteségmentes, így az iteráció tetszőleges mélységben folytatható. Másrészről, mivel a legfeljebb két attribútumot tartalmazó sémák mind BCNF-ek, így előbb-utóbb a felbontás részei BCNF sémák lesznek.

Példa: Legyen adott a *NYELV*(*DIÁKKÓD*, *DIÁKNÉV*, *OFŐNÖK*, *OFTEL*, *NYELV*, *FÉLÉVKÓD*, *OSZTÁLYZAT*) relációs séma és a funkcionális függőségek alábbi (nem minimális) halmaza:

$DIÁKKÓD \rightarrow DIÁKNÉV$

$DIÁKKÓD \rightarrow OFŐNÖK$

$OFŐNÖK \rightarrow OFTEL$

$OFTEL \rightarrow OFŐNÖK$

$DIÁKKÓD \rightarrow OFTEL$

$DIÁKKÓD, NYELV, FÉLÉVKÓD \rightarrow OSZTÁLYZAT$

Bontsuk fel BCNF alakú sémákba veszteségmentes dekompozícióval!
A séma egyetlen kulcsa a {*DIÁKKÓD*, *NYELV*, *FÉLÉVKÓD*} attribútumhalmaz. Válasszuk ki az *OFŐNÖK* → *OFTEL* függőséget, és bontsuk fel a *NYELV* relációt két részre az előbbiek szerint:

R_1(*OFŐNÖK*, *OFTEL*)

R_2(*DIÁKKÓD*, *DIÁKNÉV*, *OFŐNÖK*, *NYELV*, *FÉLÉVKÓD*, *OSZTÁLYZAT*)

Az R_1 BCNF alakú, így csak az R_2-t vizsgáljuk tovább.
Ennek függőségei:

$DIÁKKÓD \rightarrow DIÁKNÉV$

$DIÁKKÓD \rightarrow OFŐNÖK$

$DIÁKKÓD, NYELV, FÉLÉVKÓD \rightarrow OSZTÁLYZAT$,

tehát R_2 kulcsa továbbra is {*DIÁKKÓD*, *NYELV*, *FÉLÉVKÓD*}.
R_2 még mindig nem BCNF alakú, mert a *DIÁKKÓD* determináns, de nem szuperkulcs.
Az első két függőséget az alábbi formában is felírhatjuk:

$DIÁKKÓD \rightarrow DIÁKNÉV, OFŐNÖK$

Konstruáljuk meg a következő felbontást ezen függőség alapján:

R_3(*DIÁKKÓD*, *DIÁKNÉV*, *OFŐNÖK*),

R_4(*DIÁKKÓD*, *NYELV*, *FÉLÉVKÓD*, *OSZTÁLYZAT*)

Itt már R_3 és R_4 is BCNF alakú, tehát készen vagyunk. A keresett felbontás az R_1, R_3 és az R_4 sémák együttese.
Megjegyzések:

1. Ez a tétel az egyik oka annak, amiért nem érdemes minden esetben BCNF alakokra törekedni egy relációs adatbázis tervezése során. (A másik az, hogy sok kis relációból általában költségesebb, tehát adott gépen lassúbb egy lekérdezés eredményének összeállítása. Esetleg éppen a lekérdezési válaszidők csökkentése érdekében alkalmanként szándékosan redundanciát építenek bele a relációkba, ld. döntéstámogató/analitikus rendszerek.)
2. Vegyük észre, hogy más függőségeket (ill. más sorrendben) választva a felbontás alapjául az eredményül kapott sémák is más-más attribútumokat tartalmazhatnak.

3. Egy másik, kézenfekvő lehetőség BCNF sémafelbontások előállítására, hogy a 3NF, függőségőrző és veszteségmentes sémák előállítására alkalmas, fentebb leírt algoritmussal előállított felbontásból indulunk ki. Minden részsémára megvizsgáljuk, hogy az BCNF-e. Ha egy séma nem BCNF, akkor két részre bontjuk veszteségmentes dekompozícióval pontosan az imént leírt módszerrel. A módszer előnye, hogy hamarabb eredményre vezethet, mivel garantáltan 3NF sémákból indul ki.
4. A BCNF tulajdonság ellenőrzése igen költséges is lehet. Szerencsére, számos egyszerűsítésre nyílik lehetőség, amelynek a részleteire azonban nem térünk ki.

Könnyű példát mutatni arra, amikor egy séma nem bontható fel veszteségmentesen és függőségőrzően BCNF sémákba.

Vegyük ismét a már ismert $R(VÁROS, ÚT, IR\_SZÁM) = (V, U, I)$ sémát, amelyen most is az $F = \{VU \rightarrow I, I \rightarrow V\}$ függőségeket értelmeztük. Így R kulcsai UI és VU, az $I \rightarrow V$ függőség miatt R nem BCNF. Készítsük el a séma egy valódi, veszteségmentes felbontását! Könnyen ellenőrizhető, hogy $\rho = (UI, VI) = (R_1, R_2)$ az egyetlen lehetőség. A részsémák nyilván BCNF-ek. A vetített függőségek: $\pi_{R_1}(F)$ csak triviális függőségeket tartalmaz, $\pi_{R_2}(F) = \{I \rightarrow V$, triviális függőségek$\}$. Mivel a $VU \rightarrow I$ függőség a sémafelbontás során elveszett, ezért ρ nem függőségőrző.

Az előbbiek alapján a normalizálás elméletét egy másik módon is felépíthetjük: redundanciamentes relációkat akarunk létrehozni függőségőrző és veszteségmentes felbontással. A redundanciamentességéhez az szükséges, hogy minden sémán nemtriviális függés csak szuperkulcstól lehessen (BCNF). De ekkor függőségőrző és veszteségmentes felbontásokat nem feltétlenül tudunk készíteni. Ezért célszerű egy enyhébb normálforma bevezetése is. Ez lesz a 3NF, amely „éppen annyi" redundanciát tartalmaz, hogy mellette függőségőrző és veszteségmentes sémafelbontást lehessen garantálni. A 2NF jelentősége ebben a gondolatkörben marginális.