



## 7. gyakorlat – Megyei térképek készítése

A mindennapi gyakorlatban előfordulhat, hogy a rendelkezésünkre álló adatok hiányosak vagy hibásak, esetleg nem elegendőek. Ezeknek a hibáknak a javítására szolgálnak a bemutatásra kerülő lehetőségek. Most nem foglalkozunk a grafikus hibák, pontatlanságok kijavításával, ezek időigényesek. Amennyiben úgymond „élesben” készül a munka, akkor minden adatot le kell ellenőriznünk. A továbbiakban az attribútum adatok javítási lehetőségeire fektetjük a hangsúlyt. A közigazgatási adatok beszerzése kezd egyszerűsödni. Az utóbbi időben a *geo-spatial.org* oldalon lehet elérni a legfontosabb adatokat. Részleteket az adatok forrásáról szintén ott találunk. A gyakorlat során a 2017 szeptemberi közigazgatási egység szintű (ro\_uat\_poligon) – ezek lesznek a „községek” – valamint a települések pont állományait használjuk (ro\_localitati\_punct). Felhasználunk továbbá régebbi poligon típusú település adatokat (localitati\_6\_03) és egy utakat tartalmazó állományt (drum). A különböző típusú statisztikai adatok felhasználásának bemutatásához a TEMPO online rendszerből letöltött táblázatokat is használni fogunk. A gyakorlat leírása során Szilágy megye adatait fogjuk használni. Más megye esetében a műveletek azonosak, de lehet eltérés a felmerülő problémák esetében. **Bemutatásra kerülő műveletek:** adatbázis létrehozása, térbeli adategyesítés – *Spatial Join*; adatok összegzése – *Summarize*; számítások a táblázatban – *Field Calculator*.

### Az adatok betöltése

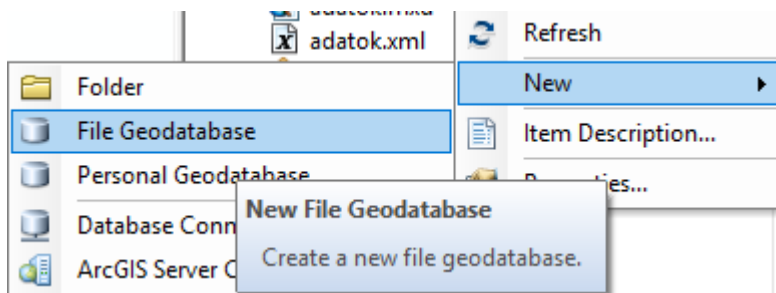
1. A Start gombtól indulva a programok között keressük meg az ArcGIS-t majd azon belül nyissuk meg az ArcMap programot. A megjelenő ablakban válasszuk a **Blank Map** (üres térkép) lehetőséget. Érdemes már az üres térképet is lementeni, ezzel megjelöljük a munkakönyvtárat és a továbbiakban a házikóra kattintva mindig ez a mappa fog megnyílni.
2. Indítsuk el az **Add Data** parancsot . Keressük meg számítógépünkön a szükséges adatokat és töltsük be őket. Jó lenne megszokni, hogy amikor új adatokkal van dolgunk, ellenőrizzük le a koordinátákat. Ezt a legjobban az **ArcCatalog**-ban tehetjük meg az állomány nevére duplán kattintva, majd a Tulajdonságok panel **XY Coordinate System** fülön tudjuk kiválasztani a megfelelő koordinátát. Ezt megtehetjük önállóan elindítva a Start menüből az ArcCatalog programot. Ha ezt elmulasztottuk, megnyithatjuk a Map-ből is a  ikonra kattintva. Még akkor is lehetőségünk van erre, ha az állomány már be van töltve a térképbe! Érdemes erre egy kis időt áldozni, sok további apró kellemetlenségtől tudunk ily módon megszabadulni! Az említett oldalról letöltött adatok azonos koordinátával rendelkeznek, ennek neve Pulkovo\_1942\_58\_Stereo70. Figyeljük meg, hogy az adatkeret (**Layers**) automatikusan átveszi ezt a rendszert. A többi adat is Stereo'70 rendszerben van csak annak más a neve. A program hibát fog jelezni, mivel a két rendszer nem egyezik meg, de valójában csak a nevük nem talál, úgyhogy a hibaüzenetet figyelmen kívül hagyhatjuk.
3. A községek és a települések poligon állománya, valamint a települések pont állománya kerül be a munkafelületre és még az utak. Miért van szükség a települések esetében kétféle állományra? Ebben az esetben azért, mert a poligon állomány kevés leíró adatot tartalmaz. A pont állomány viszont több leíró adatot tartalmaz, ezeket az adatokat szeretnénk átvinni a poligonokhoz.

### Az adatbázis létrehozása

Hozzunk létre egy adatbázist és a kiválasztott megye adatait valamint az összes további állományt helyezzük el benne. Az adatbázis használata azért előnyös, mert minden adat egy

helyen lesz tárolva és az adatbázis teljes mérete is kisebb. Érdeemes a 2. pontnál említett módon az ablak szélére helyezni a katalógust, mert többször szükségünk lesz rá.

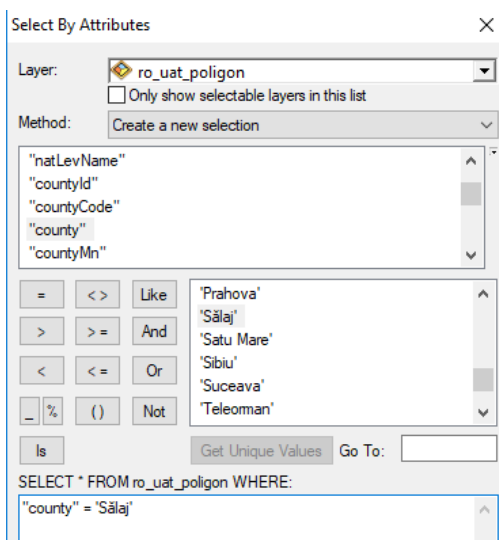
- Nyissuk meg a katalógust és kattintsunk jobb gombbal a saját mappánk nevére. A legördülő menüben válasszuk a **New** lehetőséget majd a további ablakban a **File Geodatabase** lehetőséget. A mappánkba bekerül egy új, gdb kiterjesztésű elem, aminek a nevét rákattintva meg tudjuk változtatni. Javasoljuk a Szilagy\_munka.gdb nevet.



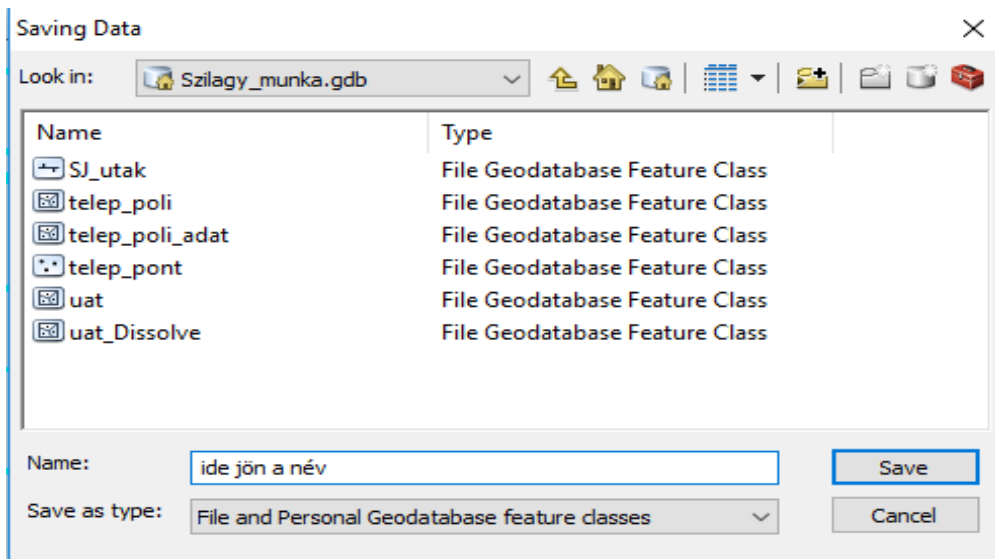
### A megyei adatok leválogatása, a megyetérkép létrehozása

Ha megnyitjuk a rendelkezésünkre álló „községek” (uat) állomány táblázatát, láthatjuk, hogy van benne több olyan mező – szöveges és számszerű is – amely egyértelműen elárulja, hogy a községeket ábrázoló poligonok melyik megyéhez tartoznak. Ennek segítségével válogatjuk le a kiválasztott megye községeit.

- Válasszuk a **Selection** menü **Select by Attributes** parancsot és végezzük el a leválogatást. A vizsgált réteg a „községek” lesz, egy új leválogatást kérünk és azokat a poligonokat keressük, amelyeknél a „county” oszlopban a kiválasztott megye neve szerepel (lásd az ábrát).



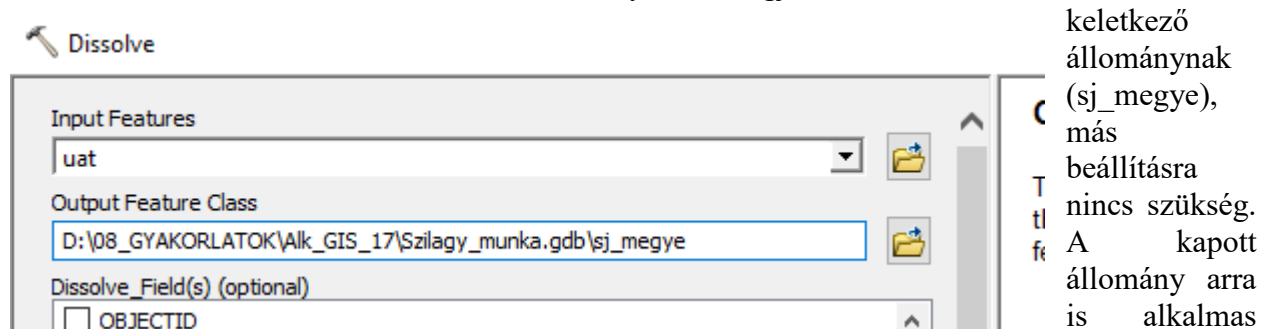
- A kijelölés a táblázatban és a térképen is láthatóvá válik. A kijelölt poligonokból egy új állományt kell létrehoznunk. Ehhez használjunk egy nagyon egyszerű lehetőséget. Kattintsunk jobb gombbal a „községek” nevére és a legördülő menüből válasszuk a **Data** lehetőséget, majd **Export Data**. Ennek hatására a kijelölt poligonok egy új shp állományba kerülnek, ennek adjunk nevet, példánkban „sj\_kozsegek”. De mivel adatbázisban szeretnénk dolgozni, most másképp kell beállítani a mentést, úgy ahogy az alábbi ábrán látható. Ahhoz, hogy az adatbázis megjelenjen, az alsó



ablakban a „File and Personal Geodatabase feature classes” kell legyen beállítva. Hagyjuk jóvá az adatok betöltésére vonatkozó kérdést, így az új állomány bekerül a

tartalomjegyzékbe. Ha ezzel végeztünk az eredeti, országos kiterjedésű község állományt ki is vehetjük a tartalomjegyzékből (jobb klikk, Remove).

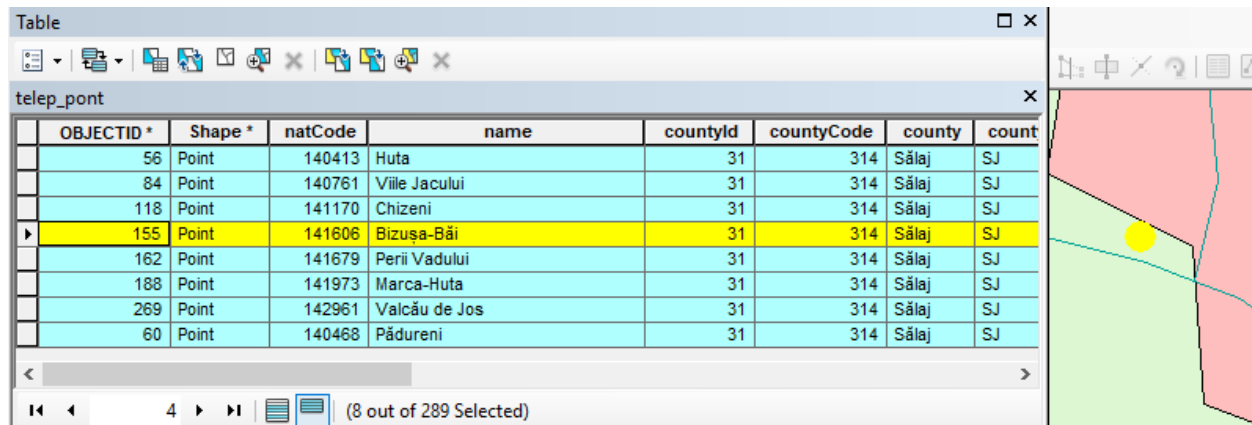
7. Végezzük el az előbbi műveletet a két település állománnyal is kiválasztva a megfelelő kereső mezőt. Miután elkészültünk, az eredeti, országos kiterjedésű állományokat töröljük ki. Az új állományok bekerülnek az adatbázisba, a katalógusban lehet látni őket.
8. Az „utak” állomány esetében nincs lehetőségünk attribútum szerint kiválogatni az adott megyében lévő utakat. Ezért egy térbeli sablont kell készítenünk, amellyel kivágjuk az országos kiterjedésű utakból azokat, amelyek az adott megyében vannak. Ehhez egy olyan állományt kell létrehoznunk, amely a megye teljes területét tartalmazza. Ehhez induljunk ki az előbb létrehozott állományból. A szükséges művelet ki van téve a **Geoprocessing** menübe, a neve **Dissolve**. Most csak a bemenő állományt kell megjelölnünk illetve nevet kell adnunk a



lesz, hogy a végleges térképünkön megjelöljük a megye határát.

9. A kivágáshoz válasszuk ugyancsak a **Geoprocessing** menüből a **Clip** eszközt. Bemenő adat a teljes kiterjedésű utak lesznek, a „vágó” eszköz (clip) az előbb létrehozott megyetérkép lesz. Ezzel a sablonnal minden olyan adatot ki tudunk vágni az adott megye területére, amely nem rendelkezik megfelelő attribútummal.
10. A fentiek elvégzése, illetve az eredeti állományok kitakarítása után a tartalomjegyzékben 5 állomány marad ...\_utak, ...\_megye, ...\_kozsegek, ...\_telep\_poli és a ...\_telep\_pont állományok.
11. Vizsgáljuk meg a településekről rendelkezésünkre álló kétféle adat táblázatát. Kiderül, hogy a pont állomány több attribútumot tartalmaz, mint a poligon. Ezért a pont állomány adatait fogjuk valahogy áthozni a poligonokhoz.
12. Érdemes azt is megnézni, hogy hány poligon és hány pont van az állományokban. Ideális esetben a két szám meg kéne egyezzen. A példának használt Szilágymegye esetében 289 pont van, de csak 286 poligon, holott mindkét állomány a településekről szól. A hibának több oka lehet. Példánkban valószínűleg több önálló településként megjelenő ponthoz csak egy poligon tartozik, ami azt jelenti, hogy a poligonok vannak hibásan létrehozva. Ezeket a hibákat mind ki kéne javítani, de mi most ezzel nem foglalkozunk.
13. Ahhoz, hogy a pontok adatait át tudjuk vinni a poligonokba, meg kell vizsgálnunk a két adat egymáshoz viszonyított térbeli helyzetét. Válasszuk a **Selection** menü **Select by Location** lehetőséget és keressük meg, hogy melyik pontok vannak a poligonokon belül (are within the source layer feature). Vigyázzunk a megfelelő beállításokra! Az elemzés hatására kiderül az adataink pontossága. A példának használt Szilágymegyében a 289 pont közül 281 van benne a poligonokban. Ez tulajdonképpen azt jelenti, hogy van olyan poligon, amelyikben 2 pont van, ugyanakkor van olyan pont, amelyik nincs benne egyetlen poligonban sem. Megyénként nagyon változó eredményt kaphatunk. Arra kell törekednünk, hogy minden pont benne legyen egy poligonban, ugyanakkor minden poligonban csakis egy pont legyen, és ez lehetőleg legyen a megfelelő pont. Ez nem minden esetben valósul meg. A továbbiakban megnézzük, hogyan tudjuk az árván maradt pontokat egy poligonhoz kapcsolni.
14. Nyissuk meg az elemzés nyomán a pontok táblázatát, ez ugye majdnem teljesen kék, hiszen nagyon sok pont kapott magának poligont. Válasszuk a táblázat tetején lévő **Switch**



**Selection** lehetőséget. Ekkor csak az a 8 pont lesz kijelölve, amely „árván” maradt. Ha alól megnyomjuk a Show ablak mellett a **Selected** gombot, akkor csak a 7 pont válik láthatóvá és a térképen is látjuk őket. Ha az egér mutatójával ráállunk a táblázat valamelyik sorának elejére és kattintunk a jobb gombbal, megnyílik egy menü. Ha kiválasztjuk a „Zoom To” lehetőséget, az illető pont a munkafelület közepére kerül alaposan kinagyítva. Ha azonban több sor van kijelölve, az előbbi hatást úgy érzük el, ha duplán kattintunk a sor elejére (az így kijelölt sor besárgul). Így láthatóvá válik, hogy az adott esetben tulajdonképpen mi is a




OBJECTID *	Shape *	natCode	name	countyId	countyCode	county	count
56	Point	140413	Huta	31	314	Sălaj	SJ
84	Point	140761	Vile Jacului	31	314	Sălaj	SJ
118	Point	141170	Chizeni	31	314	Sălaj	SJ
155	Point	141606	Bizuşa-Băi	31	314	Sălaj	SJ
162	Point	141679	Perii Vadului	31	314	Sălaj	SJ
188	Point	141973	Marca-Huta	31	314	Sălaj	SJ
269	Point	142961	Valcău de Jos	31	314	Sălaj	SJ
60	Point	140468	Pădureni	31	314	Sălaj	SJ

probléma. A pont a poligonon kívül van, bár elég közel ahhoz.


15. A probléma megoldásához a pontot el kéne költöztetni. Ezt nyugodtan megtehetjük.

Nyissuk meg az Editor eszköztárat  és indítsunk el egy szerkesztési menetet – **Start Editing**. Ezt úgy is megtehetjük, hogy az állomány nevére jobb gombbal kattintva megnyíló menüből választjuk az **Edit Features → Start editing** parancsot. Ha nem voltunk restek az elején leellenőrizni a koordinátákat, akkor most hibaüzenet nélkül indul el a dialog. Az Editor eszköztáron lévő fekete nyíllal meg tudjuk fogni a pontot és arrébb tudjuk tolni. A legbiztosabb az, hogyha az első kijelölés után feljegyezzük a pontok OBJECTID mezőben lévő értékét és a kijelölést/mozgatást egyenként végezzük. A táblázat alján van egy kis ablak, amibe be tudjuk írni a keresett elem számát, csak arra vigyázzunk, hogy a sorrend 1-el el van tolva. Mivel a FID 0-al kezdődik, az adott számú objektumhoz úgy jutunk el, hogyha 1-el nagyobb számot írunk be. Amennyiben sok ilyen, poligonon kívül elhelyezkedő pontunk van, nem kell minden pontot elmozdítanunk, így is meg tudjuk oldani a pontok és poligonok egyesítését. Arra viszont érdemes lenne időt áldozni, hogy megnézzük, hogy a megfelelő pont legyen egy pont nélküli poligon közelében. Ezt az **Identify** gomb segítségével , az adatok megtekintésével döntsük el.

16. Szilágy megye esetében olyan helyzet is van, amikor egy poligonon belül két, eltérő pont található. Kiderül, hogy a poligon valójában két települést takar, ezért azt ketté kell vágni. A poligon neve „Zimbor”, de valójában Zimboron kívül a „Chendermal” nevű pont is benne van. Ehhez a már említett szerkesztési meneten belül, a poligon kijelölése után az Editor eszköztáron található vágó eszközzel -  - vágjuk ketté a poligont.
17. Megyétől függően előfordulhat, hogy nagyon sok pont kerül a poligonokon kívülre. Ebben az esetben nem muszáj mindent tologatni, de a legbiztosabb eredményt mindenképpen akkor kapjuk, ha minden a helyén van és a számok is találnak. Most tekintsünk el a további hibáktól és folytassuk a munkát. Leállíthatjuk a szerkesztési menetet, mentjük le a módosításokat.

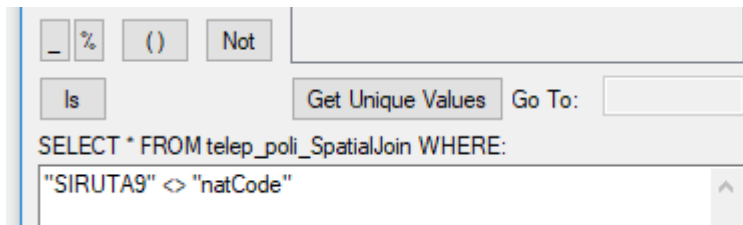
## A települési adatok párosítása

A továbbiakban elvégezzük a már emlegetett térbeli adatcsatlakozást. Ez valójában hasonlít a már bemutatott Join művelethez, de nem táblázatos adat segítségével történik, hanem térbeli helyzet alapján.

18. Kapcsoljuk be a már ismert szerszámos ládikót . Az **Analysis Tools** → **Overlay** csoportban keressük meg a **Spatial Join** eszközt. A **Target Features** a poligonállomány lesz, a **Join Features** pedig a pontállomány. Az állomány neve legyen „sj\_telep\_adat”. A **Join Operation** most nem lényeges. Az „one to one” azt jelenti, hogy ha egy poligonban több pont van, ezek adatai a lentebb lehetséges beállítások szerint összegződnek abba az egy poligonba, amelyben benne vannak. Hagyjuk ezt a lehetőséget. A **Match Option** jelen esetben lehet az **Intersect** vagy a **Contain** is. Bővebb információkat találunk a jobb oldalon megnyitható Help ablakban. Amennyiben sok „árva” pontunk van és nem tologattuk őket a helyükre, használjuk a **Closest** lehetőséget. Ekkor a poligonhoz legközelebbi pont adatai kerülnek át a poligonhoz. Hát ez sem biztos, hogy jó, de mint lehetőség jó tudni róla. Megvizsgálva az elkészült állomány táblázatát kiderül, hogy minden poligonban vannak adatok, legalábbis a példának használt megye esetében.

19. Ha nagyon igényesek vagyunk, végezhetünk egy ellenőrzést az adatok pontosságára vonatkozóan. Legalább megismerünk egy újabb műveletet. Az elvégzett adatösszevonás eredményeképpen a „sj\_telep\_adat” állomány táblázatában két olyan oszlopunk lesz, amely a SIRUTA kódot tartalmazza, az egyik a SIRUTA9 – ez származik a poligon állományból; a másik a natCode – ez származik a pont állományból. Ha összehasonlítjuk a kettőt, kiderülhet, hogy a pontok a megfelelő poligonokhoz illeszkedtek-e.

20. Válasszuk a **Selection** menü **Select by Attributes** parancsot és állítsuk be az ábrán látható



lekérdezést. Kiderül, hogy az imént kettévágott településnél van a hiba, hiszen az eredetileg egy poligonhoz most két eltérő pont adatai tartoznak. Ez most rendben van.

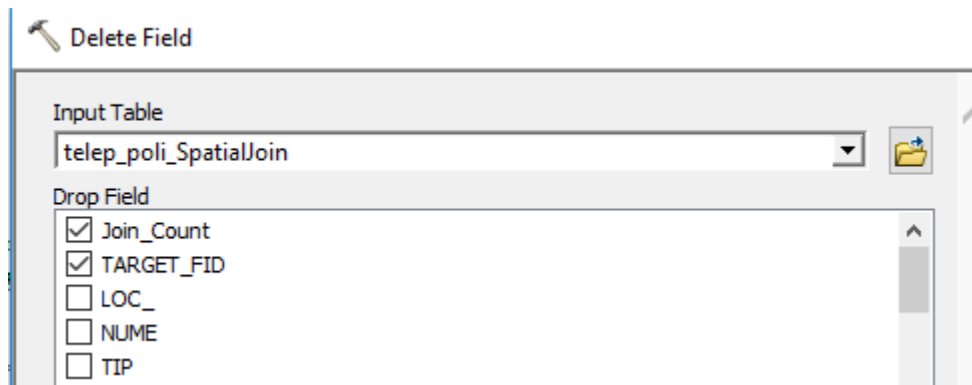
21. Más adatok, más megyék esetében előfordulhat, hogy hibaüzenet jelenik meg, miszerint nem tudja összehasonlítani az adatokat. Vajon mi lehet ennek az oka? Kiderülhet, hogy a két összehasonlított mezőben eltérő adatok vannak. Lehet, hogy az egyik oszlopban, annak ellenére, hogy számok vannak benne, ezek szöveggént vannak tárolva. Ezt ránézésre is meg tudjuk állapítani, hiszen a szöveget tartalmazó mezők balra vannak igazítva, a számok pedig jobbra. A kettőt nem lehet összehasonlítani. Ehhez létre kell hoznunk egy új, numerikus mezőt és át kell másolnunk belé a számokat. A mező létrehozható a táblázat bal felső sarkában lévő **Options** menüből (**Add Field**). Ha nem aktív ez a parancs, azt jelenti, hogy nem zártuk le a szerkesztési menetet. A másolás az új mező nevére kattintva megjelenő **Field Calculator** paranccsal végezhető el. Mivel a mező valójában számokat tartalmaz, ezek át fognak kerülni az új mezőbe és számokként jelennek meg. Ha egy valóban szöveget tartalmazó mezőt akarnánk másolni, ez nem is lenne lehetséges!

22. Ha elkészültünk ismételjük meg az előbbi összehasonlítást a két numerikus mező között. Kiderül, hogy bizony van olyan poligon, amihez nem megfelelő pont adata csatlakozott. Szilágy megye esetében az előbb említett kettévágott poligonnál van a hiba. Más megyéknél előfordulhat több hiba is. Ez is azt bizonyítja, hogy valóban sok gond van az adatokkal és helyes eredményhez mindent le kéne ellenőrizni. De most nem töltjük ezzel az időt.



## Takarítás a táblázatban

23. Az előbbieken bemutatott adatösszevonások eredményeképpen elég nagymértékű redundancia keletkezhet, ami azt jelenti, hogy több oszlopban azonos adatok ismétlődnek. Az előbb említett kódok is kétszer vannak meg. A fölösleges, ismétlődő oszlopokat érdemes kitörölni. Azokat is kitörölhetjük, amelyekre nincs szükség mivel információjuk egyértelmű, pl. megye, régió stb. Ezt könnyen megtehetjük a megnyitott táblázatban a kitörölni kívánt mező nevére kattintva jobb gombbal. A megjelenő menüben ott van a **Delete Field** lehetőség. Ez azonban csak akkor lesz aktív, amennyiben nincs megnyitott szerkesztési menet. Ezzel a módszerrel egyszerre csak egy oszlopot lehet törölni.
24. Amennyiben több oszlopot szeretnénk eltüntetni, nyissuk meg a Toolbox-ot, majd keressük meg a **Data Management Tools → Fields → Delete Field** eszközt. Az eszköz első

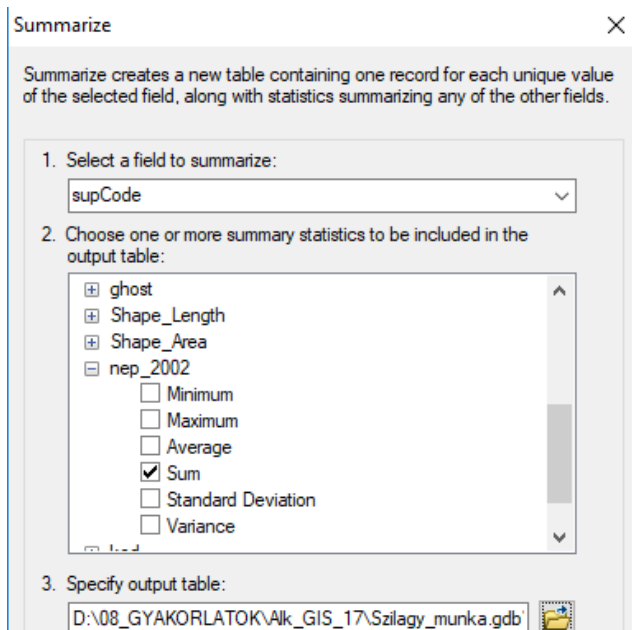


ablakában állítsuk be a kívánt táblázatot, majd alatta jelöljük ki a törölni kívánt oszlopokat. Persze figyelmesen, mert ezután a művelet után

nem lehet visszavonni a lépést!

## Hierarchikus adatösszevonás

25. Ez a művelet akkor alkalmazható, ha a rendelkezésre álló adatok valamilyen szempont szerint összevonhatók. Ez leginkább közigazgatási egységek esetében lehetséges, itt áll fenn a hierarchikus kapcsolat. Ez azt jelenti, hogy az alacsonyabb szintű közigazgatási egységek adatai beilleszkednek a magasabb szintű közigazgatási egységbe (település – község – megye – ország). Mi a települési adatokat fogjuk községi szintre összesíteni. A települések táblázatában van egy „supCode” nevű mező, amely tulajdonképpen azt jelöli, hogy egy adott település melyik községhez, vagyis magasabb rendű közigazgatási egységhez tartozik. Az adatok összevonása által gyakorlatilag a község adatait kapjuk meg.
26. Nyissuk meg a települések táblázatát és kattintsunk jobb gombbal a supCode mező nevére. A megnyíló menüből válasszuk a **Summarize** parancsot. Az első ablakban megjelenik annak a mezőnek a neve, amelyikre rákattintottunk az imént. Akár más mezőt is kiválaszthatunk. Keressük meg a NEP\_2002 mezőt, ezt szeretnénk összevonni. Válasszuk a „Sum” lehetőséget, ez biztosítja, hogy a népességi értékek összegeződjenek. Ha véletlenül nincs „Sum” lehetőség akkor valószínűleg az előbbi hibával állunk



szemben, vagyis a mező nem számokat tartalmaz. Ebben az esetben ismételjük meg az előbb – a 21. pontban – bemutatott műveletet. Nagyon figyeljünk oda munkáink során arra, hogy a számok mindig numerikus mezőkbe kerüljenek. A keletkező táblázatot mentjük saját mappánkba.

### Táblázatos adatok csatolása

Gyakran van szükségünk külső, leíró adatok térképi megjelenítésére. Valójában a GIS egyik legfontosabb lehetősége ez. Amennyiben a külső adatok térbeli objektumokra vonatkoznak, azokat a megfelelő módon az objektumokat tartalmazó állomány táblázatához csatolva meg tudjuk jeleníteni. Ehhez az objektum nevére jobb-kattintással megnyíló menü **Joins and Relates** műveletet használjuk. A továbbiakban többször meg fogjuk ismételni ezt a műveletet.

27. Ha megvizsgáljuk a községeket tartalmazó állomány táblázatát (jobb kattintás az állomány nevére, majd a legördülő menüben **Open Attribute Table**) kiderül, hogy elég sok adat van benne. Ami nagyon fontos, hogy minden községnek megvan a SIRUTA kódja (**Sistemul Informatic al Registrului Unităților Teritoriale**) a natCode mezőben. Figyeljük meg milyen formában van és ha szükséges alakítsuk numerikussá az előzőekben bemutatottak szerint. Ez lesz a **cél-táblázat**.

28. A népességre vonatkozó adatokat az előbb összesítettük a település szintű adatokból. Első lépésben ezeket csatoljuk a táblázatához. Ez lesz a **forrás-táblázat**. Ebben a táblázatban a SIRUTA-nak megfelelő mező a supCode, ezt a mezőt fogjuk az összekapcsoláshoz használni.

29. A táblázat térképhez való csatolásához válasszuk ki a községeket tartalmazó állományt, majd a jobb kattintásra megnyíló menüből válasszuk a **Joins and Relates** menüpontot. A további lehetőségek közül válasszuk a **Join**-t. A megjelenő panel felső ablakában hagyjuk az alapértelmezett beállítást, mely szerint egy táblázat adatait szeretnénk a térképhez kapcsolni.

30. Az 1.-el jelölt ablakban válasszuk ki a községek állomány poligonjainál megjelenő SIRUTA mezőt.

31. Mivel a tartalomjegyzékben csak egyetlen táblázat van betöltve, ennek neve meg is jelenik a 2. ablakban. Ha több táblázat lenne, akkor itt kéne megkeresni a megfelelőt.

32. A 3. ablakban kell megadnunk azt a mezőt, aminek alapján létrehozható a kapcsolat. A mi esetünkben ennek neve supCode. Megjegyezzük, hogy a két mező tartalma kell megegyezzen, nem pedig a neve. Az OK gomb megnyomása után látszólag nem történik semmi, ha mégis szeretnénk látni az eredményt, nyissuk meg a községek táblázatát (jobb klikk, **Open Attribute Table**). Megfigyelhető, hogy mind a cél-táblázat mind pedig a forrás-táblázat adatai megjelennek benne.

*A Join művelet eredménye csak látszólagos. Bár az adatok egy táblázatban jelennek meg, valójában a forrás-táblázat adatai nincsenek átmásolva a cél-táblázatba. Amikor a munkánkat lementjük („térkép” állomány – mxd, vagy régi nevén projekt...) majd újra megnyitjuk, a kapcsolat újra végbemegy. Ha kilépünk a munkából és a cél-táblázatot, a mi esetünkben a községek táblázatát, megnyitjuk egy táblázatszerkesztő programban, a hozzácsatolt adatok nem fognak megjelenni. Ha véletlenül kitöröljük a forrás-táblázatot a helyéről, akkor a munkánk megnyitásakor a Join nem tud végbemenni, hibaüzenetet fogunk kapni. Ha megpróbálunk szerkeszteni egy összekapcsolt táblázatot, csak a cél-táblázat adatainak módosítására lesz lehetőségünk. Az adatok lekérdezése is lassabban megy, mivel az adatok valójában nincsenek közvetlenül benne. Mindezek ismeretében célszerű lenne találni egy olyan lehetőséget, amely véglegessé teszi az adatok elérhetőségét. Mielőtt azonban ezt megnéznénk, kapcsoljunk további külső adatokat a községek poligonjaihoz. A továbbiakban a statisztikai hivatal Tempo adatbázisából letöltött adatokkal fogunk dolgozni.*

33. Az előzőekben egy olyan táblázatot csatoltunk a községekhez, amelynek adatait szintén térinformatikai adatokból állítottuk elő. Így biztosak lehetünk abban – legalábbis ha az adatok helyesek –, hogy a két táblázat sorainak a száma megegyezik. Vagyis minden egyes objektumhoz tartozik külső adat is.
34. Ezzel szemben a TEMPO adatbázissal kissé más a helyzet. Nagyon sok adatot találhatunk benne, de kissé megtévesztő az adatok szintje. Mindenütt azt írja, hogy település szintűek az adatok (localitati), de valójában a közigazgatási egységek (uat, községek) szintjére vonatkoznak. Ugyanakkor attól függően, hogy konkrétan milyen adatot keresünk, előfordul, hogy nincs adat minden közigazgatási egységre.
35. Az adatokat több formában is le lehet tölteni, de a legtöbb esetben nem tudjuk közvetlenül felhasználni. Ha Excel állományként töltjük le, mindenképpen átalakításokra van szükség. El kell tüntetni az összevont cellákat és a címeket. Szerencsére a letöltött adatok között minden esetben ott van a SIRUTA kód, de sajnos egybe van írva a települések nevével. A neveknél megjelenik a köznév is (municipiu, oras). Ezeket mind szét kell választani külön oszlopba (az Excelben – Text to Columns). A legfontosabb, hogy a SIRUTA kódok kerüljenek numerikus mezőbe. Az is nagyon fontos, hogy az ArcGIS 10.1-es verziója csak a régebbi Excel állományt ismeri fel, aminek a kiterjesztése xls. A gyakorlathoz a következő adatokat fogjuk használni: Somaj (a munkanélküliek aránya minden községre). A többi adat nem áll rendelkezésre minden községre, zárójelben az adattal rendelkező községek száma: vízhálózat (54), csatorna (15), gáz (15). A zárójelben lévő számok tulajdonképpen azt jelzik, hogy mivel nem minden községre van adatunk, a forrásként használt táblázatoknak kevesebb soruk lesz, mint a cél-táblázatnak. Ez valójában nem jelent gondot a csatolás szempontjából, de lesznek adat nélküli községek is.
36. Töltsük be a táblázatokat a munkafelületbe. Az Excel állományok több munkalappal rendelkeznek, de egyszerre csak egy munkalapot tudunk betölteni. Amennyiben az összes adatunk egy állományban van, de külön lapokon, akkor laponként kell betölteni.
37. Hajtsuk végre a táblázatok csatolását, egyenként, egymás után a már bemutatott módon. A

SIRUTA	F5	F6	F5_12	F5_12_13	Shape_Length	Shape
142122	<Null>	128	6.7	14.6	53323.164663	534083
140823	<Null>	842	1.3	22.5	40053.172905	330117
141786	<Null>	153	5.7	25.9	42238.086381	606673
141580	<Null>	369	8.7	36.9	75704.0761	873315
140244	<Null>	74	1.9	19	39492.559472	291912
142710	<Null>	135	3	14.5	67034.124297	778795
142774	<Null>	279	2.5	32	66037.477897	707411
141081	8.5	<Null>	17.3	22.1	58669.618886	645145
142612	11.6	<Null>	8.7	71	57694.61069	878518
142373	9.4	<Null>	6.6	10.5	53409.972309	525460

F5, F6... Azonban tudjuk tisztázni, hogy melyik oszlop mit jelent és át tudjuk írni az oszlopok nevét az oszlopra kattintva jobb gombbal majd a Properties... Alias sorba tudunk nevet adni az oszlopnak.

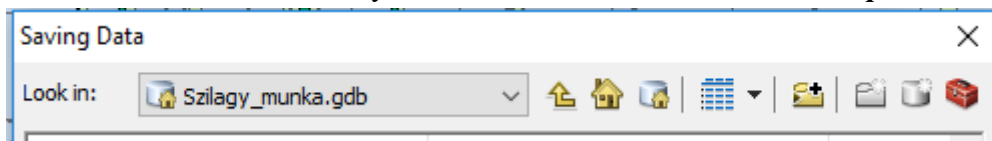
### Adatok átalakítása, mentése

A 32. pont után említett probléma legegyszerűbben úgy oldható meg, ha az összekapcsolt táblázatot tartalmazó állományt átalakítjuk, valójában lementjük más név alatt. Az így létrehozott állomány véglegesen megőrzi a csatolt adatokat. Ezt az átalakítást több módszerrel végezhetjük el.

38. Az egyik lehetőség a Toolbox. Kattintsunk a piros szerszámos ládikóra – **ArcToolbox**, majd a legördülő listából válasszuk a **Conversion Tools** csoportot. Sok lehetőség van, mi most válasszuk a **To Shapefile**-t, ezen belül pedig a **Feature Class to Shapefile** lehetőséget.

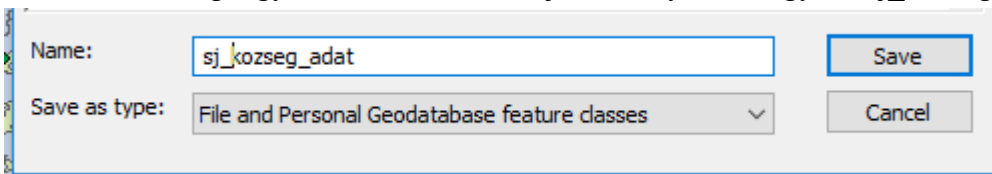


39. A megjelenő panelen 3 kis ablak van. Az elsőben a bemenő – vagyis átalakításra kerülő objektumot kell megjelölni (**Input Features**). Ha a térképünkben van már betöltve adat, akkor a kis ablak melletti legördülő jelre kattintva kiválaszthatjuk a listából a megfelelő nevet. A kiválasztott név bekerül a középső ablakba. Ha a térképünkben még nincs adat, akkor böngészővel kereshetjük meg az állományt. A harmadik ablak – **Output Folder** – kissé furcsa, ugyanis nincs lehetőségünk nevet adni az új állománynak, hanem meg kell jelölnünk azt a mappát amelybe szeretnénk menteni. A kiválasztott mappát nem kell megnyitni, hanem csak egyszer a nevére kattintani, amíg a neve bekerül az ablakba. Válasszuk a saját könyvtárunkat.
40. A másik lehetőséget, ami valójában egyszerűbb, már kipróbáltuk régebbi gyakorlatoknál. Mivel adatbázisban dolgozunk mindenképpen célszerű ezt a lehetőséget választani. Kattintsunk jobb gombbal arra az állományra, amelyhez hozzacsatoltuk az összesen 5 különböző táblázatot. A lenyíló menüből válasszuk a **Data → Export Data...** műveletet. A



megnyíló felületen jelöljük meg a mentés

helyének az adatbázisunkat, az állomány típusnál pedig ügyeljünk arra, hogy az alábbi ábrán látható lehetőség legyen kiválasztva. Az új állomány neve legyen „sj\_kozseg\_adat”.



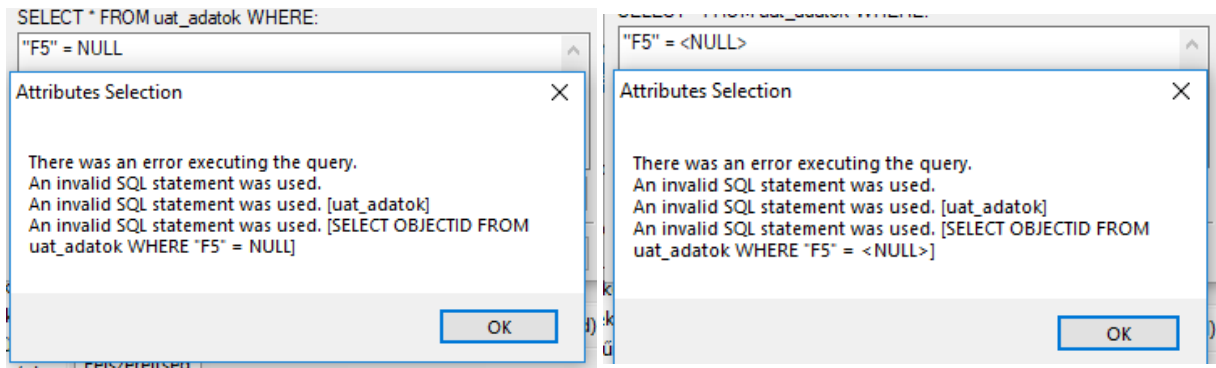
### Műveletek a táblázatban

Sok esetben a rendelkezésünkre álló, és most már megfelelő helyen és formában lévő adatok nem elegendőek a megjelenítéshez. Ezekből kiindulva újabb információkat tudunk előállítani a táblázatban belüli történő műveletekkel. Mielőtt nekifognánk, érdemes lenne egy kicsit takarítani az oszlopok között, de erről már beszéltünk. A műveleteket el lehet végezni a táblázat összes sorára, vagyis minden objektumra, vagy kijelölt sorokra, vagyis valamilyen módon kiválasztott objektumokra. Mindkét lehetőséget kipróbáljuk. Mindkét esetben szükség lesz új mezőkre, amiket a már bemutatott módon, szerkesztési meneten kívül létre tudunk hozni. Most érkezett el az a pillanat, amikor el kéne dönteni, hogy a végső térképen milyen adatokat és milyen formában szeretnénk megjeleníteni. Arra törekszünk, hogy egy térképen minél több információt jelenítsünk meg, de úgy, hogy minden értelmezhető legyen. Legtöbb adatunk a községek szintjén van, összesen 4 különböző információt fogunk megjeleníteni (népsűrűség, munkanélküliség, népesség változás és felszereltség). A települések szintjén kevés adatunk van, itt csak két félé adatot jelenítünk meg.

### Kiválasztott sorok adatfeltöltése

Amint a táblázatok csatolása után láhattuk, vannak olyan községeink, amelyekben bizonyos adatok hiányosak. Ezek a felszereltségre vonatkozó adatok, hiszen nem mindenütt van központosított vízellátás, csatornázás vagy gázellátás. Az már más kérdés, hogy több helyen van vízhálózat, mint csatornahálózat... A rendelkezésre álló adatok az illető hálózatok hosszára vonatkoznak, de mivel nagyon eltérőek a számok, nem a valódi értékeket fogjuk megjeleníteni, hanem csak az illető hálózat létezésének a tényét. Így csak egy logikai kódot kell létrehoznunk az illető községben.

41. A már többször bemutatott módon hozzunk létre három új mezőt a három felszereltségi kód számára. A nevük legyen v\_kod, cs\_kod és g\_kod. Minden mező típusa legyen short integer.
42. Indítsuk el a **Selection** menü **Select by Attributes...** parancsot. A keresést a „sj\_kozseg\_adat” állományon végezzük, hiszen ebben vannak az adatok. A keresési mező legyen valamelyik felszereltségre vonatkozó mező, mondjuk a „csatorna”. Ebben az oszlopban nincs mindenütt adat, a hiányt a <Null> szó jelöli. Ez egy kicsit meg fogja bonyolítani a lekérdezést. Ha megpróbáljuk a lekérdezésbe magát a <Null> szót bevinni ez semmiképpen sem fog sikerülni, mindenképpen hibaüzenetet fogunk kapni. A két lehetőség az alábbi ábrákon látható:

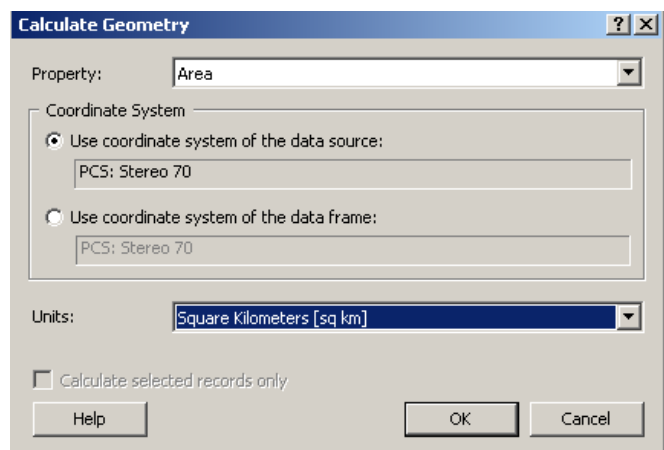


43. Így kénytelenek vagyunk a számokkal dolgozni. A lekérdezésbe írjuk azt, hogy a mező értéke legyen nagyobb, mint 0. Ennek hatására ki lesznek jelölve mindazok a sorok ahol valamilyen érték van. Nekünk pont erre van szükségünk. 15 községben van csatornahálózat.
44. Most nyissuk meg az állomány táblázatát és kattintsunk jobb gombbal az imént létrehozott 3 mező közül arra, amelyik a csatornákra vonatkozik. A legördülő menüből válasszuk a **Field Calculator** parancsot. Egyszerűen egy 1-t kell beírni a fehér felületbe. Ellenőrizzük le, hogy a panelen legyen bekapcsolva a **Calculate selected records only**. Ez tulajdonképpen automatikusan így lesz, mivel a program érzékeli, hogy vannak kijelölt sorok. Az OK megnyomása után a kijelölt sorokba bekerül az 1-es, ami ugye azt jelenti, hogy ezekben a községekben van csatornahálózat. A művelet elvégzése után szüntessük meg a kijelölést!
45. Ismételjük meg az előbbi lekérdezést a másik két felszereltségi elemre is és végezzük el a kódolást, szintén 1-es értéket adva oda ahol van víz illetve gáz. Kiderül, hogy 54 községben van vízhálózat és 15 községben van gáz. Ezzel a megjeleníteni kívánt 4 adat közül elkészültünk a felszereltséggel.

### Műveletek minden sorral, objektummal

Ebben az esetben a műveletek minden sorra érvényesek. Példaként számítsuk ki a községek népsűrűségét. Ehhez ugye a területre lesz szükségünk, lehetőleg km<sup>2</sup>-ben, és a népességre.


46. Hozzuk létre a „terület” mezőt, a típusa legyen **Float**. Kattintsunk jobb gombbal a nevére majd a menüből válasszuk a **Calculate Geometry** lehetőséget. Mivel állományaink rendelkeznek koordinátákkal, lehetőségünk van kiválasztani a számítás mértékegységét (sq km). Most már megvan a sűrűség kiszámításához szükséges terület.



47. Hozzuk létre a „suruseg” mezőt, a típusa legyen **Float**. A mező nevére jobb gombbal kattintva keressük meg a **Field Calculator** parancsot. A fehér ablakba kettős kattintással vigyük be a „pop2015” mezőt, utána az osztás jelét, majd szintén kettős kattintással az előbb kiszámított területet, amint a mellékelt ábrán látható. Az eredmény bekerül a táblázatba. Bármilyen más műveletet el tudunk végezni. Oda kell figyelni a létrehozandó mezők típusára, illetve a műveletek sorrendjére. Megjegyezzük, hogy – mivel adatbázisban dolgozunk – a létrehozott poligon kerületét (Shape\_Length) és területét (Shape\_Area) a program automatikusan kiszámítja, persze a koordinátarendszer mértékegységében, ami a mi esetünkben méter. Így akár ennek átalakításával is elő lehet állítani a számításához szükséges területet. Ezzel elkészültünk a megjeleníteni kívánt népsűrűség adattal.
48. A harmadik adat, amit megjelenítünk, nem igényel különösebb előkészítést, ez lesz a munkanélküliség.
49. A negyedik adat szintén egy számítás eredménye, majd az eredmény átkódolása során jön létre. Ehhez szintén egy új mezőt kell létrehoznunk, a neve legyen „nep\_kul” a típusa pedig Long integer. Előbb egy egyszerű műveletet fogunk végezni ebben az oszlopban kivonva a 2015-ös népességből a 2011-es népességet. Tehát a **Field Calculator** ablakába hozzuk létre a „pop2015 – pop2011” képletet. Ahol pozitív szám lesz az eredmény, ott a népesség növekedéséről beszélhetünk, ahol negatív ott pedig csökkenésről. De mivel ebben az esetben is logikai kódok kellenek, az előbb bemutatott lépésekhez hasonló módon kódoljuk át a csökkenést 1-re, a növekedést pedig 2-re. A kódok egyszerűen felülírják az ott lévő számítás eredményét, de akár új mezőbe is tehetjük a kódokat. Ezzel tulajdonképpen elkészültünk a megjeleníteni kívánt községi adatokkal.

### A települések megközelíthetőségének vizsgálata

A települések esetében az adataik között szereplő kódok segítségével fogjuk megjeleníteni a közigazgatási rangjukat. De mivel rendelkezünk egy elég jó minőségű úthálózattal is, érdemes lenne megvizsgálni a települések megközelíthetőségét. Ez valójában egy bonyolultabb térbeli elemzés, amelyet a poligon típusú település állományon kell elvégezni. A bonyodalom abból adódik, hogy az utak 4 típusba sorolhatók, a településeken pedig akár több típusú út is keresztülmehet. A településeket kódolnunk kell, ehhez a már megismert módon hozunk létre a táblázatban egy új oszlopot, ami legyen numerikus és rövid egész szám fogadására legyen képes (Short Integer).

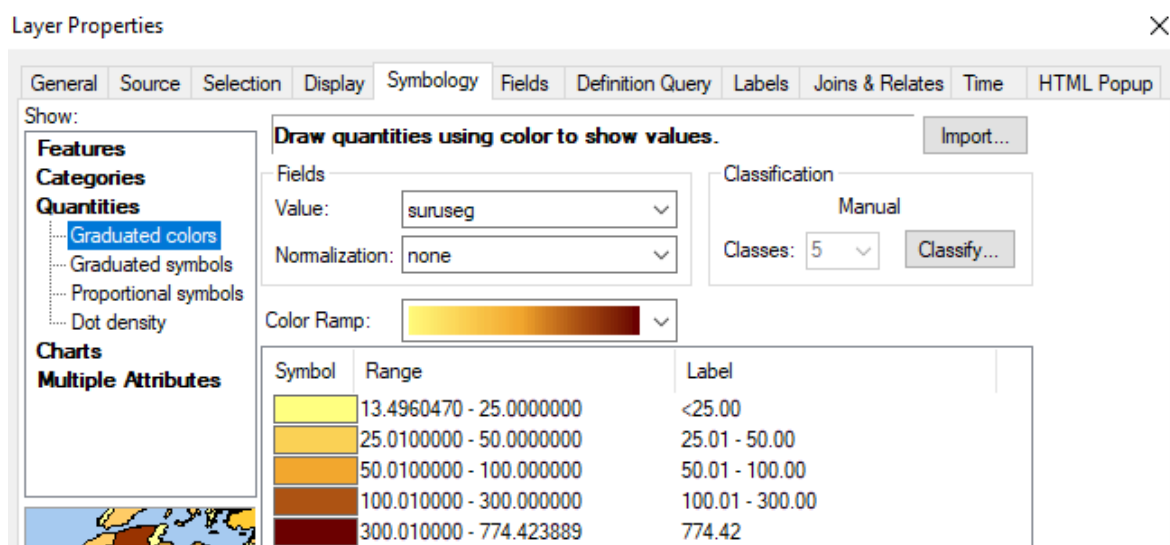
50. Első lépésben vizsgáljuk meg, hogy van-e olyan település, amelybe egyáltalán nem megy út. Indítsuk el a **Selection** menü **Select By Location** műveletet. Az első ablakban meg kell adnunk a keresés módját, ebben az esetben maradjon az alapértelmezett „select features from” lehetőség. A második ablakban lévő listából ki kell választanunk a „cél-állományt”, vagyis azt, amelyben lévő objektumok jellemzői érdekelnek. A megközelíthetőség vizsgálatánál minden esetben a települések poligon állománya lesz a cél. A harmadik ablakban meg kell adnunk, hogy melyik állományhoz viszonyítva vizsgáljuk a településeket. Ez lesz a „forrás-állomány”, a mi esetünkben az utak. A negyedik ablakban választhatjuk ki a két állomány között keresett viszony típusát. Alapértelmezetten az „intersect” van, ami azt jelenti, hogy a két állomány metszi egymást, vagy legalább egy közös pontjuk van. Ez a mi esetünkben megfelel a céljainknak. Ha most megnyomjuk az OK gombot végbemegy a keresés és kiderül, hogy 274 poligonon keresztül megy vagy érint valamilyen út. Mivel a település poligonok száma összesen 286, azt jelenti, hogy van 12 olyan település, amelyhez semmiféle út nem megy legalábbis a mi út-állományunk szerint. Ha megnyitjuk a táblázatot és megfordítjuk a kijelölést a Switch Selection gombbal -  - akkor a 12 út nélküli település lesz kijelölve.

- Most ismételjük meg a 44. pontban leírtakat és kódoljuk az út nélküli településeket 1-el az előbb létrehozott oszlopba. A kódolás után szüntessünk meg minden kijelölést.
51. A következő lépésben vizsgáljuk meg a települések és a megyei utak közötti viszonyt. Ehhez előbb az utak állományban jelöljük ki a megyei utakat, amelyek a „tip” mezőben „DJ” értékkel rendelkeznek. Az összesen 814 útszakaszból, 289 megyei útszakasz. Most ismételjük meg az 50. pontnál leírtakat, de ügyeljünk arra, hogy a **Source Layer** ablak alatt jelöljük meg a **Use Selected Features** lehetőséget. Ezzel utasítjuk a programot, hogy csak a kijelölt utakhoz viszonyítva keressen településeket. 118 ilyen települést talál, ezeket kódoljuk be 2-es értékkel.
52. Most ismételjük meg az előbbi műveletsort, de úgy, hogy az utak közül az országutakat jelöljük ki („DN”- 96 útszakasz lesz). Az eredmény 42 település lesz. Nem meglepő módon olyan település is kiválasztásra került, amely már rendelkezik 2-es kóddal. Az elején említettük, hogy vannak – főleg nagyobb - települések, amelyeken több típusú út is áthalad. Most ezeket kódoljuk 3-as értékkel.
53. Végezetül nézzük meg hány településen halad át európai rangú főút. Az ilyen „E” típusú útszakaszok száma 46, az érintett települések száma pedig 16. Ezek között is van már 2-es és 3-as kódú település. Kódoljuk ezeket 4-es értékkel. A már beírt kódok felülírása által tulajdonképpen azt értük el, hogy minden település a rajta átmenő legmagasabb rangú út kódját kapta, de ez azt is jelenti, hogy sok településen alacsonyabb rendű utak is átmennek. Ha most megvizsgáljuk a táblázat kód oszlopát kiderül, hogy 16 településen megy át európai út (4-es kód), 40 településen megy át országút (3-as kód), 98 településen megy át megyei út (2-es kód) és 12 településen nem megy át semmilyen út. A maradék 120 településen értelemszerűen csak községi út megy át. Ezek a poligonok még mindig a <Null> értéket hordozzák a kód mezőben. Vajon hogyan tudnánk ezt átkódolni mondjuk 0-ra? Az így kapott eredményeket megjeleníthetjük majd a térképen, de most arra vagyunk kíváncsiak, hogy van-e olyan községközpont, amin nem halad át megyei út!
54. A települések közigazgatási rangjának kódolása a „type” mezőben van. A községközpontok kódja 22. Egyszerű dolgunk van. Egy attribútum kereséssel megkeressük azokat a településeket ahol a „kód” =1 ÉS (AND) a „type”=22. Kiderül, hogy 4 ilyen település van. Amennyiben szándékunk van ezeket külön megjeleníteni, kódoljuk át egy másik értékre, ez legyen mondjuk 5. Természetesen az eredeti értékük 0 volt. Ezzel elkészítettük az összes, megjeleníteni kívánt adatot, fogjunk neki a térkép elkészítésének.

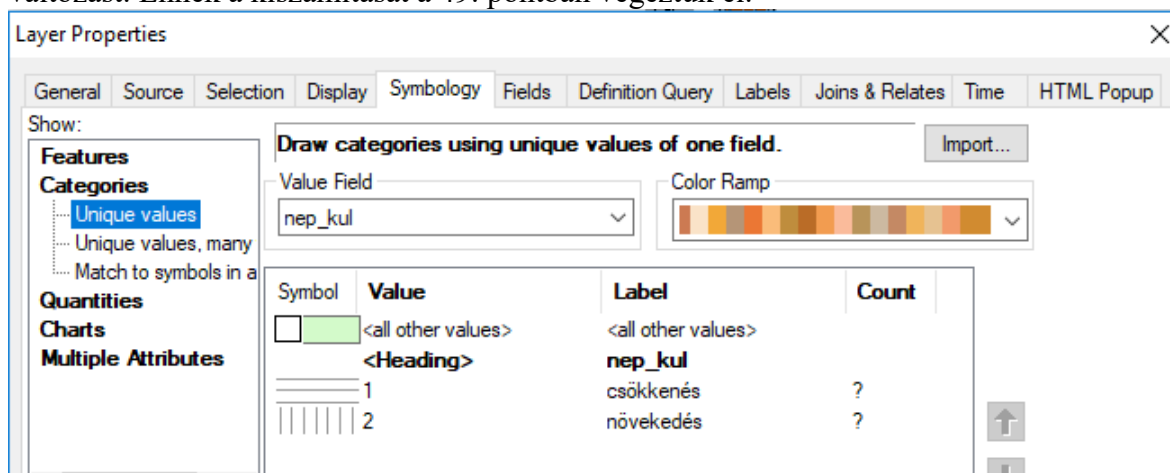
### Adatok megjelenítése, térképkészítés

Az előzőekben valahol már említettük, hogy mely adatokat szeretnénk megjeleníteni térképen. Jegyezzük meg, hogy a megjelenítés során arra kell törekednünk, hogy minden egyértelmű és jól látható kéne legyen. Nem mindegy a rétegek sorrendje és a kiválasztott szimbólum sem. Azt se feledjük el, hogy egy adott tematikus réteget többször is betölthetünk. Az egyszerűség kedvéért kevés szöveggel és több képpel illusztráljuk a további lépéseket.

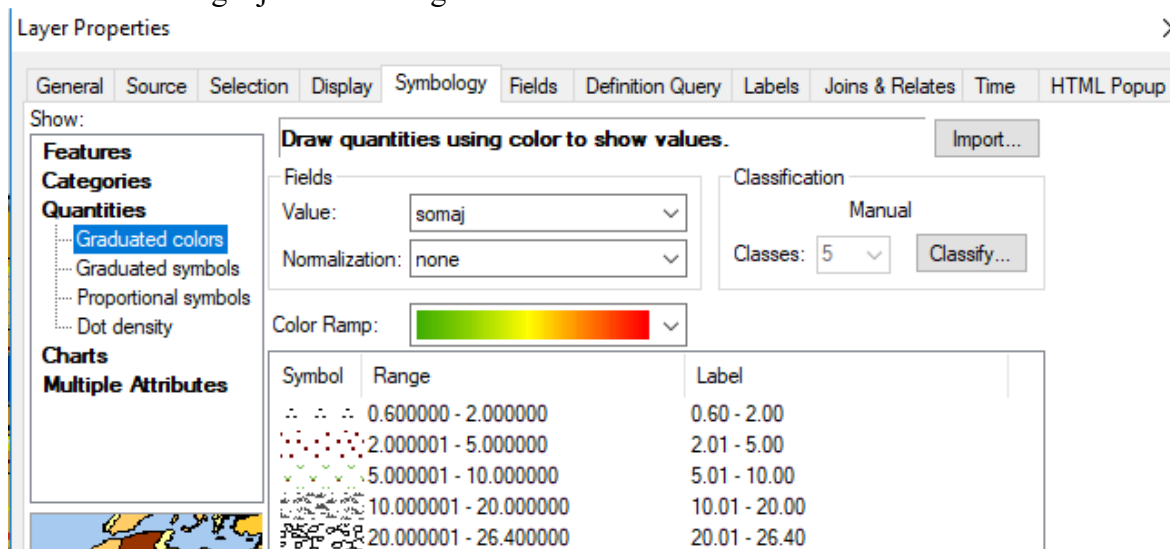
55. Készülő térképünk alapját mindenképpen a községek képezik. A rendelkezésre álló sok mező közül elsőnek a népsűrűséget fogjuk megjeleníteni valamilyen színskálával az alábbi ábrának megfelelően. Természetesen más színeket is lehet választani, de semmiképpen ne válasszunk túl telített színeket, hiszen más információkat is meg szeretnénk jeleníteni.



56. A következő réteg szintén a községek lesz, ezért töltjük be újra vagy az egyszerű réteg hozzáadással (Add Data) vagy a Copy-Paste segítségével. Most jelenítsük meg a népesség változást. Ennek a kiszámítását a 49. pontban végeztük el.

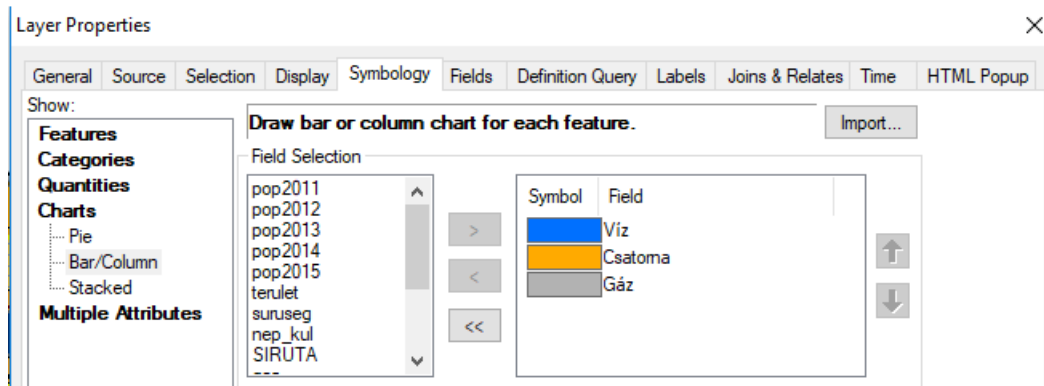


57. A harmadik adat szintén a községekhez kapcsolódik, így az újra betöltés után a munkanélküliséget jelenítsük meg.

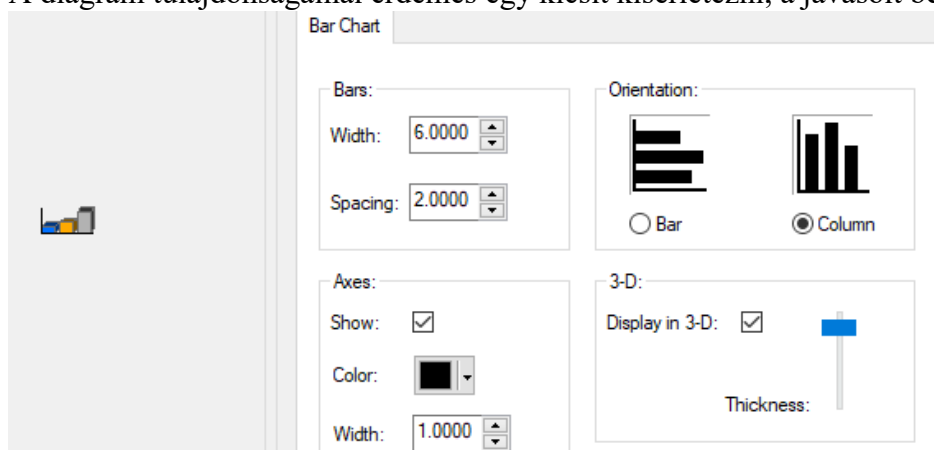




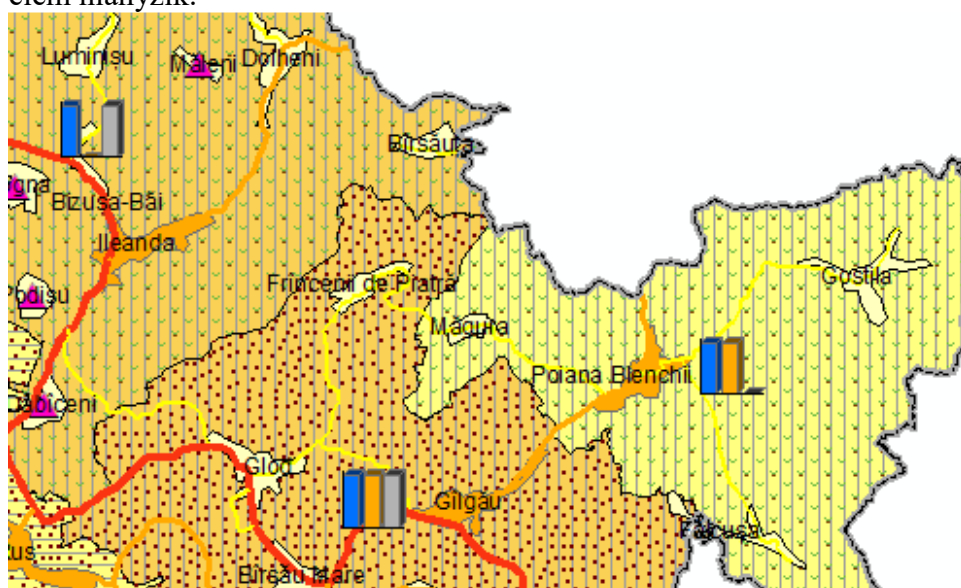
58. A következő, szintén községi szintű információ a felszereltségre vonatkozik. Ennek az előállítását a 41 – 45 pontban írtuk le. Mivel egyszerre három információt akarunk megjeleníteni és valójában csak kódjaink vannak, a diagram lehetőséget fogjuk használni. Valójában csak az adott technikai elem léte vagy nem léte érdekel, így az oszlopdiagram lesz a célszerű.



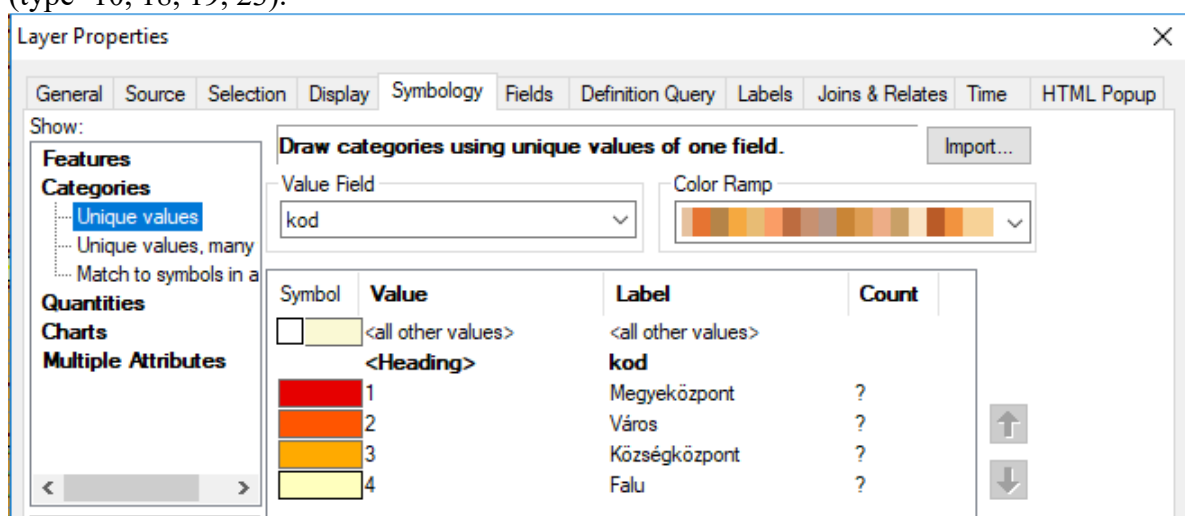
A diagram tulajdonságainál érdemes egy kicsit kísérletezni, a javasolt beállítások itt láthatók



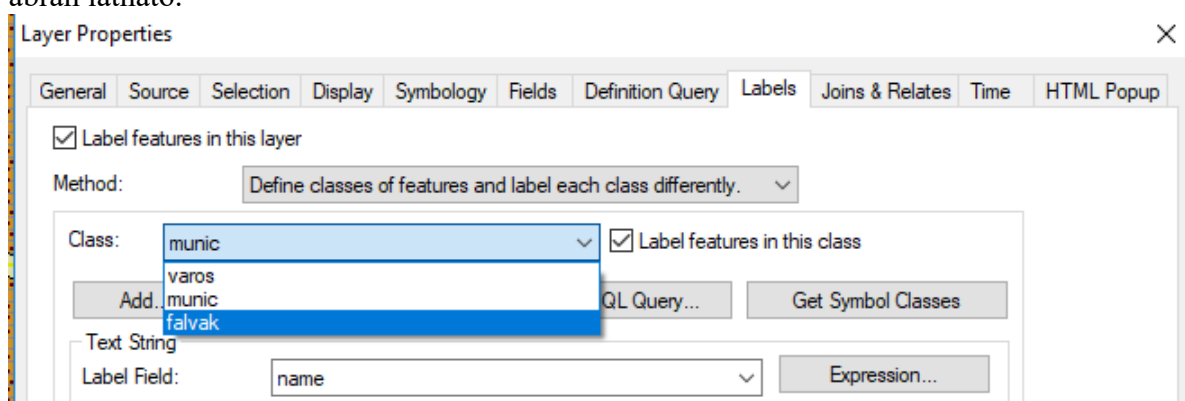
Érdekes közelebbről megnézni egy konkrét példát ahol mindhárom oszlop jelenléte arra utal, hogy az adott községben van víz, gáz és csatorna is. Ahol csak két oszlop van, ott valamelyik elem hiányzik.



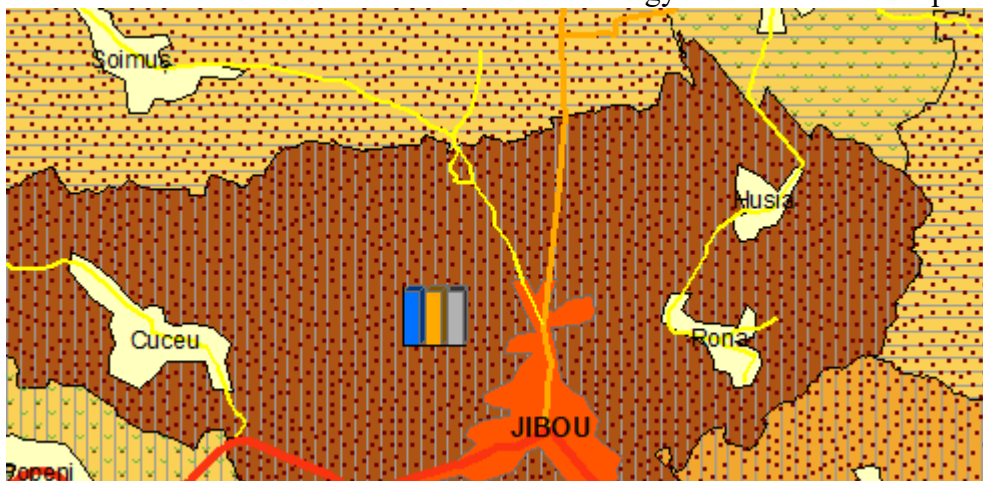
59. A következő réteg a települések poligon rétege lesz. Itt a települések közigazgatási rangját jelenítjük meg. Az ehhez szükséges információ valójában a „type” mezőben van. De mivel itt túl sok változat van, egy új „kod” mezőbe összevontuk a sok fajta típust a következők szerint: 1 – megyeközpont (type=9); 2 – város (type=17); 3 – községközpont (type=22) és 4 – falu (type=10, 18, 19, 23).



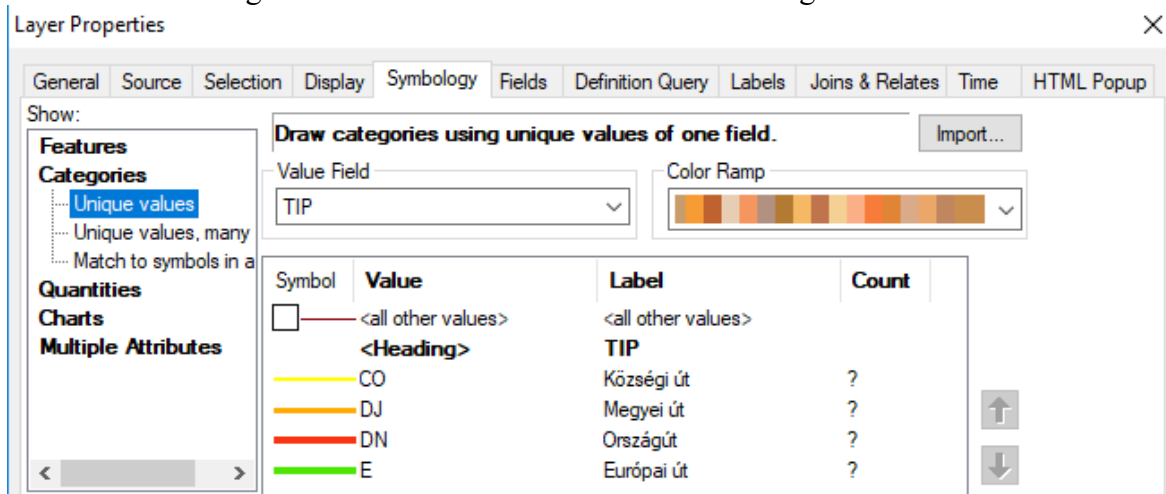
60. A településeket érdemes felcímkézni. Próbáljuk ki az eltérő címkék használatát. A beállításokat a réteg tulajdonságainál a címke (Label) fülnél lehet megtenni, amint az alábbi ábrán látható.



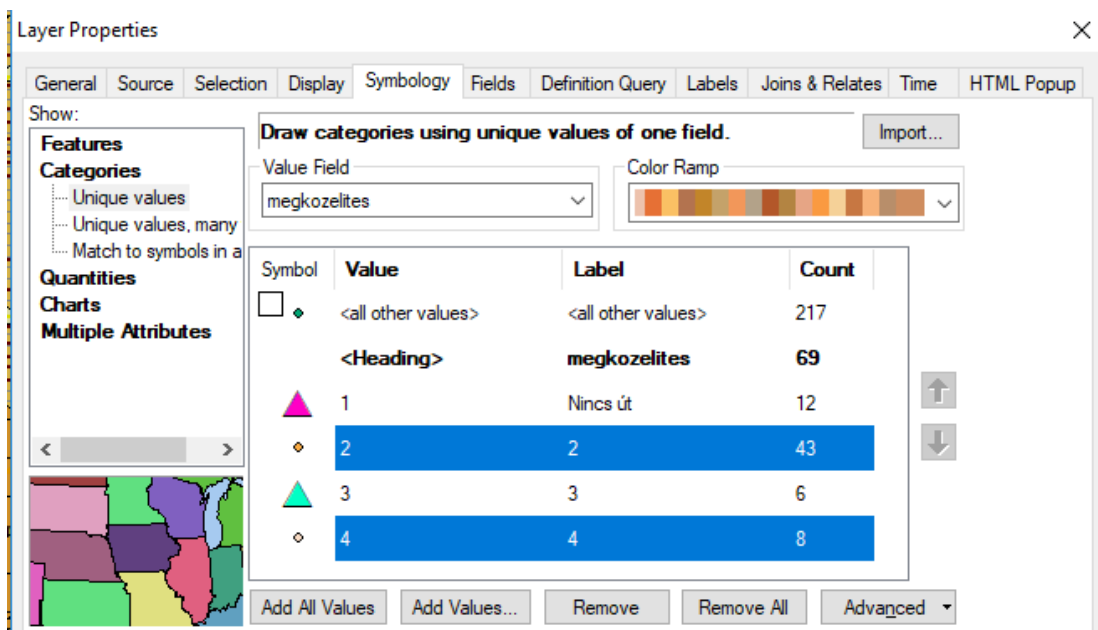
Minden címke osztályt más betűmérettel és stílussal jelenítjük meg, ez nem látszik az ábrán: municípium – 12-es méret, félkövér; város – 10-es méret, félkövér; falvak – 9-es méret, normál. A betűk méretével érdemes kísérletezni figyelembe véve a térkép végső méretét.



61. A következő réteg az úthálózat lesz a következő ábrának megfelelően:



62. Legfelülre tegyük a megyehatár réteget a vonal számára a Boundary, County vonaltípust választva.
63. De még mindig hátra van egy réteg, a bonyolult megközelítési vizsgálat eredménye. A vizsgálatot a települések polygon rétegén végeztük, oda írtuk be a kódokat is. Csakhogy a települések réteget már megjelenítettük közigazgatási rang szerint. A megközelíthetőséget eltérő pont szimbólumokkal kéne megjeleníteni, de a poligonokhoz nem tudunk pont szimbólumot rendelni. A megoldás az lesz, hogy a poligonokat ponttá alakítjuk a **Toolbox → Data Management → Features → Feature to Point** művelettel. Az átalakítás eredményét az adatbázisba mentjük majd betöltjük a térképbe. Valójában a poligonok súlypontjába kerül egy pont. Mivel ez egy pont állomány, mindenképpen a legtetejére kerül. Válasszuk ki a megközelítési kódokat és különböző szimbólummal jelenítsük meg őket. A lenti példában csak az út nélküli településeket valamint a megyei út nélküli községközpontokat jelenítjük meg. A nem kívánt értékeket – mivel az adott mezőben nem csak ez a két érték van – kijelölés után a Remove gombot megnyomva ki tudjuk venni a jelmagyarázatból.



64. Végezetül készítsük el a térképet. Minden beállítást, megírást végezzünk el a tartalomjegyzékben. Térjünk át a Layout nézetre és az oldalbeállításnál válasszuk az A3 oldalméretet. Érdekes megnézni az 1:1 lehetőséggel munkánk tényleges méretét és szükség esetén változtassuk meg a címkék betűméretét, a pontszimbólumok méretét. A továbbiakban kinagyítva látható a végleges jelmagyarázat részletekben. Ennek elkészítése már a tisztelt csapat kisujjában van...

#### Települések megközelítése

▲ Nincs út

▲ Község megyei út nélkül

#### Felszereltség

0.5

Víz

Csatorna

Gáz

#### Települések

Megyeközpont

Város

Községközpont

Falu

#### Úthálózat

Községi út

Megyei út

Országút

Európai út

#### Munkanélküliség

%

0.60 - 2.00

2.01 - 5.00

5.01 - 10.00

10.01 - 20.00

20.01 - 26.40

#### Népesség változás

csökkenés

növekedés

#### Népsűrűség

lakos/nkm

<25.00

25.01 - 50.00

50.01 - 100.00

100.01 - 300.00

774.42

