Az adat-összekapcsolás módszertana a migrációs adatbázis kialakítása során

Ez a tanulmány a Bevándorlási és Állampolgársági Hivataltól (a továbbiakban BÁH), a Közigazgatási és Elektronikus Közszolgáltatások Központi Hivatalától (a továbbiakban KEKKH), az Országos Egészségbiztosítási Pénztártól (a továbbiakban OEP) kapott, külföldi állampolgárok adatait tartalmazó adatbázisok, valamint a 2011. évi népszámlálás során a Magyarországon tartózkodó külföldiekről szóló adatbázis (továbbiakban NÉPSZ) összekapcsolásáról, és az eredmények leírásáról szól.

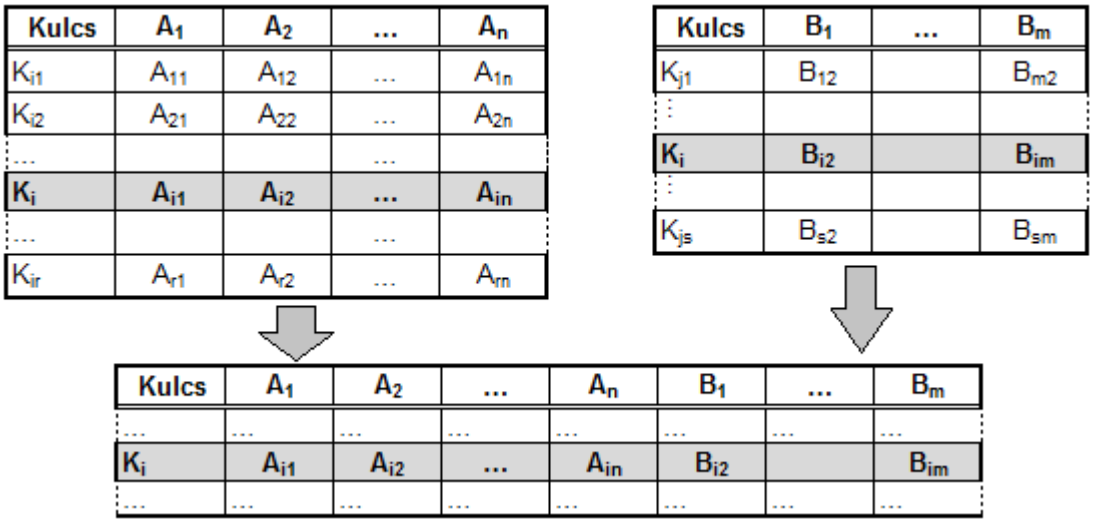
Elméleti áttekintés

Informatikai értelemben adatbázis alatt az adatok és a közöttük levő kapcsolatok valamilyen adatmodell szerint kialakított tárolását értjük. A ma használatos adatbázis-kezelők nagy részében alkalmazott relációs adatmodellt az 1970-es években kezdték kidolgozni. Ennek lényege, hogy a leírni kívánt egységeket (amelyek lehetnek személyek, tárgyak, vállalatok, országok, stb. egy szóval egyedek) több elemi adat azonosítja. Ezek az elemi adatok logikailag kétdimenziós táblázatba szervezhetők, ezeket nevezzük relációknak. A táblázat első sora a reláció fejléce, amely az oszlopok (táblázaton belül egyértelmű) azonosítóit – a reláció attribútumait – tartalmazza. A többi sorban egy adott egyedre vonatkozó adatok szerepelnek. Tekintettel arra, hogy a reláció matematikai értelemben egy halmaz, amelynek elemeit csak egyszer adjuk meg, így nem szerepelhet a táblázatban két azonos sor. Ha ez így van, akkor biztosan található oszlopok olyan összessége, amelyekben előforduló adatok együttese minden sorban különböző. Az ilyen oszlop együttest a reláció kulcsának mondják.

Determinisztikus adat-összekapcsolás

A kapcsolatok leírása a relációs adatmodellben kulcsokon keresztül valósul meg. A legegyszerűbb esetben a kulcs egyetlen oszlop, amely garantáltan minden sorban más értéket vesz fel. Ez lehet egy sorszámozás, vagy valamilyen mesterséges azonosító, mint pl. a TAJ szám, az adószám vagy a személyi szám. Ha léteznek két adattáblában ilyen kulcsok, akkor a mindkettőben előforduló személyek adatai egy új táblázatban előállíthatók. Nem kell ugyanis egyebet tenni, mint az egyik relációban szereplő egyedhez tartozó kulcs értékét kikeresni a másik relációból, és annak többi attribútumát az első mellé másolni, ahogy ezt az 1. ábra mutatja.

1. ábra



A fent leírt eljárás nyilván nem használható, ha a kulcsok különböző tartalmú azonosítók, azaz értékük egy adott táblában egyedi, de két különböző adatbázisból származó relációban nincsenek közös értékeik. Ilyen eset alakul ki, ha az egyik adatbázisban például a személyi igazolvány számát, a másikban az adószámot használták azonosításra. Előfordulhat az is, hogy az adatlistát nem valamilyen adatbázis-kezelő rendszerben vezetik, így nincs, ami betartassa a kulcsokra vonatkozó kényszereket, megszorításokat, vagy esetleg a kulcsok sérülnek, módosulnak az adatátadás során használt exportálási műveletek alatt. Ilyenkor lehetséges, hogy nincs használható kulcs, ezért az összekapcsolásra más utat kell keresni.

Megoldást jelent a valószínűségi alapon történő összekapcsolás (probabilistic matching), amelynek gyökerei az 1950-es évekig visszanyúlnak, de a precíz matematikai leírásával és elemzésével foglalkozó írások a 60-as évek vége óta jelennek meg. Bár a valószínűségi összekapcsolás eljárások egy csoportja, közös ezekben, hogy az adatlistákból olyan attribútumokat kell kiválasztani, amelyek egyezősége (vagy közel azonos értéke) azt jelzi, hogy ugyanannak az egyednek a tulajdonságai jelennek meg mindkét rekordban. Szükséges feltétel tehát, hogy legyenek (kellő számban) közös oszlopok a relációkban, amelyek tartalma és kitöltési módja azonos. Nem működhet az összekapcsolás, ha például az egyik adattáblában egy Név oszlopban szerepel egy ember teljes neve, míg a másikban van külön Vezetéknév és Utónév oszlop, vagy mondjuk a születési dátumot az egyik helyen folytonos 8 karakteres szövegként (19620427), a másikban dátumként (1962.04.27) tárolják. Az ilyen jellegű problémák megszüntetése az esetek többségében megfelelő előkészítő és adattisztító műveleteket igényel.

A sztochasztikus adat-összekapcsolás

Legyen két sokaság, jelölje ezekből vett mintákat A és B, az elemei legyenek a_i és b_j. Feltesszük, hogy vannak A-ban és B-ben közös elemek. Ennek következtében az ezeket leíró rekordok elvileg összekapcsolhatók, ha van bennük elegendő számú közös attribútum, amelyek összehasonlíthatók egymással. A gyakorlatban azonban a hibák, pontatlanságok, kulcsok hiánya miatt nem könnyű megtalálni az összetartozó rekordpárokat, így valójában két rekord vizsgálata után háromféle döntést hozhatunk:

- Az adott hibaszinten a két rekord összekapcsolható.
- Az adott hibaszinten a két rekord nem kapcsolható össze.
- Lehetséges, hogy a két rekord összekapcsolható, de ez az adott hibaszinten nem jelenthető ki.

A hibaszint azt jelenti, hogy a pontatlanságok, elírások miatt célszerű megengedni a rekord egyes mezőiben olyan eltéréseket, amelyek mellett még feltételezhetjük az azonosságot. Ha pl. egy személy címében az utcanév Kossuth Lajos utca, a másik rekordban Kossut Lajos utca, akkor a hiányzó h betű elírásnak tekinthető, így a két cím azonosként kezelhető, ha minden más részében pontos az egyezés. Ugyanakkor Kis László és Kiss László lehet ugyanaz az ember, csak egyik helyen elírták a vezeték nevét, de lehetséges, hogy valóban két különböző személyről van szó. Ezt csak további adatok ismeretében (életkor, cím, anyja neve, stb.) lehet eldönteni, 100%-os bizonyossággal talán még akkor sem. Így az eljárás során hibákat következhetnek be.

Az összekapcsolás végrehajtásához szükség van egy döntési szabályra, amely segítségével egy rekordpárról megmondható, hogy ugyanazt az egyedet írja-e le, vagy sem. 1969 decemberében jelent meg Fellegi Iván és Allan B. Santer cikke, (A theory of record linkage", Journal of the American Statistical Association 64) amelyben ismertetnek egy döntési szabály megkonstruálására használható eljárást. Ez azon alapul, hogy számszerűsíthető a rekordok „hasonlósága", így előírhatók numerikus határértékek, amelyeknél „jobban hasonlító" rekordokat össze kell, és egy másik, amelynél „kevésbé jobban hasonlítókat" nem szabad összekapcsolni. Az eljárás jelentősége az, hogy e két adott határérték mellett minimalizálja azoknak a rekordoknak a számát, amelyek összekapcsolásáról nem lehet egyértelmű döntést hozni. Az Olasz Statisztikai

Hivatal koordinálásával kifejlesztettek egy ingyenesen használható szoftvert, amely alkalmas a Fellegi-Santer modell megvalósítására, a neve RELAIS (Record Linkage At ISTAT). Ezt a programot használtuk a probabilisztikus összekapcsolás megvalósításához.

A felhasznált szoftverek bemutatása

A RELAIS 2.1 Java környezetben fut, (J2SE és legalább JDK 6.0x) szüksége van mySql szerverre, mySql ODBC 5.x vagy magasabb verziószámú driverre legalább 2.9 verziószámú R programra, és az RODBC, valamint az lpSolve R csomagokra. Fontos, hogy a mySql installálása során engedélyezzük az Anonymus felhasználót.

A RELAIS egyszerre két adatbázissal tud dolgozni, ezeket szövegfájlból olvassa be. Az első rekord kötelezően az adatbázis-sémát kell, hogy definiálja. A két adatbázis-séma lehet különböző, de az összekapcsoláshoz szükséges oszlopokat mindkettőnek tartalmaznia kell, mégpedig azonos néven. A mezőnevek megadásánál figyelni kell arra, hogy a RELAIS érzékeny a kis- és nagybetűk közötti különbségre.

A szövegfájlban az új sor karakter zárja le a rekordot, a listaelválasztó pedig egyértelműen elválasztja az egyes mezőket a rekordon belül. Így az nem fordulhat elő egyik mezőben sem, és egyedi kell, hogy legyen az egész adatbázisban.

A program a következő úton tereli végig a felhasználót:

- Kapcsolómezők megadása
  Ekkor állítjuk be azokat az attribútumokat, amelyek hasonlósága alapján döntünk a rekordok egyezőségéről.
- Összehasonlító függvények beállítása
  Ki lehet választani olyan függvényeket, amelyek vizsgálni, és számszerűsíteni tudják numerikus és szöveges adatok hasonlóságát, vagyis azt, hogy mikor tekinthetők közel azonos értékűeknek. Az előbbi típusnál ez magától értetődően a számok egyezősége, vagy eltérésük nagysága, a szövegek esetén pedig az egyéb alkalmazásokból (pl. helyesírás-ellenőrzésből) ismert metrikák, mint a Levenshtein távolság, a Dice, vagy a Jaro-Winkler távolság.
- A keresési tér előállítása
  Ez gyakorlatilag a két megadott reláció Descartes-szorzatának előállítását jelenti. A Descartes-szorzás alakítja ki azokat a rekordpárokat, amelyeket össze kell hasonlítani.
- Blokkoló változók megadása
  Ez a lehetőség a keresési tér redukálását célozza. Könnyű végiggondolni, hogy nagyobb adatbázisok összekapcsolása esetén az összehasonlítások száma hatalmas nagy lehet, mert, ha az egyik reláció n, a másik m sort tartalmaz, akkor $(n \cdot m)/2$ összehasonlítást kell elvégezni. Ennek csökkentésére a RELAIS többféle eljárást kínál.
  Egyik lehetőség egy blokkoló változó megadása, amelynek értékei alapján csoportokat alakít ki a program, és csak a csoportokon belül történik meg az összehasonlítás. Nyilván olyan attribútumot érdemes választani, amely, ha különböző értéket vesz fel a két relációban, az eleve kizárják az összekapcsolhatóságot, mint pl. a nem, vagy a születési hely.
  Másik eljárás a Sorted Neighbourhood módszer (SNM) amelynek lényege, hogy a két adathalmazt egy adott, közös változó alapján rendezni kell. Ezt követően egy fix méretű ablak szalad le az egységesen rendezett listán, és azokat a párokat kísérli meg a program összekapcsolni, amelyek beleesnek az ablakba. Ennek méretét a felhasználó határozza meg.
  Lehetőség van a fenti két módszer kombinálására is.
- Döntési modell kiválasztása
  A RELAIS képes megvalósítani a Fellegi-Sunter elméleten alapuló valószínűségi összekapcsolást, de használható determinisztikus record linkage-re is.

A RELAIS a determinisztikus összekapcsolást kétféleképpen is meg tudja valósítani. Az Equality match módszer alkalmazása esetén meg kell adnunk azokat a mezőket, amelyek a kulcsot alkotják majd. A RELAIS ennek alapján osztályozza a rekordpárt, ha minden kiválasztott kulcs egyenlő, akkor összekapcsolja azt, különben nem. Ilyenkor nem lehet az egyenlőségen kívül más összehasonlító szabályt alkalmazni.
A Rule based módszer esetén komplex szabályok definiálhatók, amelyek alszabályokba vannak szervezve. Minden egyes alszabály állhat feltételekből, amelyeket AND operátor választ el. Az egyes alszabályokat az OR operátor választja el, és egyszerű logikai feltételek is megadhatók. Ha egy rekordpár előírt változóira teljesülnek a szabályok, akkor összekapcsolódnak, egyébként nem.

Az attribútumok leírása

A rövid elméleti ismertetőből levonható egyik legfontosabb következtetés, hogy az összekapcsoláshoz az esetleges azonosítókon és közös attribútumokon keresztül vezet az út. Ennek érdekében át kell tekinteni, hogy egyáltalán mely mezők használhatók fel. Az 1. táblázat tartalmazza azokat az attribútumokat, amelyek legalább két adatbázisban előfordulnak.

1. táblázat

Attribútumok	BÁH	KEKKH	OEP	NÉPSZ
Személy leírása				
Családnév	X	X		
Utónév	X	X		
Anyja családneve	X	X		
Anyja utóneve	X	X		
Születési idő	X	X		
Nem	X	X	X	X
Családi állapot	X	X	X	X
Születési országkód	X	X		X
Születési ország		X	X	
Születési település	X	X		
Állampolg. kód		X		X
Lakóhely				
Irányítószám		X	X	
Település		X	X	X
Tartózkodási hely				
Irányítószám	X	X	X	
Megye	X			
Település	X	X	X	X
Közterület neve	X	X		
Közterület jellege	X	X		
Házszám	X	X		
Épület	X	X		
Lépcsőház	X	X		
Emelet	X	X		
Ajtó	X	X		

Az attribútumok 1. táblázatban leírt rendszerének kialakításához néhány transzformáció elvégzésére volt szükség. Ezek egészen pontosan az alábbiak:

- A KEKKH adatbázisában szerepel egy **utónév2** mező. Ebbe írták az összes utónevet, eltekintve az elsőtől. Mivel a többi adatbázisban nem szerepel ez a megoldás, a KEKKH esetén konkatenáltuk az utóneveket, így állt elő az az utónév mező, amely a BÁH adatbázisban is megtalálható.
- Ugyanezt a megoldást alkalmazták a KEKKH adatbázisban az anya utónevével kapcsolatban is.
- A NÉPSZ adatbázisban eredetileg nem volt születési idő, mert az évet, a hónapot és a napot külön oszlopokban tárolták. A BÁH és a KEKKH ezt az adatot egy 8 karakter hosszúságú szövegként tartalmazza. Így a NÉPSZ három dátumrészét is eggyé konkatenáltuk, majd töröltük a nem létező dátumokat. (Ilyen, ahol valamelyik dátumrész hiányzott).
- Az OEP adatbázis eredetileg csak a születési ország nevét tartalmazta. Ennek alapján azonban elő lehetett állítani az országkódot.
- A tartózkodási hely megyéje csak a BÁH adatbázisban szerepel. A tartózkodási hely települése viszont mindegyikben, és ha azt megbízhatóbbnak fogadjuk el, akkor javítjuk a minőséget azzal, ha a megyéket a helységnévtár alapján rendeljük a településekhez. Ez viszont mindegyik adatbázisban megtehető.

A fenti eljárások rejtenek magukban kockázatokat is. A nevek összeállítása során nem biztos, hogy az egyes név részeket az egyes adatbázisok azonos sorrendben tartalmazzák. Ez a probléma az adatbázisok eredeti állapotában is fennáll, a konkatenálás során csak az fordulhat elő, hogy az utónév2 mezőben szereplő több utónév más sorrendben szerepel, mint a BÁH adatbázisban, és így az egységes utóneveket elrontjuk. Ha viszont a BÁH-ban szereplő utónevet vágnánk két részre, ugyanilyen hiba fordulhatna elő, amennyiben első helyen nem a KEKKH utónév1 mezőjének tartalma szerepel.

A BÁH és az OEP adatbázisokban szerepelnek azonosítók, nevezetesen az **id** és a **sorszam** mezők. Valóban mindkettő egyedi azonosító abban az értelemben, hogy egyik oszlop sem tartalmazza ugyanazt az értéket kétszer, és egyik rekord esetén sem üresek. Mindkettő numerikus típusú, de más tartományban helyezkednek el. Az id legkisebb értéke 761, a legnagyobb pedig 180 331 479, míg ezek a határok a sorszam esetén 43 és 533 309. A BÁH 261 079 azonosítója közül mindössze 25 353 fordul elő az OEP azonosítói között, ebből 330 esetben fordult elő, hogy a BÁH és az OEP adatbázisokban egyaránt előforduló Nem, Családi állapot és Település mezők közül mindhárom értéke megegyezett volna. (Az irányítószámot megbízhatatlansága miatt nem vizsgáltuk). Nyilvánvaló tehát, hogy az azonosítók nem alkalmasak az összekapcsolásra, ezért nem is szerepelnek az 1. táblázatban.

Az adattisztítás végrehajtása

Az összekapcsolás sikerességéhez nem elegendő az, legyenek közös attribútumok, hanem az is fontos, hogy ezek tartalma azonos módon legyen megadva. Ezért szükség van bizonyos előkészítésre, amit adattisztításnak mondunk. Ezekben az adatbázisokban szereplő mezők két csoportba sorolhatók.

- Vannak kódjellegű attribútumok, amelyek csak jól meghatározott tartalmú értékeket vehetnek fel. Ezekben a hibás adattartalom felismerhető, esetleg javítható.
- Vannak szabad szövegeket tartalmazó attribútumok, amelyek értékeit rendszerint adottnak kell tekinteni, bizonyos esetekben nagy munkával lehet csak az írásukat egységessé tenni.

A kódjellegű attribútumok kezelése

Ezeknél az oszlopoknál a legfontosabb feladat, hogy azonos kódolást határozzunk meg, és minden adatbázisban azt használjuk. A következő mezőket kezeltük.

Születési idő

Három adatbázis tartalmazza a BÁH, a KEKKH és a NÉPSZ. Mindháromban 8 karakter hosszú szöveg tartalmazza a születési dátumot ÉÉÉÉHHNN formátumban. Az OEP csak a születési évszámot tartalmazza, ami adott esetben ellenőrzés céljára hasznos lehet. Két rekordban találtunk háromjegyű évszámot, ezt töröltük. Az adattisztítás során azt vizsgáltuk, hogy a megadott karaktersorozatokat lehet-e dátummá konvertálni, de nem találtunk hibát.

Nem

A nemre utaló kód valamennyi adatbázisban előfordul egy karakter hosszúságú szöveg típusú adatként. Nyilván csak kétféle értéket vehet fel, amit könnyű ellenőrizni, és ez valóban igaz minden adatbázisban. A kódolás az egyes forrásokban a 2. táblázat szerint valósult meg.

2. táblázat

	BÁH	KEKKH	OEP	NÉPSZ
Férfi	M	1	1	1
Nő	F	2	2	2

Az OEP és a NÉPSZ adatbázisok nem tartalmaznak neveket, így a kódokat nem lehet ellenőrizni, de nyilván feltételezhető, hogy a 2. táblázat szerinti, szokásoknak megfelelő kódolást alkalmaztak. A BÁH és a KEKKH adatbázisokban elméletileg lehetne ellenőrizni a nevek és a nem ellentmondás mentességét, de ez a sok külföldi név miatt meglehetősen nehéz feladat lenne. Így feltételeztük, hogy a nem kódolása helyes, és a BÁH adatbázisban az M-et 1-re, az F-et 2-re cseréltük.

Családi állapot

Valamennyi adatbázisban előfordul a családi állapotra utaló mező. A használt kódolást a 3. táblázat mutatja.

3. táblázat

BÁH, NEPSZ		KEKKH		OEP
Kód	**Jelentés**	**Kód**	**Jelentés**	**Jelentés**
1	Nőtlen	1	Nőtlen, hajadon	Elvált
2	Házas	2	Házas	Házas
3	Özvegy	3	Özvegy	Házassága megszűnt
4	Elvált	4	Elvált	Nem nyilvántartott
5	Bejegyzett élettárs	5	Házassága megszűnt	Nőtlen, hajadon
		6	Ismeretlen	Özvegy

Ugyanakkor ezt a mezőt nem érdemes az összekapcsoláshoz felhasználni, mert

- nem ellenőrizhető az adat helyessége, hiszen más mező tartalma alapján nem következtethetünk a kitöltés helyességére,
- könnyen megváltozhat a családi állapot, és mert az adatbázisokba nem egyszerre kerülnek be még azok sem, akik mindegyikben előfordulnak.

Ennek következében nem módosítottunk egyetlen bejegyzésen sem, pusztán annyit ellenőriztünk, hogy egy adott oszlopban tényleg csak a megengedett értékek jelennek-e meg.

Irányítószám

Különböző adatbázisokban a lakóhely és a tartózkodási hely irányítószáma szerepel. A lakóhelyhez csak települések tartoznak, így az adat nem is ellenőrizhető, hiszen egy településhez több irányítószám használatos. Elméletileg vizsgálható, hogy egy rekordban rögzített irányítószám szerepel-e az adott településhez tartozók között, ha nem, akkor viszont nehéz eldönteni,

hogy a kettő közül melyik a helyes. Ezért itt csak azt ellenőriztük, hogy van-e olyan bejegyzés, amely nem lehet irányítószám.

A tartózkodási hely fontosabb, mert a nyilvántartásokban azokra helyezik a nagyobb hangsúlyt. A BÁH, és a KEKKH az irányítószám és a település neve mellett tartalmazza a közterület nevét, jellegét, a házszámot, az épületet, a lépcsőházat és az ajtót. Egy jó címlista alapján az már ellenőrizhető lenne, hogy ezek között az adatok között van-e ellentmondás, de ha van, akkor sem lehet mindig megmondani, hogy melyik a helyes.

Tekintettel arra, hogy a címet ilyen mélységben már nem feltétlenül érdemes kapcsolómezőként használni, itt is csak arra szorítkoztunk, hogy töröljük azokat a bejegyzéseket, amelyek nem lehetnek irányítószámok.

Közterület jellege

Az ilyen mezőkben általában kódokkal adnak meg olyan adatokat, mint utca, út, lépcső, stb. A BÁH adatbázisban 69-féle kód szerepel a **kozterulettipus** mezőben, de nem állt rendelkezésünkre ezek jelentését tartalmazó leírás. A KEKKH adatbázisban a **KozterJelleg** mezőben 82-féle bejegyzés szerepel, de ezek nincsenek kódolva, így a mezőt, ha akarnánk, sem tudnánk felhasználni az összekapcsolás során. Az értékeket egyik helyen sem módosítottuk.

Házszám

A BÁH és a KEKKH adatbázisokban található mezők. A BÁH sokszor tartalmaz hibás karaktereket, amelyek nem értelmezhetők házszámként, valamint érthetetlen karaktereket. Valószínűleg a helyrajzi számok is ebbe a mezőbe kerültek. A KEKKH adatbázisban jobb minőségű a házszám mező, de összekapcsolásra nem használható.

Töröltük a házszámok közül a csillagot, a kötőjelet, a kettős kötőjelet, a vesszőt, a \N bejegyzést stb.

Egyéb címrészek

Ide tartoznak az épületet, lépcsőházat, emeletet és ajtót rögzítő adatok, amelyek a BÁH és a KEKKH adatbázisban vannak. Közös jellemzőjük az alacsony kitöltöttség, hiszen nem is feltétlenül részei egy címnek. Éppen ezért nem is használjuk összekapcsolásra, így csak a nyilvánvaló hibákat töröltük az adatbázisokból.

A szabad szöveget tartalmazó mezők kezelése

Ezeknél a mezőknél a legfontosabb, hogy a különböző adatok (nevek, települések, ország nevek stb.) leírása egységes legyen. Az adattisztításban a legnagyobb nehézséget éppen az okozza, hogy sokszor nem egyértelmű, milyen írásmódot kellene követni, bármelyiket is választjuk, nem zárható ki teljes biztonsággal, hogy egy másik adatbázisban nem másikat használtak.

Családnevek

Családnevek a BÁH és a KEKKH adatbázisokban fordulnak elő. Közös gond, hogy a hosszabb családi nevet is egy mezőbe írták, így nem ritka a kettő, három vagy többtagú név. Vannak szóközhalmozódások, és nem nevekbe illő karakterek. Sok esetben tudományos fokozatot, vagy egyéb rövidítést is beleírtak a családi névbe. Máskor rövidítéseket tartalmaznak, amelyekről sok esetben nem is állapítható meg, hogy mit jelentenek: GEB, HTL, OEC, RER, ING, stb. Az adattisztítást a következő módon célszerű végrehajtani:

- Megszüntetjük a szóközök duplikálódását, mert az alább leírt tisztító program csak egyszeres szóközökre van felkészítve.

Ez követően lefuttatunk egy tisztító programot, amely

- Törli a 4. táblázatban felsorolt tudományos fokozatokat és az egyéb rövidítéseket az azokat határoló egyéb karakterekkel – rendszerint pontokkal – együtt a név elejéről, végéről és belsejéből, helyére szóközt ír.

4. táblázat

CSC	DR	ING	KMF	NAT	PROF
DENT	EOC	IUR	MAG	OEC	RER
DIPL	ÉP	JR	MED	OE	RNDR
DIR	GEB	JUR	MGR	PHARM	SCH
DKFM	HTL	JUDR	MUDR	PHD	SOC
DOC	HTM	KFM	MVDR	PHIL	UNIV

- Törli a következő betűcsoportokat a név elejéről, végéről és belsejéből, majd szóközzel helyettesíti: E/V, E/W, M/E, P/V, V/D, W/V
- A nevek elején, végén és belsejében levő (FH) karakternégyest szóközre cseréli.
- Kicseréli a ' karaktersorozatot aposztrófra, a szóköz és aposztróf, illetve az aposztróf és szóköz kombinációkat aposztrófra, a pontot, a vesszőt, a pontosvesszőt, az alsó aláhúzás jelet és a kötőjelet szóközre.
- A bejegyzések elején és végén levő pontot, vesszőt, pontosvesszőt, alsó aláhúzás jelet, kötőjelet, / jelet és \ jelet szóközre cseréli.

Ezt követően meg kell szüntetni a dupla szóközöket és újra futtatni a tisztító makrót. Ezt addig kell csinálni, amíg 0 nem lesz a változtatások száma. Ma ez megtörténik, akkor célszerű manuálisan még a következőket megtenni:

- Ellenőrizni, hogy a ' valamilyen töredéke maradt-e a névben, és ha igen, vagy törölni, vagy aposztrófra cserélni.
- Megnézni, hogy maradt-e a névben csillag vagy kérdőjel karakter. A csillag rendszerint egyedül fordul elő, (ritkán két vagy három csillag szerepel névként) ez nyilván törölhető. A kérdőjel a név belsejében egy vagy több karakter helyett szerepel, ilyenkor a hasonló nevek alapján javítunk. Ha nem voltak hasonló nevek, inkább töröltük a bejegyzést.
- Ellenőrizni, hogy maradt-e zárójel a nevekben, és ezt megfelelő módon kezelni. Előfordult, hogy egy név mellett zárójelben egy másik név olt, akkor azt – zárójelek nélkül – új oszlopba vittük. Pl.: YAMAMOTO (HORVATH) névből a HORVATH új oszlopba került, de az összekapcsolás során ezt nem használtuk fel.
- Átalakítani a bejegyzéseket nagybetűssé az egységes megjelenés érdekében, (mivel a nevek döntő többsége eleve az volt), törölni a szóközöket a nevek elejéről és végéről, megszüntetni a szóközök duplikálódását a nevek belsejében.
- Törölni a nullahosszúságú bejegyzéseket.

Az eredeti vezetékneveket eredeti tartalmukkal meghagytuk az adatbázisban, de az összekapcsoláshoz a fenti eljárás után kapott nevet használtuk.

Utónevek

Az utóneveken is végrehajtottuk a családneveknél leírt adattisztító eljárásokat. A sokféle leírás számát úgy tudtuk csökkenteni, hogy az ékezetes betűket, valamint az Ä betűt ékezet nélkülire cseréltük, továbbá a Zsuzsát Zsuzsannára, a Beát Beátára, a Katát és a Katit Katalinra változtattuk. Az eredeti utónevek megmaradtak az adatbázisokban, de az összekapcsolás során a tisztítottakat használtuk fel. A nevek cseréje során vigyázni kell arra, hogy csak akkor szabad a bejegyzést módosítani, ha a módosítani kívánt szöveg kiadja a mező teljes tartalmát. (Nem szabad a Katalin elején levő Kata szöveget is Katalinra cserélni, mert akkor Katalinlin lesz az eredmény).

Országkódok, ország nevek és településnevek

Ennek a három mezőnek az együttes tárgyalását az indokolja, hogy egymással szoros összefüggésben állnak és a tisztításuk is csak együtt lehetséges. Itt valójában a vizsgálatok az ellentmondás mentességet és az országok, illetve települések nevének egységes leírását célozzák.

2. ábra

BÁH	KEKKH	OEP	NÉPSZ
Országkód	Országkód	Országkód	
Ország	Ország	Ország	
Település	Település		Település

A 2. ábrán láthatóak az egyes adatbázisokban rendelkezésre álló attribútumok. A szürkével jelölteket elő lehet állítani a rendelkezésünkre álló adatokból, a satírozottakat nem. Figyelembe véve, hogy a települések pontosabban azonosítanak, mint az országok, a BÁH és a KEKKH esetén érdemes lenne a településeket használni az összekapcsoláshoz. Ekkor elsődleges fontosságú a települések azonos módon történő leírása. Ez azonban sajnos alig valósítható meg, mert sokféle települést kellene ellenőrizni. A BÁH adatbázisban eredetileg 34695-féle településnév volt, amelyek közül 240-et sikerült egységesen leírni. Ezek kb. 24 000 rekordban szerepelnek, de ez még mindig csak az adatbázis 10%-a. Ezért a születési települések valójában nem alkalmasak jelenleg arra, hogy kapcsolók legyenek, ennek ellenére megpróbáltuk javítani a minőséget, amiért is a következő lépéseket tettük:

- Töröltük a fölösleges szóközöket és a bejegyzéseket nagybetűsre konvertáltuk.
- A \ és a / előtt és utáni szóközöket megfelelő cserék segítségével töröljük, továbbá a nyitó zárójel, majd szóköz, valamint a szóköz majd csukó zárójel karakterpárokat nyitó és csukó zárójelre cseréltük. Azért, hogy minden zárójel előtt legyen szóköz, a nyitózárójelet szóköz plusz nyitózárójel, a csukó zárójelet csukó zárójel plusz szóköz karaktersorozatra cseréltük.
- Töröltük a nem megfelelő típusú bejegyzéseket, pl. dátumokat, csillagokat, kérdőjeleket. Ezek javított formáját tabelláljuk, és később használjuk fel.
- A pontot, vesszőt, kettőspontot úgy cseréltük le, hogy utána egy szóköz is következzen.
- Megszüntettük a szóközök duplikálódását.

Ez követően lefuttatunk egy tisztító programot, amely

- Törli a ' karaktereket, a vesszőt, a pontot, a kettőspontot és a kötőjelet a nevek elejéről.
- Szóközzel helyettesíti az 5. táblázatban felsorolt rövidítéseket.

5. táblázat

JUD	OBL	COM	SAT	CUB
BAC	CON	MUN	ORS	CD

Végül manuálisan elvégeztük a következő tevékenységeket.

- Újra megszüntettük a szóközök, zárójelek duplikálódását, mert a tisztítás futása során keletkezhettek ilyenek.
- Manuálisan ellenőriztük a zárójeleket, / jeleket és \ jeleket tartalmazó neveket, megállapítottuk, hogy melyik a település neve, és ezt is táblázatba foglaltuk. Ezt használtuk fel a településnevek leírásának további egységesítésére, de az esetek nagy száma miatt még nem teljes.
- A táblázat alapján módosítottuk a település neveket, és ha szükséges volt, az országkódokat is.

Sajnos az országok kódolása adatbázisonként eltérő, és nem felel meg egyetlen nemzetközi szabványnak sem. Fontos tehát közös kódolásra áttérni, amihez az ISO3166 szabványban található három karakter hosszú szöveges kódot választottuk. Az eredeti adatbázisokban a következő országkódokat használták:

BÁH: három karakter, és a 6. táblázatban szereplő kódok.

6. táblázat

007	009	010
011	012	013

KEKKH: három numerikus karakterből álló szöveg típusú kód.
NÉPSZ: öt numerikus karakterből álló szöveg típusú kód.

A születési települések nevét és azok módosításait tartalmazó táblázatban megadtuk az azokhoz tartozó országot és országkódot is, ha szükséges volt, ennek alapján módosítottuk az eredeti országkódot. Ezt nem lehetett mindenhol megtenni, mert mind a BÁH, mind a KEKKH tartalmazott olyan országkódokat, melynek nem felel meg az ISO3166 szabványból semmi. Ezek javítása esetén az őket tartalmazó, nagyobb terület kódját használtuk, ha szükséges volt, bevezettünk ilyet. Az adatbázisokban például a szerbiai és koszovói területek nincsenek egyértelműen elkülönítve, így a Jugoszlávia számára bevezetett YUG kódot használtuk ilyenkor. Azok a kódok, amelyeket át kellett alakítani a BÁH adatbázisban az alábbiak voltak:

- **004**: 1 rekordban szerepel, de ennek alapján az ország pontosan nem azonosítható, így töröltük.
- **007**: 3 rekordban szerepel, és a települések neve alapján Palesztina, így a PSE kód került ezek helyére
- **009**: Sok a Szerbiában található település, de mindegyiket nem lehetett ellenőrizni, így a YUG kódot használtuk.
- **010**: Az átvizsgált települések neve Szlovákiára utal. A Csehszlovákiára bevezetett CSE kódot írtuk be
- **011**: A települések neve alapján Németország, így a DEU kódot jelenítettük meg.
- **012**: A települések nevéből ítélve Németország, szintén a DEU kódra cseréltük le.
- **013**: A települések zömmel a volt Szovjetunió tagállamaiban vannak, ezért az USR kódra cseréltük le ezeket. Ezt a kódot úgyis használták, és valószínűleg a Szovjetunió kódjaként. (A Szovjetunió angol nevének rövidítése egyébként USSR – Union of Soviet Socialist Republics, ez is arra utal, hogy jó felé tapgatódzunk).
- **014**: A települések Horvátországban és Szerbiában vannak, így a YUG kódot használtuk.
- **999**: A kód nem rendelhető egyetlen országhoz. Ahol lehetett, a település neve alapján módosítottuk, a többi esetben töröltük.
- **GAZ**: A település Gáza, így ország Palesztina, az országkód pedig PSE.
- **SRB**: A települések neve alapján a YUG kódot használtuk.
- **USR**: A települések neve miatt itt is a Szovjetunió lehet az ország.
- **XXK**: Valószínűleg Koszovó. A YUG kódra tértünk át.
- **XXP**: 51 rekordban szerepel, amelyek több országból származó településeket tartalmaznak. Ahol sikerült megállapítani a település alapján az ország nevét ott megadtuk azt, ahol nem, (9 esetben) ott üresen hagytuk. Egy helyen cseréltünk nevet, ugyanis Alquds Jeruzsálem arab neve. Ezt átírtuk, és országkódnak megadtuk az ISR-t.
- **YUG**: a volt Jugoszlávia.
- **ZAR**: A Kongói Demokratikus Köztársaságra utal. Albertville ugyan francia város, de Kongóban is volt ilyen, csak ma már Kalemie.

Végül töröltük a /N bejegyzést, amely 170 rekordban szerepelt. A későbbiekben a települések neve alapján esetleg meg lehet kísérelni az országkód megállapítását. Azért, hogy megfeleljünk a szabványnak, a ROM kódot a Romániának megfelelő ROU-ra cseréltük, továbbá a REU-t is ROU-ra változtattuk, mert az előbbi a Réunion kódja, de az ott szereplő települések Romániához tartoznak. Az MNE valószínűleg Koszovóra utal, de a YUG kódot alkalmaztuk helyette.

A KEKKH adatbázisban az országkódokat a következő módon állítottuk elő:

- Egy erre a célra készített táblázat alapján hozzárendeltük az országok hivatalos nevét az adatbázisban megadottakhoz.
- Az így kapott ország nevek mellé egy új oszlopba beírtuk a hivatalos országkódokat.
- Töröltük a 003-as kódot, aminek a jelentése az, hogy az ország neve nem ismert.

Azért, hogy mindenütt meg lehessen adni országkódot, fel kellett használnunk a 7. táblázatban szereplő új azonosítókat. A KEKKH tartalmaz ugyanis olyan kódokat, amelyekhez tartozó országok már nem léteznek, vagy soha nem is léteztek, de az OEP adatbázisban szintén előfordulnak.

7. táblázat

Fiktív ország neve	KEKKH kód	Saját kód
AFRIKA (EGYÉB)	302	AFE
AMERIKA (EGYÉB)	303	USE
ÁZSIA (EGYÉB)	305	ASE
BUKOVINA	130	BKV
CSEHSZLOVÁKIA	141	CSE
DÉL-JEMEN	147	YED
DOMINIKA	151	DMM
EURÓPA (EGYÉB)	306	EUE
JUGOSZLÁVIA	195	YUG
KOREA	203	KRR
MORVAORSZÁG	231	MVR
NÉMETALFÖLD	240	NMA
OSZTRÁK-MAGYAR MONARCHIA	250	OMM
SZOVJETUNIÓ	280	USR

Ugyanakkor az ezekhez az országokhoz tartozó települések alapján közülük többet lehet azonosítani, ami növeli a BÁH adatbázissal való összekapcsolhatóságot, mert ott nem szerepelnek fiktív országkódok. Ezért két országkód listát hoztunk létre, az egyik a fent leírt, az OEP adatbázissal való összekapcsolás miatt, a másik csak szabványos országkódokat tartalmaz, amelyeket az azonosítható településekhez tartozó országok alapján oda is beírtunk, ahol egyébként a fenti fiktív országkódok szerepelnének.

A KEKKH adatbázisa még egy helyen használ országkódokat, egy **ApOKod** nevű mezőben, ugyanis a jelenlegi állampolgárságot adó ország kódját tárolja. Ezek között azonban vannak olyanok, amelyekről nem lehetett megállapítani, hogy mely országot jelentik. Mivel ezt a mezőt úgy sem lehetne felhasználni, nem módosítottuk a tartalmát.

Az OEP adatbázisban szereplő országok nevét egységesítettük, majd hozzájuk rendeltük az országkódot. Itt is vannak olyan országok, amelyekhez nem rendelhető kód, ezek nevét és előfordulásuk számát a 8. táblázat tartalmazza. Ezek közül az országok közül egyedül a Hontalan fordul elő a NÉPSZ adatbázisban, így annak a HNT kódot adtuk.

8. táblázat

Név	Előfordulások száma
AMERIKA	4
ARÁBIA	1
HONTALAN	6
KONGÓ	29
VOLT MAGYARORSZÁGI TERÜLETEK	27

A többi esetben kicsi az előfordulások száma, és a további mezők tartalma alapján nem lehet azonosítani, hogy pontosan melyik létező ország lehet a születési ország. Ezért ezekhez a területekhez nem tartozik kód.

A NÉPSZ adatbázisban nincs születési országgal kapcsolatos adat.

Tartózkodási hely, lakóhely település és megye

A tartózkodási hely mezőben a település javítását lehetett megkísérelni. Problémát okozott, hogy sok esetben településrészek nevét adták meg, azokat egy erre a célra kialakított szótár segítségével átalakítottuk szabványos településnevekké. Az eredeti oszlopok természetesen megmaradtak. A településnevekhez a helységnév-tár alapján rendeltük a megyéket.

Az OEP adatbázisban a tartózkodási hely nagyon alacsony szinten kitöltött, és sok külföldi város is szerepel benne, így használhatósága kétséges. Itt nem is állítottunk elő megyéket.

A NÉPSZ tartalmaz a lakóterületre és a tartózkodási helyre utaló településkódokat, amelyek alapján a települések nevét könnyen elő lehetett állítani. Egyedül a 39999 kódot nem sikerült településnévnek megfeleltetni, de ez a tartózkodási hely esetén csak két rekordban szerepel, a lakóhely esetén viszont egyetlen egyben sem. A lakóterület településkódja egyébként minden rekordban megegyezik az **Terul** mezőben szereplő településkóddal.

Egyéb adatok

A közterületek nevét nem tudtuk vizsgálni, mert nem áll rendelkezésünkre olyan megbízható lista, amelyből kiderülne, hogy egy adott településen létezik-e az adott közterület, és hogy helyesen van-e leírva. Ugyanakkor a NÉPSZ adatbázisból összerakhatók címkódok a TERUL, a SZLOK és a CIMSSZ alapján, amelyek segítségével megjeleníthető a közterület neve. Ez a BÁH és a NÉPSZ összekapcsolásánál hasznos lehet, bár a BÁH közterületneveinek tisztítása szintén nagyon sok időt venne igénybe, így teljesen nem oldottuk meg.

Egyéb információ híján az adattisztítás itt is csak azt jelenti, hogy ellenőrizzük az adatokat, és törekszünk az egységes írásmódra. Ezzel kapcsolatban célszerű kidolgozni valamilyen egységes rendszert pl. a helyrajzi számot tartalmazó címek egységes írására.

A többi adat zömmel csak egy adatbázisban fordul elő, így azok vizsgálatára, tisztítására nem fordítottunk figyelmet.

Az ismétlődések törlése

A tisztítás utolsó mozzanataként töröltük a duplázásokat és az egyéb nem megfelelő rekordokat. Két rekordot akkor tekintettünk egyenlőnek, ha valamennyi mező tartalma – eltekintve az egyedei azonosítóktól – pontosan megegyezett. Ahol volt tisztított oszlop is, ott annak tartalmát használtuk az összehasonlítás során, hiszen azt adjuk majd meg kapcsolómezőnek is. A BÁH esetén – mivel azt két adatlistából kellett összefűzni – külön-külön töröltük az azonos rekordokat, elértük, hogy az eredeti azonosítót újra egyedi legyen, majd az összefűzött állományból újra töröltük az azonos rekordokat. Ezek elvégzése után a tisztított változat 557 756 rekordot tartalmaz. A KEKKH-ban 213 102, a NÉPSZ-ben 143 197, az OEP adattáblában pedig 320 036 rekord maradt meg.

Az adattáblák összekapcsolása

Azért, hogy a RELAIS tudjon dolgozni, létrehoztunk összekapcsolás előtti állományokat, amelyeket a művelet során felhasználunk. Ezek csak a tisztított, összekapcsoláshoz felhasznált oszlopokat tartalmazzák, hogy ne zavarjanak olyan karakterek, (vessző, pontosvessző vagy kettőspont), amelyek esetleg listaelválasztónak vannak megadva. A közös attribútumok puszta léte mellett az eredményesség szempontjából azok kitöltöttsége sem lényegtelen. A következő pontokban szereplő táblázatok tartalmaznak erre vonatkozó információkat is. Ezekben a **Kitöltött** oszlop a nem üres cellák százalékos arányát mutatja.

A BAH és a KEKKH összekapcsolása

A 9. táblázat felsorolja, hogy a BÁH adatbázisból melyek azok az attribútumok, amelyet felhasználhatók az összekapcsolás során.

9. táblázat

Mező	Jelentés	Nem üres	Kitöltött (%)
myid	Egy általunk bevezetett egyedi azonosító, amely egyszerű numerikus sorszámozás.	552 465	100.0
id	Az eredeti azonosító	552 465	100.0
szcsnev1_k	Születéskori családi név, az eredeti adatbázisban levő nevek tisztított változata.	552 423	100.0
szunev1_k	Születéskori utónév, az eredeti adatbázisban levő nevek tisztított változata.	552 391	100.0
acsnev1_k	A személy anyjának családi neve, az eredeti adatbázisban levő nevek tisztított változata.	549 846	99.5
aunev1_k	A személy anyjának utóneve, az eredeti adatbázisban levő nevek tisztított változata.	549 522	99.5
sz_szulido	A személy születési ideje 8 karakteres szöveges adat ÉÉÉÉHHNN formátumban.	552 465	100.0
szulev	A személy születési éve négykarakteres numerikus formátumban.	552 465	100.0
sz_nem_k	A személy neme karakteres típusú adatként, az 1 jelenti a férfit, a 2 a nőt.	552 465	100.0
szulorsz_k	A születési ország ISO3166 szabvány szerinti kódja, felhasználva néhány olyan kódot, ami a többi adatbázis miatt kell. Ezek: YUG, (Jugoszlávia), USR, (Szovjetunió) és CSE (Csehszlovákia).	552 021	99.1
szultel_k	A születési település neve, az eredeti adatbázisban levő nevek tisztított változata.	551 502	99.8
lt_helyseg_k	A tartózkodási hely településének neve, az eredeti adatbázisban szereplő nevek tisztított változata.	550 209	99.6
lt_megye_k	A tartózkodási hely településének megyéje, amelyet a település alapján állítottunk elő.	550 229	99.6
kozterulet_k	Közterület neve a tartózkodási helyen.	548 862	99.3

A 10. táblázat mutatja a KEKKH adatbázisból felhasznált mezőket. Tekintettel arra, hogy a kitöltöttség jóval magasabb a lakóhely esetén, és a tartózkodási hely, illetve a lakóhely fogalma

nem pontosan fedi egymást a különböző nyilvántartó helyek szóhasználatában, így az összekapcsolást a lakóhely települése alapján végezzük el.

10. táblázat

Mező	Jelentés	Nem üres	Kitöltött (%)
myid	Egy általunk bevezetett egyedi azonosító, amely egyszerű numerikus sorszámozás.	212 244	100.0
szcsnev1_k	Születéskori családi név, az eredeti adatbázisban levő nevek tisztított változata.	212 244	100.0
szunev1_k	Születéskori utónév, az eredeti adatbázisban levő nevek tisztított változata.	212 244	100.0
acsnev1_k	A személy anyjának családi neve, az eredeti adatbázisban levő nevek tisztított változata.	212 244	100.0
aunev1_k	A személy anyjának utóneve, az eredeti adatbázisban levő nevek tisztított változata.	212 244	100.0
sz_szulido	A személy születési ideje 8 karakteres szöveges adat ÉÉÉÉHHNN formátumban.	212 244	100.0
szulev	A személy születési éve négykarakteres numerikus formátumban.	212 244	100.0
sz_nem_k	A személy neme karakteres típusú adatként, az 1 jelenti a férfit, a 2 a nőt.	212 244	100.0
szulorsz_o	A születési ország ISO3166 szabvány szerinti kódja, kiegészítve néhány olyan kóddal, ami az OEP és a NÉPSZ miatt kell. (Ld. 7. táblázat.)	212 010	99.9
szulorsz_k	A születési ország ISO3166 szabvány szerinti kódja, amely a BÁH-ban használt kódokat tartalmazza, ahol lehetett, a településnév alapján módosítottuk a 7. táblázatban szereplő kódokat.	212 003	99.9
szultel_k	A születési település neve, az eredeti adatbázisban levő nevek tisztított változata.	213 243	100.0
lt_helyseg_k	A lakóhely településének neve, az eredeti adatbázisban szereplő nevek tisztított változata	211 684	99.7
lt_megye_k	A lakóhely településének megyéje, amelyet a település alapján állítottunk elő.	211 684	99.7

Először megpróbáltunk determinisztikus módon összekapcsolni amiket lehetett, hiszen az mindig megbízhatóbb, mint bármelyik a sztochasztikus módszer. Ennek érdekében mindkét input adatbázist pontosvesszővel határolt txt kiterjesztésű szövegfájllá alakítottuk, majd a következő eljárást hajtottuk végre:

- Beolvastuk az adatállományokat és beállítottuk határoló karakternek a pontosvesszőt. Ekkor a RELAIS automatikusan a **Dataset A** (DSA) és a **Dataset B** (DSB) neveket rendeli az adatforrásokhoz. A munka során a BAH volt a DSA és a KEKKH-nak jutott a DSB. Bármikor lekérdezhető, hogy melyik adathalmaz melyik nevet kapta. A sikeres beolvasás visszaigazolásaként a RELAIS kiírja a beolvasott rekordok számát.
- Lehetséges, hogy az általunk kiválasztott változókról különféle tájékoztató adatokat kérjünk, ezt a **Data Profiling/Matching Variables** menüpontban lehet kezdeményezni. Elkészíttettük például a változók gyakoriság eloszlását, hogy lássuk, melyek a gyakran előforduló adatok, illetve megkapjuk a 9. és a 10. táblázatokban közölt kitöltöttségi arányokat.

- A **Decision Model/Deterministic/Equality Match** parancs kiadása után lehetett megadni, hogy mely változókat használjuk az összekapcsoláshoz. A 11. táblázatban leírtakat választottuk:

11. táblázat

Változó neve	Metrika	Küszöb
SZCSNEV1_K	Equality	1
SZUNEV1_K	Equality	1
ACSNEV1_K	Equality	1
AUNEV1_K	Equality	1
SZ_SZULIDO	Equality	1
SZ_NEM_K	Equality	1
SZULORSZ_K	Equality	1

  A táblázatot a RELEAS generálta, az első oszlop tartalmazza a kapcsoló változók nevét, a második az összehasonlításukhoz használt relációt, ami most az egyenlőség, és a küszöbértéket, amely a pontos egyezőség esetén egy.
- A változók beállítása után lefut az eljárás, és előállnak az összekapcsolt táblázatok, valamint a reziduális adatbázisok, amelyek az eredeti listák azon rekordjait tartalmazzák, amelyeket nem sikerült összekapcsolni. Ezeket txt formátumban lehet menteni.
- Az összekapcsolt tábla automatikusan a Match.txt nevet kapja, szerkezetét pedig a 12. táblázat mutatja. (A neveket valamilyen karaktersorozattal helyettesítettük, de az azonos sorozatok azonos neveket takarnak. Helytakarékosságból csak az elő néhány oszlopot mutatjuk.)

12. táblázat

DS	KEY_DS	MYID	SZCSNEV1_K	SZUNEV1_K	ACSNEV1_K
A	2	2	XXX	YYY	AAA
B	140173	140178	XXX	YYY	AAA
A	3	3	ZZZ	WWW	BBB
B	140404	140409	ZZZ	WWW	BBB

  Az első oszlopba írja a RELAIS az adatbázis azonosítóját az A vagy B betűt. A második oszlopban a RELAIS által létrehozott, az adott rekordhoz tartozó egyedi azonosító értékei látszódnak. Ezek egyszerű sorszámok. A harmadik oszloptól jönnek a közös mezők értékei. Az összekapcsolás kb. egy perc alatt megtörtént, és 145 760 rekordpárt sikerült egymásnak megfeleltetni. Ezek azonban csak olyan értelemben képeznek párokat, hogy a kapcsolómezők értékei egymással egyenlők. Valójában nem biztos, hogy egymás megfelelői, mert a többi argumentum értékében eltérhetnek egymástól. Csak 132 793 olyan rekordpárt sikerült elkülöníteni, amely egy-egy kapcsolatban áll egymással. Amely rekordokra ez nem igaz, azt külön táblába gyűjtöttük.
- A RELAIS automatikusan a ResidualDSA.txt és a ResidualDSB.txt nevet adja mentéskor azoknak a fájloknak, amelyek az A és B adatbázisok nem összekapcsolható rekordjait tartalmazzák. Ezek szerkezete megegyezik az eredeti adattáblák szerkezetével. Esetünkben a reziduális adattáblák a BAH esetén 407 623, a KEKKH esetén pedig 80 633 rekordot tartalmaznak. Amiatt, hogy egyes rekordok több másikkal is párba álltak a kapcsolt és a reziduális táblák sorainak összege nem adja ki az eredeti tábla sorainak összegét. A reziduális táblákban nincsenek ismétlődő sorok.

A RELAIS egyébként nem tud minden ékezetes betűvel megbirkózni, az Ő helyére például kérdőjel került. Az eredményhalmazban levő azonosítók (DS, MYID) segítségével az eredeti adatbázisokból le tudjuk kérdezni a teljes rekordot, és létre tudjuk hozni az outputot. Ezt a munkát már Access segítségével végeztük el.

Ezt követően logikus lépésnek látszik a maradék rekordok sztochasztikus összekapcsolása, de az ehhez szükséges Descartes-szorzat elkészítését reménytelen feladat megkísérelni. Ezért szükség volt a keresési tér csökkentésére. Először szétválasztottuk mindkét reziduálist a Romániában, volt Jugoszláviában és a volt Szovjetunióban (kivéve Ukrajna), illetve az egyéb országban született személyeket tartalmazó részekre. Ezeken belül az első csoportban a születési évet, és a nemet, a másodikban ezeken felül a születési ország kódját is felhasználtuk blokkoló változónak, mert ezeknek majdnem minden rekordra létezik értékük, és különbözőségük kizárja az összekapcsolhatóságot.

A táblákat újra betöltöttük, majd az alábbi lépéseket hajtottuk végre:

- Megvizsgáltuk a kiválasztott blokkosító változót. (**Data Profiling/Blocking Variables**)
  Lehetőség van arra, hogy ellenőrizzük a kitöltöttségét, gyakoriság eloszlását, és egy adott blokk dimenzióra (ez a blokkba eső párok száma, amely a RELAIS alapértelmezése szerint 1 000 000) kiszámoljunk egy Blocking Adequacy mutatót, amely a blokkdimenziónál kisebb számú rekordpárt tartalmazó blokkok és az összes blokk számának hányadosa.
- Létrehoztuk azt a keresési teret, amelyben a Fellegi-Sunter módszer szerint vizsgáljuk majd a rekordpárokat. (**Search Space Creation/Search Space Reduction/Blocking**)
  A keresési tér viszonylag gyorsan elkészült, 213 blokk kicsit több mint 35 millió rekordpárt tartalmaz, miközben a Descartes-szorzat 5.3 milliárd párból állna. A párosítás kb. 4 óra alatt futott le.
- Megadtuk az összekapcsoláshoz használt változókat, nevüket, az összehasonlításuk vizsgálatához használt függvényt és a hozzájuk rendelt küszöbértékeket a 13. táblázat tartalmazza. (**Decision Model/Probabilistic/Matching Variables/Variables Selection**)
- Beállítottuk az összehasonlításhoz használt metrikát (Similarity metric) és a hibaszintet (**Decision Model/Probabilistic/Matching Variables/Metrics and Threshold Setting**)

13. táblázat

Első modell (D-3G)		
Változó	**Hasonlító metrika**	**Hibaszint**
SZCSNEV1_K	Dice	0.9
SZUNEV1_K	3Grams	0.9
ACSNEV1_K	Dice	0.9
AUNEV1_K	3Grams	0.9
SZ_SZULIDO	Equality	1.0
SZULTEL_K	3Grams	0.8
Második modell (3G-09)		
SZCSNEV1_K	3Grams	0.9
SZUNEV1_K	3Grams	0.9
ACSNEV1_K	3Grams	0.9
AUNEV1_K	3Grams	0.9
SZ_SZULIDO	Equality	1.0
SZULTEL	3Grams	0.9
Harmadik modell (L-09)		
SZCSNEV1_K	Levenstein	0.9
SZUNEV1_K	Levenstein	0.9
ACSNEV1_K	Levenstein	0.9
AUNEV1_K	Levenstein	0.9
SZ_SZULIDO	Equality	1.0
SZULTEL	Levenstein	0.9

Valamennyi változó kitöltöttsége nagyobb, mint 90%, de a teljesen azonos tartalmú rekordok már hiányoznak. Az összekapcsolásra három modell készült, amelyeket az összehasonlító függvények különböztetnek meg. A tapasztalatok alapján a harmadik modellt használtuk valamennyi sztochasztikus összekapcsolás során.

- Ezt követően elő kell állítani az összehasonlító vektor komponenseit, és az előfordulások gyakoriságait tartalmazó kontingencia táblát. Ezt megtehetjük akár egyetlen blokkra is, amit előtte meg kell adnunk, (**Decision Model/Probabilistic/Fellegi-Sunter/One Block/Block Selection**, majd **Contingency table)** vagy mindre (**Decision Model/Probabilistic/Fellegi-Sunter/Contingency table)**. Ennek előállítása hosszabb időt vehet igénybe, a futási idő nagy része erre megy el..
- Ezt követően optimalizáltuk a megoldást 1:1 kapcsolatot feltételezve optimalizáltuk a megoldást (**Linkage 1:1/Reduction/ to 1:1/Optimal solution**)
- Lefuttattuk a megoldást az összekapcsolhatóságot és a nem összekapcsolhatóságot jelző küszöbértékek beállításával (**Linkage result/Threshold**).
- Létrehoztuk az output táblákat (**Linkage result/1:1 Result/** és a szükséges táblák kiválasztása)
- Beállítottuk az output mappát, és mentettük az eredményeket. (**Save**) Ezek szöveges (txt kiterjesztésű) állományokban állnak elő, amelyek további szoftverekkel könnyen szerkeszthetők.

A jó minőségű országkód helyett, most a rosszabb minőségű születési települést használtuk kapcsolómezőnek. Természetesen amiatt, hogy az összehasonlításkor nem követeltük meg a teljes egyezést, vannak eltérések, az egyes mezők tartalmában, de ezek valóban nem jelentősek. Pl. összekapcsolódott két rekord, amelyben a születési település az egyikben PISLOLT, a másikban PISCOLT, vagy egymásra talált a LOYOS és a LAJOS, és megfelelt a KSZENIIA a KSZENYIJA-nak. Összesen mintegy 1000 biztosan összekapcsolható rekordot eredményezett a szochasztikus összekapcsolás a Romániában, a volt Jugoszláviában vagy a volt Szovjetunióban születettek esetén, és arányaiban hasonló eredményt adott a többi országban születettek összekapcsolása is. A látszólag kicsi számnak az az oka, hogy a determinisztikus összekapcsolás más a pontosan illeszkedő rekordokat kivette a folyamat elején, és nagyrészt ugyanazokkal a kapcsolómezőkkel tudtuk a párosítást folytatni. Ez persze nem baj, mert a determinisztikus összekapcsolás mindig jobb minőségű, mint a sztochasztikus.

Összekapcsolások a NÉPSZ vagy az OEP adatbázissal

Amikor az összekapcsolt adatbázisok egyike a NÉPSZ vagy az OEP, akkor abba a sajátos problémába ütközünk, hogy nagyon kicsi a közös attribútumok száma. Ennek az a következménye, hogy

A NÉPSZ adatbázisból az összekapcsoláshoz felhasználható adatokat a 14. táblázat tartalmazza.

14. táblázat

Mező	Jelentés	Nem üres	Kitöltött (%)
myid	Egy általunk bevezetett egyedi azonosító, amely egyszerű numerikus sorszámozás.	143 197	100.0
sz_szulido	A személy születési ideje 8 karakteres szöveges adat ÉÉÉÉHHNN formátumban.	143 197	100.0
szulev	A személy születési éve négykarakteres numerikus formátumban.	143 197	100.0

Mező	Jelentés	Nem üres	Kitöltött (%)
sz_nem_k	A személy neme karakteres típusú adatként, az 1 jelenti a férfit, a 2 a nőt.	143 197	100.0
lt_helyseg_k	A lakóhely településének neve, az eredeti adatbázisban szereplő nevek tisztított változata.	143 197	100.0
kozterulet_k	Az lt_helyseg_k mezőhöz tartozó közterület neve	36 748	25.7
szulorsz_k	A születési ország ISO3166 szabvány szerinti kódja	143 197	100.0

A probléma az, hogy ezek az adatok a legtöbb esetben nem azonosítanak egyetlen személyt sem, így amikor kiválasztunk egy sort a BAH vagy a KEKKH adatbázisból, akkor ahhoz nagyon sok rekord rendelhető a NEPSZ vagy az OEP adatbázisból. Így valójában a reziduális táblák informatívak, amelyek azokat a rekordokat tartalmazzák, amelyek nem kapcsolhatók össze. Hasznos lehet azoknak a rekordoknak az ismerete is, amelyekhez csak két-három további rekord kapcsolható.

Megoldást jelentene az OEP és a NÉPSZ esetleges olyan további rekordjainak ismerete, amelyek pontosítják, hogy mely személyekről van szó, és felhasználhatók az összekapcsolás során.

Az outputok

Valamennyi outputot Excel formátumban állítottuk elő.

BAH_KEKKH: Az összekapcsolható rekordok, jelöltük, hogy melyek keletkeztek determinisztikusan és melyek sztochasztikusan.

R_BAH_KEKKH_B: A BAH azon rekordjai, amelyeket nem lehetett bevonni az összekapcsolásba.

R_BAH_KEKKH_K: A KEKKH azon rekordjai, amelyeket nem lehetett bevonni az összekapcsolásba.

R_BAH_KEKKH_M: Azok az összekapcsolható rekordok, amelyek többszörös kapcsolat részei, vagy nem dönthető el, hogy kapcsolhatók, vagy nem.

Hasonló jelentéssel a következő állományok álltak elő:

BAH_OEP
R_BAH_OEP_B
R_BAH_OEP_O
BAH_NEPSZ
R_BAH_NEPSZ_B
R_BAH_NEPSZ_N
KEKKH_OEP
R_KEKKH_OEP_O
R_KEKKH_OEP_K
KEKKH_NEPSZ
R_KEKKH_NEPSZ_N
R_KEKKH_NEPSZ_K

Tapasztalatok

A RELAIS ahhoz képest, hogy ingyenes igen sokat tud, és kényelmesen használható. Túl nagy adatbázisok esetén lassú, és sajnos nem lehet benne egy folyamatot megállítani. Nem jelzi azt sem, hogy hol tart a feldolgozás, így nagyjából tapasztalatok alapján dönthető el, hogy kb. meddig tart egy folyamat. Az is kényelmetlen, hogy az output állományokat csak egyesével lehet menteni, vagy MySQL programozási ismeretek segítségével a háttérben használt adatbázisból nyerhetők ki. (Nem teszteltük a batch módú használatot, de a leírás szerint ezen a problémán az nem segítene). Végül azt is meg kell jegyeznünk, hogy a RELAIS kézikönyve a háttérben folyó számítások matematikai részleteit illetően elég szűkszavú, így nehézséget okoz valamennyi output jelentésének pontos megértése.