

# Közgazdász hallgatók informatikai kompetenciájának fejlesztése

**Takács Viktor László**

*tanársegéd*

Debreceni Egyetem, Gazdaságtudományi Kar, Alkalmazott Informatika és Logisztika  
Intézet, Üzleti Informatika Tanszék  
E-mail: takacs.viktor@econ.unideb.hu

DOI: [10.29180/978-615-6342-90-4\\_28](https://doi.org/10.29180/978-615-6342-90-4_28)

**Összefoglalás:** Az általam vezetett pénzügyi információfeldolgozás és adatvizualizáció kurzusokon a tananyag oktatása a klasszikus információ rendszerekre jellemző Forrásrendszer – Adatátalakítás – Vizualizáció hármas tagozódás szerint történik. Ennek során változó részletességgel kerülnek tárgyalásra az egyes részterületek. A tanulmány célja, hogy bemutassam a szükséges kompetenciák fejlesztésének egy lehetséges felépítését.

**Kulcsszavak:** Informatikai kompetenciák fejlesztése, Közgazdászképzés, Rendszerszemlélet, Analitikus gondolkodásmód.

**Abstract:** In the courses I teach, the curriculum is taught according to the Source System - Data Transformation - Visualization triple structure, typical of classical information systems. During the course, the individual subfields are discussed in varying detail. The aim of this study is to present a possible structure of the education of the necessary competences.

**Keywords:** Development of IT competences, Economist education, Systems approach, Analytical thinking.

## 1. Bevezetés

A Debreceni Egyetem gazdasági adminisztrációjában eltöltött 12 év tapasztalataival (gazdálkodási adminisztráció, később SAP bevezetés és kulcsfelhasználói tapasztalat, majd SAP BW, illetve HANA) felvértezve 2013-tól a Gazdaságtudományi Karon segítem hallgatóinkat a munkaerőpiacon szükséges informatikai kompetenciák elsajátításában.

Az eltelt évek során sikerült a képzés számára egy letisztult elméleti és gyakorlati tudásanyagot összeállítanom.

Jelenleg négy, szorosan a témához kapcsolódó kurzus oktatásában veszek részt: Szakmai és pénzügyi információfeldolgozási alapismeretek (FOSZ), Gazdasági információrendszerek (FOSZ), Üzleti informatika (BSc) és Haladó Adatvizualizáció (MSc). A kurzusokon a tananyag oktatása a klasszikus

információ rendszerekre jellemző Forrásrendszer – Adatátalakítás - Vizualizáció hármas tagozódás szerint történik, melynek során változó részletességgel kerülnek tárgyalásra az egyes részterületek.

A képzés során változatos technikákat alkalmazunk a közös feladatmegoldások mellett. A bevezető kurzusokon az önálló feladatmegoldási gyakorlatok vannak a tananyagba építve, valamint megosztunk rövid technikai videókat, hogy többféle megközelítéssel segítsük a hallgatókat. Projekt-jelleggel több gyakorlaton átívelő, komplexebb rendszereket is készítünk.

Minden kurzus esetében a cél a hallgatók három kulcskompetenciájának fejlesztése az adott kurzus képzési kimeneti követelményének megfelelően, segítve az adatelemzés felé orientálódást. Ez a három készség a problémamegoldó készség, az analitikus gondolkodásmód és a rendszerszemlélet. Ezek esszenciálisan mind a számítógépes gondolkodás részei [9]. A számítógépes gondolkodás kifejezést először Seymour Papert használta [5], de az elmúlt tíz évben ez a fogalomkör az informatikai módszertan fontos kutatási területévé vált [4]. Számos definíciót hoztak létre, átértelmezték, magyarázták, de mindenki egyetértett abban, hogy a számítógépes gondolkodásnak nemcsak az informatikai szakemberek, hanem mindenki kulcskompetenciájának kell lennie.

Az elmúlt év a generatív mesterséges intelligencia előretörését hozta. Számos publikáció foglalkozik az előnyeivel és a veszélyeivel. Jelen publikáció szempontjából legérdekesebb újdonság a promptolás, prompt engineering, mint új tevékenység megjelenése az MI területén. A prompt engineering a bemeneti lekérdezések vagy "promptok" tervezésének és finomításának folyamata, amely a kívánt válaszokat váltja ki a nagy nyelvi modellekből (LLM). A promptok döntő fontosságúak abban, hogy az LLM-ek hasznos és releváns kimeneteket hozzanak létre [1]. A promptolás minőségét megalapozza kiemelten a problémamegoldó készség, analitikus gondolkodásmód és a rendszerszemlélet.

A problémamegoldó készség nem technológiafüggő, a problémamegoldást elsősorban a közoktatás matematika óráin fejlesztik. Tipikus eszköze a matematikai szöveges feladatok.

Felismertük a hasonlóságokat a szöveges feladatok megoldása és a vezetői kérdések megválaszolása között [2], és Pólya módszerét szakterület-specifikus szöveges feladatokon keresztül adaptáltuk OLAP (On-Line Analytical Processing) technológiára.

Pólya problémamegoldó módszere [6] alapján az adatelemzőnek néhány lépésen kell keresztülmennie: *A probléma megértése, terv kidolgozása, a terv végrehajtása, általánosítás.*

Kurzusaink során az elejétől fogva különböző komplexitású feladatokon keresztül mutatjuk be a hallgatóknak a problémák részproblémákra bontását, a

részproblémák megoldásán keresztül az összetett megoldások szintetizálását. Mindeközben szem előtt tarjuk a Halassy-féle adatmodellezés [7] alapelveit, vagyis hogy a megoldások során úgy gondolkodjanak rendszerben a hallgatók a problémáról (vezetői kérdésekről), hogy azok, mint koncepcionális terv összhangban legyen a tervezés logikai és fizikai szintjével, illetve a fizikai megoldással a kialakításra kerülő rendszer folyamatmodelljeként.

Amikor először kezdjük el a munkát a hallgatókkal, világossá tesszük, hogy általános célját tekintve kétféle információs rendszer létezik. Egyik a tranzakcióorientált (OLTP) rendszer, a másik pedig az analitika-orientált (OLAP) rendszer. A különbség a rendszer célja és felépítése.

A Microsoft Excel OLAP elemeket is tartalmaz, ami nagymértékben demokratizálta az adatelemzést. Hogy pontosan mely elemek és hogyan, azt az 1. táblázatban foglaltam össze.

1. Táblázat  
Az Excel OLAP technológiához kapcsolódó objektumai

OLAP rendszer	Objektum/eszköz	
Adatmodell	Power Pivot Táblázat Nevesített tartomány	Power Pivot eszközben klasszikus csillag, hópehely és galaxis sémák Táblázatban kiterített kockaként
Adatkinyerés	Power Query Power Pivot	Az adatok kinyeréséhez lekérdezéseket készítünk, a Power Query esetében sokkal többféle forráshoz kapcsolódhatunk
Átalakítás	Power Query Power Pivot Tömbfüggvények	A PowerQuery M adatmanipulációs nyelv rendelkezik a legszéleskörűbb funkcionalitással, beleértve a no-code technikát is. A Power Pivot esetében a DAX nyelv egy szűkebb halmazával oszlop műveletekkel operálunk. Táblázat adatforrásra való hivatkozással használt tömbfüggvények esetében bármelyik excel függvényt használhatjuk új megközelítésben.
Vizualizáció	Kimutatás (Pivot tábla) Kimutatás diagram Szeletelő Tömbfüggvények	A vizualizáció során a leglátványosabb eszköz a kimutatáson alapuló diagram, de a tömbfüggvényekkel előállítható válaszok szabadsága is említésre érdemes.

*Forrás: Saját szerkesztés*

A hallgatóknak meg kell tanulniuk az alapvető adatátalakítási és adatgazdagítási módszereket.

Az Excel fenti BI eszköztárát integrálta és azóta is havi rendszerességgel fejleszti a Microsoft Power BI Desktop néven, amit 2015-ben ingyenesen hozzáférhetővé tett. A Power BI-nak létezik egy felhő alapú szolgáltatás verziója is, ahol közzé tehető a vezetői dashboardok, (ugyan ingyenes

verzióban csak a teljes nyilvánosság számára nyilvános linken keresztül), ám létezik fizetős Professional és Enterprise verziók SaaS rendszerben, ahol már megadott felhasználói körrel is meg lehet osztani az elemzéseket, valamint elérhető az ETL folyamat automatizációja is, amely EduID fiókkal folyamatosan megújuló trial verzióban használható a felsőoktatásban.

Bevezető informatika kurzusokon (Szakmai és pénzügyi információfeldolgozási alapismeretek (FOSZ), illetve Üzleti informatika (BSc)) jellemzően Excel és adatbázis ismereteket oktatunk. A 2023-24. tanév őszi félévében először az Ms Access szoftvert és Power BI-t oktattuk párhuzamosan. Gazdasági információrendszerek (FOSZ), illetve Haladó Adatvizualizáció (MSc) kurzusok esetében pedig komplex OLAP rendszerek tervezésével és megvalósításával foglalkozunk.

## 2. Bevezető informatikai problémamegoldás

Bevezető kurzusokon a cél az alapok elsajátítása, illetve a hallgatók gondolkodásának, szemléletmódjának formálása, hogy képesek legyenek magasabb szintű struktúrákban gondolkodni és azokkal műveleteket végezni. A cél érdekében már az Excel oktatás során az adatbázis kezelőkre jellemző objektumokkal dolgozunk: táblázatokkal azok oszlopaival, valamint nevesített tartományokkal, jellemzően egy cellás változók hivatkozásával.

Az adatbázis-tervezési módszertanhoz [7] hasonlóan általánosított lépéseket (eszközfüggetlen logikai tervet) készítünk a vezetői kérdések megoldására. A következő három lépés elengedhetetlen az adattárházak és vezetői információs rendszerek tervezésében:

- koncepcionális tervezés,
- logikai tervezés,
- fizikai tervezés.

A fenti módszer alapjait Excel függvények használatával bemutattuk a [2] publikációban.

Automatizálási, illetve automatizált OLAP rendszer szempontból az egyik legfontosabb érdekesség, hogy táblázat objektum használata esetén élesen szétválasztható a funkcionalitás (folyamatmodell) és az adatmodell. Lássunk egy példát, ahol a feladat az „összes rendelési érték” meghatározása. Az adatokat egy Rendelések nevű tábla Érték nevű oszlopában tárolva a kérdés megválaszolható a  $=SZUM(Rendelések[Érték])$  képlettel, ami az adatok számosságának változása esetén is helyes marad. Ugyanezt a feladatot oldjuk meg a  $=SZUM(Rendelések!E2:E300)$  képlettel, ahol a Rendelések az Excel egy munkalapja, viszont ez a megoldás csak egy adott pillanatban jelentheti a fizikai modelljét az „összes rendelési érték” koncepciónak.

Az oktatás során észrevettem, hogy a hallgatók szeretik a képleteket abban a cellában látni, amit szerkesztenek. A vezetői információs rendszerek esetében

használt Excel munkafüzetekben külön munkalapon tároljuk az adatokat (mint forrás) és a vezetői kérdésekre adott válaszokat. A képzés és a számonkérés során is ezt az elvet erősítjük. Cellahivatkozások használata esetén a hallgatók kénytelenek ugrálni a munkalapok között és nagyon nehezen rögzül, hogy hiába jelölik ki a képletben jól az adatforrás munkalapjáról a szükséges cellatartományt, ha a követő záró zárójel vagy pontosvessző megadása helyett visszaugranak a válasz munkalapra és direktben utasítják az Excelt a tartományhivatkozás munkalap összetevőjének cseréjére. Ez táblázat objektum használata esetén automatikusan kiküszöbölésre kerül, hiszen a táblázat nevének gépelésekor felajánlja az Excel a függvények mellett a táblázatokat és nevesített tartományokat saját ikonos előtaggal. A [ jel megadása után megkapják a táblázat oszlopainak listáját, amiből választhatnak, csupán arra kell figyelniük, hogy a hivatkozást le kell zárniuk a ] jellel.

A táblázat további egyértelmű előnye az „aktuális sor” hivatkozása @ jellel, illetve a képletmásolás, mint művelet automatizálása, amely a táblázat új sora esetén is automatikusan alkalmazásra kerül, ahogy érdekes módon a cellaszintű feltételes formázás és a cellakorlátozás is. A megoldások helyességének ellenőrzésekor a táblázat használata szintén egy magasabb szintet jelent, ugyanis a „mindenhol jó, vagy valahol hibás” igaz állítás átalakul a „vagy mindenhol jó, vagy mindenhol hibás” állítássá, így pontosan egy helyen elég ellenőrizni a képletet. Az oszlopképletek alkalmazása esetén az ellenőrzéshez hasonlóan elég pontosan egy elemet javítani ahhoz, hogy a táblázat oszlopának összes elemére automatikusan vonatkozzon a módosítás. Ehhez kapcsolódó utolsó gondolatként tekintsünk rá a 2. táblázatban levő tervekre, ahol egy másik példát mutatok be a koncepcionális terv egy általános struktúrájára. A feladat ebben az esetben például egy lépésszámláló alkalmazásban előforduló „Mennyi a pulzusa az 51 éves betegeknek?” kérdés megválaszolása.

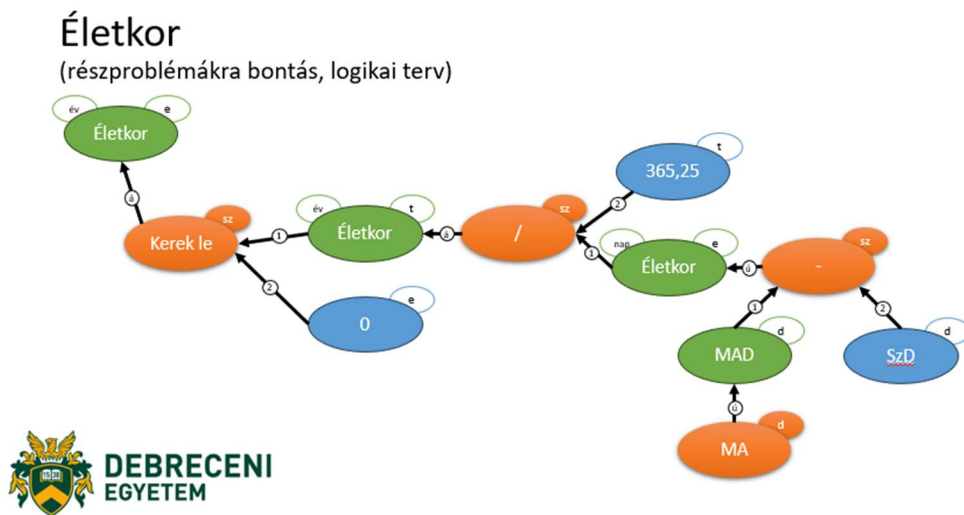
2. Táblázat  
Az 51 éves betegek átlagos pulzusa

Adatgyűjtés	af	AVERAGE
	A/M	Pulse
	C	Age = 51
Koncepcionális terv		$result\{=af(IF(C,M))\}$
Logikai terv		$result\{=SUM(IF(Age=51,Pulse))\}$
Fizikai terv		$K5\{=SUM(IF(B4:B105=51,I4:I102))\}$
Fizikai terv (táblázat adatforráson)		$K5\{=SUM(IF(dataset[Age]=51,dataset[Pulse]))\}$

Forrás: [2]

Az életben, munka során számos esetben fordul elő, hogy periodikusan kell előállítani repetitív jelentéseket, adatszolgáltatásokat, esetlegesen más emberek munkáját kell folytatni, értelmezni az Excelben levő képleteket, azaz a rendszer funkcionalitását. Megfigyelhető, hogy a táblázat objektum használata mellett a fizikai terv (valós Excel képlet) sokkal közelebb van a logikai tervhez, mint cellaképletek használata esetén. Ilyen módon kisebb kognitív terhelést jelent a „fordítása”, azaz értelmezni, hogy mit jelent a képlet. Könnyen belátható továbbá, hogy a munka során használt Excel esetében a munkafüzetünk nem csupán egy képletet tartalmaz, így már olyan egyszerű változások esetében is, amikor az adatok számossága változik ellenőrizni kell, hogy annak milyen továbbgyűrűző hatása van a rendszer funkcionalitására cellaképletek használata esetén. Ezért is kerülendő a cellaképlet használat és javasolt helyette az általam bemutatott objektumok használatának alapos megismerése.

A bevezető informatikai kurzusok esetében természetesen nem csak formalizálási technikákkal operálunk. Az egyszerűsített folyamatmodelleket is megtervezzük a hallgatókkal közösen, mint ami az 1. ábrán látható Életkor probléma egy logikai modelljét mutatja, kiegészítve a Power Query fizikai adat- és művelet típusaira és a mutatószám mértékegységére vonatkozó információkkal. A kék szín a rendelkezésre álló adatokat jelöli, a zöld a kiszámítandó részeredményt, a narancs a műveletet logikai szinten. A nyilakon a műveletek esetében a paraméter sorrendet jelöljük, a nyilak a klasszikus input-output irányokat mutatják.



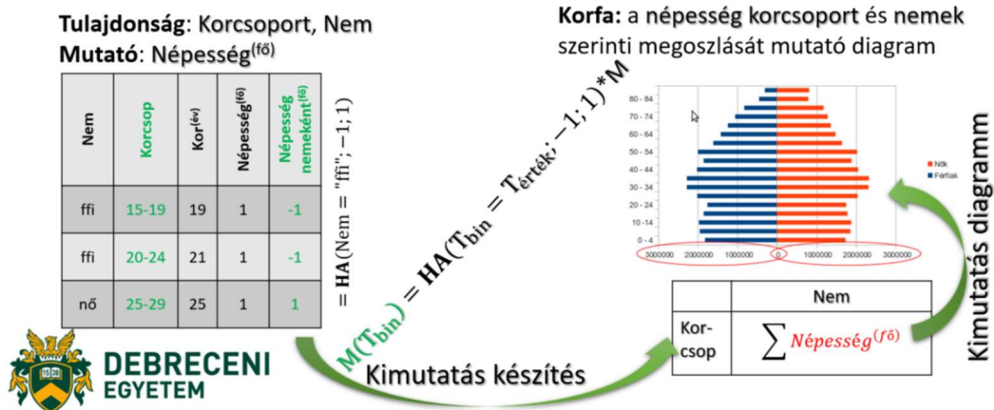
Forrás: saját szerkesztés

1. ábra

A kurzus zárásakor az információ rendszer elméletéhez kapcsolódó problémaként megkapják az Excelben nem triviális „korfa” problémát. A problémamegoldás során a „Mit mutat meg?” kérdéssel indítva a kapcsolódó fogalomtár elem, a kívánt vizualizáció, a vizualizációhoz minimálisan szükséges adatok, illetve a forrás kerül közös megtárgyalásra. Végül megvalósul a probléma általánosítása mind koncepcionális, logikai és fizikai modell szinten. Az eredmény koncepcionálisan (szövegesen) megfogalmazva: a korfa diagram alkalmas egy tetszőleges mutató egy bináris és egy kategorikus tulajdonság szerinti bemutatására. A tanulási folyamat és a megvalósítási folyamat vizuális bemutatása látható a 2. ábrán, ahol a nyilak a követelmény elemzés irányának ellentétesen mutatják a probléma megoldásának folyamatát.

## Rendszerszemlélet

(Egy érdekesebb probléma a „Korfa” diagram, diszkusszió)



Forrás: saját szerkesztés

2. ábra

Ahogy a cikk bevezetőjében említésre került, az adatbázis ismeretek oktatása során kísérleti jelleggel párhuzamosan mutattuk meg az Ms Access szoftvert és a Microsoft Power BI Desktop szoftvert. Tettük ezt bevezető jelleggel a későbbi OLAP megvalósítások technikai megalapozása érdekében. Ezeken a kurzusokon még hasonló kérdésekkel dolgoztunk, amelyekkel akár a középfokú oktatás során is találkozhattak. Egyrészt vizsgáltuk, hogy mekkora hasonlóságot mutatnak a Power BI M, illetve DAX nyelvek az Access SQL-jéhez, másrészt, hogy az adatmodell-tábla-lekérdezés-jelentés objektumok adatmodell-tábla-vizualizáció párhuzamok felfedezhetőek-e, megvalósíthatóak-e.

Az eredményeket a 3. táblázatban foglaltam össze.

3. Táblázat  
MS Access és Microsoft Power BI Desktop összehasonlítása

Feladat	Ms Access	Microsoft Power BI Desktop
Adatmodell	Összetett kulcs (több mezős)	Egyszerű kulcs (egy mezős)
Tábla nézet	Van számított mezőkkel	Van DAX képlettel kiegészíthető
Lekérdezés	SQL nyelven QBE ráccsal segítve	M nyelven no code megvalósítással segítve
	Választó	Tábla vagy Kártya vizualizáció
	Keresztábrás	Mátrix vizualizáció
	Frissítő	M nyelven Átalakítás típusú művelet
	Tábla készítő	M nyelven no code megvalósítással segítve
	Hozzáfűző	M nyelven no code megvalósítással segítve
	Unió	M nyelven no code megvalósítással segítve
Űrlap	Adatbevitelhez	Nincs, tekintve az OLAP rendszer sajátosságait

*Forrás: Saját szerkesztés*

A lekérdezések relációs műveletei tekintetében a QBE rácsot részben felváltják a Power BI vizualizációk, amiből 100-nál több egyedi van, az SQL-t pedig az M nyelv a betöltés előtt és a DAX nyelv a betöltés után az ETLT folyamatnak (Extract, Transform, Load, Transform) részeként.

- A projekció megfeleltethető a vizualizációk adatigényének.
- A szelekció a vizualizációk (vagy a jelentéslap, akár az egész jelentés) szűrőjének. Ez alapvetően egyszerűbb, nem ad akkor szabadságot, mint az SQL, vagy a QBE rács, de természetesen az alapvető filozófiai különbözőségek elfogadásával - gondolok specifikusan az OLAP kocka szeletelésére -, a reláció kiterjeszhető egy új logikai tulajdonsággal, amely tetszőlegesen összetett logikai kifejezést tartalmazhat.
- Kiterjesztésben a Power BI erősebbnek érződik, alapvetően a relációk (táblák) transzformációjára is ad lehetőséget, az egyszerű, DAX nyelven megvalósuló képlet alapú oszlopbővítés mellett az említett M nyelven is. Táblák összekapcsolására mind adatmodell szinten, mind M nyelvi transzformációkon keresztül lehetőségünk van, akár az OLAP kocka dimenzióit is ki tudjuk teríteni a központi ténytáblá(k)ba.
- Aggregáció alapértelmezettként áll rendelkezésre a vizualizációkban. Az OLAP szemlélet miatt elvégezhető akár a vizualizációk szintjén, akár a táblák szintjén M nyelven.
- Rendezés tekintetében a Power BI vizualizációk lehetőségei pontosan egy megjelenített mezőre korlátozódnak. Rendelkezik viszont egy érdekes



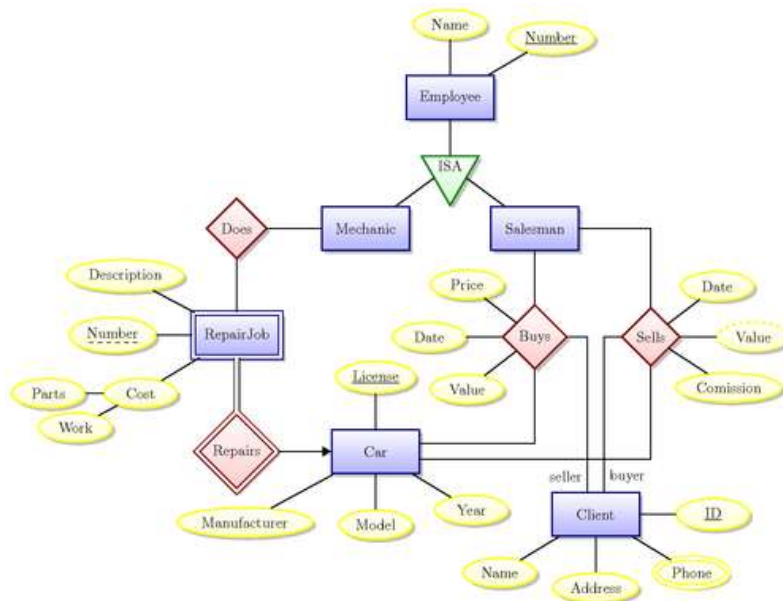
konceptióval, ugyanis a táblázat oszlopának rendezettsége definiálható egy másik oszlop alapján. Például a hónapok nevei szöveges oszlop esetében a hónap sorszáma oszlop alapján. Ez elrejthető a vizualizációs felületen. Általánoságban egy, akár szöveges oszlop rendezettsége definiálható egy vele 1:1 kapcsolatban levő szám típusú oszlop alapján. Ezt kihasználva úgynevezett ranking problémára visszavezethető a rendezés, melynek során az SQL-hez hasonlóan megadható a sorrendben több oszlopot érintő rendezés. Annyit kell még tennünk csupán, hogy ki kell egészíteni a táblát a rendezésnek megfelelő „rank” kulcsozloppal, amit természetesen támogat a beépített M nyelv.

Összegzésképpen megjegyezhető, hogy a kísérlet során a relációs adatbázis kezelés a közgazdász hallgatók számára bevezető informatikai kurzuson Access helyett a Microsoft Power BI szoftverrel is bemutatható, ha az OLTP, illetve OLAP szemléletmód különbözőségét elfogadjuk. Továbbá, amennyiben figyelembe vesszük, hogy hamarabb fognak Excelt és valamilyen adatvizualizációs eszköznek mondott szoftvert használni a munkájuk során, akkor célszerű is meglépnünk ezt a cserét végrehajtani a tananyagban (vagy legalább párhuzamosan használni a két eszközt).

A képzés 13 hetéből 2 ZH hét, 1 pótlási hét van, a maradék 7-3 arányban bomlik Excel, illetve adatbázis ismeretek oktatására. Excel esetén minden páros héten a gyakorlatok keretében projekt munka van. Ekkor a hallgatók önállóan, az addig tanultaknak megfelelően oldanak meg feladatokat. Természetesen amennyiben elakadnak, az oktató segít. Az adatbázis ismeretek esetében 1-1 gyakorlat jut oktatói vezetéssel Access és Power BI gyakorlásra, a 3. alkalom pedig szintén projekt jellegű önálló feladatmegoldás, amikor a hallgató választhat, hogy melyik szoftverrel dolgozik. A projekt munkák adják a hallgatók jegyének 20%-át 12%-8% megoszlásban, az Excel ZH a 45%-ot, míg az adatbázis a maradék 35%-ot.

### **3. Információs rendszerrel kapcsolatos problémamegoldás**

Az információs rendszerrel kapcsolatos kurzusom legfontosabb adatbázisa a 3. ábrán látható kibővített egyed-kapcsolati modellen alapul. Az eltelt évek kurzusai során ezen keresztül mutattam be az OLTP és OLAP rendszerek tervezésének, fejlesztésének és implementálásának lépéseit. Az ábrán látható rendszermodellhez tanulják meg a hallgatók az értékteremtő folyamatok folyamatmodelljeinek kidolgozását ARIS keretrendszerben.



Forrás: <https://everythingstudies.files.wordpress.com/2017/08/erdiagram.png>

3. ábra „Cartrader erm”

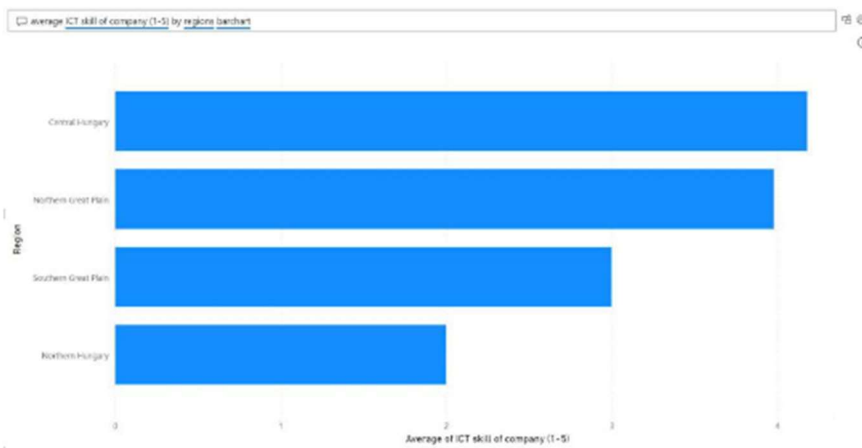
A modellezést az időközben többször átalakult [8] eszközzel végezzük, mely a nagyfokú szabadsága és ingyenessége mellett alkalmas a megvalósított rendszerhez kapcsolódó dokumentáció rendszerben való kezelésére is, támogatva a közös munkát.

Módszertanilag a bevezető kurzuson használt egyszerűbb transzformációs technikák mellett a [3] cikkben publikált hibrid adattárház tervezési módszertan szerinti minimális modellt is meghatározzuk. Továbbá kiemelt fókuszot kap az OLTP és OLAP rendszer adatmodellje közötti transzformáció. Nem elégszünk meg egy-egy folyamati mutató kiszámításával, hanem összetettebb modelltranszformációt igénylő csillag-, hópehely-, illetve galaxis sémát építünk. A fenti tranzakció orientált rendszerből remekül számítható egy profit mutatót tartalmazó ténytáblához a kapcsolódó autó dimenzió kor dimenziós mutatója. Kiegészítjük a szükséges fogalomtár elemekkel, hogy pontosan mit és *milyen dátumon* tekintünk profitnak az információs rendszerben. Például az *eladott* autók beszerzési ára csökkentve a felmerült költségekkel, vagy hasonlóan az autó kora az *eladáskori dátumon*, nem pedig ma. Számítható klasszikusan készlet forgási sebesség (eladott autókra), készlet mennyiségét és értékét tartalmazó központi ténytábla, közös és egyedi dimenzionalitással.

A képzés során figyelmet fordítunk a hallgatók kérdés megfogalmazási készségének fejlesztésére, melynek során az adatok szövegből való formalizálásakor a [3] cikkben publikált kérdésekre keressük a válaszokat:

- Mi a mutató?
- Melyik összesítésben?
- Mi a mértékegység?
- Melyik vizualizációt szeretnénk látni?
- Milyen felbontás(ok)ban?
- Van-e szeletelő?

A vezetői kérdések megfordításaként bemutatásra kerül a Microsoft Power BI beépített Q&A vizualizációja, amely szoros összefüggést mutat a bevezetőben említett promptolás, prompt engineering tevékenységgel. Ennek során angol nyelven megfogalmazott formális kérdésből a szoftver alapszinten képes a 4. ábrán látható vizualizáció előállítására.



Forrás: [10]

3. ábra Az „average ICT skill by regions barchart” kérdésre adott Q&A vizualizációs eredmény

A képzés során törekszünk bevonnunk a hallgatókat a közös munkába. Az elméleti számonkérés kérdéseit a Covid előtti időszakban a hallgatók által Google kérdőívben megszerkesztett tesztkérdések jelentették, amely a jegyük 20%-át adta. Figyelniük kellett a kérdés megfogalmazására, illetve a helyes válasz/válaszok jelölésére is. A 20%-ot a kérdés kidolgozottságára, pontosságára kapták, nem pedig a nehézségére. A hallgatók számára a vizsga előtt az összes, félévenként nagyjából 160 kérdés a válaszokkal együtt rendelkezésre állt. A vizsgát papíron írták és természetesen nem használhattak segédeszközt. Oktatói szempontból érdekes volt látni, hogy az adott évfolyamon a hallgatók mit értettek meg az előadásomból. A Covid természetesen megváltoztatott sok mindent. A múltbéli tapasztalatok alapján az összegyűlt nagy számú kérdésből témakörökre bontott eLearning tesztet készítettünk. Napjainkban a kurzushoz tartozó 7 előadásból 6 első 10 percében maximum 6 percen az addig tárgyalt témaköröknek megfelelően kapnak a

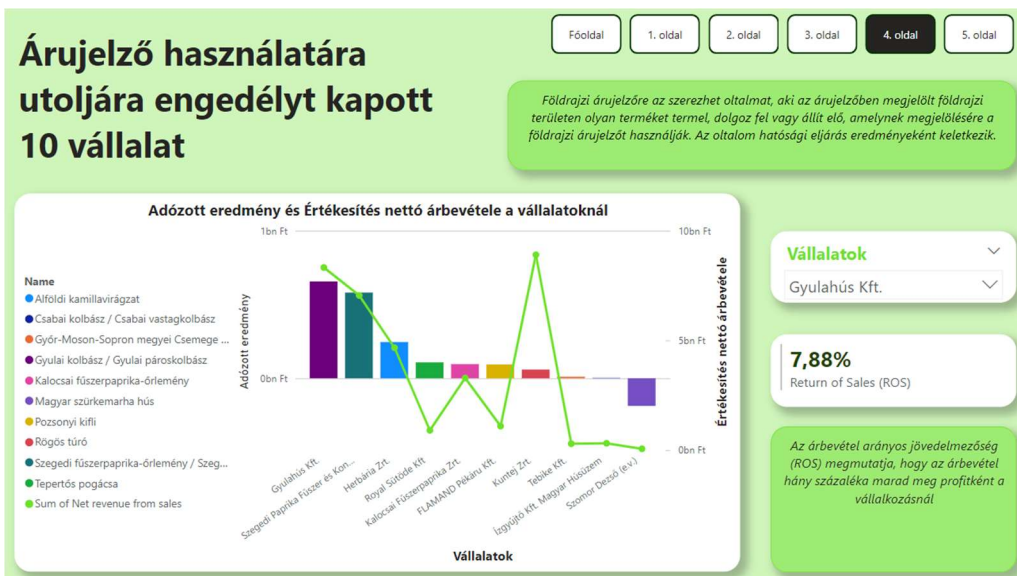
hallgatók az éles vizsgakérdésekből 5-öt 5 pontért. Ezen tesztek eredményeit felhasználva az előadások első 20-30 percében a hibás válaszokból szemezgetve ismétlésre kerülnek a jó válaszokat megalapozó tananyagrészek, kiegészítő magyarázatokkal. Ezen villámteszteken elért eredmény beleszámít az elméleti vizsga 45 pontos eredményébe, mégpedig úgy, hogy a vizsga elméleti teszten elért pontja arányosításra kerül. Az elérhető 45 pontból kivonjuk a villámteszteken elért eredményeket (a villámtesztek megírása opcionális).

A kurzushoz tartozó gyakorlati vizsgán a hallgatók 45 pontot szerezhetnek. A gyakorlaton használt tranzakció orientált adatforrásból kell kinyerniük és megfelelően átalakítani az adatokat 10 pontért, valamint vezetői kérdésekre kell adniuk vizualizált válaszokat 2 alap, 2 közepes és 1 emelt besorolású véletlenszerűen választott kérdésre. Ezeket a kérdéseket a hallgatók állítják össze 10 pontért. Az 5 kérdés mindegyike 7 pontot ér, nehézségtől függetlenül, hiszen az értékelés már eleve progresszív. A gyakorlati részhez is szerezhetnek a hallgatók opcionális pontokat. A Microsoft ingyen hivatalos, minőségi tananyagokat [11] nyújt számos nyelven a Power BI szoftverhez. A hallgatók be tudnak lépni és különböző modulok teljesítésével jelvényeket szerezni. Ezzel a gyakorlati részhez tudnak maximum 4 modul elvégzésével modulonként 5 pontot szerezni. A tapasztalataim azt mutatják, hogy még FOSz szinten is van néhány hallgató, aki a kurzushoz tartozó minimum elváráson felül érdeklődik és 15-20-nál is több modul jelvényeit megszerzi. A többség megelégszik a minimummal. Természetesen ebben az esetben is a gyakorlati vizsga eredménye arányosításra kerül a hiányzó gyakorlati pontokra. A gyakorlati vizsgarész kiváltható önálló, egyedi beadandóval, ahol a hallgatóknak önállóan kell keresniük adatforrást, azon be kell mutatniuk a gyakorlaton tanult adatátalakítások széles skáláját, illetve látványos adatvizualizációs jelentéslapokat kell előállítaniuk. Egy FOSz-os hallgató jobban sikerült munkájának egy része látható az 5. ábrán, valamint egy MSC-s hallgatóé a 6. ábrán.



Forrás: hallgatói munka

5. ábra FOSz képzésben résztvevő hallgató beadandójának részlete



Forrás: hallgatói munka

6. ábra MSc képzésben résztvevő hallgató beadandójának részlete

## Összegzés

A közgazdász képzésben kiemelten fontosnak tartjuk, hogy a hallgatók már az elejétől kezdve információs rendszer elméleti megközelítésben kapják a tananyagot, és az Excel ne egy számológép helyettesítő eszköz legyen csak számukra. Ennek megfelelően fejlesztjük a hallgatók probléma megoldó képességét, analitikus gondolkodásmódját, valamint rendszerszemléletét, megalapozva az egyre fontosabbá váló promptolási készséget.

## Irodalomjegyzék

- [1] Marvin, G., Hellen, N., Jjingo, D., Nakatumba-Nabende, J.: *Prompt Engineering in Large Language Models*. In: Jacob, I.J., Piramuthu, S., Falkowski-Gilski, P. (eds) *Data Intelligence and Cognitive Informatics, ICDICI 2023, Algorithms for Intelligent Systems*, Springer, Singapore, 2024; [https://doi.org/10.1007/978-981-99-7962-2\\_30](https://doi.org/10.1007/978-981-99-7962-2_30);
- [2] Takács V., Bubnó K.: *Mathability in Business Education*. *Acta Polytech. Hung.* 19 (1), 9-29, 2022, <https://doi.org/10.12700/APH.19.1.2022.19.2>;
- [3] Takács, V., Takácsné Bubnó, K., Ráthonyi, G., Bácsné Bába, É., Szilágyi, R.: *Data Warehouse Hybrid Modeling Methodology*. *Data Sci. J.* 19 (1), 1-23, 2020, <https://doi.org/10.5334/dsj-2020-038>;
- [4] Millwood R., Bresnihan N., Walsh D., Hooper J.: *Review of literature on computational thinking*. NCCA, 2018, [https://ncca.ie/media/3557/primary-coding\\_review-of-literature-on-computational-thinking.pdf](https://ncca.ie/media/3557/primary-coding_review-of-literature-on-computational-thinking.pdf);
- [5] Papert S.: *Mindstorms. Children, computers and powerful ideas*. Basic Books, 1980, <https://dl.acm.org/doi/pdf/10.5555/1095592>;
- [6] Pólya G.: *How to solve it*. 2nd ed., Doubleday, 1957;
- [7] Halassy B.: *Adatmodellezés*. 2000, <https://mek.oszk.hu/11100/11144/11144.pdf>;
- [8] <https://app.diagrams.net/>;
- [9] Wing J. M.: *Computational thinking*. *Communications of the ACM*, 2006, 49.3: 33-35, <https://doi.org/10.1145/1118178.1118215>
- [10] Szilágyi, R., Botos, S., & Felföldi, J. (2023). *Application of smart visualisation in the analysis of Hungarian Agriculture ICT*. *Journal of Agricultural Informatics*, 13(2). <https://doi.org/10.17700/jai.2022.13.2.681>,
- [11] <https://learn.microsoft.com/hu-hu/training/browse/?products=power-bi>