



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

NEUMANN JÁNOS
INFORMATIKAI KAR

SZAKDOLGOZAT

OE-NIK

2022

Hallgató neve:

Hallgató törzskönyvi száma:

Pádár Miklós

T/005470/ FI12904/N

Óbudai Egyetem
Neumann János Informatikai Kar
Kiberfizikai Rendszerek Intézet

SZAKDOLGOZAT FELADATLAP

Hallgató neve:	Pádár Miklós
Törzskönyvi száma:	T/005470/F112904/N
Neptun kódja:	

A dolgozat címe:

Felsőoktatási Felvételi Eljárás BI rendszerének kialakítása, adattárházzal
Higher Educational Process in Hungary with BI system and DWH

Intézményi konzulens:	Dr. Fleiner Rita
Külső konzulens:	

Beadási határidő:	2022. május 15.
-------------------	-----------------

A záróvizsga tárgyai:	Számítógép architektúrák Big Data és üzleti intelligencia specilaizáció
-----------------------	---

A feladat

A szakdolgozat során a már évtizedek óta működő felsőoktatási felvételi eljárás során keletkező adatok elérhetőségének javítása a cél a szakmai kollégák részére. A kitűzött cél BI rendszer kialakításával érhető el. Ennek a fejlesztésnek a hatására könnyebben, gyorsabban hozzáférnek a megfelelő személyek és szervezeti egységek az oktatáspolitikai irányításához és tervezéséhez szükséges fontosabb adatokhoz.

Feladatunk a dolgozat során a meglévő, meghatározott adatforrásokból és informatikai rendszerekből az adatok átvitelének megtervezése és megvalósítása ETL rétegek segítségével adattárházba, majd az adattárházból automatikusan frissülő rétegek, kimutatások készítése a felsővezetők számára. További teendő egy BI portál kialakítása, mely segítségével az adatok vizualizációja lesz megvalósítható megfelelő jogosultság ellenében.

A dolgozatnak tartalmaznia kell:

- az adatforrás réteg meghatározását, illetve a rendszerek és a kívánt adatok/táblák kiválasztását,
- ETL rétegek részletes megtervezését és megalkotását,
- adattárház tervezését és megvalósítását,
- az adatpiac számára a megfelelő információk kiválasztását és megfelelő formára alakítását,
- BI alkalmazás/portál megtervezését és megalkotását,
- adatvizualizáció megvalósítását.



Fleiner Rita
.....
Dr. Fleiner Rita
intézetigazgató

A szakdolgozat elévülésének határideje: **2024. május 15.**
(OE TVSz 55.§ szerint)

A dolgozatot beadásra alkalmasnak tartom:

.....
külső konzulens

Fleiner Rita
.....
intézményi konzulens

A feladat

A szakdolgozat során a már évtizedek óta működő felsőoktatási felvételi eljárás során keletkező adatok elérhetőségének javítása a cél a szakmai kollégák részére. A kitűzött cél BI rendszer kialakításával érhető el. Ennek a fejlesztésnek a hatására könnyebben, gyorsabban hozzáférnek a megfelelő személyek és szervezeti egységek az oktatáspolitikai irányításához és tervezéséhez szükséges fontosabb adatokhoz.

Feladatunk a dolgozat során a meglévő, meghatározott adatforrásokból és informatikai rendszerekből az adatok átvitelének megtervezése és megvalósítása ETL rétegek segítségével adattárházba, majd az adattárházból automatikusan frissülő rétegek, kimutatások készítése a felsővezetők számára. További teendő egy BI portál kialakítása, mely segítségével az adatok vizualizációja lesz megvalósítható megfelelő jogosultság ellenében.

A dolgozatnak tartalmaznia kell:

- az adatforrás réteg meghatározását, illetve a rendszerek és a kívánt adatok/táblák kiválasztását,
- ETL rétegek részletes megtervezését és megalkotását,
- adattárház tervezését és megvalósítását,
- az adatpiac számára a megfelelő információk kiválasztását és megfelelő formára alakítását,
- BI alkalmazás/portál megtervezését és megalkotását,
- adatvizualizáció megvalósítását.



Fleiner Rita
.....
Dr. Fleiner Rita
intézetigazgató

A szakdolgozat elvégzésének határideje: **2024. május 15.**
(OE TVSz 55.§ szerint)

A dolgozatot beadásra alkalmasnak tartom:

.....
külső konzulens

.....
intézményi konzulens



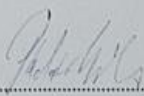
ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar

HALLGATÓI NYILATKOZAT

Alulírott hallgató kijelentem, hogy a szakdolgozat / diplomamunka saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült szakdolgozatomban / diplomamunkámban található eredményeket az egyetem és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja.

Budapest, 2022. május 15.


.....
hallgató aláírása



KONZULTÁCIÓS NAPLÓ

Hallgató neve: Neptun kód: Tagozat:
Pádár Miklós ESTI.....
Telefon: Levelezési cím (pl: lakcím):

.....

Szakdolgozat / Diplomamunka¹ címe magyarul:

Felsőoktatási Felvételi Eljárás BI rendszerének kialakítása adattárházzal.....

Szakdolgozat / Diplomamunka² címe angolul:

Higher Educational Process in Hungary with BI system and DWH.....

Intézményi konzulens: Külső konzulens:

Dr. Fleiner Rita
.....

Kérjük, hogy az adatokat nyomtatott nagybetűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	2021.09.26.	Szakdolgozat témájának megbeszélése	Fleiner Rita
2.	2021.10.05.	Feladatlap megbeszélése	Fleiner Rita
3.	2021.11.29.	Irodalomkutatás megbeszélése	Fleiner Rita
4.	2021.12.06.	Dokumentumok beadásának megbeszélése	Fleiner Rita

A Konzultációs naplót összesen 4 alkalommal, az egyes konzultációk alkalmával kell láttamoztatni bármelyik konzulenssel.

A hallgató a Szakdolgozat I. / Szakdolgozat II. (BSc) vagy Diplomamunka 1 / Diplomamunka 2 / Diplomamunka 3 / Diplomamunka 4³ tantárgy követelményét teljesítette, beszámolóra / védésre⁴ bocsátható.

Budapest, 2021.12.10.

Fleiner Rita
intézményi konzulens



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar

KONZULTÁCIÓS NAPLÓ

Hallgató neve: PÁDÁR MIKLÓS Neptun kód: [REDACTED] Tagozat: ESTI

Telefon: [REDACTED]

Szakdolgozat címe magyarul:

FELSŐOKTATÁSI FELVÉTELI ELJÁRÁS BI RENDSZERÉNEK KIALAKÍTÁSA
ADATTÁRHÁZZAL

Szakdolgozat címe angolul:

HIGHER EDUCATION PROCESS IN HUNGARY WITH BI SYSTEM AND DWH

Intézményi konzulens: FELINER RITA Külső konzulens: -

Kérjük, hogy az adatokat nyomtatott nagybetűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	2022.02.26.	Irodalomkutatás véglegesítése	Fleiner Rita
2.	2022.03.05.	Követelmény specifikáció, rendszerterv	Fleiner Rita
3.	2022.04.29.	Kódolás, eredmények összegzése	Fleiner Rita
4.	2022.05.06.	Dokumentumok beadásának megbeszélése	Fleiner Rita

A Konzultációs naplót összesen 4 alkalommal, az egyes konzultációk alkalmával kell láttamoztatni bármelyik konzulenssel.

A hallgató a Szakdolgozat I. / Szakdolgozat II. (BSc) vagy Diplomamunka 1 / Diplomamunka 2 / Diplomamunka 3 / Diplomamunka 4¹ tantárgy követelményét teljesítette, beszámolóra / védésre² bocsátható.

Budapest, 2022.05.10.

Fleiner Rita
intézményi konzulens

Tartalomjegyzék

1. Bevezetés.....	10
2. Az adattárház létrehozásának fontossága a felsőoktatási felvételi eljárásban	11
2.1. Központi adatbázis	12
3. Használt eszközök	12
4. Adattárház koncepciói Ralph Kimball és Bill Inmon által	13
5. A felvételi környezet	15
6. A feladatra választott megoldási módszer	17
7. A modellek/rendszer specifikációja	17
7.1. Use-Case meghatározása.....	17
7.2. Az adatok forrása	18
7.2.1. A forrás adatbázis tábláinak leírása	19
7.3. Dimenzionális modell	19
7.3.1. Dimenziók.....	19
7.3.2. Ténytábla.....	20
7.3.3. Csillagséma	20
8. Tervezés lépései.....	21
8.1. ETL folyamatok.....	22
8.1.1. Extract.....	22
8.1.2. Transform.....	24
8.1.3. Load.....	26
9. Riportkészítés.....	35
9.1. A felsőoktatásba jelentkezők területi megoszlása nemenként (Magyarország).....	36
9.2. Intézményenként és képzési területenként a jelentkezések száma.	37
9.3. Jelentkezők átlagéletkora megyénként informatika képzési területen.....	38
9.4. Összpontszámok átlagának alakulása intézményenként és képzési területenként.	39
9.5. Jelentkezők nyelvvizsgájának számossága intézményenként, megyénként, kor és nem szerint.....	40
9.6. Adott szakra (mérnökinformatikus) jelentkezők megoszlása megye és intézmény szerint.	40
9.7. Alapfokú végzettséggel rendelkező felsőoktatásba jelentkezők számossága, akik pedagógus végzettségűek, megyei bontásban.	41
9.8. Intézményenként az 1-nél több nyelvvizsgával rendelkezők, nyelvvizsga megszerzésének dátuma szerint.	41

10. A megvalósítás elemzése, továbbfejlesztési lehetőségek számbavétele	42
11. Összefoglaló	44
12. Summary.....	44
13. Irodalomjegyzék.....	45
14. Mellékletek	48
1. sz. melléklet	48
2. sz. melléklet	52
3. sz. melléklet	54
4. sz. melléklet	56
5. sz. melléklet	65
6. sz. melléklet	66

Felsőoktatási Felvételi Eljárás BI rendszerének kialakítása, adattárházzal - Higher Educational Process in Hungary with BI system and DWH

1. Bevezetés

Magyarországon évente szerveznek felsőoktatási felvételi eljárást (eljárás). Ennek során rengeteg adatot tárolnak a jelentkezőkről, a választott szakhoz tartozó intézményekről, illetve az eljárás lefolytatása során számos adat újabb táblákat, adatbázisokat is generál. Ilyet eredményez például az intézmény tanulmányi osztályának adatfeldolgozása (alkalmassági vizsga, oklevélbíráló) vagy a központi adatfeldolgozás, például az érettségi eredmények feldolgozása.

További adatok keletkeznek a vonalhúzás (ponthatár-megállapítás) következtében, melynek során a jelentkezőket, jelentkezési sorrendjük alapján (ahogy megjelölték az intézményt és annak szakjait) besorolják; akik pedig nem nyertek felvételt, azok jogorvoslattal élhetnek eredményük felülvizsgálatát kérve.

Maga az adott eljárás – egy évben három eljárás kerül lebonyolításra: általános, keresztféléves, pót – a felvételi tájékoztató szerkesztésével kezdődik, mely során az Oktatási Hivatal (OH), az Innovációs és Technológiai Minisztérium és a felsőoktatási intézmények kollégái dolgoznak együtt, a korábbi eljárások statisztikáit felhasználva, hogy a jelentkezőknek felvételi tájékoztatót készítsenek. Pusztán ez a folyamat számos adatbázist generál.

A felsőoktatási felvételi eljárás adatbázisát 1996-ban alkották meg az oktatási minisztériumi, illetve informatikus kollégák. Apróbb változtatások kivételével – melyeket maga a jogszabályi környezet változása eredményezett – mind a mai napig az akkor kialakított struktúra van használatban.

A bonyolult eljárásnak csak egy kis adathalmazán, azaz a felsőoktatási felvételi eljárás központi adatbázisán fogjuk modellezni, hogy hogyan tudunk a már meglévő adatokkal egy adattárházat megtervezni, megvalósítani, ETL (extract, transform, load) folyamatokat véghezvinni, ami végül a BI (business intelligence) vizualizációjában csúcsosodik ki. Figyelve a kiépített rendszer távlati üzemeltetésére, továbbfejlesztésére. A tervezésnél figyelembe kell venni, hogy a jelentkezők által megadandó adatok körét jogszabály írja elő, ennél sem többet, sem kevesebbet nem lehet tőlük kérni.

2. Az adattárház létrehozásának fontossága a felsőoktatási felvételi eljárásban

A felsőoktatási felvételi eljárás során, szinte a kezdetek óta igény van a nagy mennyiségű adatok gyors feldolgozására, az azokból kinyert információk helyes vizualizációjára, értelmezésére. Az adatok jellege az eljárás előrehaladtával változik. Míg a jelentkezési időszak lezárásával, február 15. az összes jelentkező e-felvi adata átöltésre kerül a központi adatbázisba, főként az érdeklő döntéshozókat, hogy mely szakot, képzési területet [1] mennyi jelentkező jelölte. Addig a ponthatár megállapításakor és utána, főként az elért eredményeik a fontosak.

A felsőoktatási felvételi eljárás adatbázisa ugyan sok éve működik a jelenlegi struktúrájában, azonban a jelen dolgozatban kipróbálandó új környezet (DB2- ről, Microsoft termékekre váltás) miatt, egy új adatbázis tervezése indokolt, ami magában foglalja a régi rendszer optimalizálását is. Az adatbázisrendszert a meglévő adatok, használt sémák egy releváns, de kisebb részének kiemelésével fogjuk megvalósítani. Ennek során törekedni kell arra, hogy az adatbázis lényegét megőrizzük, mivel az egy integrált, logikailag összetartozó információk összessége, adatok és az adatok közötti összefüggések egységes rendszere. Tehát a jogszabály által előírt [2] tárolandó adatokra épülő adatbázist, illetve az eljárások során kialakult egyéb adatbázis-szerkezetet igyekszünk megtartani.

Az eljárások során rengeteg adatkérés érkezik az informatikára a minisztérium, egyetemek és a szakma felől is. Jelenleg ezeket a kéréseket ad hoc lekérdezésekkel végzik az informatikus kollégák. Az adatokat többnyire excel formájában továbbítják felhasználásra, melyből a kollégák kimutatásokat tudnak készíteni. Ezek a lekérdezések nagyon sok esetben azonos alapokra épülnek, azonos kérdéskört vetnek fel, ami felveti annak lehetőségét, szükségességét, hogy ezek automatizálásán elgondolkozzunk.

Az is megemlítenéd, hogy az adatok a szigorúan vett központi adatbázisba több irányból érkeznek. Ilyen például az e-felvi, a Gólya (egyetemek jelentkezői adminisztrációs alkalmazása), Felsőoktatási Felvételi Tájékoztató (FFT), postán érkező dokumentumok, amik digitalizált formában kerülnek betöltésre, az Ofix az Oktatási Hivatal (OH) kollégái számára készített, központi jelentkezői adminisztrációs alkalmazás, a vonalhúzás felülete és még egyéb kisebb programok által generált adatok, melyek az eljárás működéséhez szükségesek.

A fentiek figyelembevételével a központi adatbázisban meglévő adatok kisebb halmazával egy teljes extract, transform, load (ETL) folyamat megvalósítása kézenfekvő megoldásnak tűnik, a feladat magasabb fokú automatizálására, mely egy adattárház felépítésével valósítható meg.

2.1. Központi adatbázis

A központi adatbázis pár szóban magyarázatra szorul a dolgozat ezen kezdeti pontján. Az OH –ban központi adatbázisnak nevezik a jelentkezők 4T adatait, jelentkezéseik, hozott dokumentumaiknak a gyűjteményét, az eljárás előrehaladtával a pontszámaikat (törzspont, többletpont), alkalmassági vizsgák eredményét, vonalhúzás (ponthatár megállapítás) adatait, intézmény által meghirdetett, meghirdethető (nyilvántartásba vett képzéseket).

Tehát nagy vonalakban, több felületről importálásra kerülnek az adatok ide, melyek az adott eljárás főbb, jelentkezőhöz tartozó adatait tartalmazzák.

3. Használt eszközök

A dolgozat során Microsoft termékeket fogok használni. A jelenleg használt rendszer (központi adatbázis) alapvetően az IBM DB2 adatbázisra épül, de egyes alkalmazások MySQL adatbázist használnak (felvi.hu).

A Microsoft termékekre való átállást a felvételi eljárásban növelheti az a tény, hogy az OH-ban más szakrendszerek már hosszabb ideje használják a cég termékeit, konkrétan az:

- SQL Server Integration Services-t (SSIS)
- SQL Server Analysis Services (SSAS)
- SQL Server Reporting Services (SSRS)
- SQL Server Management Studio (SSMS)
- Power BI-t.

Ezért mind a technológia, mind pedig az adatbázisfejlesztői személyi erőforrások is rendelkezésre állnak a szervezetnél.

Magas rendelkezésre állás tekintetében számos lehetőséget nyújt a Microsoft. Ilyen például a clusteres megoldás (aktív, passzív clusterek), redundáns storage biztosítása, adatbázis tükrözés. [3]

Az SSMS beépített funkciói, mint például az SQL Server Agent, melyben jobok stepekre való bontása is lehetséges, illetve ez a funkció összekapcsolható a Management Database Mail funkciójával, mely további előnyökkel járhat az optimalizált folyamat kialakításában. Továbbá a fentiek használatával, maga a folyamat, vagy annak egyes részei jól delegálhatók lennének az adatbázis fejlesztőtől a DBA (adatbázis adminisztrátor) felé, jobb munkamegosztást, csapatmunkát létrehozva ezzel.

Magát az ETL, azaz extract, transform, load folyamatokat az SQL Server Integration Services (SSIS) használatára szeretnénk felépíteni. Szoftvergyártón belüli programok kompatibilitási problémáktól jobbára mentesek, szoftverfrissítéskor kevesebb probléma

merülhet fel., illetve a két alkalmazás (az SSIS és az SSMS) összekapcsolásával belső fejlesztésű program(ok) is kiválthatók lehetnének, melyek amellet, hogy előnyösek, hiszen házon belüli termékek lévén, módosításukra nagy szabadságot kaphatnak a fejlesztők, az új igények könnyen specifikálhatók, megbeszélhetők, azonban mindez hátrányuk is, mert a kornak megfelelő támogatásuk, fejlesztésük nagy erőforrásokat vesz el munkaórában. Ezek a belső fejlesztésű alkalmazások a fejlődés gátjai is lehetnek, mert sok esetben azért használatosak eljárásról eljárásra, mert korábban is így volt.

A dolgozat során a Microsoft Power BI rendszerét fogom használni a részletes adatelemzésre, vizualizációra, mely képes vállalati üzleti intelligencia platformmal csatlakozni az adatokhoz és megjeleníteni azokat. Segítségével könnyedén együtt dolgozhatnak a szervezet döntéshozói másokkal ugyanazokon az adatokon és jelentéseken, és elemzési eredményeket oszthatnak meg egymással, így a szervezet megfelelő tagjai képesek lehetnek a gyorsan rendelkezésre álló adatokon alapuló döntéshozatalra, ami elősegítheti a stratégiai műveletek végrehajtását.[\[12\]](#)

4. Adattárház koncepciói Ralph Kimball és Bill Inmon által

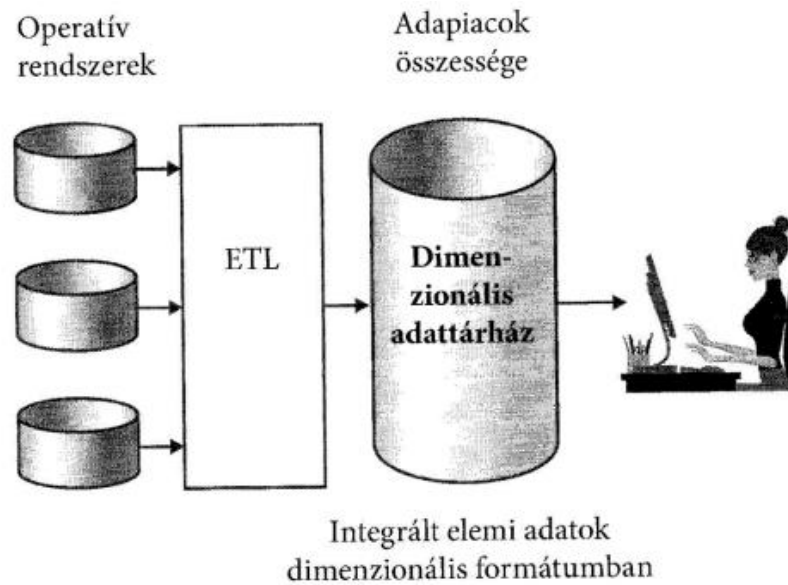
Az adattárházzal kapcsolatos munkák során nem hagyható ki a két fenti szerző munkásságának tanulmányozása.

A téma két nagy alakja Ralp Kimball és Bill Inmon, akiktől az alábbi definíciók származnak az adattárházra vonatkozóan.

Kimball *(4.1. ábra: Kimball EDW architektúrája)*:

"The conglomeration of an organization's data warehouse staging and presentation areas, where operational data is specifically structured for query and analysis performance and ease-of-use."

Tehát, az adott szervezet adatgyűjtő/szolgáltató része, ahol az adatokat újrarendszerezik riportolási, elemzési feladatokhoz. [\[4\]](#)

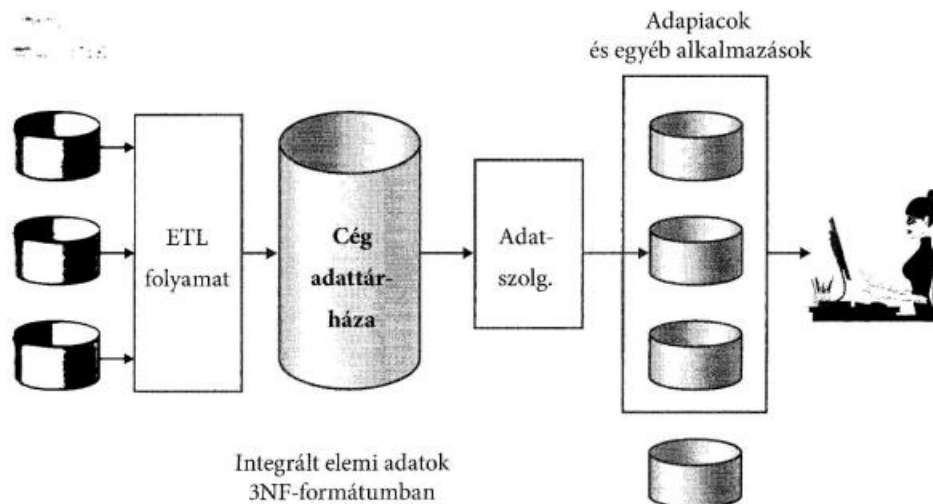


4.1. ábra: Kimball EDW architektúrája [6]

Inmon (4.2. ábra: Inmon egyszerűsített CIF architektúrája):

"A data warehouse is a subject oriented, integrated, nonvolatile, and time variant collection of data in support of management's decisions."

Tehát az adattárház itt tárgyorientált, integrált, tartós, időfüggő adatok gyűjteménye. A témaorientáltság egy fontos tényező, mivel itt először meg kell határozni a fő elemzési témákat, melyekre a vizualizációt építjük, mely üzleti értékkel bírhat. [5]



4. 2. ábra: Inmon egyszerűsített CIF architektúrája [6]

A fentiek értelmében a DW/BI rendszer(t) [4]:

- Elő kell mozdítsa az információk, adatok könnyű elérését, melyek értelmezhetőek kell legyenek az azt felhasználóknak.
- Az információknak konzisztensnek kell lenniük
- Fel kell készíteni a változások adaptálására
- Az információ/adat igény gyors kiszolgálására
- Az adatok biztonságos tárolására
- Helyes adatokra, melyekből helyes döntések születhetnek
- Felhasználók által is szívesen használhatónak kell lennie

A felvételi eljárás során, az inmon architektúrát igyekszünk megvalósítani, azért mert a témaorientáltság, az egyes adatok tartóssága ezt a módszert igényli jobban.

5. A felvételi környezet

A felsőoktatási felvételi eljárás hivatalos tájékoztatója a Felsőoktatási Felvételi Tájékoztató (FFT), ami a 423/2012. (XII. 29.) Kormány rendelet alapján, a jelentkezők számára emészthető formában taglalja a sikeres jelentkezéshez, felvételhez szükséges információkat.

Az alábbiakban erről a folyamatról írok.

Az adott évben, három eljárás zajlik. Időrendben keresztféléves, ami főként mesterképzésre jelentkezőknek, általános ezt nevezhetjük a „fő” eljárásnak, mely körülbelül 100 ezer jelentkezőt érint, pótfelvételi, mely az általános eljárás kiegészítésének tekinthető, így azon szabályok szerint történik a lebonyolítása is, de csak egy képzést jelölhetnek a jelentkezők.

Az általános eljáráshoz az FFT december végén jelenik meg a felvi.hu-n, hivatalos kiegészítése (főként intézmények által eszközölt módosítások) pedig január utolsó napján. A tájékoztató tartalmazza a jelentkezés szabályait, magyarázatokat, hasznos táblázatokat, meghirdetett képzéseket.

Az alap, osztatlan képzés, felsőoktatási szakképzés pontszámítása alapvetően a tanulmányi pontra [középiskolai osztályzatok (utolsó két tanult év végi), érettségi átlageredmény], érettségi pontokra (két érettségi tárgy százalékos eredménye), illetve többletpontokra épül. A kettő közül tehát, hogy tanulmányi pontok és többletpontok vagy érettségi pont kétszerezése és többletpontok alapján kerül kiszámításra a jelentkező pontszáma, az dönti el, hogy számára melyik eredményez több pontot (maximálisan 500 pont szerezhető). Más pontszámítás vonatkozik például a művészet- és

művészetközvetítés képzési terület Bsc tárgyaira, mivel itt alkalmassági, gyakorlati vizsga kerül megszervezésre. Alkalmassági vizsga sikertelensége esetén természetesen minden más eredményétől függetlenül a jelentkező pontszáma 0.

Mesterképzéseknél a jogszabályi előírások betartásával az intézmények határozzák meg a pontszámításukat.

Többletpontok esetében egyre nagyobb jelentőséggel bír a nyelvtudásért, illetve az emelt szintű érettségi vizsgáért adandó pontok. Erős döntéshozói szándék van arra, hogy minél több jelentkező rendelkezzen ezen eredményekkel. Ezek mellett számos egyéb többletpontra jogosító dokumentum is van.

A megszerezhető pontok tekintetében, amint fentebb szóba került, maximálisan 500, minimum 280 pont szerezhető Bsc képzések esetében, mesterképzéseknél minimum 50 maximum 100.

A folyamatos digitalizációnak köszönhetően, mára kevés dokumentum jön be postai úton, komoly igyekezet van, hogy digitálisan lehessen minél több folyamatot elvégezni. Ennek következtében a legnagyobb tömeget érintő adathalmaz az érettségi eredmények is központilag kerülnek áttöltésre a központi érettségi nyilvántartásból (KÉNY), de a nyelvvizsga dokumentumok mind nagyobb száma is központilag kezelt, a nyelvvizsga akkreditációs központ (NYAK) által. [7]

Összegezve a fentieket, felvételt nyerhet az a jelentkező, aki:

- érvényes jelentkezést nyújtott be
- megfelelő végzettséggel rendelkezik
- teljesíti a felvételi követelményeket (pl.: emelt szintű érettségi)
- rendelkezik az adott tárgyon szükséges érettségi tárgyakkal
- eléri a minimum pontot
- eléri a ponthatár megállapításakor megszülető ponthatárt az adott szakra
- egyéb feltételeknek megfelel (pl.: alkalmassági vizsga)

A jelen dolgozatban egy általános eljárás valamennyi Bsc képzésén törekszünk modellezni a fenti komplex folyamat figyelembevételével az eljárás végső állapotát, tehát a vonalhúzást követő állapotot, amikor már minden adat rendelkezésre áll és a felvételi folyamat – a jogorvoslat kivételével – lezárult.

6. A feladatra választott megoldási módszer

Egy adott általános felvételi eljárást választva, az ott elérhető adattömeg optimalizációjára és az ETL folyamatok, illetve vizualizáció megvalósítására, a Microsoft termékek használatára esett a választásom. [13] Emellett főként az szól, hogy a Hivatalban már régebb óta használják az általam is használni kívánt eszközöket más területeken, így mind a technológiai, mind a szakemberi gárda rendelkezésre áll a dolgozatban megalkotni kívánt béta verzió későbbi továbbfejlesztésére. További előnyök természetesen a szinte végtelen lehetőségek tárháza, hiszen, ami esetleg nincs lefejlesztve például az SSIS – ben, az C# -ban lefejleszthető a későbbiekben és azt a .dll-t fel lehet használni az integration services-ben, ha az „üzleti” folyamatok megkívánják (a szakemberi gárda erre is adott a Hivatalban). További előny a rendszeres, validált frissítések, Microsoft általi fejlesztések. A Hivatalban belső fejlesztésű programok valósítják meg jelenleg ezt a feladatot, csupán adatmigráció tekintetében a több évtizede megírt program, egy dobozos termékre cserélésének ötlete, már évek óta, komolyabban mozgatja a döntéshozókat.

Összegezve tehát, Microsoft termékek használatával igyekszem modellezni a problémát és arra megoldást találni, amely lehetőséget ad az adatbázisok optimalizált [9],[14] létrehozására, ETL folyamatokra, vizualizációra és a későbbi továbbfejlesztés által rengeteg egyébre is.

Munkám során generált, mockolt adatokkal fogok dolgozni. Az adatok excel és .csv file-okból kerülnek betöltésre.

7. A modellek/rendszer specifikációja

7.1. Use-Case meghatározása

A felvételi eljárások folyamata során, a felvételi tájékoztató szerkesztésétől a jogorvoslat lezárultáig, számos adatkérés érkezik a szakma, minisztériumi, intézményi kollégák, külső cégek felől, mely során a kapott adatokat (jellemzően xlsx) igyekeznek értelmezni, vizualizálni.

Az adatok elemzési irányvonalai többfélék; egyfelől a már lezárult eljárások adatainak értelmezésével törekszenek például a jövő felvételi eljárásait, az ország igényeinek megfelelően pozitív irányba alakítani, befolyásolni kimutatások segítségével is, másfelől az éppen zajló eljárást zökkenőmentesen végigvinni a korábbi eljárások eredményei tükrében, akár befolyásolni.

Mindezek miatt felmerült az igény egy már meglévő, de elavult adatbázisrendszer módosítására, kiépítésére, illetve egy erre épülő business intelligence (BI) rendszer létrehozására riportokhoz, melyeket a fentiek átgondolásával az alábbi fiktív munka során mutatunk be.

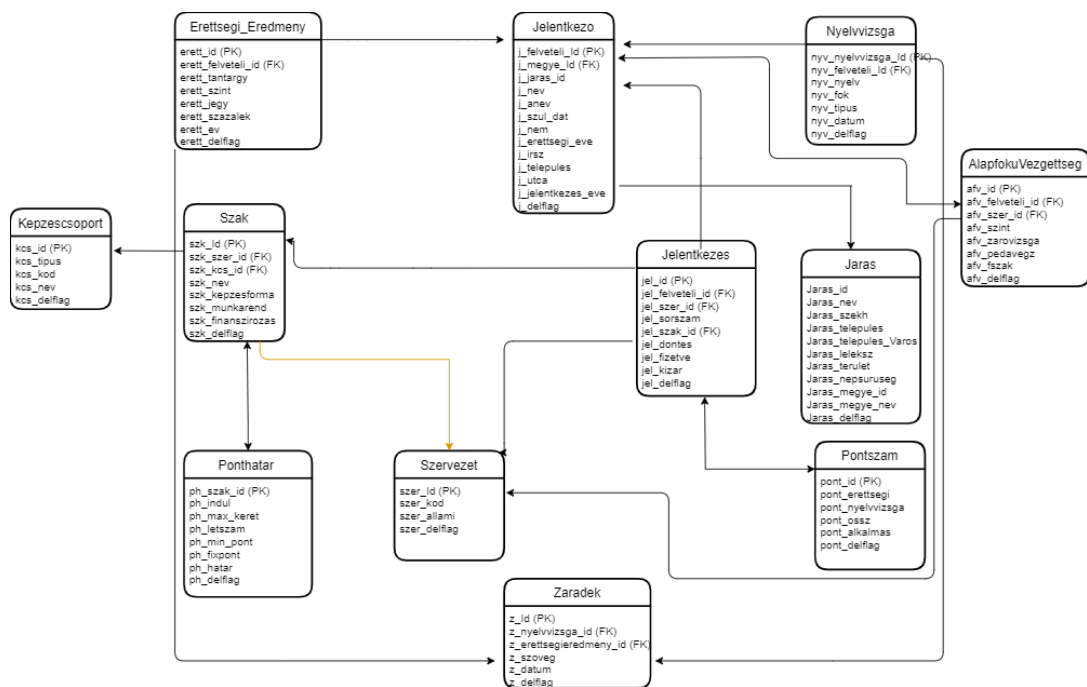
Minthogy a főbb elemzési kérdések, az alapképzésre jelentkezők területi megoszlására (megye), érettségi adataira-, eredményeire, megjelölt szakjaira, előzetes többletpontra jogosító eredményeire (pl.: nyelvvizsga) vonatkoznak, ezért ebben a munkában is ezekre a részekre helyezzük a hangsúlyt.

A megrendelő a következőket szeretné tudni elemezni [11]:

1. A felsőoktatásba jelentkezők területi megoszlása nemenként (Magyarország).
2. Intézményenként és képzési területenként a jelentkezések száma.
3. Jelentkezők átlagéletkora megyénként informatika képzési területen.
4. Összpontszámok alakulása intézményenként és képzési területenként.
5. Jelentkezők nyelvvizsgájának számossága intézményenként, megyénként, kor és nem szerint
6. Adott szakra (mérnök-informatikus) jelentkezők megoszlása megye és intézmény szerint.
7. Alapfokú végzettséggel rendelkező felsőoktatásba jelentkezők számossága, akik pedagógus végzettségűek, megyei bontásban.
8. Intézményenként az 1-nél több nyelvvizsgával rendelkezők, nyelvvizsga megszerzésének dátuma szerint.

7.2. Az adatok forrása

A jelentkezők főbb adatait; jelentkezéseiket, dokumentumaik eredményét, lakóhelyüket az alábbi forrásadatokat tartalmazó kép mutatja (7. 2.1. ábra: Forrásadatok modellje).



7. 2.1. ábra: Forrásadatok modellje

7.2.1. A forrás adatbázis tábláinak leírása

A forrástáblák a központi eljárásnak azon entitásai, melyek a jelentkező személyes adatain túl, a hozott dokumentumait, jelentkezéseit (az ezekhez tartozó adatokat), illetve a vonalhúzás eredményeit tartalmazzák.

Megjegyzendő még, hogy a forrástáblák mindegyikébe teszünk időkövetést is a változások követése miatt, hogy a későbbi továbbfejlesztési lehetőségek közül a delta áttöltést is meg lehessen valósítani, a dátum figyelésével:

```
alter table [dbo].[ZARADEK]
    add sys_startdate datetime2 generated always as row start not null
DEFAULT GETUTCDATE()
    sys_enddate datetime2 generated always as row end
hidden not null DEFAULT CONVERT(DATETIME2, '9999-12-31
23:59:59.9999999')
    period for system_time (sys_startdate, sys_enddate) ;
```

7.3. Dimenzionális modell

A forrásadatok és a use-case alapján a továbbiakban ebben a fejezetben látható dimenziókat és tényeket különböztetjük meg.

7.3.1. Dimenziók

Date: A dátumokat tartalmazza napi bontásban, több évre előre.

Jelentkezo: Egy jelentkező 4T adataiból tartalmazza a szükséges adatokat.
Forrásul szolgálnak a Nyelvvizsga, Jelentkezo, AlapfokuVegzettseg táblák.

Jaras: A lokalizációhoz szükséges megyéket, járásokat tartalmazza; a jelentkező lakhely szerinti megyéje. Forrása a Jaras.

Szak: A jelentkező a felsőoktatási intézmény által meghirdetett szakra jelentkezését, illetve a szak képzéscsoport szerinti besorolását tartalmazza.
Forrás táblái a Szak, Ponthatar, Kepzescsoport.

Szervezet: A felsőoktatási intézményeket tartalmazó entitás.

7.3.2. Ténytábla

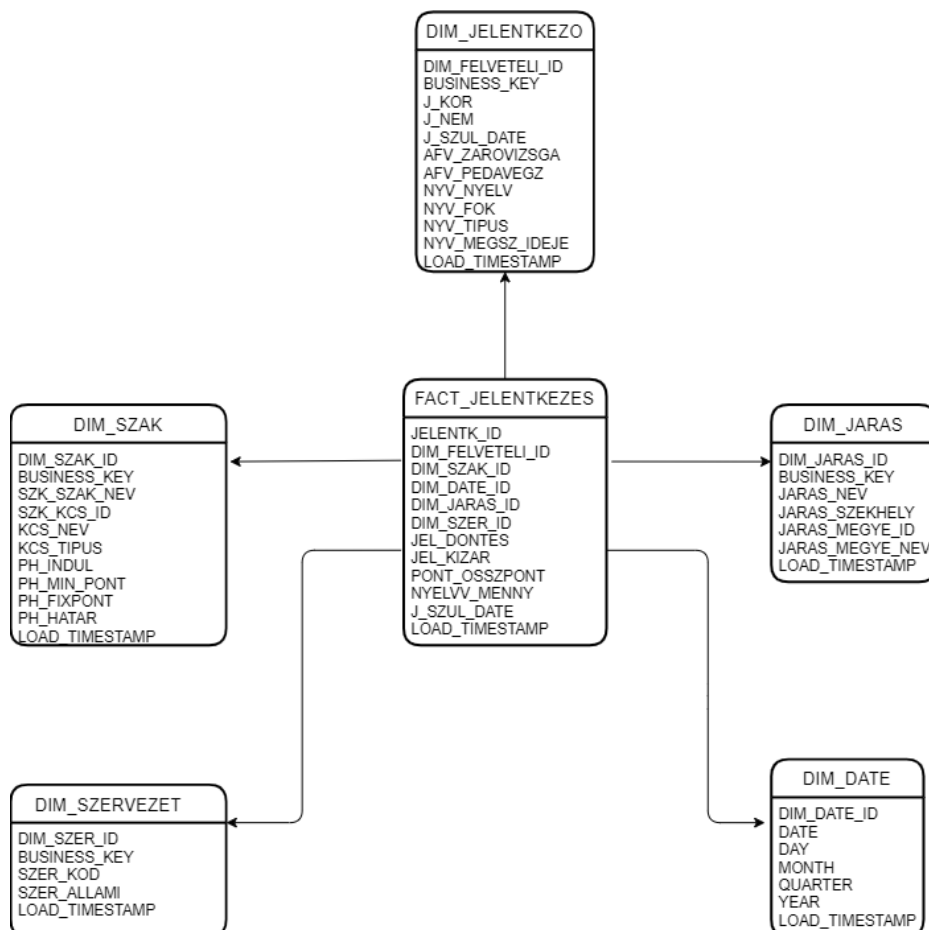
Jelentkezes: Egy jelentkező jelentkezéseit tartalmazza, szakra, jelentkezés szintű granularitással. Forrása a Jelentkezes, Pontszám, Nyelvvizsga táblák.

7.3.3. Csillagséma

Egy denormalizált séma, célja a dimenzionális szemlélet fizikai vizualizációja. A dimenziótáblák a szűrést és a csoportosítást, míg a ténytáblák az összegzést támogatják.

[8] Csillagsémát választottunk a feladat megoldásához, mert a use case-ek megvalósításához ez az egyszerű adatmodell kevés join művelettel a legmegfelelőbb, illetve a szakmai oldalon a vezetőknek, munkatársaknak ezen demo verziójú adattárház bemutatása így tud a leggyorsabban megtörténni.

A fent felsorolt dimenziókból és ténytáblából a következő csillagsémát hoztuk létre (7.3.3. ábra: Csillagséma modellje). A séma közepén látható a fact tábla, amihez idegen kulcsokkal kapcsolódnak a dimenziótáblák.



7.3.3. ábra: Csillagséma modellje

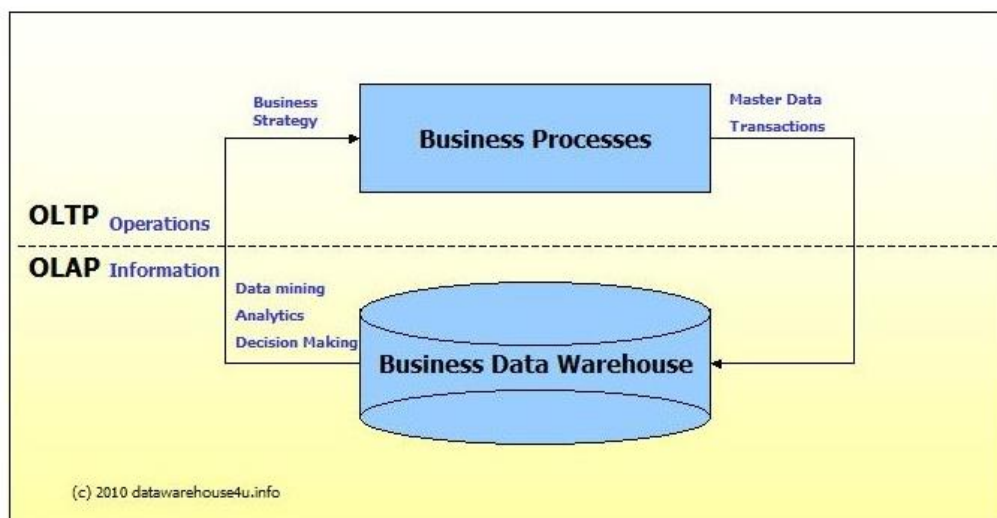
8. Tervezés lépései

OLTP és az OLAP rendszerek rövid áttekintése

Különböző adatbázisok különböző használati eseteket feltételeznek. Amikor a rendelkezésre álló adathalmazmal nap, mint nap tranzakciókat, tehát CRUD műveleteket szeretnénk végezni, menedzselni a folyamatban lévő üzleti folyamatokat, akkor az online transaction processing (OLTP) egy jó megoldás erre. Abban az esetben, amikor olyan múltbéli eseményekre keressük a választ, hogy mennyi pénzt költöttek el lokálisan a vásárlók egy adott időpontban/intervallumban egy bevásárlóközpontban, melyik volt ekkor a legkelendőbb árucikk, tehát analizáljuk a historikus adatokat, azokon aggregációkat végzünk, akkor jó megoldás az online analytical processing (OLAP).

A kettő közötti különbség is a fentiekből ered. Amíg az OLTP rendszerek felépítése jellemzően a 3NF-et követik (normalizált) és sémákba szervezve tárolják a jelen adatait tranzakcionális adatbázisba, addig OLAP esetében data mining megoldásokhoz jól használható multi dimenzionális szerkezetbe (pl.: csillagséma) szervezve tárolják az adatokat (de-normalizált).

Jellemzően feltételezhető, hogy egy OLTP rendszer biztosítja az adatokat az adattárházhoz, míg az OLAP segítségünkre lehet annak *elemzésében* (8.1. ábra OLTP vs OLAP). [15]



8.1. ábra OLTP vs OLAP [15]

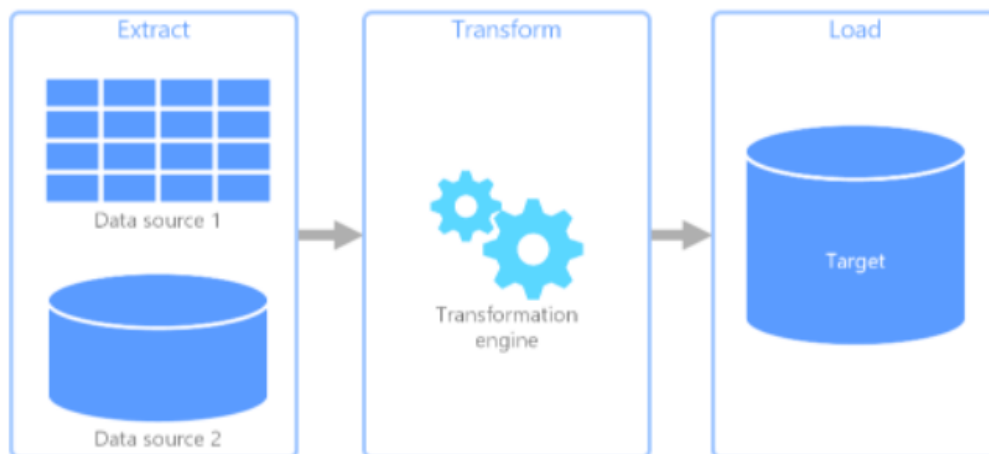
A forrásadatok a korábban említetteknek megfelelően egy Microsoft SQL Server adatbázisban vannak tárolva.

A következőkben a dimenzionális modell tábláiba való adatbetöltéshez szükséges ETL folyamatokat, valamint az egyes rétegek adatbázisait ismertetem.

8.1. ETL folyamatok

Az ETL, tehát extract, transform, load folyamatok (8.1.1. ábra: ETL folyamatábrája), egy olyan probléma megoldására születtek a relációs adatbáziskezelők esetében (struktúrált adatok), hogy mivel az adatok több forrásból származnak, emiatt több formátumban is történik a tárolásuk (pl.: dátum), ezért a forrásból a célba áttöltés problematikus. Szükséges emiatt az adatok transzformálása mielőtt áttöltésre kerülnek a végső célhelyre. Továbbá az sem mindegy, hogy mennyi idő alatt lehet az adattárházba áttölteni a forrásból az adatokat. Minthogy a gyakorlatban a forrásadatbázison számtalan adatmanipuláció zajlik, szükséges egy olyan időpontot találni az áttöltésre, amikor egyéb folyamatokat/usereket ez nem zavar, illetve még gyors is legyen, amennyire ez lehetséges.

A transzformáció számos dolgot jelent, mint például szűrés, rendezés, adattisztítás, érvényesítés, adattípusok közötti konverzió stb.



8.1.1. ábra: ETL folyamatábrája [16]

Pár szóban említést igényel az ELT folyamat, mely sorrendjében tér el az általunk használttól. Ebben az esetben, az adatok átalakítása akkor történik meg, amikor az ténylegesen elemzésre is kerül, szükség van az adatra. Ez a megközelítés jellemzően nem – vagy félig struktúrált adatokra használatos. Tipikus használati esetei a big data területe.

Az ETL, mint struktúrált és az ELT mint nem – vagy félig struktúrált adatok a DataLake-ben szervezhetők, gyúrhatók a BI rétegnek felhasználható formába. [16]

8.1.1. Extract

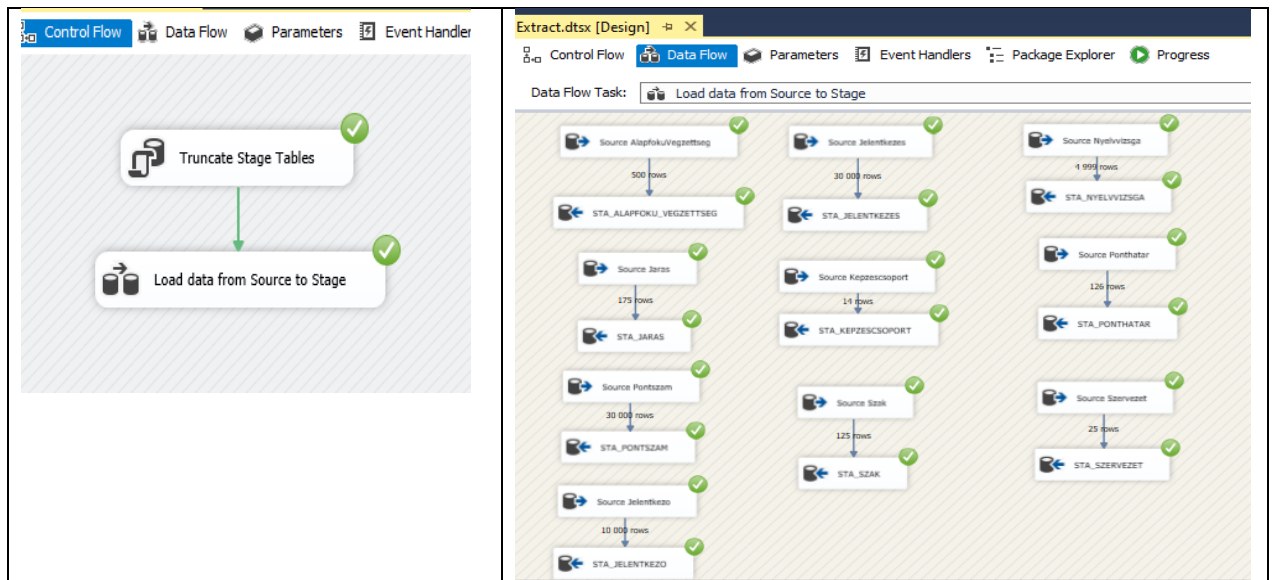
Ezen réteg szolgál a későbbiekben a Stage adatbázis létrehozására, melynek során az összes adat változatlanul kerül átemelésre a forrásból, figyelve az adatok minél

gyorsabban történő mozgatására, hogy ne terheljük a rendszert. Emiatt nvarchar-ban kerülnek ebbe a rétegbe a táblák oszlopai. [16]

Azonban szükséges megjegyezni, hogy éles rendszerénél érdemes csak azokat az adatokat áthozni a forrástáblákból, melyekre ténylegesen szükségünk van. [4] Ezt, illetve csak a változások követését (delta áttöltés) fogja majd elősegíteni a forrástáblákban elhelyezett időkövetés.

Ebben a rétegben a forrásadatokat típuskonverzió nélkül, szöveges formában töltjük be a STAGE nevű adatbázis tábláiba annak érdekében, hogy a forrásrendszer a lehető legkevesebb ideig legyen az áttöltéssel terhelve, minél kevésbé akadályozza a szolgáltatások hatékony működését.

A forrásadatokat adatbázisból érem el, OLE DB Source elem segítségével, a betöltést megelőzően pedig kiürítem a STAGE adatbázis tábláit az EXEC sp_MSForEachTable 'TRUNCATE TABLE ?' segítségével (8.1.1.ábra: Stage táblák kiürítése és feltöltése a forrásból).



8.1.1.ábra: Stage táblák kiürítése és feltöltése a forrásból

Change Data Capture, Change Tracking

Egy másik megoldás lett volna a változásokövetésre például a Change Data Capture (CDC), amely egy SQL server mechanizmus, ami képes követni a loginformációk alapján (tranzakciós napló), hogy milyen változások voltak az adatbázisban, legyen az akár nettó adatváltozás. Használatát tekintve, először engedélyezni kell az adatbázisban, majd egy táblára is, beépített tárolt eljárás segítségével.

Ezt követi a log sequence numbers meghatározása beépített függvénnyel, tehát a két szám közti változások, amikre kíváncsiak vagyunk. Természetesen lehet, hogy nem

történt az intervallumban változás, az így visszajövő null értékeket például egy sima 'if' – el lekezeljük.

Utolsó lépésként egy beépített függvénnyel kivesszük a nettó változásokat a meghatározott intervallumból, adott táblából. [17], [18] A CDC magát a törölt/frissített rekordot is mutatja.

A Change Tracking (CT) szintén követi az adatbázis oszlopokban történt változásokat. Egy verziószámot rendel az adathoz. Hasonló a fentihez, de ez csupán a változást tudja jelezni, hogy történt-e, viszont magukat a változásokat nem látjuk, nem menti el az adatokat nekünk. Használata a CDC-vel hasonlatos. Mikor Change Tracking-et használunk, érdemes snapshot izolációt készíteni az adat konzisztencia megőrzésre, ugyanis előfordulhat, hogy miközben kinyerjük az adatokat, updatelik azokat, ennek következménye pedig, hogy a verziózás miatt – amit végez a CT – duplikált adatbetöltésünk lesz a végén. Tehát snapshottal rögzítjük, hogy egy konkrét tranzakcióból vegyük ki az adatokat, más ne tudjon 'beleszemetelni' ebbe. [19]

8.1.2. Transform

Ebben a rétegben a STAGE adattábláinak tartalmát töltjük át az adattárházba. Ebben a rétegben már megjelennek a többféle adattípusok (konverziók), transzformációk és az adatok historizálása is. A típusok a forrásadatbázisnak megfelelően, de optimalizálva lesznek megválasztva, tehát próbáljuk például az nvarchar mezőket is szűkíteni, ahol ez lehetséges.

Mivel több tábla adathalmaz meghaladja a 10 MB-ot (~10 000 sor), emiatt az 'AutoAdjustBufferSize' méretét az összes containerben 'True'-ra állítottuk (8.1.2.2. ábra: A módosított adat DW-be áttöltéskor megjelenik).

A nyomon követés alapvető fontosságú, hiszen a jelentkezők dokumentumokat töltenek fel az eljárás folyamán, a szakmai, intézményi kollégák dokumentumokat bírálnak el, tehát az eljárás előrehaladtával a forrásadatbázis dinamikusan változik, ezért Slowly Changing Dimension (SCD) Type 1-es 2-t használunk.

Type 1-et ott használunk, ahol nem érdekel minket a korábbi állapota az adatnak, simán felülírjuk, például a minimum pontja a szaknak.

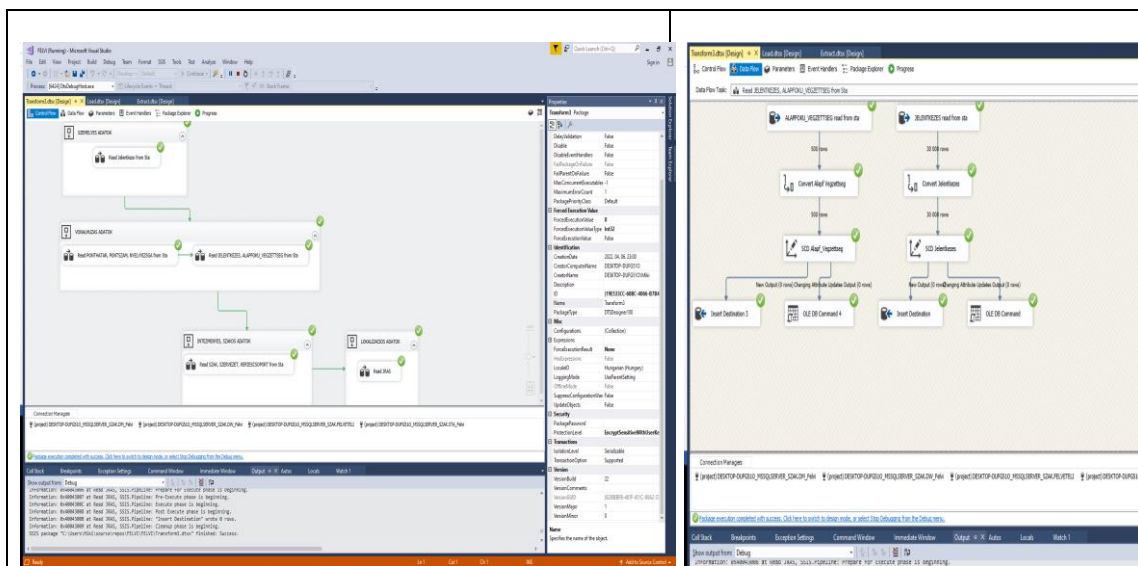
Type 2-t használunk például a szervezet esetében, hogy ezt a változást a ValidFrom-To segítségével nyomon tudjuk követni.

SCD-t azért szükséges minden táblán használni itt, mert azon felül, hogy a type 1 és 2-vel a változásokat szeretnénk megfelelően nyomon követni, az adatok rátöltését is szeretnénk elkerülni.

Összegezve tehát, az SCD a következőket végzi el nekünk:

- Összeparásítja a bejövő sorokat a lookup táblában (cél) lévővel, hogy meghatározza az új és a létező sorokat.
- Meghatározza a bejövő sorokat, amelyek olyan változásokat tartalmaznak, melyek nem engedélyezettek (fix értékek).
- Meghatározza az esetlegesen előforduló ‘inferred member’ értékeket, amiket frissíteni kell. Olyan korábban betöltött adat a ténytáblába, ami még nem létezik a dimenzióban (ezt a töltési lehetőséget mi nem használjuk).
- Meghatározza azon sorokat, melyek historikus változásokat tartalmaznak, emiatt új rekord insertálása, illetve a létezők/lejártak frissítése szükséges.
- Meghatározza az olyan bejövő sorokat, melyek változásokat hoznak, ami miatt a létező rekordokat frissíteni kell, beleértve a lejártakat is. [20]

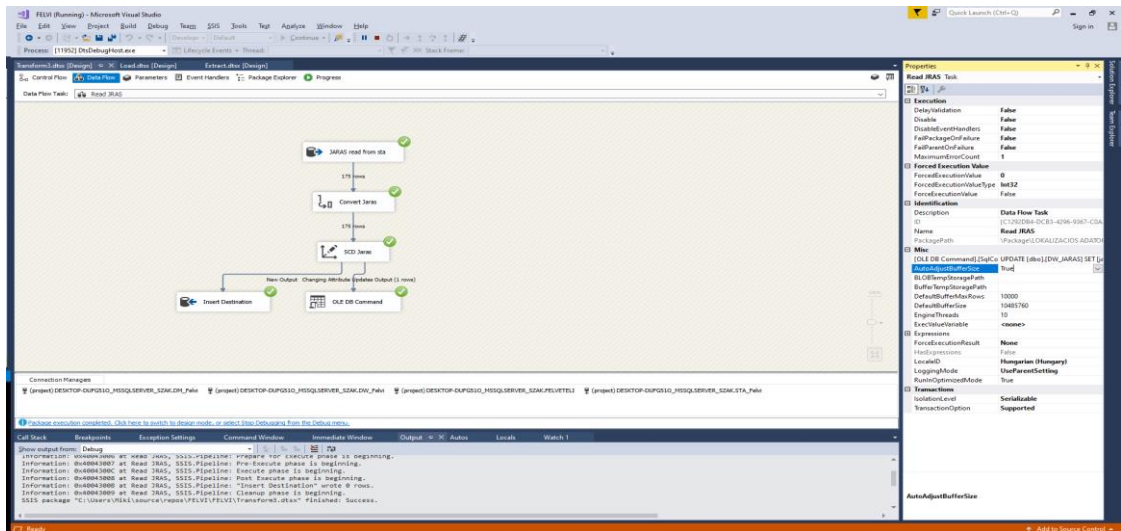
Alább az SCD használatával a DW rétegbe betöltött adatok láthatók (8.1.2.1. ábra: SCD használatával DW-be töltött adatok).



8.1.2.1. ábra: SCD használatával DW-be töltött adatok

Az áttöltés után ismételt töltésnél SCD-t használva, 1 érintett sort töltött a DW céltáblába, amit a Stage-ben korábban módosítottunk (8.1.2.2. ábra: A módosított adat DW-be áttöltéskor megjelenik):

```
update dbo.STA_JARAS set jaras_delflag='jira_456' where jaras_id=1;
```

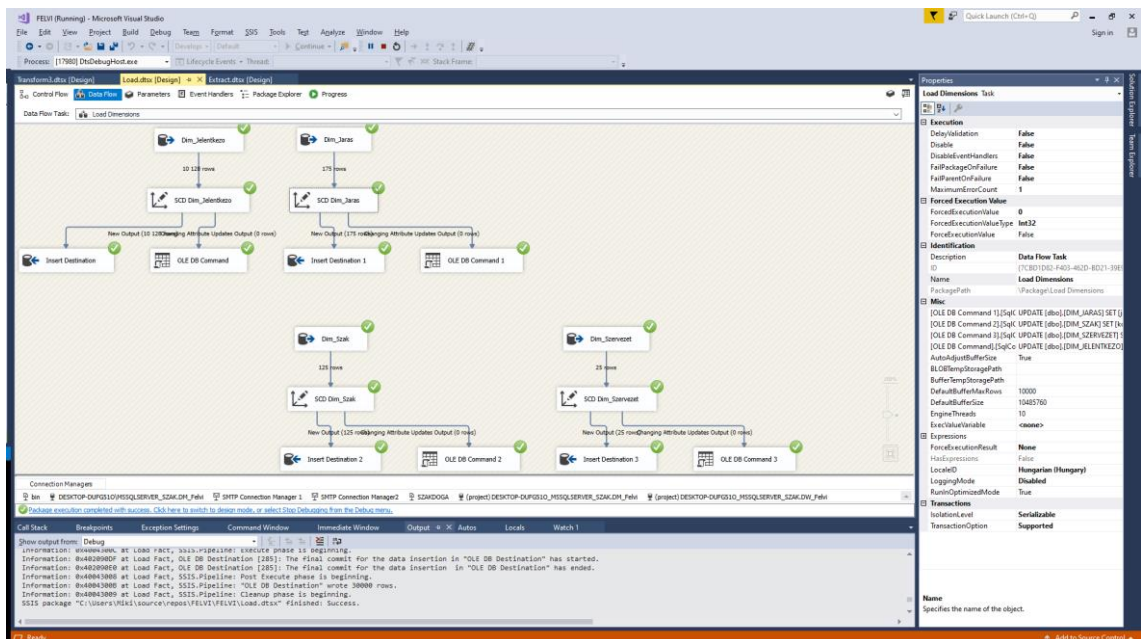


8.1.2.2. ábra: A módosított adat DW-be átöltéskor megjelenik

8.1.3. Load

Az ETL folyamat Load rétegében történik meg a csillagséma feltöltése az adattárház alapján (8.1.3.d. ábra: Ideiglenes tényítábla és valós feltöltése lookup-olással). Ennek során a dátum dimenzió – mivel a többi dimenziótól eltérően ez egy adott dátum intervallumban generált tábla – feltöltését tárolt eljárással valósítjuk meg, míg a többi dimenziót, a csillagsémában meghatározott tábla definíciók alapján szükséges értékekkel töltjük fel a DW rétegből.

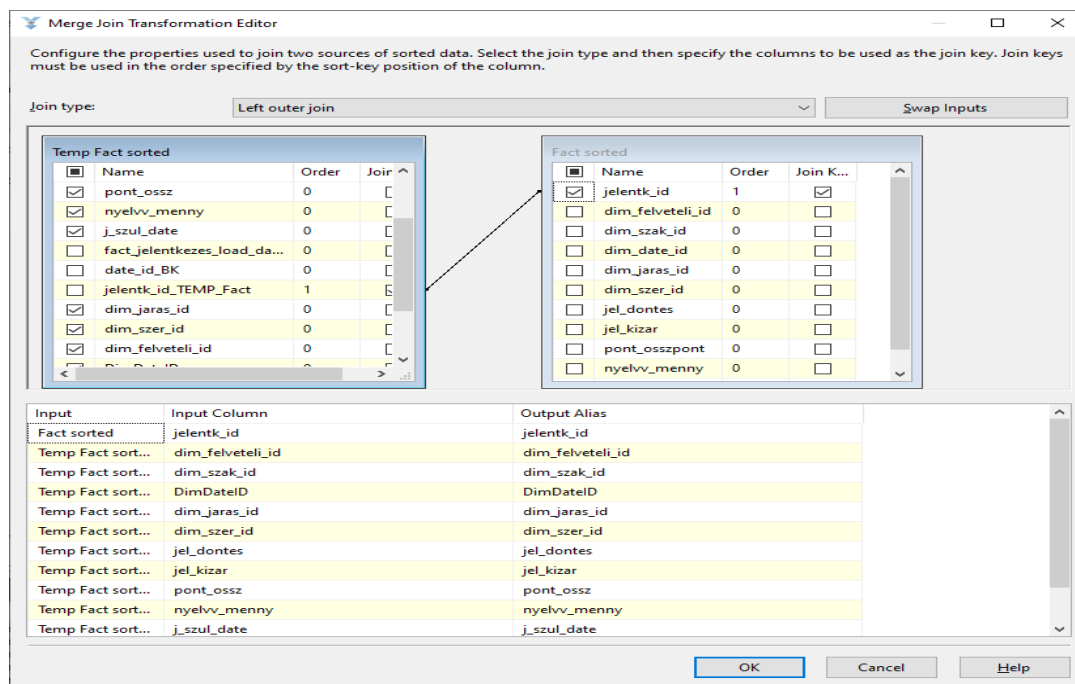
Tehát a dátum dimenzió feltöltése után SQL kód segítségével először feltöltjük a dimenziókat adatokkal (8.1.3.a. ábra: Dimenziók betöltése):



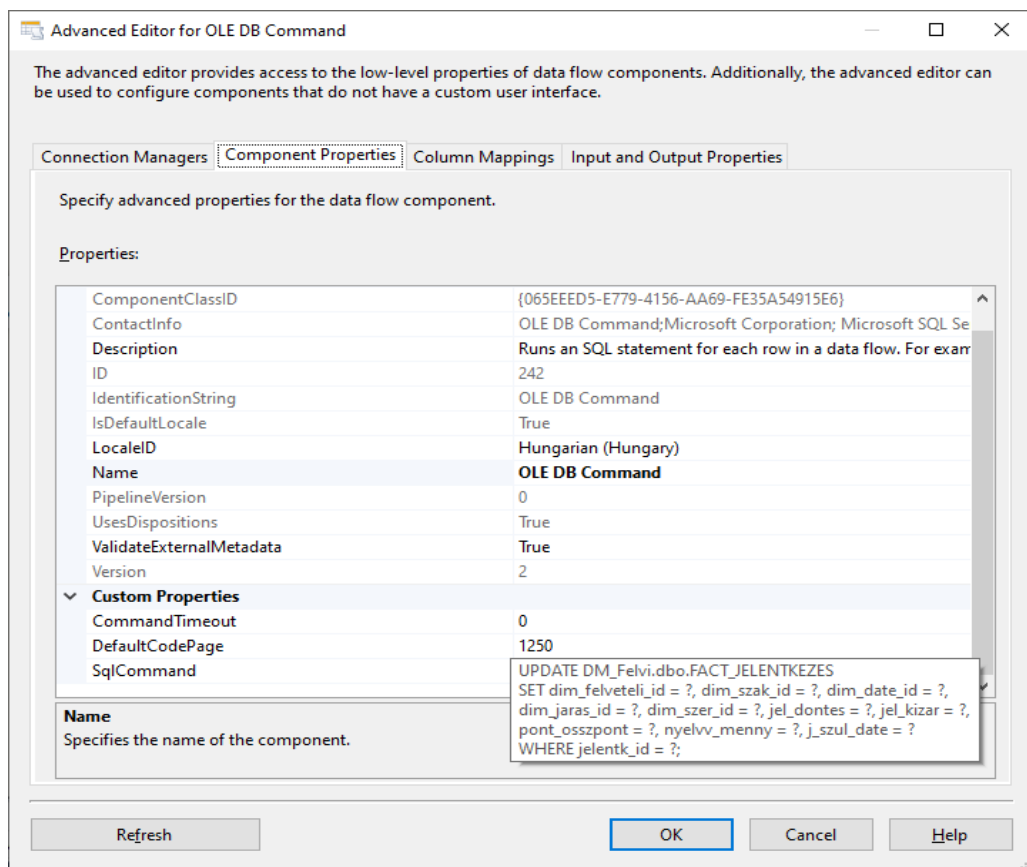
8.1.3.a. ábra: Dimenziók betöltése

A dimenzió táblák feltöltését követően, a ténytáblát is feltöltjük adatokkal. Ehhez először beolvassuk az ideiglenes ténytáblába SQL kóddal az adatokat, majd Lookup elemek segítségével minden tény rekordhoz megkeressük a hozzá tartozó dimenziók elsődleges kulcsát, a business key-k segítségével.

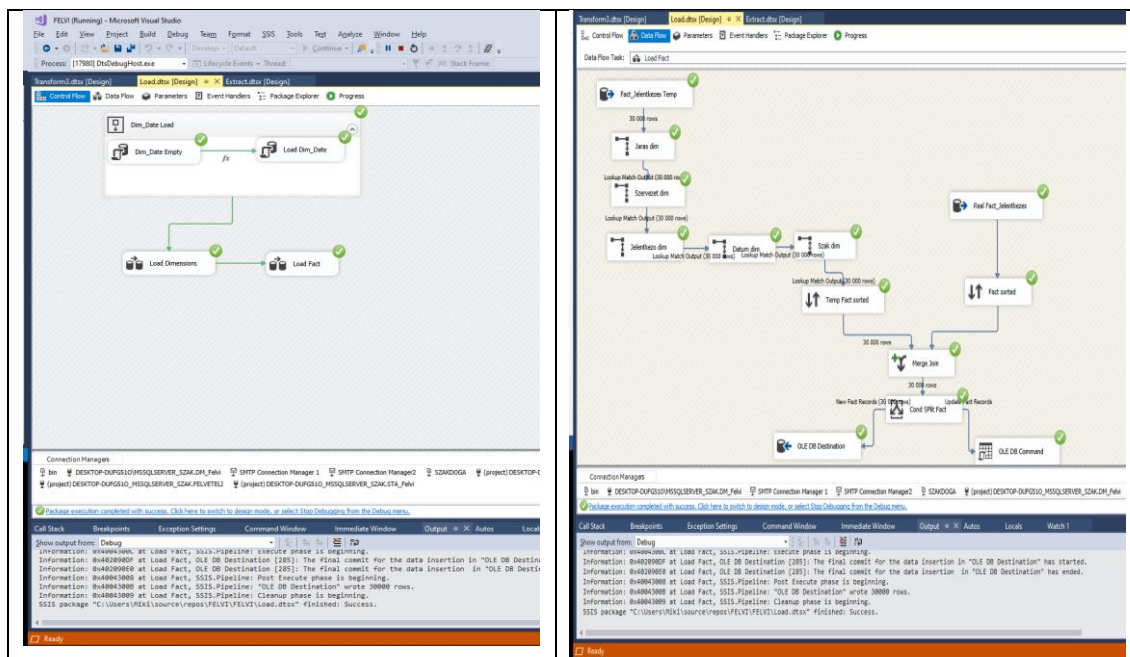
Szükséges a továbbiakban rendeznünk egy sort az értékeket, hogy aztán használhassuk a merge join, mely csak rendezett adatokon végezhető el (8.1.3.a. ábra: Sort a merge join használatához), illetve paraméterezzük is (8.1.3.b. ábra: Paraméterek átadása).



8.1.3.a. ábra: Sort a merge join használatához



8.1.3.b. ábra: Paraméterek átadása



8.1.3.d. ábra: Ideiglenes táblázat és valódi feltöltése lookup-olással

SSIS logging

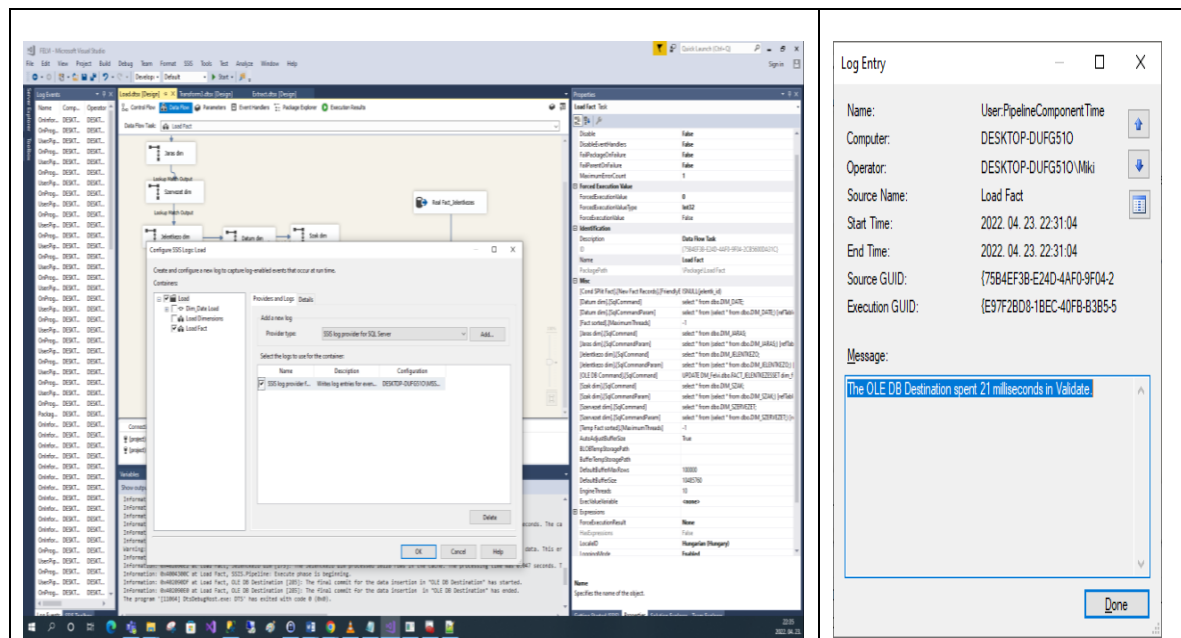
A Microsoft az SSIS package-ek futtatása során lehetőséget kínál a csomag futása során képződő különböző információk logolására. Ennek segítségével, könnyedén nyomon tudjuk követni az általunk fontosnak ítélt eseményeket, amire logolást beállítva, azok rögzülnek. Mi alább, a rendelkezésre álló opciók közül, most az 'OnInformation', 'OnProgress', illetve a 'OnPipelineComponentTime' (task futási idő) eseményekre tettünk logolást, amikről bővebb információkat szeretnénk látni (8.1.3.2. ábra: Logging beállítása a Load folyamatra) és a (8.1.3.1. ábra: LogEvent type beállításai). Ha az 'OnInformation'–re is teszünk logolást, akkor viszonylag sok logbejegyzésre kell számítsunk! Ez a jelen demo esetben nem jelent problémát. A logolt mezőneveket alapértelmezetten hagytuk, tehát az összes lehetséges értéket logoljuk.

A logbejegyzések alapértelmezettként, ahogy itt is látható az aktuális adatbázis 'System Table'–be fognak kerülni, egy táblába. Ezt 'ssislog' néven találjuk meg. A mi esetünkben (8.1.3.3. ábra: LogeventType-ok értékeinek sysssislog-ba írása):

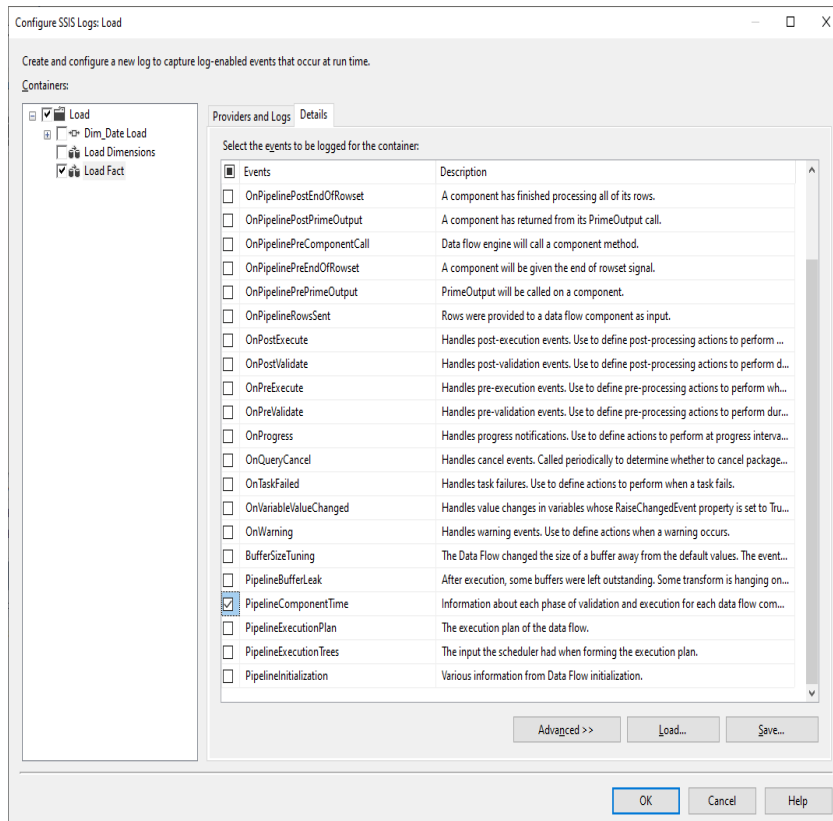
[DM_Felvi].[dbo].[sysssislog]

Ez azért jó, hogy ide kerül, mert éles rendszerre történő megvalósítás esetén a fejlesztő részletes információkat beállítva a logolásra, a log nézegetést le tudja delegálni az üzemeltető kollégák felé, ezáltal ő mentesül a nem feltétlenül fejlesztői tevékenység alól.

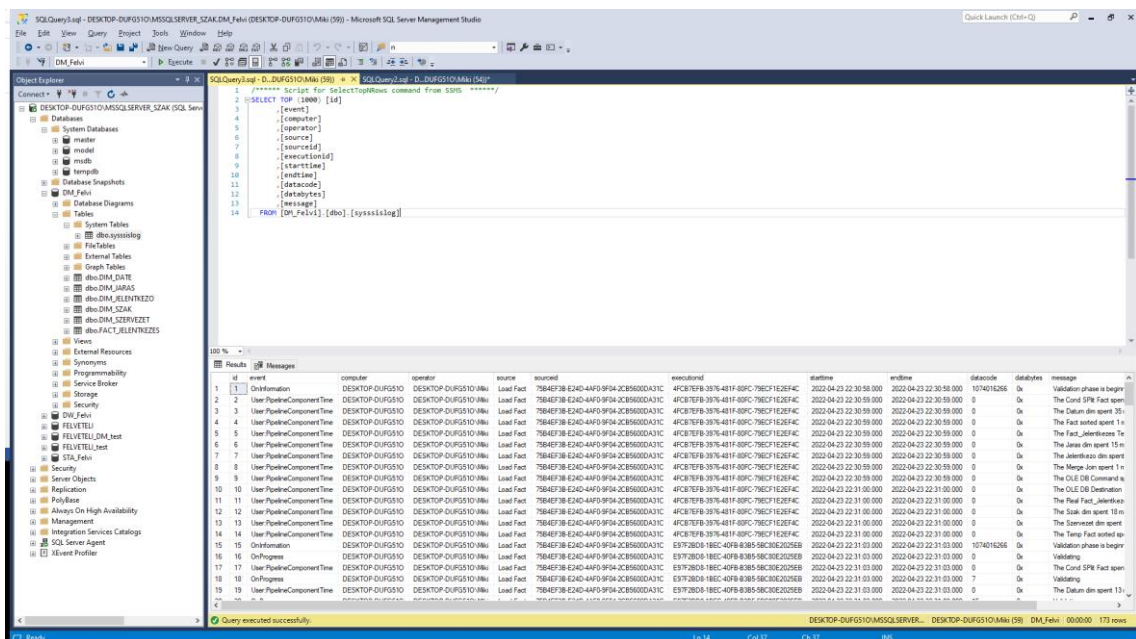
[21]



8.1.3.1. ábra: LogEvent type beállításai



8.1.3.2. ábra: Logging beállítása a Load folyamatra



8.1.3.3. ábra: LogeventType-ok értékeinek sysssislog-ba írása

Project Deployment Model és Package Deployment Model

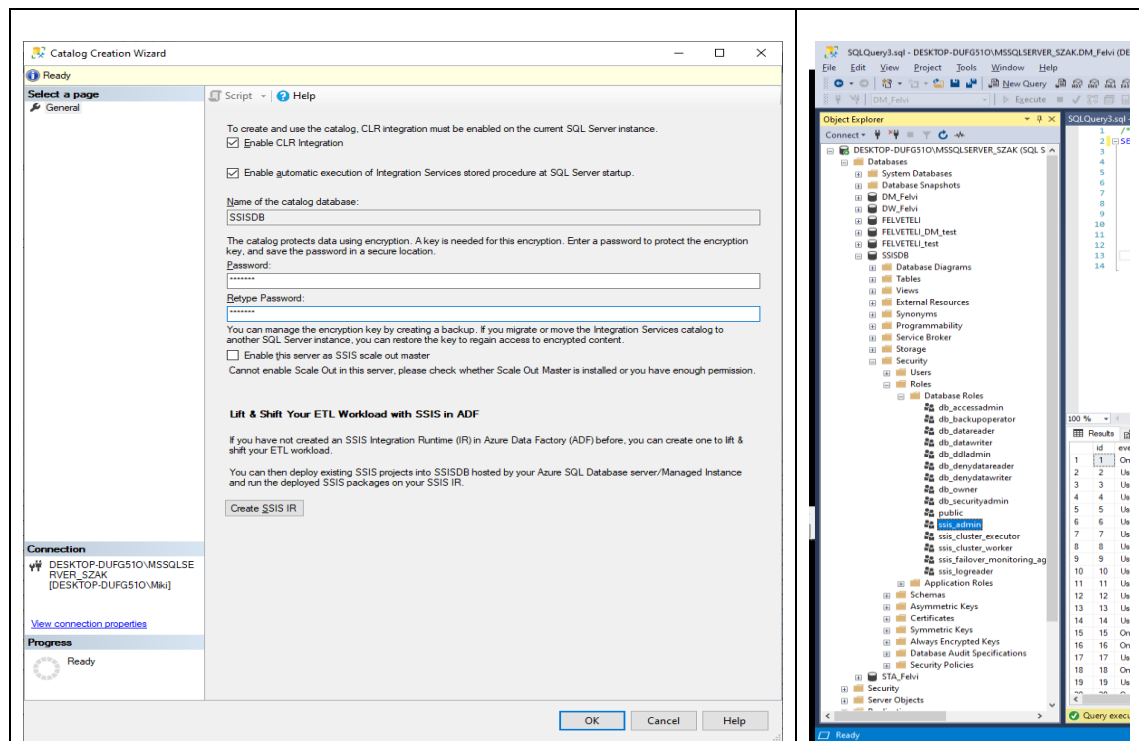
Az SQL Server 2012-től érhető el a Project Deployment Model, ami lehetővé teszi számunkra, hogy az egész solution-t tudjuk deploy-olni, amit esetünkben használunk, hogy kikerüljön az Integration Services Catalogs-ba (ISC) az összes csomagunk. Amennyiben az egész projektet szeretnénk kitelepíteni – mint most – ez a megoldás nagyban megkönnyíti az életünket (8.1.3.4. ábra: Project Deployment Model telepítése).

A Package Deployment Model esetében lehetőségünk van külön .dtsx csomagonként elvégezni a telepítést, a mi esetünkben ez többletmunka lenne.

Ahhoz, hogy az imént említetteket megtehessük, létre kell hozni egy SSISDB-t az ISC-ben SSMS-t használva, amennyiben még nem létezik.

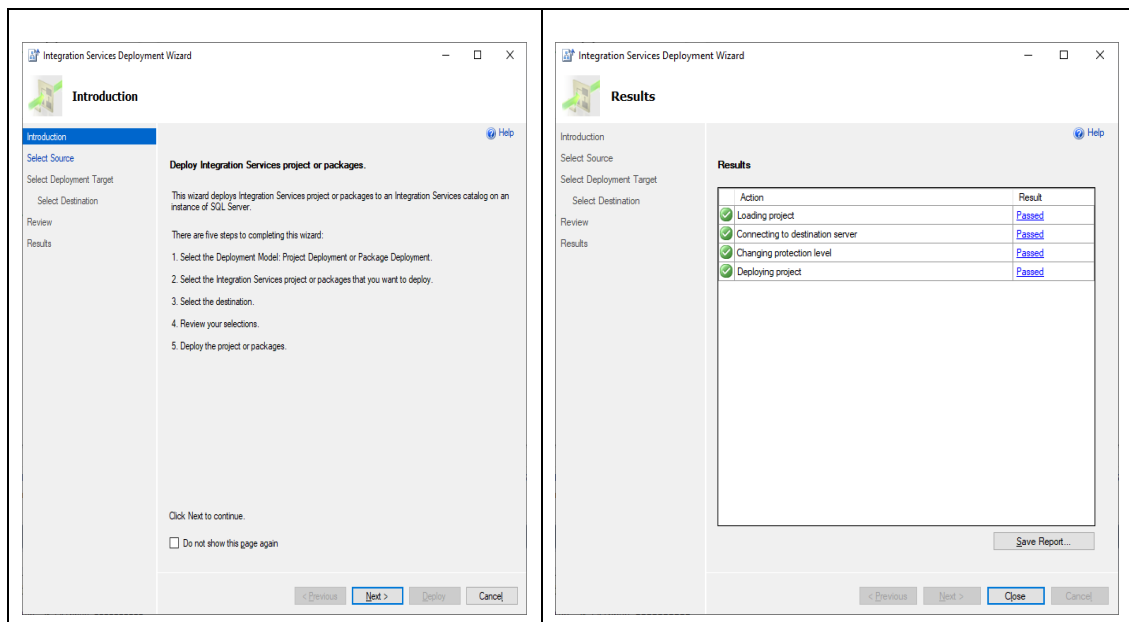
A telepítés során fontos figyelni arra, hogy jelöljük az 'enable automatic execution of...', mivel ezzel fogjuk engedélyezni a Catalog karbantartását, automatikus logtakarítást.

Telepítést követően létrejön egy SSISDB, mind az ISC-ben, mind pedig az adatbázisok között létrejön. Megemlítendő, hogy azon felhasználókat, akiket szeretnénk, hogy adminisztrálják a Catalog-ot, hozzá kell rendeljük a Databases/SSISDB/Security/Roles/ssid_admin szerepkörhöz (8.1.3.4. ábra: Project Deployment Model telepítése). Éles rendszernél érdemes a telepítés után ezzel folytatni.



8.1.3.4. ábra: Project Deployment Model telepítése

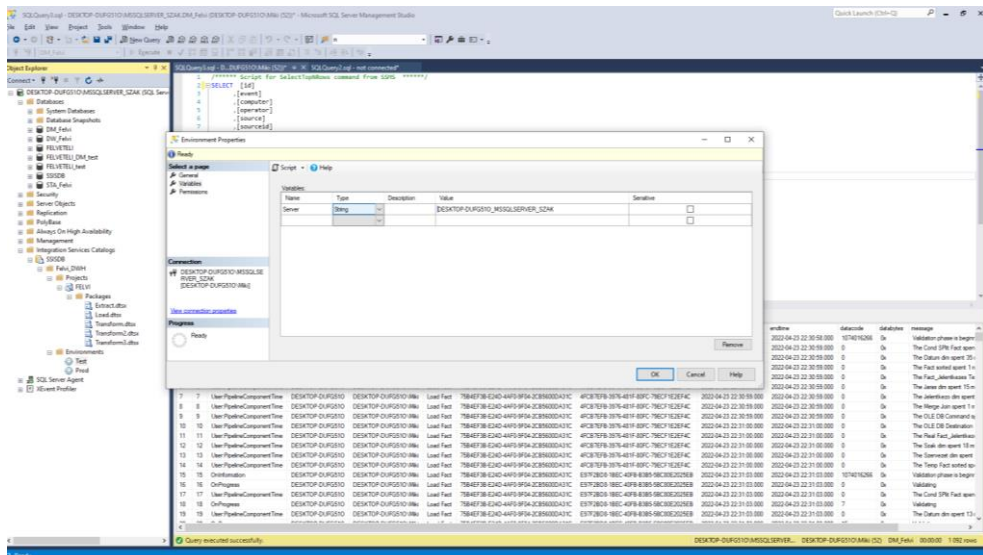
Az ISC /SSISDB-n létrehozunk egy mappát. 'Projects' és 'Enviroments' mappákat fog tartalmazni, ezt követően megyünk vissza az SSIS-be és buildet követően deploy-t teszünk a solution-ön (8.1.3.5. ábra: *Project Deployment Model SSIS/ Deploy*). Beállítjuk a .ispac fájlt, majd az imént létrehozott, mappába telepítjük a soulution-ünket.



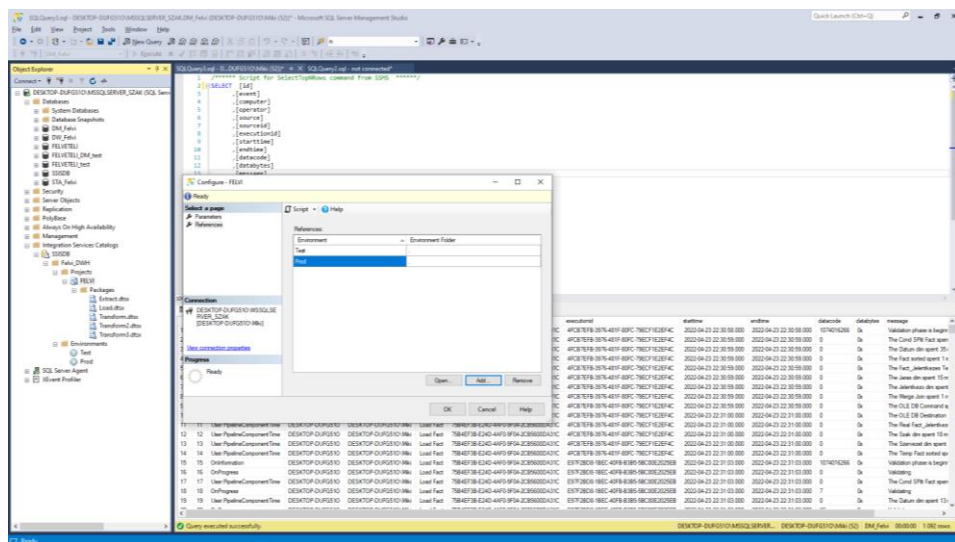
8.1.3.5. ábra: *Project Deployment Model SSIS/ Deploy*

Ezt követően egy refresh az SSISDB-n és láthatjuk Management Studioban a csomagjainkat (8.1.3.6. ábra: *Prod/Test környezet beállításai*).

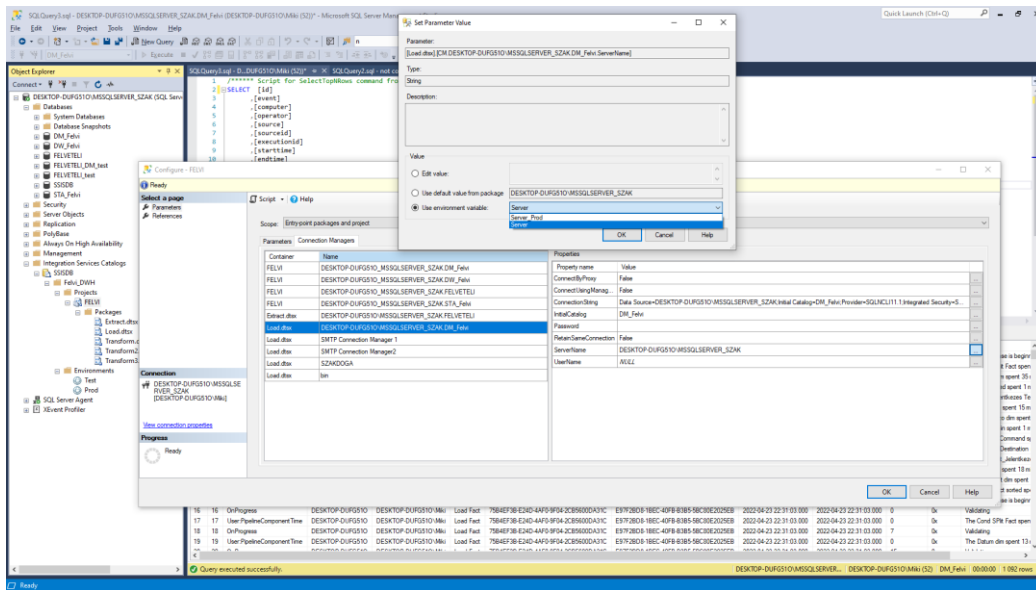
Ezt követi a futtatási környezetek, Test és Prod kialakítása (Environments) (8.1.3.6. ábra: *Prod/Test környezet beállításai*). Ennek a jelentősége, hogy a két környezetet külön tudjuk paraméterezni és a csomag futtatásakor, csupán kiválasztjuk a szükséges környezetet ehhez. Ezt követően a környezetekben változókat definiálunk, mi most csak egy string változót, ami a szerver nevét fogja paraméterben átadni. Ezeket a környezeteket Configure segítségével hozzárendeljük a projektünkhöz, ahol is referenciákat (Test, Prod) adunk hozzá (8.1.3.7. ábra: *A két környezet hozzárendelése a projekthez, referenciaként*). Végül pedig be is kell, hogy map-eljük a paramétereket (8.1.3.8. ábra: *A környezetek map-elése*).



8.1.3.6. ábra: Prod/Test környezet beállításai

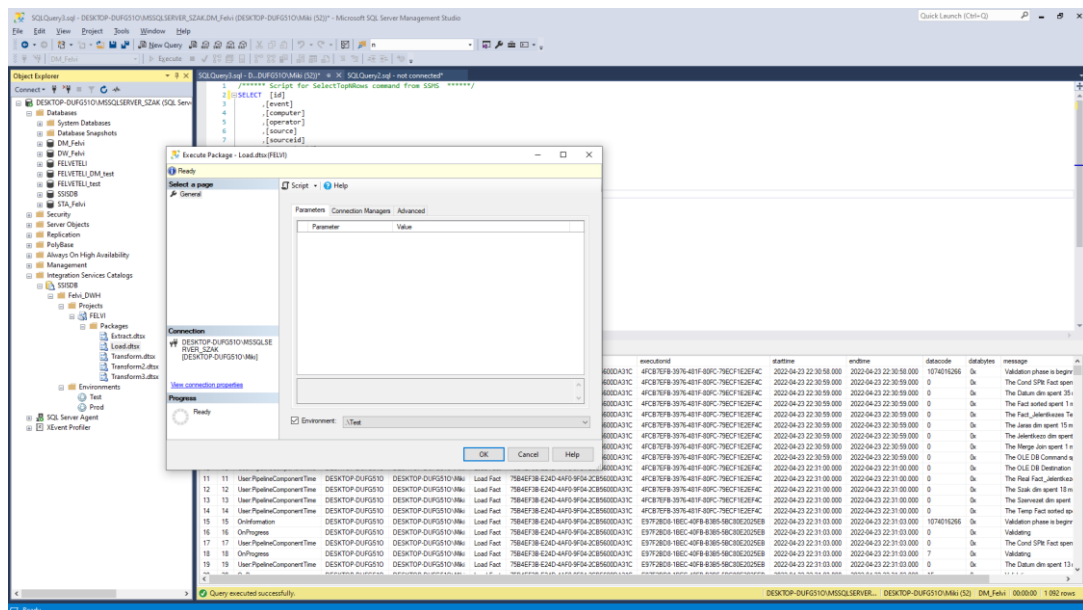


8.1.3.7. ábra: A két környezet hozzárendelése a projekthez, referenciaként

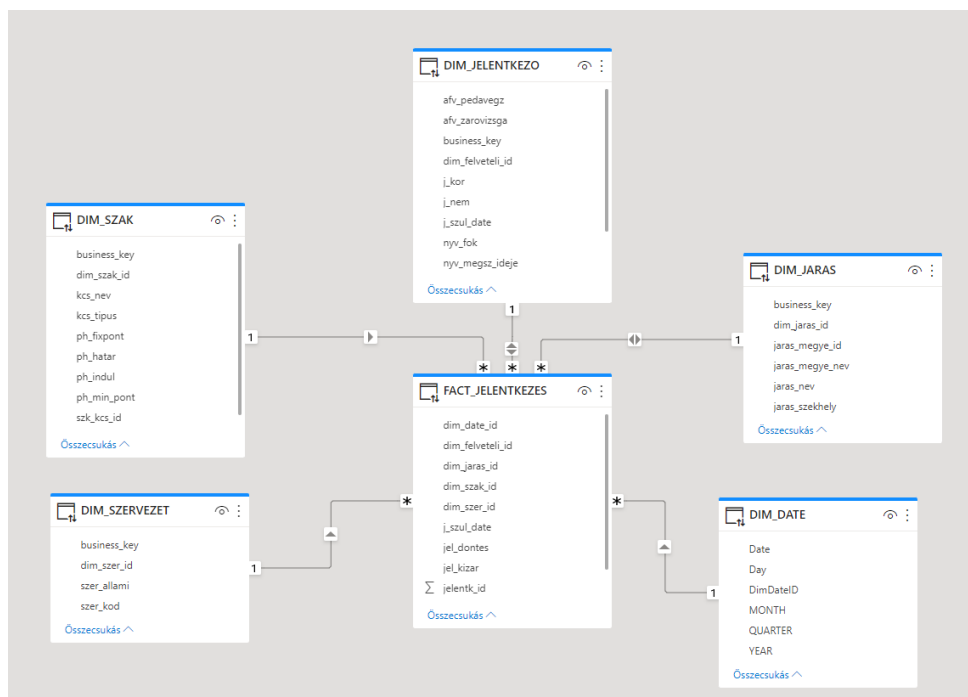


8.1.3.8. ábra: A környezetek map-elése

Ezek után indíthatunk egy execute-ot a Load csomagon kiválasztva, hogy melyik környezetből fusson (8.1.3.9. ábra: A Test környezetből futtatjuk). A felugró ablakban megtekinthetjük az 'Overview Reportot', ami az automatikus naplózás (8.1.3.10. ábra: Az automatikus naplózás megtekintéséről felugró ablak, csomag lefutását követően).



8.1.3.9. ábra: A Test környezetből futtatjuk



9.1. ábra: Tábla betöltés DirectQuery-vel

A use-case-ekben meghatározottak alapján, a szakma az alábbi kérdésekre keresi a választ:

1. A felsőoktatásba jelentkezők területi megoszlása nemenként (Magyarország).
2. Intézményenként és képzési területenként a jelentkezők száma.
3. Jelentkezők átlagéletkora megyénként, informatika képzési területen.
4. Összpontszámok, átlagának alakulása intézményenként és képzési területenként.
5. Jelentkezők nyelvvizsgájának számossága, intézményenként, megyénként, kor és nem szerint
6. Adott szakra (mérnökinformatikus) jelentkezők megoszlása megye és intézmény szerint.
7. Alapfokú végzettséggel rendelkező, felsőoktatásba jelentkezők számossága, akik pedagógus végzettségűek, megyei bontásban.
8. Intézményenként az 1-nél több nyelvvizsgával rendelkezők, nyelvvizsga megszerzésének dátuma szerint.

9.1. A felsőoktatásba jelentkezők területi megoszlása nemenként (Magyarország).

Az alábbi képen az első use case-re táblázatos formában mutatjuk meg a jelentkezői megoszlást járásonként, megyénként (9.1. ábra: A jelentkezők területi megoszlása).

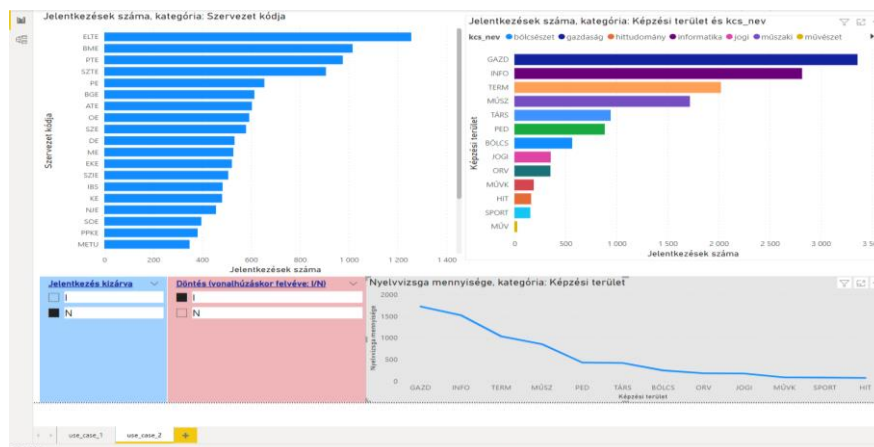
Felhasznált mérték: MES_felsooktba_jelentkezo = COUNTROWS(DIM_JELENTKEZO)

Vágólap		Adatok	
jaras_megye_nev	jaras_nev	j_nem	felsooktba_jelentkezo
Bács-Kiskun megye	Bácsalmási járás	F	25
Bács-Kiskun megye	Bácsalmási járás	N	33
Bács-Kiskun megye	Bajai járás	F	29
Bács-Kiskun megye	Bajai járás	N	29
Bács-Kiskun megye	Jánoshalmi járás	F	27
Bács-Kiskun megye	Jánoshalmi járás	N	31
Bács-Kiskun megye	Kalocsa járás	F	25
Bács-Kiskun megye	Kalocsa járás	N	33
Bács-Kiskun megye	Kecskeméti járás	F	25
Bács-Kiskun megye	Kecskeméti járás	N	33
Bács-Kiskun megye	Kiskőrösi járás	F	22
Bács-Kiskun megye	Kiskőrösi járás	N	36
Bács-Kiskun megye	Kiskunfélegyházi járás	F	31
Bács-Kiskun megye	Kiskunfélegyházi járás	N	27
Bács-Kiskun megye	Kiskunhalasi járás	F	28
Bács-Kiskun megye	Kiskunhalasi járás	N	30
Bács-Kiskun megye	Kiskunmajsai járás	F	28
Bács-Kiskun megye	Kiskunmajsai járás	N	30
Bács-Kiskun megye	Kunszentmiklósi járás	F	32
Bács-Kiskun megye	Kunszentmiklósi járás	N	26
Bács-Kiskun megye	Tiszakécskei járás	F	28
Bács-Kiskun megye	Tiszakécskei járás	N	30
Baranya megye	Bólyi járás	F	22
Baranya megye	Bólyi járás	N	36
Baranya megye	Hegyháti járás	F	24
Baranya megye	Hegyháti járás	N	34
Baranya megye	Komlói járás	F	34
Baranya megye	Komlói járás	N	24
Baranya megye	Mohácsi járás	F	28
Baranya megye	Mohácsi járás	N	30
Baranya megye	Pécsi járás	F	28
Baranya megye	Pécsi járás	N	28
Összesen			10128

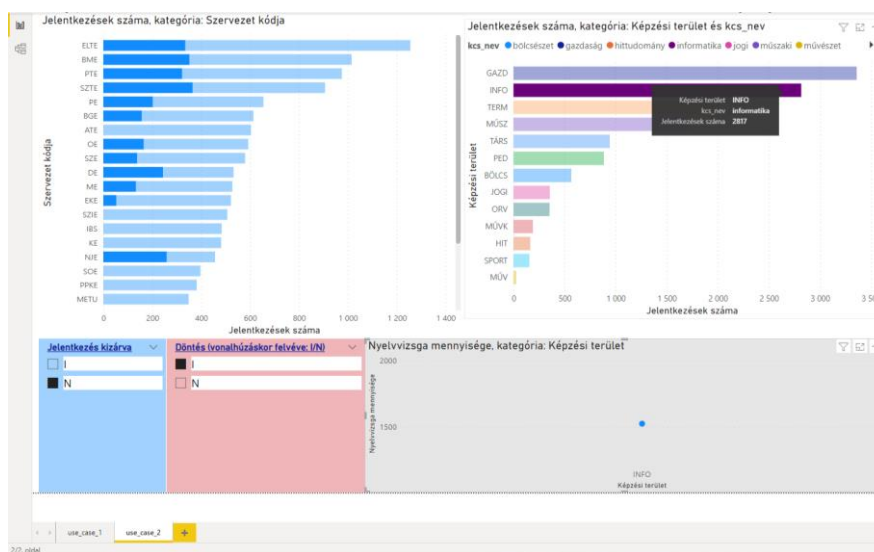
9.1. ábra: A jelentkezők területi megoszlása

9.2. Intézményenként és képzési területenként a jelentkezések száma.

Az alábbi dashboard-okon, sávdigramokon látható a jelentkezéseknek a use-case-ben megfogalmazottak szerinti megoszlása. Szeletelő segítségével lehet módosítani kizárt, illetve felvett jelentkezők között. Az első képen összes (9.2.a. ábra: Megoszlás, intézményenként, képzési területenként), míg a másodikon az 'INFO' képzési területre jelentkezők láthatók, akiket felvettünk a vonalhúzás során (9.2.b. ábra: Megoszlás, intézményenként, képzési területenként).



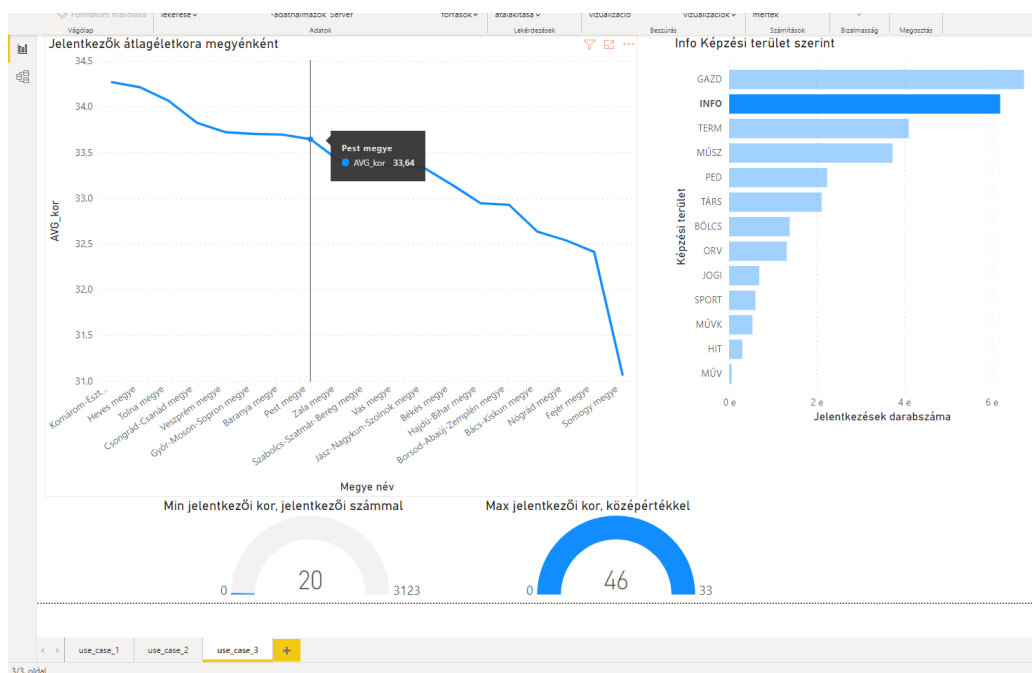
9.2.a. ábra: Megoszlás, intézményenként, képzési területenként



9.2.b. ábra: Megoszlás, intézményenként, képzési területenként

9.3. Jelentkezők átlagéletkora megyénként informatika képzési területen.

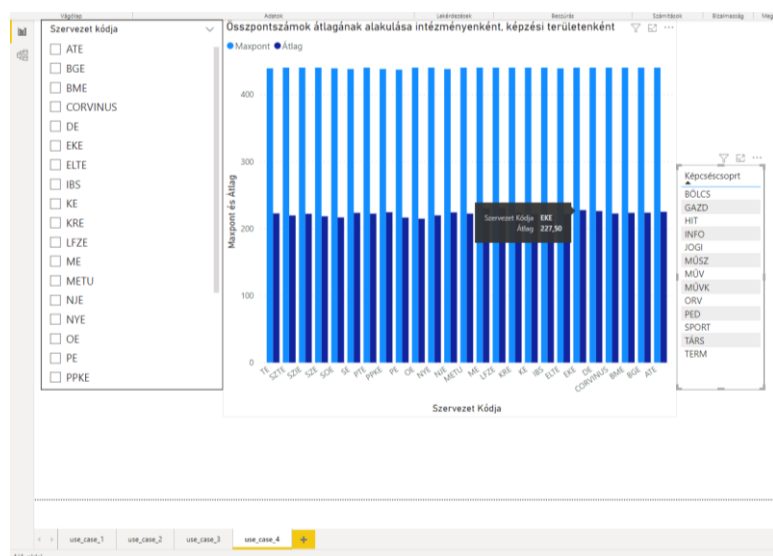
Az alábbiakban vonal illetve sávdiaagramon látható az informatikai területre jelentkezők területi megoszlása, mérővel a legfiatalabb, legidősebb jelentkezőkkel (9.3. ábra: *Jelentkezők kor, terület, képzési területi megoszlása*).



9.3. ábra: Jelentkezők kor, terület, képzési területi megoszlása

9.4. Összpontszámok átlagának alakulása intézményenként és képzési területenként.

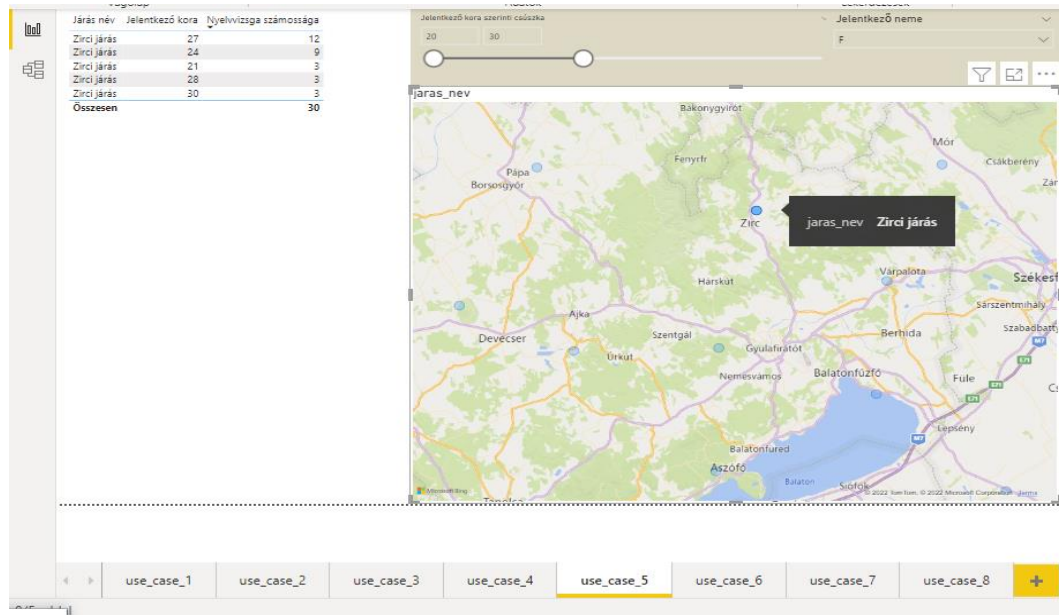
A következőkben a jelentkezők által elért maximális pontszámot, illetve ezek átlagát mutatja az oszlopdiagram, szűrési lehetőséggel intézményre, képzési területre (9.4. ábra: Összpontszámok, intézményenként, képzési területenként).



9.4. ábra: Összpontszámok, intézményenként, képzési területenként

9.5. Jelentkezők nyelvvizsgájának számossága intézményenként, megyénként, kor és nem szerint.

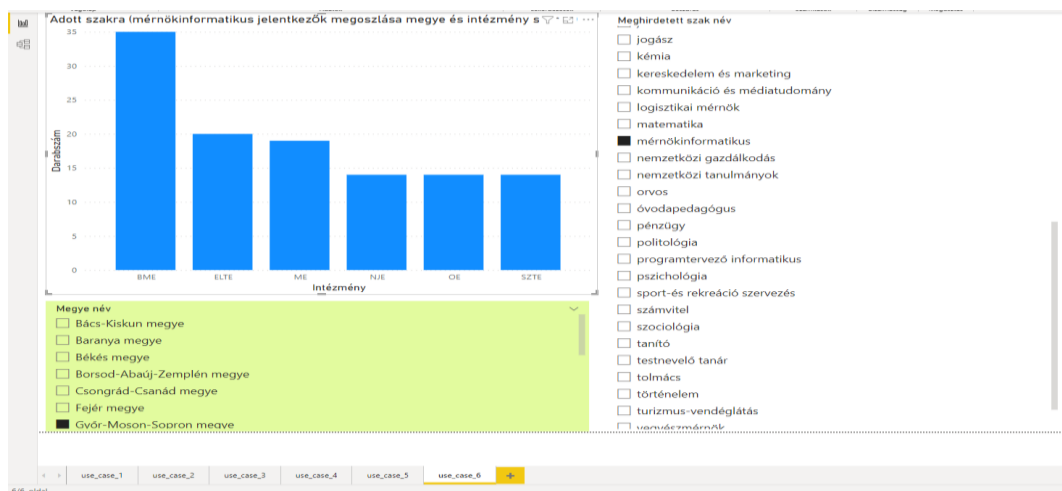
Az alábbi dashboard-on láthatjuk a jelentkezői nyelvvizsgák megoszlását, egy adott járásra leszűkítve, szeletelő csúszkán bizonyos korú jelentkezőkre szűrve, térképes megjelenítéssel (9.5. ábra: *Jelentkezők nyelvvizsgájának megoszlása*).



9.5. ábra: Jelentkezők nyelvvizsgájának megoszlása

9.6. Adott szakra (mérnökinformatikus) jelentkezők megoszlása megye és intézmény szerint.

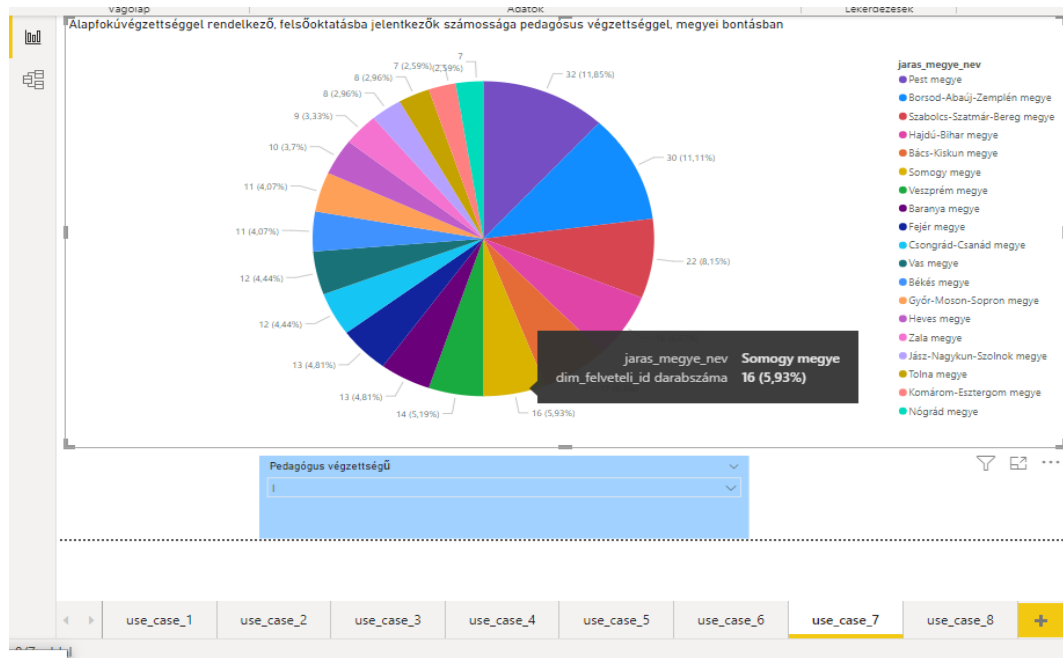
Az alábbi dashboard-on szeletelőkkel, illetve oszlopdiagram segítségével láthatjuk a use case-ben megfogalmazott kérdésre a választ (9.6. ábra: *Mérnökinformatikus szakra jelentkezők*).



9.6. ábra: Mérnökinformatikus szakra jelentkezők

9.7. Alapfokú végzettséggel rendelkező felsőoktatásba jelentkezők számossága, akik pedagógus végzettségűek, megyei bontásban.

Az alábbi dashboard-on tortadiagram segítségével mutatjuk be a pedagógus Bsc diplomával rendelkező jelentkezők számosságát (9.7. ábra: Pedagógus diplomával jelentkezők)

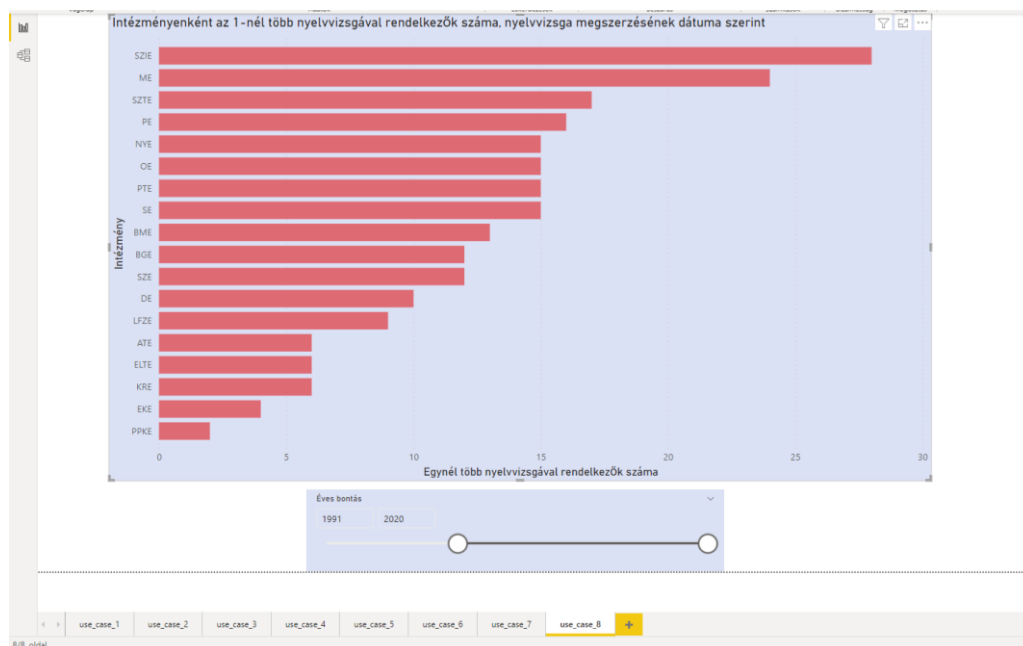


9.7. ábra: Pedagógus diplomával jelentkezők

9.8. Intézményenként az 1-nél több nyelvvizsgával rendelkezők, nyelvvizsga megszerzésének dátuma szerint.

Az alábbi dashboard-on láthatjuk az utolsó use-case-ben megfogalmazott kérdésre a választ, sávdiaagram és egy évválasztó csúszka segítségével (9.8. ábra: Egnél több nyelvvizsgások).

Felhasznált mérték: MES_nyv_menny2_nagyobb =
`CALCULATE(SUM(FACT_JELENTKEZES[nyelv_menny]),FILTER(FACT_JELENTKEZES,FACT_JELENTKEZES[nyelv_menny]>1))`



9.8. ábra: Egynél több nyelvvizsgások

10. A megvalósítás elemzése, továbbfejlesztési lehetőségek számbavétele

A eddigi fejezetek során felépítettünk – a felvételi eljárás vonalhúzását követően kialakult adatokból – egy adattárházat. Ezzel a fejlesztéssel egyben az is bemutatásra került, hogy milyen komplex folyamat az adattárház építése, a megfelelő használati esetek megfogalmazása, a megfelelő fejlesztő rendszer kiválasztása, megfelelő adatkörök meghatározása, logikai/fizikai tervezés.

A munka során létrehoztunk egy olyan rendszert, mely bizonyos időközönként – amit a szakmával szükséges meghatározni – a forrásrendszerek adataira támaszkodva, frissíti az adattárházat, illetve az adatpiacot. Megvalósítottunk egy részletes logolási, tesztelési, monitorozási mechanizmust, melynek segítségével fejlesztői eszköz hiányában (ssis), nyomon követhető a csomagok futásának hatásai, megfelelő joggal. Ezek a csomagok le is futtathatók Management Studio-ból, ezáltal ledelegálhatók bizonyos feladatok az alkalmazásüzemeltetés felé.

Ez a demo verzió a szakmai oldal számára is szeretné bemutatni a DWH alkalmazhatóságát a felvételi eljárásnak egy általános eljárása során, annak előnyeit, hátrányait bemutatva. Ilyen utóbbi például a strukturáltságból adódóan létrejövő korlátozás, a séma kötöttsége.

Ebből következik, hogy az adattárház fejlesztése nem ér véget a megrendelőnek történő átadással, a kialakított rendszert egyfelől támogatni (support) kell. Másfelől figyelembe szükséges azt is venni, hogy az adatok, forrás sémák változhatnak, aminek következtében

az adattárházba új táblák kerülhetnek be, emiatt új dimenziók, ténytáblák létrehozása lehet szükséges, a meglévőket megeshet, hogy bővíteni kell. Így a riport táblákat is ezek függvényében kell módosítanunk.

Ami a továbbfejlesztési lehetőségeket illeti, elsőként azt kell megemlítenünk, hogy a jelen munka során egyetlen fő eljárásra összpontosítottunk. Ezt viszont korábbi eljárásokkal is célszerű lenne kiegészíteni több évre visszamenőleg, amikor ez az adattárház éles adatokon épülne fel a jövőben! Emiatt viszont olyan problematikára is figyelni kell majd, mint például az intézményi átalakulások kérdése – hogy csak egyet említsünk. Ez az eset nagyon is jellemző és ennek minden verziójára fel kell készülnünk, és meg kell oldani az adattárházban ezek egyértelmű beazonosítását. Ilyen változás volt például a közelmúltban:

SZIE-ből lett MATE, EKE-ből lett EKE, THE

Továbbfejlesztés lehetne még első között, jobok létrehozása, melyek segítségével a szakmával, rendszerüzemeltetéssel egyeztetett időpontokban, automatikusan futnának az SSIS csomagok.

Mindenképpen megoldandó élesre átálláskor a jogok és hozzáférések beállítása, hogy a csomagok hiba nélkül tudjanak futni. A felvételi eljárásban jelenleg több adatbáziskezelő működik, a központi adatbázis DB2-n, a felvi.hu MySQL-en alapszik, ezeket meg kell oldani a rendszerüzemeltetőkkel, hogy MSSQL felé problémamentesen működhessen minden.

Az imént említettek közül következik, hogy webes adatforrásoknál http request kiépítése szükséges. Connection types, tehát milyen user/password kellenek, (credential and permission).

Különösen fontos az adat formátuma, például dátum kezelése a korábban említett adatbáziskezelőknél, amire szintén figyelni kell.

Szükséges azt is megtervezni, hogy mikor töltsük át az adatokat, amikor azok a legjobb minőségűek és ezzel nem zavarunk más folyamatokat, sem felhasználókat. Tervet kell létrehozni arra is, hogy mi történik, ha nem sikerül az áttöltés a forrásrendszerből? Később ismét megpróbáljuk? Kimarad az aznapi töltés? Ekkor adatkonzisztenciára is kiemelten kell figyelni.

Ki kell alakítani éles rendszerrel továbbá az architektúrát is, amit a rendszergazdákval közösen kell elvégeznünk:

- Memória: az adattárház leggyakrabban lekérdezett adatszelete (például egy főeljárás) beleférjen.
- Processzor: terhelésteszték szükségesek annak megállapítására, hogy az adattárház lekérdezései mennyire számításigényesek.

- Storage: írás/olvasás sebessége, mekkora sávszélesség áll rendelkezésére a rendszerüzemeltetésnek.
- Lemez: ennek tekintetében, minél több legyen a parallel olvasás lehetősége.

Érdemes továbbfejlesztés tekintetében átgondolni a Data Quality Service (DQS) és a Master Data Management (MDM) megvalósítását.

DQS segítségével az SSIS-ben adatellenőrzéseket tudunk végezni, mielőtt töltjük a forrásból az adatokat. A Creating Matching Policy-vel (similarity, weight, prerequisite) szabályokat lehet felállítani, amit finomítani tudunk, egészen a végleges állapotig. Ennek segítségével mindenre lehet ellenőrzést tenni, duplikációkat kiszűrni, adakonzisztenciára figyelni, hogy az adattárházunk szépen, tisztán legyen tartva.

MDM segítségével kiszűrhető, ha több rendszerben ugyanarról tárolunk adatot és ezek mindegyik rendszerben máshogy jelennek meg. Felvételiben csak abban az esetben szükséges, ha a FIR-hez (Felsőoktatási Információs Rendszer) kötése is megtörténne, mivel ott bizonyos szakok máshogy vannak elnevezve.

Összefoglalva, a továbbfejlesztések több terület, csoport együttműködését igénylik (szakma, adatbázisfejlesztők, rendszerüzemeltetés, alkalmazásüzemeltetés). A fejlesztés megtervezését követően, egy folyamat menedzser tudna irányítani a feladatok pontos végrehajtását.

11. Összefoglaló

A magyarországi felsőoktatási felvételi eljárásra készített jelen demo adattárház hiánypótló. A szükséges adatkörökkel kiegészítve, éles megvalósításával nagyban hozzájárulhat a felvételi adatok gyors kiértékeléséhez, elősegítve a vezetők gyors döntéshozatalát.

A felépített adattárház létjogosultságát az is megalapozhatja, hogy a felvételi környezetbe több rendszerből érkeznek az adatok, ilyen például a Gólya (intézmények felvételi adatrögzítő felülete), KÉNY (központi érettségi nyilvántartó rendszer), felvi.hu (jelentkezők jelentkezését lehetővé tevő felület) és sok más belső fejlesztésű program.

Munkám során ugyan fiktív adatokkal dolgoztam, de ettől függetlenül, igyekeztem úgy felépíteni a rendszert, hogy a szakmai kollégák számára is könnyen megérthető legyen, a fejlesztők, folyamat menedzserek láthassák, hogy mindez adoptálható lenne egy a jövőben létrejövő felvételi adattárházba.

12. Summary

This demo DWH (data warehouse) for the higher education admission procedure of Hungary is a niche. Expanded it with the needed real data this product would be efficient to forward decisions of leaders by rapid evaluation of the process/data.

The validation of this DWH can be founded on the fact that the data in higher educational environment come from different systems, like GÓLYA (the system for the university's HR to manage the applicants data), KÉNY (central program to store applicants final examination data), FELVI.HU (web site where applicants can apply for higher education) and many other programs what was developed by Educational Authority.

Although this thesis is based on mock data, I tried to build this system that it could be easily understandable for the professional colleagues. Furthermore, it would be possible for developers and scrum masters to adapt the process to DWH of higher education admission procedure of the future.

13. Irodalomjegyzék

[1] Felvi.hu egyetemek, főiskolák meghirdetett képzései,
(https://www.felvi.hu/felveteli/egyetemek_foiskolak/!IntezmenyiOldalak/szervezet.php?szer_i d=6&oldal=2&elj=22k), utoljára megtekintve: 2021. 11. 01.

[2] Felvi.hu Felsőoktatási felvételi tájékoztató,
(https://www.felvi.hu/felveteli/jelentkezes/korabbi_elj_archivum/felveteli_tajekoztatok/FFT_2021A/1_teendok_hataridok/14_feltoltendok), utoljára megtekintve: 2021. 11. 06.

[3] Adatbázis tükrözés, Microsoft dokumentáció,
(<https://docs.microsoft.com/hu-hu/sql/database-engine/database-mirroring/force-service-in-a-database-mirroring-session-transact-sql?view=sql-server-2017>), utoljára megtekintve: 2021. 11. 06.

[4] Ralph Kimball, Margy Ross: The Data Warehouse Toolkit- 3rd Edition.
John Wiley & Sons, Inc. 10475 Crosspoint Boulevard Indianapolis, IN 46256

[5] Összefoglaló az adattárházak témaköréről, Elte,
(<http://scs.web.elte.hu/Work/DW/adattarhazak.htm>), utoljára megtekintve: 2021.11.06.

[6] Bánné Varga Gabriella: Az adattárház-készítés technológiája. Typotex, 2012

[7] Oktatas.hu, a Felsőoktatási Felvételi Eljárás, rövidebb összefoglalása,
(https://www.oktatas.hu/pub_bin/dload/kozoktatas/pok/Budapest/2020-as_felsooktatasi_felveteli_eljaras_Petho_Judit.pdf), utoljára megtekintve: 2021. 11. 07.

[8] A csillagséma és a Power BI –ban játszott szerepének bemutatása, Microsoft dokumentáció,
(<https://docs.microsoft.com/hu-hu/power-bi/guidance/star-schema>), utoljára megtekintve: 2021. 11. 27.

- [9] Radványi, T.: Adatbázisrendszerek,
(https://dtk.tankonyvtar.hu/xmlui/bitstream/handle/123456789/3476/2011-0038_22_radvanyi_hu.pdf?sequence=1&isAllowed=y), utoljára megtekintve: 2021.11.01.
- [10] Adatbázisrendszer megvalósítása,
(<https://people.inf.elte.hu/branyi/ora/gyak2/adatbazisrendszerek/Adatbazisrendszerek%20megvalositasa%200-249old.PDF>), utoljára megtekintve: 2021.11.01.
- [11] Felvételi ponthatárok, statisztikák,
(https://www.felvi.hu/felveteli/ponthatarok_statisztikak), utoljára megtekintve: 2021.11.10.
- [12] MS Power BI alkalmazási területei. Microsoft dokumentáció,
(<https://powerbi.microsoft.com/hu-hu/what-is-power-bi/>), utoljára megtekintve: 2021.11.04.
- [13] SSMS rendelkezésre állás, Microsoft dokumentáció,
(<https://docs.microsoft.com/hu-hu/mem/configmgr/core/servers/deploy/configure/high-availability-options>), utoljára megtekintve: 2021.11.04.
- [14] Ullman J.D., Widom J.: Adatbázisrendszerek; alapvetés, 2. kiadás. PANEM Kiadó, 2008
- [15] OLTP és OLAP rendszerek ,
(<https://www.datawarehouse4u.info/OLTP-vs-OLAP.html>), utoljára megtekintve: 2022. 04. 20.
- [16] Az adattárház ETL folyamatai, Microsoft dokumentáció,
(<https://docs.microsoft.com/hu-hu/azure/architecture/data-guide/relational-data/etl>),
utoljára megtekintve: 2022. 04. 20.
- [17] Change Data Capture, Microsoft dokumentáció,
(<https://docs.microsoft.com/en-us/sql/relational-databases/system-stored-procedures/sys-sp-cdc-enable-db-transact-sql?view=sql-server-ver15>), utoljára megtekintve: 2022. 04. 15.
- [18] Change Data Capture, Microsoft dokumentáció,
(<https://docs.microsoft.com/en-us/sql/relational-databases/track-changes/about-change-data-capture-sql-server?view=sql-server-ver15>), utoljára megtekintve: 2022. 04. 15.
- [19] Change Tracking, Microsoft dokumentáció,
(<https://docs.microsoft.com/en-us/sql/relational-databases/track-changes/about-change-tracking-sql-server?view=sql-server-ver15>), utoljára megtekintve: 2022. 04. 15.
- [20] SCD, Microsoft dokumentáció,
(<https://docs.microsoft.com/en-us/sql/integration-services/data-flow/transformations/slowly-changing-dimension-transformation?view=sql-server-ver15>), utoljára megtekintve: 2022. 04. 20.

[21] SSIS logging, Microsoft dokumentáció,
(<https://docs.microsoft.com/en-us/sql/integration-services/performance/integration-services-ssis-logging?view=sql-server-ver15>), utoljára megtekintve: 2022.04.28.

[22] SSIS Project Deployment Model és Package DEployment Model, Microsoft dokumentáció,
(<https://docs.microsoft.com/en-us/sql/integration-services/packages/deploy-integration-services-ssis-projects-and-packages?view=sql-server-ver15>), utoljára megtekintve: 2022. 04. 28.

[23] Power BI DirectQuery, Microsoft dokumentáció,
(<https://docs.microsoft.com/hu-hu/power-bi/connect-data/desktop-use-directquery>), utoljára megtekintve: 2022. 04. 28.

1. sz. melléklet

Mapping (Adatbázisok közötti adatváltozásokat, kulcsokat leíró táblázat):

[illegible]

A forrástáblák leírása.

Jelentkezo:

A felvételi eljárásban jelentkezést beadott jelentkező személyes adatait tartalmazza.

j felvételi idő (PK) – egyedi rekordazonosító

j_megye_id (FK) – lakóhely megyéje

j_nev – jelentkező neve
j_anev – jelentkező anyja neve
j_szul_dat – jelentkező születési dátuma
j_nem – jelentkező neme
j_erettsegi_eve – jelentkező érettségi bizonyítvány megszerzésének éve
j_irszt – jelentkező lakóhelyének irányítószáma
j_telepuless – jelentkező lakóhelyének a települése
j_utca – jelentkező lakóhelyének utcája
j_jelentkezes_eve – jelentkezés éve
j_delflag – rekord logikai törlésére szolgáló mező

Erettsegi_Eredmeny:

A felvételi eljárásban jelentkezést beadott jelentkező érettségi eredményeit részletező tábla.

(Minden jelentkező jogszabály szerint min 5 érettségi eredménnyel rendelkezik, ennél lehet több is pl. ha előrehozott érettségit, színtemelő érettségit stb. tesz a jelentkező és tanúsítványt kap, illetve kivételek is vannak pl. a táncművészek már 16 évesen felvételre kerülnek az egyetemre, érettségi nélkül.

erett_id (PK) – egyedi rekordazonosító
erett_felveteli_id (FK) – idegen kulcs a Jelentkezo tábla PK-re
erett_tantargy – érettségi tantárgy megnevezése
erett_szint – érettségi tárgy szintje
erett_jegy – érettségi tárgy elért eredménye, jegyben
erett_szazalek – érettségi tárgy elért eredménye, százalékban
erett_ev – érettségi tárgy éve
erett_delflag – rekord logikai törlésére szolgáló mező

Nyelvvizsga:

A felvételi eljárásban jelentkezést beadott jelentkező nyelvvizsga eredményeit részletező tábla.

nyv_id (PK) – egyedi rekordazonosító
nyv_felveteli_id (FK) – idegen kulcs a Jelentkezo tábla PK-re
nyv_nyelv – nyelvvizsga nyelve
nyv_fok – nyelvvizsga foka (K,F)
nyv_tipus – nyelvvizsga típusa (pl.: érettségi tanúsítvány, külföldi oklevél)
nyv_datum – nyelvvizsga megszerzésének dátuma
nyv_delflag - rekord logikai törlésére szolgáló mező

Szak:

A felvételi eljárásban a Szervezet által meghirdetett szakokat részletező tábla.

szk_id (PK) – egyedi rekordazonosító
szk_szer_id (FK) – idegen kulcs a Szervezet PK -re
szk_kcs_id (FK) – idegen kulcs a Képzéscsoport PK-re
szk_nev – szak neve
szk_kepzesforma – Bsc, Msc képzési szint
szk_munkarend – milyen időszámban zajlik az oktatás
szk_finanszirozas – államilag támogatott vagy önköltség
szk_delflag – rekord logikai törlésére szolgáló mező

Jelentkezés:

A felvételi eljárásban jelentkezést beadott jelentkező jelentkezéseit részletező tábla.

jel_id (PK) – egyedi rekordazonosító
jel_felveteli_id (FK) – idegen kulcs a Jelentkezo PK-re
jel_szer_id (FK) – idegen kulcs a Szervezet tábla PK-re
jel_sorszam – a jelentkezés sorszáma; hányadik helyen jelölte a képzést
jel_szak_id (FK) – idegen kulcs a Szak PK-re
jel_dontes – a vonalhúzás döntése; felvéve: I/N
jel_fizetve – az öt ingyenesen jelölhető képzés felett jelölt képzések után fizettet: I/N
jel_kizar – kizárásra került-e az adott jelhelyről pl: alkalmassági miatt: I/N
jel_delflag – rekord logikai törlésére szolgáló mező

Járas:

A felvételi eljárásban jelentkezést beadott jelentkező lakóhely megyéjét , járást részletező tábla.

jaras_id (PK) egyedi rekordazonosító, járás
jaras_nev – járás neve
jaras_szekh – járáshoz tartozó székhely (város)
jaras_telepules – járáshoz tartozó település
jaras_telpules_varos – településhez tartozó város
jaras_leleksz – jarasban lakók számossága
jaras_terulet – terület adatok
jaras_nepsuruseg – járáshoz tartozó lakosságszám
jaras_megye_id – jaras, mely megyébe tartozik
jaras_megye_nev – jaras megyéjének neve
jaras_delflag – rekord logikai törlésére szolgáló mező

Ponthatar:

A felvételi eljárásban jelentkezést beadott jelentkező által megjelölt szak ponthatárát részletező tábla.

ph_szak_id (PK) – egyedi rekordazonosító

ph_indul – indul-e a szak; I/N

ph_max_keret – a szakon az intézmény által meghatározott max keretszám

ph_letszam – a szakon jelentkezők száma

ph_min_pont – vonalhúzáskor meghatározott minimum pont, ami a felvételhez kell

ph_fixpont – legmagasabb jelentkezői pontszám

ph_hatar – vonalhúzás által meghatározott maximális ponthatár

ph_delflag – rekord logikai törlésére szolgáló mező

Szervezet:

A felvételi eljárásban jelentkezést beadott jelentkező által megjelölt intézmény adatait részletező tábla.

szer_id (PK) – egyedi rekordazonosító

szer_kod – intézmény karakteres kódja

szer_allami – állami fenntartású intézmény-e; I/N

szer_delflag – rekord logikai törlésére szolgáló mező

Zaradek:

A felvételi eljárásban jelentkezést beadott jelentkező érettségi, illetve nyelvvizsgában esetleg szereplő bejegyzést részletező tábla.

z_id (PK) – egyedi rekordazonosító

z_nyelvvizsga_id (FK) – idegen kulcs a Nyelvvizsga PK-re

z_erettsegieredmeny_id (FK) – idegen kulcs az Erettsegi_Eredmenyek PK-re

z_szoveg – a záradék leírása; pl: nem volt áram a szerverteremben, ezért nem 8:00 kor kezdődött az érettségi informatikából, hanem délután 3-kor

z_datum – záradék dátuma

z_delflag – rekord logikai törlésére szolgáló mező

Pontszam:

A felvételi eljárásban jelentkezést beadott jelentkező által megjelölt szakon kialakult pontszáma.

pont_id (PK) - egyedi rekordazonosító

pont_erettsegi – az érettségi tárgyakból számolt pontszám

pont_nyelvvizsga – a nyelvvizsgákból számolt pontszám
pont_ossz – a nyelvvizsga és az érettségi pontok összege
pont_alkalmas – alkalmassági vizsgán megfelelt-e; I/N
pont_delflag – rekord logikai törlésére szolgáló mező

Alapfoku_Vegzettseg:

A felvételi eljárásban jelentkezést beadott jelentkező által benyújtott korábbi alapfokú végzettséget igazoló diploma; Bsc végzettség.

afv_id (PK) – egyedig rekordazonosító
afv_felveteli_id (FK) – idegen kulcs a Jelentkezo PK-re
afv_szer_id (FK) – idegen kulcs a Szervezet PK-re
afv_szint – a végzettség szintje; B/M
afv_zarovizsga – rendelkezik-e záróvizsgával; I/N
afv_pedavegz – végzettség pedagógus végzettség-e; I/N
afv_fszak – a felsőfokú végzettség szakjának megnevezése
afv_delflag – rekord logikai törlésére szolgáló mező

Kepzescsoport:

Az adott eljárásban meghirdetett szak jogszabály alapján nyilvántartásba vett képzési területe (pl.: MŰSZ = műszaki, PED = pedagógus)

kcs_id (PK) – egyedi rekordazonosító
kcs_tipus – képzéscsoport típusa; (pl.: PED)
kcs_kod – képzéscsoport kódja
kcs_nev – képzéscsoport megnevezése (pl. pedagógus)
kcs_delflag – rekord logikai törlésére szolgáló mező

2. sz. melléklet

A DM rétegre írt lekérdezések:

--1. A felsőoktatásba jelentkezők területi megoszlása nemenként (Magyarország)

```
select jar.jaras_megye_nev, jar.jaras_nev, j.j_nem, COUNT(j.j_nem) as nemenkent
from dbo.FACT_JELENTKEZES f
join dbo.DIM_JELENTKEZO j on j.dim_felveteli_id=f.dim_felveteli_id
join dbo.DIM_JARAS jar on jar.dim_jaras_id=f.dim_jaras_id
group by jar.jaras_megye_nev, jar.jaras_nev, j.j_nem
order by 1,2
;
```

--2. Intézményenként és képzési területenként a jelentkezések száma

```
select sz.szer_kod, s.kcs_nev, COUNT(1) as jelentkezések_szama
from dbo.FACT_JELENTKEZES f
join dbo.DIM_SZAK s on s.dim_szak_id=f.dim_szak_id and s.ph_indul='I'
join dbo.DIM_SZERVEZET sz on sz.dim_szer_id=f.dim_szer_id
group by sz.szer_kod, s.kcs_nev
order by 1
;
```

--3. Jelentkezők átlagéletkora megyénként informatika képzési területen

```
select jar.jaras_megye_nev, j.j_kor --avg(j.j_kor) as atlag_kor
from dbo.FACT_JELENTKEZES f
join dbo.DIM_JARAS jar on jar.dim_jaras_id=f.dim_jaras_id
join dbo.DIM_JELENTKEZO j on j.dim_felveteli_id=f.dim_felveteli_id
join dbo.DIM_SZAK s on s.dim_szak_id=f.dim_szak_id
where s.szk_kcs_id=4 and s.ph_indul='I'
group by jar.jaras_megye_nev ,j.j_kor
;
```

--4. Összpontszámok átlagának alakulása intézményenként és képzési területenként.

```
select sz.szer_kod, s.kcs_nev, f.pont_osszpont --atlagolni
from dbo.FACT_JELENTKEZES f
join dbo.DIM_SZAK s on s.dim_szak_id=f.dim_szak_id and s.ph_indul='I'
join dbo.DIM_SZERVEZET sz on sz.dim_szer_id=f.dim_szer_id
group by sz.szer_kod, s.kcs_nev , f.pont_osszpont
order by 1
;
```

--5. Jelentkezők nyelvvizsgájának számossága járásonként, kor szerint;

```
select jar.jaras_nev, j.j_kor, COUNT(f.dim_felveteli_id) --count(j.dim_felveteli_id) nyelv_darab
from dbo.FACT_JELENTKEZES f
join dbo.DIM_JARAS jar on jar.dim_jaras_id=f.dim_jaras_id
join dbo.DIM_JELENTKEZO j on j.dim_felveteli_id=f.dim_felveteli_id
join dbo.DIM_SZAK s on s.dim_szak_id=f.dim_szak_id and s.ph_indul='I'
where j.nyv_nyelv is not null
group by jar.jaras_nev, j.j_kor --, j.nyv_nyelv
order by 1,2
;
```

--6. Adott szakra (mérnök-informatikus) jelentkezők megoszlása megye és intézmény szerint.

```

select sz.szer_kod, jar.jaras_megye_nev, COUNT(*) as jelentkezes_szama_mernokinfo
from dbo.FACT_JELENTKEZES f
join dbo.DIM_JARAS jar on jar.dim_jaras_id=f.dim_jaras_id
--join dbo.DIM_JELENTKEZO j on j.dim_felveteli_id=f.dim_felveteli_id
join dbo.DIM_SZAK s on s.dim_szak_id=f.dim_szak_id and s.ph_indul='I'
join dbo.DIM_SZERVEZET sz on sz.dim_szer_id=f.dim_szer_id
where s.szk_szak_nev = 'mérnökinformatikus'
group by sz.szer_kod, jar.jaras_megye_nev
order by 1
;

```

--7. Alapfokú végzettséggel rendelkező felsőoktatásba jelentkezők számossága, akik pedagógus végzettségűek, megyei bontásban.

```

select jar.jaras_megye_nev, COUNT(1) as szamossag
from dbo.FACT_JELENTKEZES f
join dbo.DIM_JELENTKEZO j on j.dim_felveteli_id=f.dim_felveteli_id
join dbo.DIM_JARAS jar on jar.dim_jaras_id=f.dim_jaras_id
where j.afv_zarovizsga='I' and j.afv_pedavegz='I'
group by jar.jaras_megye_nev
order by 1
;

```

--8. Intézményenként a 1-nél több nyelvvizsgával rendelkezők, nyelvvizsga megszerzésének dátuma szerint.

```

select distinct s.szer_kod, f.nyelvv_menny, d.YEAR
from dbo.FACT_JELENTKEZES f
join dbo.DIM_SZERVEZET s on s.dim_szer_id=f.dim_szer_id
join dbo.DIM_DATE d on d.DimDateID=f.dim_date_id
where f.nyelvv_menny > 1 --f.dim_felveteli_id=311748
order by 1
;

```

3. sz. melléklet

DM layer insert:

```

use DM_Felvi
go

```

```

--DW Layer
--dim_jelentkezo
select
j.j_felveteli_id as business_key

```



```

,j.j_kor
,j.j_nem
,j.j_szul_dat
,a.afv_zarovizsga
,a.afv_pedavegz
,n.nyv_nyelv
,n.nyv_fok
,n.nyv_tipus
,n.nyv_datum as nyelvv_megsz_ideje
,CURRENT_TIMESTAMP as dim_jel_load_date
from DW_Felvi.dbo.DW_JELENTKEZO j
left join DW_Felvi.dbo.DW_ALAPFOKU_VEGZETTSEG a on a.afv_felveteli_id=j.j_felveteli_id
left join DW_Felvi.dbo.DW_NYELVVIZSGA n on j.j_felveteli_id=n.nyv_felveteli_id
order by 1
;

```

```

--dim_jaras
select
jaras_id as business_key,
jaras_nev,
jaras_szekh,
jaras_megye_id,
jaras_megye_nev,
CURRENT_TIMESTAMP as dim_jaras_load_date
from DW_Felvi.dbo.DW_JARAS
order by 1
;

```

```

--dim_szak
select
szk_id as business_key,
s.szk_nev ,
k.kcs_id,
k.kcs_nev,
k.kcs_tipus,
p.ph_indul,
p.ph_min_pont,
p.ph_fixpont,
p.ph_hatar,
CURRENT_TIMESTAMP as dim_szak_load_date
from DW_Felvi.dbo.DW_SZAK s
left join DW_Felvi.dbo.DW_KEPZESCSOPORT k on k.kcs_id=s.szk_kcs_id
left join DW_Felvi.dbo.DW_PONTHATAR p on p.ph_szk_id=s.szk_id
where s.szk_valid_to is null
order by 1

```

```

;

--dim_szervezet
select
s.szer_id as business_key,
s.szer_kod,
s.szer_allami,
CURRENT_TIMESTAMP as dim_szer_load_date
from DW_Felvi.dbo.DW_SZERVEZET s
where s.szer_valid_to is null
order by 1
;

--fact_jelentkezés
select
jel.j_felveteli_id as j_felveteli_id_BK,
s.szk_id as szak_id_BK,
YEAR(j_szul_dat) * 10000 + MONTH(j_szul_dat) * 100 + DAY(j_szul_dat) as date_id_BK,
jar.jaras_id as jaras_id_BK,
sz.szer_id as szer_id_BK,
p.pont_ossz,
ny.nyelv_mennyiseg as nyelv_menny,
jel.j_szul_dat as j_szul_date,
CURRENT_TIMESTAMP as fact_jelentkezés_load_date
from DW_Felvi.dbo.DW_JELENTKEZES j
left join DW_Felvi.dbo.DW_PONTSZAM p on p.pont_id=j.jel_id
left join DW_Felvi.dbo.DW_JELENTKEZO jel on jel.j_felveteli_id=j.jel_felveteli_id
left join DW_Felvi.dbo.DW_JARAS jar on jar.jaras_id=jel.j_jaras_id
left join DW_Felvi.dbo.DW_SZAK s on s.szk_id=j.jel_szak_id
left join DW_Felvi.dbo.DW_SZERVEZET sz on sz.szer_id=j.jel_szer_id
left join DW_Felvi.dbo.VW_NYELVV_MENNY ny on ny.nyv_felveteli_id=j.jel_felveteli_id
--where j.jel_id=562
;

```

4. sz. melléklet

STA_DDW_DM_ddl:

```

create database STA_Felvi COLLATE Hungarian_CI_AS
go
create database DW_Felvi COLLATE Hungarian_CI_AS
go
create database DM_Felvi COLLATE Hungarian_CI_AS
go

```

```

--STAGE LAYER
use STA_Felvi
go

create table STA_Felvi.dbo.STA_JELENTKEZO(
j_felveteli_id NVARCHAR(256),
j_megye_id NVARCHAR(256),
j_jaras_id NVARCHAR(256),
j_nev NVARCHAR(256),
j_anev NVARCHAR(256),
j_nem NVARCHAR(256),
j_erttsegi_eve NVARCHAR(256),
j_irsz NVARCHAR(256),
j_telepules NVARCHAR(256),
j_utca NVARCHAR(256),
j_jelentkezes_eve NVARCHAR(256),
j_delflag NVARCHAR(256),
j_szul_dat NVARCHAR(256)
);
--drop table dbo.STA_JELENTKEZO

CREATE table STA_Felvi.dbo.STA_SZAK(
szk_id NVARCHAR(256),
szk_szer_id NVARCHAR(256),
szk_kcs_id NVARCHAR(256),
szk_nev NVARCHAR(256),
szk_kepzesforma NVARCHAR(256),
szk_munkarend NVARCHAR(256),
szk_finanszirozas NVARCHAR(256),
szk_delflag NVARCHAR(256),
);
--drop table dbo.STA_SZAK

CREATE TABLE STA_Felvi.dbo.STA_JELENTKEZES(
jel_id NVARCHAR(256),
jel_felveteli_id NVARCHAR(256),
jel_szer_id NVARCHAR(256),
jel_sorszam NVARCHAR(256),
jel_szak_id NVARCHAR(256),
jel_dontes NVARCHAR(256),
jel_fizetve NVARCHAR(256),
jel_kizar NVARCHAR(256),
jel_delflag NVARCHAR(256)
);

```

```

create table STA_Felvi.dbo.STA_ALAPFOKU_VEGZETTSEG(
afv_id NVARCHAR(256),
afv_felveteli_id NVARCHAR(256),
afv_szer_id NVARCHAR(256),
afv_szint NVARCHAR(256),
afv_zarovizsga NVARCHAR(256),
afv_pedavegz NVARCHAR(256),
afv_fszak NVARCHAR(256),
afv_delflag NVARCHAR(256)
);

```

```

create table STA_Felvi.dbo.STA_SZERVEZET(
szer_id NVARCHAR(256),
szer_kod NVARCHAR(256),
szer_allami NVARCHAR(256),
szer_delflag NVARCHAR(256)
);
--drop table dbo.STA_SZERVEZET

```

```

create table STA_Felvi.dbo.STA_JARAS(
jaras_id NVARCHAR(256),
jaras_nev NVARCHAR(256),
jaras_szekh NVARCHAR(256),
jaras_telepules NVARCHAR(256),
jaras_telepules_varos NVARCHAR(256),
jaras_leleksz NVARCHAR(256),
jaras_terulet NVARCHAR(256),
jaras_nepsuruseg NVARCHAR(256),
jaras_megye_id NVARCHAR(256),
jaras_megye_nev NVARCHAR(256),
jaras_delflag NVARCHAR(256)
);
--drop table dbo.STA_JARAS

```

```

create table STA_Felvi.dbo.STA_PONTSZAM(
pont_id NVARCHAR(256),
pont_ereztsegi NVARCHAR(256),
pont_nyelvvizsga NVARCHAR(256),
pont_ossz NVARCHAR(256),
pont_alkalmas NVARCHAR(256),
pont_delflag NVARCHAR(256)
);
--drop table dbo.STA_PONTSZAM

```

```

create table STA_Felvi.dbo.STA_PONTHATAR(

```

```

ph_szk_id NVARCHAR(256),
ph_indul NVARCHAR(256),
ph_max_keret NVARCHAR(256),
ph_min_letszam NVARCHAR(256),
ph_min_pont NVARCHAR(256),
ph_fixpont NVARCHAR(256),
ph_hatar NVARCHAR(256),
ph_delflag NVARCHAR(256)
);
--drop table dbo.STA_PONTHATAR

create table STA_Felvi.dbo.STA_NYELVVIZSGA(
nyv_id NVARCHAR(256),
nyv_felveteli_id NVARCHAR(256),
nyv_nyelv NVARCHAR(256),
nyv_fok NVARCHAR(256),
nyv_tipus NVARCHAR(256),
nyv_datum NVARCHAR(256),
nyv_delflag NVARCHAR(256)
);
--drop table dbo.STA_NYELVVIZSGA

create table STA_Felvi.dbo.STA_KEPZESCSOPORT(
kcs_id NVARCHAR(256),
kcs_tipus NVARCHAR(256),
kcs_kod NVARCHAR(256),
kcs_nev NVARCHAR(256),
kcs_delflag NVARCHAR(256)
);

--DW LAYER
use DW_Felvi
go

create table DW_Felvi.dbo.DW_JELENTKEZO(
j_felveteli_id INT,
j_megye_id INT,
j_jaras_id INT,
j_nev NVARCHAR(256),
j_anev NVARCHAR(256),
j_nem NVARCHAR(10),
j_kor INT,      --aggregáljuk SSIS-ben
j_erttsegi_eve INT,
j_irs INT,
j_telepules NVARCHAR(256),

```

```

j_utca NVARCHAR(256),
j_jelentkezés_eve NVARCHAR(256),
j_delflag NVARCHAR(256),
j_szul_dat DATE,
j_load_time_stamp TIMESTAMP
);
--drop table dbo.DW_JELENTKEZO

```

```

CREATE table DW_Felvi.dbo.DW_SZAK(
szk_id INT,
szk_szer_id INT,
szk_kcs_id INT,
szk_nev NVARCHAR(100),
szk_kepzesforma NVARCHAR(10),
szk_munkarend NVARCHAR(10),
szk_finanszirozas NVARCHAR(10),
szk_delflag NVARCHAR(256),
szk_valid_from DATE,
szk_valid_to DATE,
szk_load_time_stamp TIMESTAMP
);
--drop table dbo.DW_SZAK

```

```

--SCD, fix
CREATE TABLE DW_Felvi.dbo.DW_JELENTKEZES(
jel_id INT,
jel_felveteli_id INT,
jel_szer_id INT,
jel_sorszam INT,
jel_szak_id INT,
jel_dontes NVARCHAR(10),
jel_fizetve NVARCHAR(10),
jel_kizar NVARCHAR(10),
jel_delflag NVARCHAR(256),
jel_load_time_stamp TIMESTAMP
);

```

```

--SCD, fix
create table DW_Felvi.dbo.DW_ALAPFOKU_VEGZETTSEG(
afv_id INT,
afv_felveteli_id INT,
afv_szer_id INT,
afv_szint NVARCHAR(10),
afv_zarovizsga NVARCHAR(10),
afv_pedavegz NVARCHAR(10),

```

```
afv_fszak NVARCHAR(256),
afv_delflag NVARCHAR(256),
afv_load_time_stamp TIMESTAMP
);
```

```
create table DW_Felvi.dbo.DW_SZERVEZET(
szer_id INT,
szer_kod NVARCHAR(10),
szer_allami NVARCHAR(10),
szer_delflag NVARCHAR(256),
szer_valid_from DATE,
szer_valid_to DATE,
szer_load_time_stamp TIMESTAMP
);
--drop table dbo.DW_SZERVEZET
```

```
--SCD, fix
create table DW_Felvi.dbo.DW_JARAS(
jaras_id INT,
jaras_nev NVARCHAR(256),
jaras_szekh NVARCHAR(256),
jaras_telepules NVARCHAR(256),
jaras_telepules_varos NVARCHAR(256),
jaras_leleksz INT,
jaras_terulet NVARCHAR(256),
jaras_nepsuruseg INT,
jaras_megye_id INT,
jaras_megye_nev NVARCHAR(256),
jaras_delflag NVARCHAR(256),
jaras_load_time_stamp TIMESTAMP
);
--drop table dbo.DW_JARAS
```

```
--SCD, fix
create table DW_Felvi.dbo.DW_PONTSZAM(
pont_id INT,
pont_erettsegi INT,
pont_nyelvvizsga INT,
pont_ossz INT,
pont_alkalmas NVARCHAR(10),
pont_delflag NVARCHAR(256),
pont_load_time_stamp TIMESTAMP
);
--drop table dbo.DW_PONTSZAM
```

```

--SCD, fix
create table DW_Felvi.dbo.DW_PONTHATAR(
ph_szk_id INT,
ph_indul NVARCHAR(10),
ph_max_keret INT,
ph_min_letszam INT,
ph_min_pont INT,
ph_fixpont INT,
ph_hatar INT,
ph_delflag NVARCHAR(256),
ph_load_time_stamp TIMESTAMP
);
--drop table dbo.DW_PONTHATAR

--SCD, fix
create table DW_Felvi.dbo.DW_NYELVVIZSGA(
nyv_id INT,
nyv_felveteli_id INT,
nyv_nyelv NVARCHAR(50),
nyv_fok NVARCHAR(10),
nyv_tipus NVARCHAR(10),
nyv_datum DATE,
nyv_delflag NVARCHAR(256),
nyv_load_time_stamp TIMESTAMP
);
--drop table dbo.DW_NYELVVIZSGA

--SCD, TYPE_1 felülírjuk
create table DW_Felvi.dbo.DW_KEPZESCSOPORT(
kcs_id INT,
kcs_tipus NVARCHAR(50),
kcs_kod INT,
kcs_nev NVARCHAR(256),
kcs_delflag NVARCHAR(256),
kcs_load_time_stamp TIMESTAMP
);

go

-- use DW_Felvi
-- go
-- EXEC sp_MSForEachTable 'TRUNCATE TABLE ?'
-- ;

--VIEW nyelvvizsga aggregáláshoz, DW feltöltése után jön létre

```



```

create or alter view VW_NYELVV_MENNY as
select nyv_felveteli_id, COUNT(*) as nyelv_mennyiseg
      from DW_Felvi.dbo.DW_NYELVVIZSGA
      group by nyv_felveteli_id
;

```

```

--DM LAYER
use DM_Felvi
go

```

```

CREATE TABLE DM_Felvi.dbo.FACT_JELENTKEZES (
      jelentk_id INT IDENTITY(1,1),
      dim_felveteli_id INT,
      dim_szak_id INT,
      dim_date_id INT,
      dim_jaras_id INT,
      dim_szer_id INT,
      jel_dontes NVARCHAR(10),
      jel_kizar NVARCHAR(10),
      pont_osszpont INT,
      nyelv_mennyi INT,
      j_szul_date DATE,
      load_timestamp TIMESTAMP,
);

```

```

CREATE TABLE DM_Felvi.dbo.DIM_JARAS (
      dim_jaras_id INT IDENTITY(1,1),
      business_key INT,
      jaras_nev NVARCHAR(256),
      jaras_szekhely NVARCHAR(256),
      jaras_megye_id INT,
      jaras_megye_nev NVARCHAR(256),
      load_timestamp TIMESTAMP,
);

```

```

CREATE TABLE DM_Felvi.dbo.DIM_JELENTKEZO (
      dim_felveteli_id INT IDENTITY(1,1),
      business_key INT,
      j_kor INT,
      j_nem NVARCHAR(10),
      j_szul_date DATE,
      afv_zarovizsga NVARCHAR(10),
      afv_pedavegz NVARCHAR(10),
      nyv_nyelv NVARCHAR(50),
      nyv_fok NVARCHAR(10),

```

```

        nyv_tpus NVARCHAR(100),
        nyv_megsz_ideje DATE,
        load_timestamp TIMESTAMP,
    );

CREATE TABLE DM_Felvi.dbo.DIM_SZERVEZET(
    dim_szer_id INT IDENTITY(1,1),
    business_key INT,
    szer_kod NVARCHAR(10),
    szer_allami NVARCHAR(10),
    load_timestamp TIMESTAMP,
);

CREATE TABLE DM_Felvi.dbo.DIM_SZAK(
    dim_szak_id INT IDENTITY(1,1),
    business_key INT,
    szk_szak_nev NVARCHAR(100),
    szk_kcs_id INT,
    kcs_nev NVARCHAR(256),
    kcs_tpus NVARCHAR(50),
    ph_indul NVARCHAR(10),
    ph_min_pont INT,
    ph_fixpont INT,
    ph_hatar INT,
    load_timestamp TIMESTAMP,
);

CREATE TABLE DM_Felvi.dbo.DIM_DATE (
    DimDateID INT,
    Date DATE,
    Day INT,
    MONTH INT,
    QUARTER INT,
    YEAR INT,
    LoadTimeStamp TIMESTAMP
);

-----
-----
--DATUM DIMENZIÓ INSERTJE
CREATE OR ALTER PROCEDURE LoadDimDate
@StartDate DATE = '1976-01-28',
@EndDate DATE = '2020-11-03'

AS

```

```

BEGIN
    SET XACT_ABORT ON;
    TRUNCATE TABLE dbo.DIM_DATE;

BEGIN TRAN;

-- amíg nem pörgetjük végig a kezdődátumtól a végdátumig
WHILE @StartDate < @EndDate
BEGIN
    INSERT INTO dbo.DIM_DATE (
        [DimDateID],
        [Date],
        [Day],
        [Month],
        [Quarter],
        [Year]
    )
    SELECT [DimDateID] = YEAR(@StartDate) * 10000 + MONTH(@StartDate) * 100 +
    DAY(@StartDate),
        [Date] = @StartDate,
        [Day] = DAY(@StartDate),
        [Month] = MONTH(@StartDate),
        [Quarter] = DATEPART(q, @StartDate),
        [Year] = YEAR(@StartDate)

    -- növeljük a kezdődátum értékét 1 nappal (következő nap)
    SET @StartDate = DATEADD(DD, 1, @StartDate)
END

SELECT * FROM dbo.DIM_DATE
COMMIT;

END;

--futtajuk
EXEC LoadDimDate
    @StartDate = '1976-01-28',
    @EndDate = '2020-11-03'
    ;

```

5. sz. melléklet

--CONDITIONAL SPLIT, UPDATE FACT:

```

UPDATE DM_Felvi.dbo.FACT_JELENTKEZES
SET dim_felveteli_id = ?, dim_szak_id = ?, dim_date_id = ?,
dim_jaras_id = ?, dim_szer_id = ?, jel_dontes = ?, jel_kizar = ?,

```

```
pont_osszpont = ?, nyelv_v_menny = ?, j_szul_date = ?  
WHERE jelentk_id = ?;
```

JELENTKEZŐ KORÁNAK KISZÁMOLÁSA SSIS-BEN:

```
DATEDIFF( "yy", (DT_DATE)@[User::fixDate], (DT_DATE)[j_szul_dat]) )
```

6. sz. melléklet

A teljesség igénye nélkül, a generált fiktív adatokon végzett főbb módosítások, hogy a use-case feltételek értelmesebb adatot, vizualizációt adjanak (a teljes script az állományhoz csatolva):

```
-- exec sp_rename 'SZERVEZET.kod','szer_kod', 'column';  
-- exec sp_rename 'SZERVEZET.allami','szer_allami', 'column';  
-- exec sp_rename 'SZERVEZET.delflag','szer_delflag', 'column';  
  
create table FELVETELI.dbo.JELENTKEZO(  
    j_felveteli_id INT,  
    j_megye_id INT,  
    j_jaras_id INT,  
    j_nev NVARCHAR(250),  
    j_anev NVARCHAR(250),  
    j_születési_dat NVARCHAR(250), --DATUM LEGYEN!!!!!!!!!!!!!! alter table jelentkezo add column ddd -> ....  
    j_nem NVARCHAR(50), --CHAR(1), SSIS nem szereti STA -ba nem megy be simán  
    j_érettségi_éve INT,  
    j_írsz NVARCHAR(26),  
    j_település NVARCHAR(250),  
    j_utca NVARCHAR(250),  
    j_jelentkezés_éve INT,  
    j_delflag NVARCHAR(256),  
);  
--előbb az insert  
alter table jelentkezo add j_szul_date;  
update dbo.JELENTKEZO set j_szul_dat= cast(j_születési_dat as date);  
alter table dbo.jelentkezo drop column j_születési_dat;  
  
update dbo.KEPZESCSOPORT set kcs_típus='TÁRS', kcs_nev='társadalomtudomány' where kcs_id=0;  
update dbo.KEPZESCSOPORT set kcs_nev='sport' where kcs_id=1;  
update dbo.KEPZESCSOPORT set kcs_nev='természettudomány' where kcs_id=2;  
update dbo.KEPZESCSOPORT set kcs_nev='pedagógia' where kcs_id=3;  
update dbo.KEPZESCSOPORT set kcs_nev='informatika', kcs_kod=5 where kcs_id=4;  
update dbo.KEPZESCSOPORT set kcs_nev='művészetkövetítés' where kcs_id=5;  
update dbo.KEPZESCSOPORT set kcs_nev='agrár' where kcs_id=6;  
update dbo.KEPZESCSOPORT set kcs_kod=7 where kcs_id=7;  
update dbo.KEPZESCSOPORT set kcs_típus='MŰV', kcs_kod=1, kcs_nev='művészet' where kcs_id=8;  
update dbo.KEPZESCSOPORT set kcs_típus='MŰSZ', kcs_nev='műszaki' where kcs_id=12;  
update dbo.KEPZESCSOPORT set kcs_nev='hittudomány' where kcs_id=11;  
update dbo.KEPZESCSOPORT set kcs_típus='BÖLCS', kcs_nev='bölcselet' where kcs_id=10;  
update dbo.KEPZESCSOPORT set kcs_típus='MŰV' where kcs_id=9;  
update dbo.KEPZESCSOPORT set kcs_típus='AGRÁR' where kcs_id=6;  
  
insert into dbo.KEPZESCSOPORT values (13,'ORV',12,'orvos',0);  
insert into dbo.KEPZESCSOPORT values (14,'JOGI',14,'jogi',0);  
  
update dbo.SZAK set szk_nev='állatorvos', szk_kcs_id=2 where szk_szer_id=1;  
update dbo.SZAK set szk_nev='biológia' where szk_id in(1,76);  
update dbo.SZAK set szk_nev='történelem' where szk_szer_id=10;  
update dbo.SZAK set szk_nev='pszichológia' where szk_id in(85,10);  
update dbo.SZAK set szk_nev='ének-zene' where szk_szer_id=11;  
update dbo.SZAK set szk_nev='fuvola' where szk_id in(61,36);  
update dbo.SZAK set szk_nev='anyagmérnök' where szk_id=87;  
update dbo.SZAK set szk_nev='politológia' where szk_szer_id=13;  
update dbo.SZAK set szk_nev='anglisztika', szk_kcs_id=8 where szk_id in(63,38);  
update dbo.SZAK set szk_kcs_id=8 where szk_id in(63,38,13,88);  
update dbo.SZAK set szk_kcs_id=7 where szk_id=32;  
  
select *  
from SZERVEZET s  
join SZAK sz on sz.szk_szer_id=s.szer_id  
;  
select * from SZAK order by 2 asc;  
  
--16 OE  
update dbo.SZAK set szk_kcs_id=7 where szk_id in(16);  
update dbo.SZAK set szk_kcs_id=4, szk_nev='mérnök-informatikus' where szk_id in(41);  
update dbo.SZAK set szk_kcs_id=12, szk_nev='villamosmérnök' where szk_id in(66);  
update dbo.SZAK set szk_kcs_id=7, szk_nev='pénzügy' where szk_id in(91);  
  
--17 EE  
update dbo.SZAK set szk_kcs_id=2 where szk_id in(67);  
update dbo.SZAK set szk_kcs_id=7 where szk_id in(42);  
update dbo.SZAK set szk_kcs_id=12, szk_nev='vegyészmérnök' where szk_id in(17);  
update dbo.SZAK set szk_kcs_id=4 where szk_id in(92);  
  
--18 PPKE  
update dbo.SZAK set szk_kcs_id=10, szk_nev='történelem' where szk_id in(93);  
update dbo.SZAK set szk_kcs_id=8, szk_nev='tölmács' where szk_id in(18);  
update dbo.SZAK set szk_kcs_id=8, szk_nev='anglisztika' where szk_id in(43);  
update dbo.SZAK set szk_kcs_id=14, szk_nev='jogász' where szk_id in(68);
```

```

--19 PTE
update dbo.SZAK set szk_kcs_id=2 where szk_id in(69);
update dbo.SZAK set szk_kcs_id=14 where szk_id in(44);
update dbo.SZAK set szk_kcs_id=4 where szk_id in(19);
update dbo.SZAK set szk_kcs_id=9 where szk_id in(94);
insert into dbo.SZAK values(101,19,4,'gazdaságinformatikus','A','N','A',0);
insert into dbo.SZAK values(102,19,5,'vizuális tervezés','A','L','A',0);
insert into dbo.SZAK values(103,19,13,'orvos','O','N','A',0);

--2 BGE
update dbo.SZAK set szk_kcs_id=4, szk_nev='gazdaságinformatikus' where szk_id in(77);
update dbo.SZAK set szk_kcs_id=7, szk_nev='turizmus-vendéglátás' where szk_id in(52);
update dbo.SZAK set szk_kcs_id=7, szk_nev='számvitel' where szk_id in(27);
update dbo.SZAK set szk_kcs_id=7, szk_nev='kereskedelem és marketing' where szk_id in(2);

--20 SE
update dbo.SZAK set szk_kcs_id=13, szk_nev='orvos' where szk_id in(20);
update dbo.SZAK set szk_kcs_id=13, szk_nev='fogorvos' where szk_id in(45);
update dbo.SZAK set szk_kcs_id=13, szk_nev='ápolás és betegellátás' where szk_id in(70);
update dbo.SZAK set szk_kcs_id=13, szk_nev='egészségügyi gondozó' where szk_id in(95);

--21 SOE
update dbo.SZAK set szk_kcs_id=3, szk_nev='tanító' where szk_id in(96);
update dbo.SZAK set szk_kcs_id=3, szk_nev='andragógia' where szk_id in(71);
update dbo.SZAK set szk_kcs_id=7, szk_nev='számvitel' where szk_id in(46);
update dbo.SZAK set szk_kcs_id=7, szk_nev='kereskedelem és marketing' where szk_id in(21);

--22 SZE
update dbo.SZAK set szk_kcs_id=4, szk_nev='mérnök-informatikus' where szk_id in(22);
update dbo.SZAK set szk_kcs_id=12, szk_nev='gépészmérnök' where szk_id in(47);
update dbo.SZAK set szk_kcs_id=7, szk_nev='számvitel' where szk_id in(72);
update dbo.SZAK set szk_kcs_id=7, szk_nev='kereskedelem és marketing' where szk_id in(97);

--23 SZIE
update dbo.SZAK set szk_kcs_id=12, szk_nev='gépészmérnök' where szk_id in(98);
update dbo.SZAK set szk_kcs_id=7, szk_nev='számvitel' where szk_id in(73);
update dbo.SZAK set szk_kcs_id=12, szk_nev='élelmiszer-mérnök' where szk_id in(48);
update dbo.SZAK set szk_kcs_id=7, szk_nev='kereskedelem és marketing' where szk_id in(23);
insert into dbo.SZAK values(104,23,12,'épitőmérnök','A','N','A',0);

```

```

--24 SZTE
update dbo.SZAK set szk_kcs_id=12, szk_nev='gépészmérnök' where szk_id in(24);
update dbo.SZAK set szk_kcs_id=7, szk_nev='számvitel' where szk_id in(49);
update dbo.SZAK set szk_kcs_id=4, szk_nev='programtervező informatikus' where szk_id in(74);
update dbo.SZAK set szk_kcs_id=10, szk_nev='történelem' where szk_id in(99);
insert into dbo.SZAK values(105,24,12,'járműmérnök','A','N','A',0);
insert into dbo.SZAK values(106,24,3,'ének-zene','A','L','A',0);
insert into dbo.SZAK values(107,24,12,'gépészmérnök','A','N','A',0);
insert into dbo.SZAK values(108,24,14,'jogász','O','N','A',0);

--25 TE
update dbo.SZAK set szk_kcs_id=3, szk_nev='testnevelő tanár' where szk_id in(100);
update dbo.SZAK set szk_kcs_id=1, szk_nev='edző' where szk_id in(75);
update dbo.SZAK set szk_kcs_id=1, szk_nev='sport-és rekreáció szervezés' where szk_id in(50);
update dbo.SZAK set szk_kcs_id=3, szk_nev='gyógytestnevelő' where szk_id in(25);

--3 BME
update dbo.SZAK set szk_kcs_id=12, szk_nev='gépészmérnök' where szk_id in(3);
update dbo.SZAK set szk_kcs_id=7, szk_nev='számvitel' where szk_id in(28);
update dbo.SZAK set szk_kcs_id=12, szk_nev='épitőmérnök' where szk_id in(53);
update dbo.SZAK set szk_kcs_id=10, szk_nev='kommunikáció és médiatudomány' where szk_id in(78);
insert into dbo.SZAK values(105,3,12,'járműmérnök','A','N','A',0);
insert into dbo.SZAK values(109,3,12,'villamosmérnök','A','L','A',0);
insert into dbo.SZAK values(110,3,12,'logisztikai mérnök','A','N','A',0);
insert into dbo.SZAK values(111,3,2,'biológia','A','N','A',0);
insert into dbo.SZAK values(112,3,2,'matematika','A','N','A',0);

--4 BCE
update dbo.SZAK set szk_kcs_id=9, szk_nev='szociológia' where szk_id in(76);
update dbo.SZAK set szk_kcs_id=7, szk_nev='számvitel' where szk_id in(54);
update dbo.SZAK set szk_kcs_id=9, szk_nev='nemzetközi tanulmányok' where szk_id in(29);
update dbo.SZAK set szk_kcs_id=10, szk_nev='kommunikáció és médiatudomány' where szk_id in(4);
insert into dbo.SZAK values(113,4,4,'gazdaságinformatikus','A','N','A',0);
insert into dbo.SZAK values(114,4,7,'turizmus-vendéglátás','A','L','A',0);

--5 DE
update dbo.SZAK set szk_kcs_id=9, szk_nev='szociológia' where szk_id in(5);
update dbo.SZAK set szk_kcs_id=7, szk_nev='számvitel' where szk_id in(30);
update dbo.SZAK set szk_kcs_id=9, szk_nev='nemzetközi tanulmányok' where szk_id in(55);
update dbo.SZAK set szk_kcs_id=10, szk_nev='kommunikáció és médiatudomány' where szk_id in(80);
insert into dbo.SZAK values(115,5,4,'programtervező informatikus','A','N','A',0);
insert into dbo.SZAK values(116,5,7,'turizmus-vendéglátás','A','L','A',0);
insert into dbo.SZAK values(117,5,2,'biológia','A','N','A',0);

```

```

--6 EKE
update dbo.SZAK set szk_kcs_id=0, szk_nev='tanító' where szk_id in(6);
update dbo.SZAK set szk_kcs_id=7, szk_nev='számíté' where szk_id in(31);
update dbo.SZAK set szk_kcs_id=0, szk_nev='nemzetközi tanulmányok' where szk_id in(56);
update dbo.SZAK set szk_kcs_id=10, szk_nev='kommunikáció és médiatudomány' where szk_id in(81);

--7 ELTE
update dbo.SZAK set szk_kcs_id=0, szk_nev='szociológia' where szk_id in(5);
update dbo.SZAK set szk_kcs_id=7, szk_nev='számíté' where szk_id in(30);
update dbo.SZAK set szk_kcs_id=0, szk_nev='nemzetközi tanulmányok' where szk_id in(55);
update dbo.SZAK set szk_kcs_id=10, szk_nev='kommunikáció és médiatudomány' where szk_id in(80);
update dbo.SZAK set szk_kcs_id=0, szk_szer_id=7, szk_nev='fizika' where szk_id in(93);
insert into dbo.SZAK values(118,7,4,'programtervezé informatikus','A','N','A',0);
insert into dbo.SZAK values(119,7,7,'turizmus-vendéglátás','A','L','A',0);
insert into dbo.SZAK values(120,7,2,'biológia','A','L','A',0);
insert into dbo.SZAK values(121,7,3,'tanító','A','N','A',0);
insert into dbo.SZAK values(122,7,10,'fordító tolmács','A','N','A',0);
insert into dbo.SZAK values(123,7,2,'matematika','A','N','A',0);
insert into dbo.SZAK values(124,7,2,'földrajz','A','N','A',0);

--8 IBS
update dbo.SZAK set szk_kcs_id=7, szk_nev='gazdálkodási és mmenedzsment' where szk_id in(58);
update dbo.SZAK set szk_kcs_id=7, szk_nev='nemzetközi gazdálkodás' where szk_id in(83);
update dbo.SZAK set szk_kcs_id=0, szk_nev='nemzetközi tanulmányok' where szk_id in(8);

--9 KE
update dbo.SZAK set szk_kcs_id=7, szk_nev='gazdálkodási és mmenedzsment' where szk_id in(9);
update dbo.SZAK set szk_kcs_id=7, szk_nev='nemzetközi gazdálkodás' where szk_id in(34);
update dbo.SZAK set szk_kcs_id=0, szk_nev='tanító' where szk_id in(55);
update dbo.SZAK set szk_kcs_id=0, szk_nev='gyógypedagógia' where szk_id in(84);

update dbo.SZAK set szk_kepesforma='0' where szk_id in(14,39,44,68);

--500 alapfokú végzettség
update dbo.ALAPFOKU_VEGZETTSAG set afv_pedavegz='N' where afv_fszak like 'közgazdászt';
update dbo.ALAPFOKU_VEGZETTSAG set afv_pedavegz='I' where afv_fszak like 'óvodák';
update dbo.ALAPFOKU_VEGZETTSAG set afv_pedavegz='I' where afv_fszak like 'tanító4';
update dbo.ALAPFOKU_VEGZETTSAG set afv_pedavegz='N' where afv_fszak like 'biológ';
update dbo.ALAPFOKU_VEGZETTSAG set afv_pedavegz='I' where afv_fszak like 'mérnökinf4';
update dbo.ALAPFOKU_VEGZETTSAG set afv_pedavegz='N' where afv_fszak like 'angliszt4';

update dbo.JELENTKEZO set j_erettsegi_ave=2018 where j_jaras_id in(2,3);
update dbo.JELENTKEZO set j_erettsegi_ave=2017 where j_jaras_id=7;
update dbo.JELENTKEZO set j_erettsegi_ave=2016 where j_felveteli_id between 15 and 55;
update dbo.JELENTKEZO set j_erettsegi_ave=2015 where j_felveteli_id in(101,102,111,234,555,657,4453,765,678);
update dbo.JELENTKEZO set j_erettsegi_ave=2014 where j_felveteli_id in(7658,9997,3451);

```

```

--do.PONTISZAM 90k
update dbo.PONTISZAM set pont_osszp=pont_erettsegi + pont_yelvelisege;
--update dbo.PONTISZAM set pont_erettsegi= cast (pont_erettsegi as int) , pont_osszp= cast(pont_osszp as int);
update dbo.PONTISZAM set pont_alkalmas='I' where pont_id in
3(
SELECT pont_id FROM dbo.PONTISZAM WHERE pont_alkalmas = 'N' and pont_osszp between 50 and 410
);

--PONTISZAM
update dbo.PONTISZAM set ph_indok='I' where ph_szk_id in(6,7,8,11,12,14,15,16,17,18,19,20,11,13,15,16,17,31,34,37,39,41,43,46,47,49,50,51,53,57,61,63,64,67,
,83,85,87);

--szabla felvett szakk mindkétre máshoz is kellene!
insert into dbo.PONTISZAM values(101,'I',50,10,405,0,450,0);
insert into dbo.PONTISZAM values(102,'I',15,5,305,0,350,0);
insert into dbo.PONTISZAM values(103,'I',100,5,450,0,450,0);
insert into dbo.PONTISZAM values(104,'I',15,5,405,0,450,0);
insert into dbo.PONTISZAM values(105,'I',150,5,435,0,470,0);
insert into dbo.PONTISZAM values(106,'I',15,5,305,0,350,0);
insert into dbo.PONTISZAM values(107,'I',150,5,445,0,450,0);
insert into dbo.PONTISZAM values(108,'I',100,5,0,450,0);
insert into dbo.PONTISZAM values(109,'I',150,5,435,0,450,0);
insert into dbo.PONTISZAM values(110,'I',150,5,435,0,450,0);
insert into dbo.PONTISZAM values(111,'I',50,5,305,0,350,0);
insert into dbo.PONTISZAM values(112,'I',10,5,350,0,350,0);
insert into dbo.PONTISZAM values(113,'I',150,100,435,0,450,0);
insert into dbo.PONTISZAM values(114,'I',150,115,325,0,350,0);
insert into dbo.PONTISZAM values(115,'I',15,5,435,0,450,0);
insert into dbo.PONTISZAM values(116,'I',150,100,325,0,350,0);
insert into dbo.PONTISZAM values(117,'I',150,5,435,0,375,0);
insert into dbo.PONTISZAM values(118,'I',150,5,435,0,450,0);
insert into dbo.PONTISZAM values(119,'I',50,5,305,0,350,0);
insert into dbo.PONTISZAM values(120,'I',150,5,335,0,350,0);
insert into dbo.PONTISZAM values(121,'I',150,5,315,0,350,0);
insert into dbo.PONTISZAM values(122,'I',15,5,305,0,350,0);
insert into dbo.PONTISZAM values(123,'I',150,5,305,0,350,0);
insert into dbo.PONTISZAM values(124,'I',150,5,315,0,350,0);
insert into dbo.PONTISZAM values(125,'I',150,5,305,0,350,0);
insert into dbo.PONTISZAM values(126,'I',150,5,435,0,450,0);

update dbo.PONTISZAM set ph_fizpont=105, ph_min_pont=0, ph_hatar=0 where ph_szk_id in(14,39,44,68,103); -- jogszabály fizpont -> állam végzett a pontot ide
update dbo.PONTISZAM set ph_min_hatar=50, ph_max_hatar=100 where ph_szk_id in
3(
6,33,36,41,50,57,62,64,65,66,68,70,75,83,94,99
);

```

```

update dbo.PONTHATAR set ph_hatar=85, ph_min_pont=75 where ph_szk_id=1;
update dbo.PONTHATAR set ph_hatar=95, ph_min_pont=85 where ph_szk_id=2;
update dbo.PONTHATAR set ph_hatar=65, ph_min_pont=50 where ph_szk_id=5;
update dbo.PONTHATAR set ph_hatar=80, ph_min_pont=65 where ph_szk_id=6;
update dbo.PONTHATAR set ph_hatar=85, ph_min_pont=75 where ph_szk_id=8;
update dbo.PONTHATAR set ph_hatar=95, ph_min_pont=75 where ph_szk_id=9;
update dbo.PONTHATAR set ph_hatar=87, ph_min_pont=75 where ph_szk_id=10;
update dbo.PONTHATAR set ph_hatar=98, ph_min_pont=85 where ph_szk_id=12;
update dbo.PONTHATAR set ph_hatar=88, ph_min_pont=75 where ph_szk_id=18;
update dbo.PONTHATAR set ph_hatar=87, ph_min_pont=78 where ph_szk_id=21;
update dbo.PONTHATAR set ph_hatar=90, ph_min_pont=85 where ph_szk_id=23;
update dbo.PONTHATAR set ph_hatar=85, ph_min_pont=70 where ph_szk_id=25;
update dbo.PONTHATAR set ph_hatar=85, ph_min_pont=75 where ph_szk_id=28;
update dbo.PONTHATAR set ph_hatar=95, ph_min_pont=85 where ph_szk_id=29;
update dbo.PONTHATAR set ph_hatar=415, ph_min_pont=400 where ph_szk_id=32;
update dbo.PONTHATAR set ph_hatar=355, ph_min_pont=325 where ph_szk_id=37;
update dbo.PONTHATAR set ph_hatar=355, ph_min_pont=325 where ph_szk_id=40;
update dbo.PONTHATAR set ph_hatar=460, ph_min_pont=425 where ph_szk_id=41;
update dbo.PONTHATAR set ph_hatar=465, ph_min_pont=440 where ph_szk_id=43;
update dbo.PONTHATAR set ph_hatar=450, ph_min_pont=400 where ph_szk_id=50;
update dbo.PONTHATAR set ph_hatar=95, ph_min_pont=75 where ph_szk_id=51;
update dbo.PONTHATAR set ph_hatar=95, ph_min_pont=75 where ph_szk_id=52;
update dbo.PONTHATAR set ph_hatar=95, ph_min_pont=85 where ph_szk_id=53;
update dbo.PONTHATAR set ph_hatar=465, ph_min_pont=420 where ph_szk_id=55;
update dbo.PONTHATAR set ph_hatar=95, ph_min_pont=85 where ph_szk_id=57;
update dbo.PONTHATAR set ph_hatar=475, ph_min_pont=425 where ph_szk_id=58;
update dbo.PONTHATAR set ph_hatar=75, ph_min_pont=70 where ph_szk_id=62;
update dbo.PONTHATAR set ph_hatar=85, ph_min_pont=75 where ph_szk_id=63;
update dbo.PONTHATAR set ph_hatar=75, ph_min_pont=65 where ph_szk_id=65;
update dbo.PONTHATAR set ph_hatar=95, ph_min_pont=85 where ph_szk_id=67;
update dbo.PONTHATAR set ph_hatar=95, ph_min_pont=85 where ph_szk_id=69;
update dbo.PONTHATAR set ph_hatar=395, ph_min_pont=350 where ph_szk_id=71;
update dbo.PONTHATAR set ph_hatar=95, ph_min_pont=85 where ph_szk_id=76;
update dbo.PONTHATAR set ph_hatar=85, ph_min_pont=75 where ph_szk_id=78;
update dbo.PONTHATAR set ph_hatar=454, ph_min_pont=400 where ph_szk_id=80;
update dbo.PONTHATAR set ph_hatar=87, ph_min_pont=75 where ph_szk_id=81;
update dbo.PONTHATAR set ph_hatar=98, ph_min_pont=75 where ph_szk_id=84;
update dbo.PONTHATAR set ph_hatar=97, ph_min_pont=75 where ph_szk_id=85;
update dbo.PONTHATAR set ph_hatar=88, ph_min_pont=75 where ph_szk_id=87;
update dbo.PONTHATAR set ph_hatar=96, ph_min_pont=80 where ph_szk_id=88;
update dbo.PONTHATAR set ph_hatar=397, ph_min_pont=380 where ph_szk_id=89;
update dbo.PONTHATAR set ph_hatar=100, ph_min_pont=80 where ph_szk_id=91;
update dbo.PONTHATAR set ph_hatar=85, ph_min_pont=75 where ph_szk_id=95;
update dbo.PONTHATAR set ph_hatar=95, ph_min_pont=75 where ph_szk_id=100;

```

```

update dbo.JELENTKEZES set jel_kizar='N' where jel_id not in
(
    select jel_id from dbo.JELENTKEZES where jel_kizar='I' and jel_sorszam = 5
);

update dbo.JELENTKEZES set jel_fizetve='I';
update dbo.JELENTKEZES set jel_fizetve='N' where jel_id in(2,3,12,184,188,395,398,655,656,689,6
update dbo.JELENTKEZES set jel_szak_id=105 where jel_id in
(
    105,205,305,405,505,605,705,805,905,1005,1105,1205,1305,1405,1505,1605,1705,1805,1905,2005,2105
    3905,4005,4105,4205,4305,4405,4505,4605,4705,4805,4905,5005,5105,5205,5305,5405,5505,5605,5705,
    7505,7605,7705,7805,7905,8005,8105,8205,8305,8405,8505,8605,8705,8805,8905,9005,9105,9205,9305,
    10905,11005,11105,11205,11305,11405,11505,11605,11705,11805,11905,12005,12105,12205,12305,12405
    13905,14005,14105,14205,14305,14405,14505,14605,14705,14805,14905,15005,15105,15205,15305,15405
    16905,17005,17105,17205,17305,17405,17505,17605,17705,17805,17905,18005,18105,18205,18305,18405
    19905,20005,20105,20205,20305,20405,20505,20605,20705,20805,20905,21005,21105,21205,21305,21405
    22905,23005,23105,23205,23305,23405,23505,23605,23705,23805,23905,24005,24105,24205,24305,24405
    25905,26005,26105,26205,26305,26405,26505,26605,26705,26805,26905,27005,27105,27205,27305,27405
    28905,29005,29105,29205,29305,29405,29505,29605
);
update dbo.JELENTKEZES set jel_szak_id=104, jel_szer_id=3 where jel_id in
(
    6710,6810,6910,7010,7110,7210,7310,7410,7510,7610,7710,7810,7910,8010,8110,8210,8310,8410,8510,
    10810,10910,11010,1110,11210,11310,11410,11510,11610,11710,11810,11910,12010,12110,12210,12310,12410,
    13310,13410,13510,13610,13710,13810,13910,14010,14110,14210,14310,14410,14510,14610,14710,14810
    16310,16410,16510,16610,16710,16810,16910,17010,17110,17210,17310,17410,17510,17610,17710,17810
    19310,19410,19510,19610,19710,19810,19910,20010,20110,20210,20310,20410,20510,20610,20710,20810
    22310,22410,22510,22610,22710,22810,22910,23010,23110,23210,23310,23410,23510,23610,23710,23810
    25310,25410,25510
);
update dbo.JELENTKEZES set jel_szak_id=106 where jel_id in
(
    810,410,510,610,710,810,910,1010,1110,1210,1310,1410,1510,1610,1710,1810,1910,2010,2110,2210,23
    4110,4210,4310,4410,4510,4610,4710,4810,4910,5010,5110,5210,5310,5410,5510,5610,5710,5810,5910
);
update dbo.JELENTKEZES set jel_szak_id=107, jel_szer_id=4 where jel_id in
(
    9213,9313,9413,9513,9613,9713,9813,9913,10013,10113,10213,10313,10413,10513,10613,10713,10813,1
    12313,12413,12513,12613,12713,12813,12913,13013,13113,13213,13313,13413,13513,13613,13713,13813
    15313,15413,15513,15613,15713,15813,15913,16013,16113,16213,16313,16413,16513,16613,16713,16813
    18313,18413,18513,18613,18713,18813,18913,19013,19113,19213,19313,19413,19513,19613,19713,19813
    21313,21413,21513,21613,21713,21813,21913,22013,22113,22213,22313,22413,22513,22613,22713,22813
    24313,24413,24513,24613,24713,24813,24913,25013,25113,25213,25313,25413,25513,25613,25713,25813
    27313,27413,27513,27613,27713
);

```