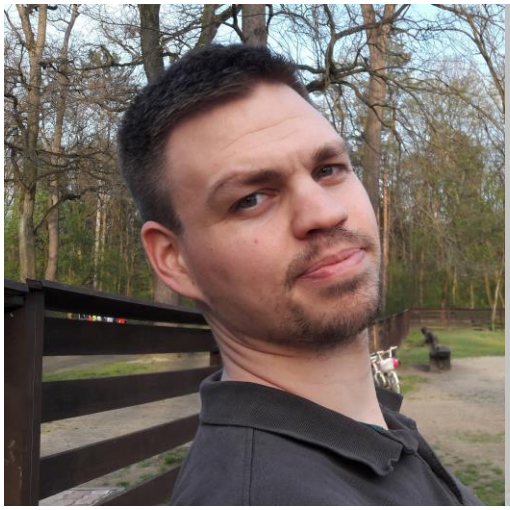



Dr. Hegedűs Péter, Dr. Ferenc Rudolf

Nagyméretű adatbázisok

Jelen tananyag a Szegedi Tudományegyetemen készült az Európai Unió támogatásával.

Projekt azonosító: EFOP-3.4.3-16-2016-00014

Szegedi Tudományegyetem
Cím: 6720 Szeged, Dugonics tér 13.
www.u-szeged.hu
www.szechenyi2020.hu

Adat feldolgozás és transzformáció

Összefoglalás

Ez az olvasó lecke bevezet az adatfeldolgozás és transzformáció világában. Megismerjük az adat transzformációk két leggyakoribb megközelítését, amelyet az adattudósok alkalmaznak: ETL és data wrangling. Az egyes módszerek megismerése mellett azt is megtanulja az olvasó, hogy mik a főbb közös tulajdonságaik és mik a különbözőségeik. Információt szerezhetünk azokról a konkrét eszközökről is, amelyek rendelkezésre állnak ETL vagy data wrangling feladatok megoldásához.
A lecke fejezetei:

- 1. fejezet: **Adatok lehetséges formátumai, transzformáció szükségessége és főbb típusai (olvasó)**
- 2. fejezet: **Az ETL módszer jellemzői és eszközei (olvasó)**
- 3. fejezet: **Az data wrangling módszer jellemzői és eszközei (olvasó)**

Téma típusa: **elméleti**
Olvasási idő: **40 perc**

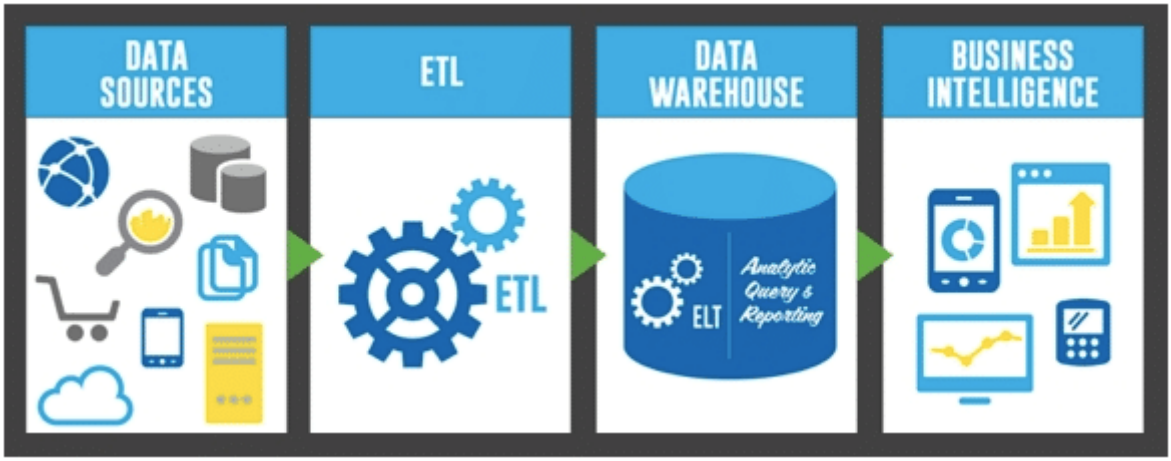
📖 1. fejezet
Adatok formátuma, transzformációs megközelítések

A korábbi leckékben átnéztük, hogy a BigData adatforrásokból milyen típusú adatok kerülhetnek ki:

- Strukturált
- Nem strukturált
- Részben strukturált

A BigData 4V jellemzőinek egyike a `variety`, azaz a megjelenő adatok formátuma heterogén, sokszor előfordul, hogy sok egymástól teljesen különböző formában kapjuk meg a bemenő adatainkat, amiket a tárolás és feldolgozás módjától függően transzformálnunk, egységesítenünk kell mielőtt használni tudnánk egy BigData megoldásban. Alapvetően két fajta adat transzformációt különböztethetünk meg attól függően, hogy az adatok milyen forrásból érkeznek, és mi a transzformáció célja:

- Egy adat mozgatása egyik BigData rendszerből a másikba, vagy hagyományos adat forrásból valamilyen BigData rendszerbe. Ez a lépés az adatok kinyerését igényli, azok transzformálását, majd a célhelyen történő betöltését, ami ún. ETL (Extract Transform Load) eszközökkel végezhető el. Ezek az eszközök jórészt automatizáltak, és kezdetben adattárházak közti adatcsere megvalósítására szolgáltak, ám napjainkban már sokkal szélesebb körű a felhasználásuk módja.

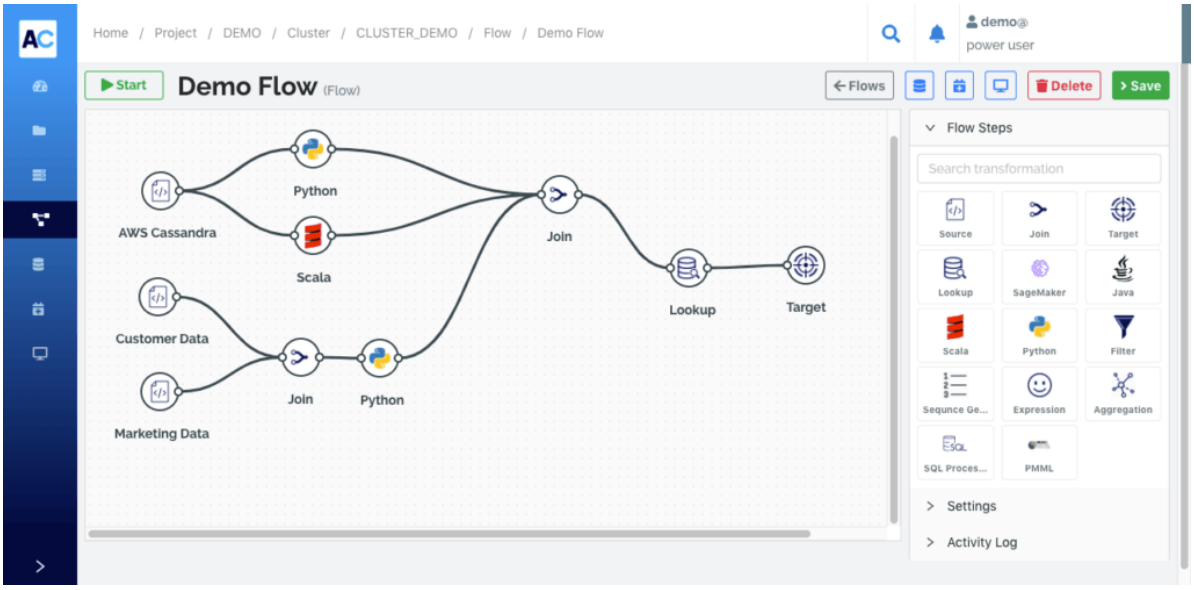


> https://research.aimultiple.com/wp-content/uploads/2018/04/etl-processes.png

- A különböző adatforrásból érkező adatok összefésülése/egységesítése tárolás vagy feldolgozás előtt. Tipikusan manuális programozói munka, hiszen a feldolgozandó adatok formátuma teljesen tetszőleges lehet, akár nem strukturált is. Ezt a módszert `data wrangling`-nek nevezzük.

```
Példa: adott egy kérdőív, amit egy webalkalmazás szolgál ki. Az egyes
kitöltéseket a rendszer egy NoSQL adatbázisba gyűjti, és óránként automatikusan
kiexportálja az elmúlt egy óra kitöltéseit egy-egy JSON fájlban. A feladatunk az
összes eddigi kitöltés adatainak elemzése és vizualizálása. Ehhez szükséges lehet
egy lépés, ami során az összes JSON fájlt összefésüljük egy CSV táblába, amit
aztán elemzünk/vizualizálunk.
```

Természetesen a fenti két dolog nem teljesen elkülöníthető, és sokszor vegyesen is használhatók. Például lehetnek olyan ETL eszközök, melyeknek a transzformációs lépésébe saját `data wrangling` program illeszthető, mint például a lenti ábrán szemléltetett folyamat. Igaz ez a másik irányban is, azaz az ETL eszközöket használhatják például arra, hogy az adatokat elérhetővé tegyék az üzleti elemzők számára későbbi `data wrangling` elvégzéséhez.



> https://avikcloud.com/wp-content/uploads/2020/01/Avik-Flow-copy-1024x500.png

A két módszer közti alapvető különbségeket így lehetne összefoglalni:

- **Felhasználók**

    - `Data wrangling`: az adatokat feldolgozó, azt értő és abból értékes információt kinyerő szereplők használják (megfelelő programozói támogatás hiányában ezt korábban spreadsheet vagy BI eszközökkel végezték)
    - `ETL`: IT infrastruktúrát üzemeltetők használják, hogy az adatokat a specifikált formában elérhetővé tegyék a megfelelő platformon, de az üzleti szereplők nagyon ritkán alkalmazzák
- **Adatok**
    - `Data wrangling`: tetszőleges formátumú (strukturált és nem strukturált) és méretű adatok feldolgozhatók a megfelelő eszközökkel
    - `ETL`: leginkább jól strukturált adatok mozgatására tervezték, a nagyon intenzív adat formátum transzformálás nem az erőssége
- **Használati esetek**
    - `Data wrangling`: felfedező adatelemzés, új források feltérképezése, analizálási lehetőségek feltárása a cél, vagy már meglévő elemzések hatékonyabbá tétele
    - `ETL`: elsődlegesen adatok központi/vállalati tárhelyre mozgatása a cél adat elemzések készítéséhez és üzleti alkalmazásokhoz

📖 2. fejezet

Extract Transform Load (ETL)

Egy általános folyamat (és az azt megvalósító eszköz), amely célja az adatok különböző forrásból történő begyűjtése és átmásolása egy célhelyre, amely a forrástól különböző formában és/vagy környezetben tárolja az adatokat. Nem újkeletű dolog, az 1970-es években vált népszerűvé, és leginkább az adattárházakban alkalmazták kötegelt adatfeldolgozás keretein belül. Az ETL eszközök három fő funkciót ötvöznek:

- **Extract:** adatok beolvasása az adatforrás(ok)ból
- **Transform:** adatok konvertálása a cél adattár formátumára (szabály alapú transzformáció, összefésülés, stb.)
- **Load:** konvertált adat betöltése a cél adattárba

Az ETL szerepe azonban folyamatosan változik, és manapság sokkal szélesebb körben használt, mint a csak adattárházakon belüli adatmozgatás. Az ETL rendszerek manapság támogatják a különböző adattárak, BI platformok (üzleti intelligencia alkalmazások), felhő vagy éppen a Hadoop klaszterek közötti adatmozgatást. Az ETL nem első sorban BigData technológia, de napjainkra már a BigData platformokat is támogatják. Az ETL működése kötegelt és párhuzamosított. Mivel általában hatalmas mennyiségű adatot kell mozgatni, ezért amint egy extract fázis véget ér és elkezdődik az adatok transzformálás, azzal párhuzamosan egy újabb köteg extract indul. Ugyanígy amint előállt egy kötegelt transform lépés eredménye a betöltéssel párhuzamosan újabb transzformálás kezdődik.

Extract

Az ETL folyamat első és egyben legkritikusabb része. Az adatok megfelelő kinyerése az összes későbbi fázis sikeres végrehajtást befolyásolja. A legtöbb esetben több különböző adatforrás használata szükséges, amelyek teljesen különböző formában szolgáltathatják az adatokat, pl. relációs adatbázisok, XML, JSON, hagyományos fájlok, stb. Ezek mellett azonban egyéb, akár kívülről érkező adatforrásokat is használhat, mint például web oldalak beolvasása. Az extract lépés két féle módon történhet:

1. Az adatokat először közös formátumra hozzuk és eltároljuk egy ideiglenes tárban (ez a tipikus eset)

2. Nincs ideiglenes tárolás, a kinyert adatokat stream szerűen rögtön a cél adattárba töltjük

Az extract lépés tartalmazza az adatok validálását, minőségi ellenőrzését is. Ebben a fázisban ellenőrizzük, hogy a betöltött értékek megfelelnek-e az adott domain-re jellemző szabályoknak. Amennyiben nem megfelelő a formátuma vagy tartalma az adatnak, azt nem dolgozzuk fel a továbbiakban, és lehetőség szerint a forrás rendszer felé is visszajelzést küldünk, hogy az adatok javítása megtörténhessen.

Transform

A transzformáció során szabályok vagy függvények egy sorozatát alkalmazzuk a kinyert és egységesített köztes adatokra, amellyel előállítjuk a cél adattárba betöltendő végleges adatot. Néhány tipikus transzformációs lépés:

- Bizonyos adatoszlopok kiválasztása/szűrése
- Kódolt értékek helyettesítése (pl. az adatforrás "1"-es értékkel jelöli a "Férfi" opciót, "2"-vel a "Nő" opciót, de a cél adattárban "F", "N" kódolás kell)
- Szabad formátumú adat kódolása (pl. "Férfi" -> "F")
- Új számított érték készítése (pl. érték = darab * egység ár)
- Adatok rendezése
- Több különböző adat összefésülése
- Aggregált értékek számítása
- Sor-oszlop csere (transzponálás)
- Egy oszlop több oszlopra történő bontása (pl. vesszővel elválasztott értékek egy string-ben -> több külön oszlop)
- Adatok ellenőrzése/validálása referencia alapján

Load

Ha elkészültek a transzformált adatok, a load lépés során töltjük be ezeket a cél adattárba. Ez az adattár lehet egy egyszerű tagolt formátumú sima fájltól kezdve egy adattárházig szinte bármi. A load sokszor nem üres adattárba tölti az értékeket, hanem egy már ott lévő adathalmazt frissít. Ez a frissítés lehet akár napi, heti, havi, stb. rendszerességű, és a load a korábbi értékeket felülírhatja, vagy egyszerűen minden korábbi értéket megőrizve betölti a legfrissebb adatokat.

Tipikus ETL felhasználási esetek

Az alábbiakban az ETL néhány tipikus használatát mutatjuk be példákon keresztül:

- Tegyük fel, hogy egy pénzügyi cég több részlege is nyilvántartást vezet az ügyfelekről. Minden részleg más-más adatot tárolhat és más-más szempontok szerint listázhatja az ügyfeleket. Az ETL nagyon jól használható arra, hogy ezeket az adatokat begyűjtse, egységesítse és rendszeresen egy központi adatbázisba töltse.
- Az ETL hasznos lehet akkor is például, ha egy vállalat egyik alkalmazásról átáll egy másikra, de az adatokat szeretnék megőrizni. A régi és új alkalmazás teljesen más adattárolási módszert alkalmazhat, ezért az ETL használható az adatok migrálására.
- Egy számlázó rendszerben tárolt adatokat többnyire a számla kiállító program használja, de az ETL segítségével értékes információkat juttathatunk el a nyers adatokról más részlegek felé, ahol például HR vagy üzemeltetési célokra is szolgálhat.

ETL megvalósítások

A teljesség igénye nélkül felsorolunk néhány ETL eszközt, amelyek a fenti funkcionalitással rendelkeznek:

- *Hevo*: https://hevodata.com/?utm_source=softwaretestinghelp&utm_medium=etl_tools&utm_campaign=listing
- *Xplenty*: https://www.xplenty.com/?utm_source=STH&utm_medium=Referral&utm_campaign=Best_ETL_Tools
- *Skyvia*: https://skyvia.com/?utm_source=softwaretestinghelp.com&utm_medium=referral&utm_campaign=cc_listing_softwaretestinghelp.com
- *DBConvert Studio*: https://dbconvert.com/dbconvert-studio
- *Voracity*: http://www.iri.com/products/voracity
- *IBM Infosphere*: https://www.ibm.com/us-en/marketplace/infosphere-information-server
- *Oracle Data Integrator*: https://www.oracle.com/middleware/data-integration/management-pack-for-odi/
- *Microsoft SSIS*: https://docs.microsoft.com/en-us/sql/integration-services/sql-server-integration-services
- *Talend*: https://www.talend.com/products/talend-open-studio/
- *SAP Business Object Integrator*: https://www.sap.com/india/products/data-services.html
- *Apache Sqoop (Hadoop specifikus)*: https://sqoop.apache.org/

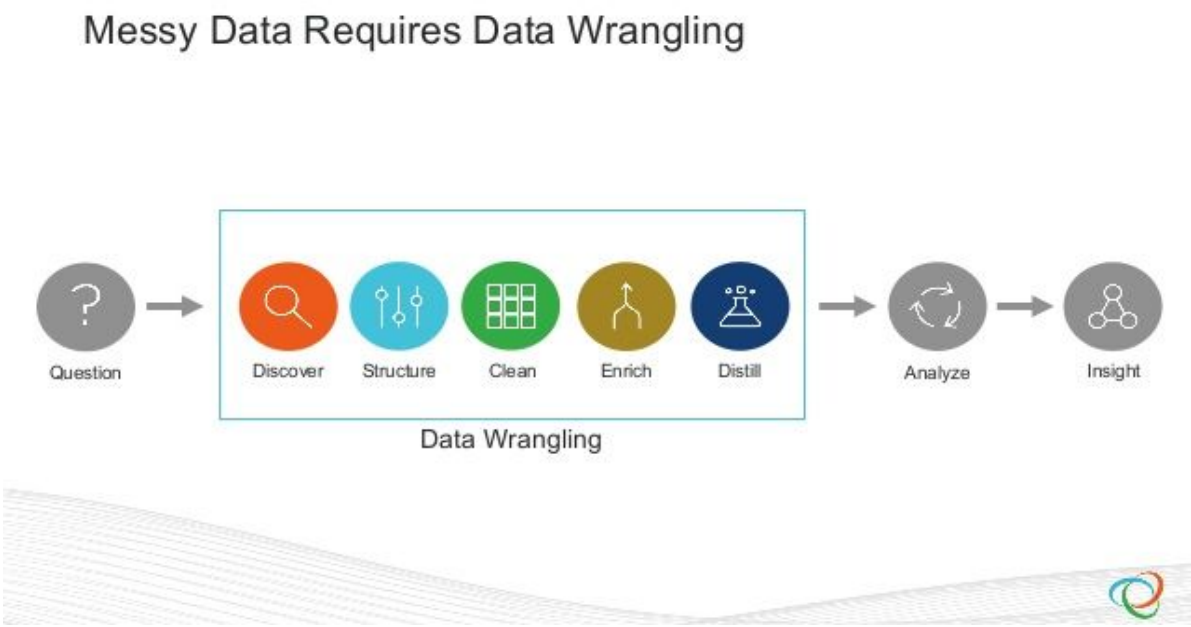
📖 3. fejezet

Data Wrangling

Ahogy már a bevezetőben láttuk, maga a data wrangling folyamat nagyon hasonló az ETL folyamatához, csak más céllal történik. Data wrangling esetében (szokták még `data munging` néven is illetni) az elsődleges célunk a különböző forrásból érkező adatok "nyers" adatokat olyan formára hozni, amely alkalmasabb üzleti alkalmazások készítéséhez, elemzésekhez, vizualizációhoz, stb. Data wrangling esetében is megtörténik az extract és transform lépés, mint az ETL esetén, de legtöbbször az eredményeket nem eltároljuk, hanem felhasználjuk az adatok elemzéséhez, alkalmazások építéséhez. Egy data wrangling program az alábbi tipikus műveleteket hajthatja végre az adatokon:

- Adatkinyerés
- Adatok elemzése/parsing
- Adatok összeillesztése/joining
- Értékek standardizálása
- Adatok augmentálása (más forrásból érkező adattal történő kiegészítés)
- Adattisztítás
- Szűrés és egységesítés

Az alábbi ábra azt mutatja meg, hogyan illeszkedik bele a data wrangling egy tipikus BigData alkalmazás építésébe.



> https://i.pinimg.com/736x/2a/39/02/2a390250ced51ae14222563687616ba5.jpg

Tipikus data wrangling eszközök

Míg az ETL viszonylag jól formalizált folyamat számos vizuális, adatfolyam leíró eszköz támogatással, addig a data wrangling műveletet tipikusan kézzel szokták elvégezni (értsd ez alatt saját adatfeldolgozó szkriptek készítését). Természetesen ebben az esetben is számos olyan könyvtár áll rendelkezésre, amely megkönnyíti ennek a feladatnak az elvégzését, de persze léteznek olyan eszközök, amelyek kifejezetten data wrangling programozás nélküli elvégzését támogatják.

Néhány konkrét módszer és eszköz a data wrangling elvégzéséhez:

- *Excel*: igen, egy klasszikus táblázatkezelő szoftver is alkalmas lehet data wrangling feladatok elvégzéséhez, sőt gyors adatelemzésekhez kiváló kiindulási alap lehet.
- *OpenRefine*: szofisztikáltabb módszereket ad, de már programozói tudás szükséges hozzá
- *Google DataPrep*: feltáráshoz, adattisztításhoz és előkészítéshez használható eszköz
- *Tabula*: fizetős eszköz, cserébe minden féle adatfeldolgozási feladathoz használható
- *DataWrangler*: adattisztításhoz és transzformációhoz használható eszköz
- *CSVKit*: adatkonverziót támogató eszköz
- *Szkriptnyelvek*: néhány szkriptnyelv, első sorban az **R** és **Python** az adatfeldolgozás de facto eszköze. A nyelv jellege mellett számos olyan könyvtár létezése teszi kifejezetten vonzóvá az adatfeldolgozáshoz, mint a NumPy, Pandas, Matplotlib, Plotly, Theano, stb. Python esetén vagy a Dplyr, Purrr, Splitstackshape, JSOnline, Magrittr R esetén

Data wrangling alkalmazása BigData kontextusban

A data wrangling nem kifejezetten BigData technológia, az adattudomány egy fontos eszköze, amelyet tetszőleges méretű adathalmazok esetén bevethetünk. BigData kontextusban alapvetően a következő problémákra nyújt megoldást:

- *Adat feltérképezés (exploration):* segít a rendelkezésre álló adat megértésében és annak megállapításában, hogy milyen érték nyerhető ki belőle (Value, Veracity)
- *Adategységesítés/strukturálás*: BigData esetén az adatok mindig zavarosan, mindenféle formátumban állnak rendelkezésre, amit data wrangling segítségével rendszerezhetünk, hogy átlássuk milyen adat is áll a rendelkezésünkre, és mire is lehet azt használni (Variety)

- *Adatok zaj, hiba és hiányzó információ szűrése*: hibák és hiányzó értékek mindig vannak az adatainkban, amik emberi hibák, félrecímkézések vagy technikai hibák nyomán kerülnek be, amik data wrangling során kiszűrhetők (Veracity)

Ellenőrző kérdések

1. Miért van szükség az adatok transzformációjára?
2. Mi az adat transzformáció két fő megközelítése? Mondj 1-1 példát is rájuk!
3. Mik a két fő adat transzformációs módszer közti fő különbségek?
4. Mi az ETL folyamat három fő lépése?
5. Mondj néhány tipikus esetet, amikor az ETL használata indokolt!
6. Mondj néhány tipikus esetet, amikor a data wrangling használata indokolt!
7. Milyen ETL eszközöket ismersz?
8. Milyen data wrangling eszközöket ismersz?

Referenciák

[1] https://www.dummies.com/programming/big-data/data-science/the-role-of-traditional-etl-in-big-data/

[2] https://tdwi.org/articles/2017/02/10/data-wrangling-and-etl-differences.aspx

[3] https://www.dummies.com/programming/big-data/data-science/the-role-of-traditional-etl-in-big-data/

[4] https://www.talend.com/resources/data-wrangling-vs-etl/

[5] https://en.wikipedia.org/wiki/Extract,_load,_transform

[6] https://en.wikipedia.org/wiki/Data_wrangling

[7] https://theappsolutions.com/blog/development/data-wrangling-guide-to-data-preparation/