



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

**NEUMANN JÁNOS
INFORMATIKAI KAR**



SZAKDOLGOZAT

**OE-NIK
2021**

Hallgató neve:
Hallgató törzskönyvi száma:

**Dobi Fanni
T/005077/FI12904/N**

SZAKDOLGOZAT FELADATLAP

Hallgató neve:
Törzskönyvi száma:
Neptun kódja:

Dobi Fanni
T/005077/F112904/N

A dolgozat címe:

Telekommunikációs cég adatainak elemzése és vizualizációja
Analysis and visualization of telecommunication company data

Intézményi konzulens:
Külső konzulens:

Simon-Nagy Gabriella

Beadási határidő:

2021. december 15.

A záróvizsga tárgyai:

Számítógép architektúrák
Big Data és üzleti intelligencia
specializáció

A feladat

Tervezzon és valósítson meg egy olyan rendszert, amely lehetővé teszi egy telekommunikációs cég forgalmi adatainak elemzését, vizualizációját. Kutassa fel és vizsgálja meg a feladat megoldásához használható technológiákat és adatforrásokat. Ismertesse a hasonló rendszerek működését és képességeit. Valósítson meg egy olyan tesztrendszert, amely lehetővé teszi adathalmazok importálását, elemzését, releváns információk, összefüggések egyszerű kinyerését és vizualizációját.

A dolgozatnak tartalmaznia kell:

- a feladat pontos ismertetését,
- a hasonló rendszerek működését és képességeit,
- a feladat megoldásához alkalmazható technológiák és adatforrások tömör bemutatását,
- a választott megoldások indoklását,
- a tervezés és megvalósítás részletes leírását,
- a tesztelés tervét és eredményeit,
- az elkészített rendszer eredményeinek részletes bemutatását és értékelését,
- a továbbfejlesztési lehetőségek számba vételét.



.....
Fc n
Dr. Fleiner Rita
intézetigazgató

A szakdolgozat elévülésének határideje: **2023. december 15.**
(OE TVSz 55.§ szerint)

A dolgozatot beadásra alkalmasnak tartom:

.....
külső konzulens

.....
JH
intézményi konzulens



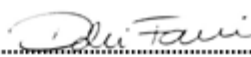
ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar
7. sz. melléklet

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, 2021.12.14|


.....
hallgató aláírása



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar

KONZULTÁCIÓS NAPLÓ

Hallgató neve: Dobi Fanni
Neptun Kód: [REDACTED]
Tagozat: Nappali

Telefon: [REDACTED]
Levelezési cím (pl.: lakcím): [REDACTED]

Szakdolgozat címe magyarul:

Telekommunikációs cég adatainak elemzése és vizualizációja

Szakdolgozat címe angolul:

Analysis and visualization of telecommunication company data

Intézményi konzulens: SIMON-NAGY GABRIELLA
Külső konzulens:

Alk.	Dátum	Tartalom	Aláírás
1.	2021. 02. 26.	A téma meghatározása	[Signature]
2.	2021. 03. 19.	A kutatási területek áttekintése	[Signature]
3.	2021. 04. 16.	Az irodalomkutatás összegzése	[Signature]
4.	2021. 05. 10.	A rendszerspecifikáció véglegesítése	[Signature]

A hallgató a Szakdolgozat I. tantárgy követelményét teljesítette, beszámolóra bocsátható.

[Signature]

.....
Intézményi konzulens

Budapest, 2021. május 14.



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar

KONZULTÁCIÓS NAPLÓ

Hallgató neve: Neptun Kód: Tagozat:

Dobi Fanni [REDACTED] Nappali

Telefon: Levelezési cím (pl.: lakcím):

[REDACTED]

[REDACTED]

Szakdolgozat címe magyarul:

Telekommunikációs cég adatainak elemzése és vizualizációja

Szakdolgozat címe angolul:

Analysis and visualization of telecommunication company data

Intézményi konzulens:

Külső konzulens:

SIMON-NAGY GABRIELLA

Alk.	Dátum	Tartalom	Aláírás
1.	2021. 09. 21.	Megvalósítási lehetőségek áttekintése	[REDACTED]
2.	2021. 10. 19.	Fejlesztési fázis ellenőrzése	[REDACTED]
3.	2021. 11. 22.	Tesztelés ellenőrzése	[REDACTED]
4.	2021. 12. 10.	Eredmények, összegzés	[REDACTED]

A hallgató a Szakdolgozat II. tantárgy követelményét teljesítette, védelemre bocsátható.

[REDACTED]

Intézményi konzulens

Budapest, 2021. december 10.

Tartalom

1.	Bevezetés	10
2.	Irodalomkutatás	11
2.1.	Adatbázis	11
2.1.1.	Fogalma	11
2.2.	Relációs adatbázis	12
2.2.1.	Kialakulása, fogalma	12
2.2.2.	Tárolt adatok	13
2.2.3.	Kapcsolatok	13
2.3.	Relációs adatbázis-kezelő rendszerek	13
2.3.1.	Oracle.....	13
2.3.2.	MS-SQL.....	14
2.3.3.	MySQL	15
2.3.4.	PostgreSQL	16
2.3.5.	SQLite.....	16
2.4.	Adattárház	16
2.4.1.	Kimball	17
2.4.2.	Inmon	17
2.5.	OLAP	18
2.6.	Extract Transform Load	19
2.6.1.	Extract.....	19
2.6.2.	Transform.....	20
2.6.3.	Load	20
2.7.	ETL eszközök.....	20
2.7.1.	Talend Open Studio	20
2.7.2.	OracleData Integrator	20
2.7.3.	SQL Server Integration Service	21

2.8.	Üzleti intelligencia	21
2.9.	Riport készítő eszközök	22
2.9.1.	Tableau.....	22
2.9.2.	Cognos Analytics.....	23
2.9.3.	Power BI	24
2.9.4.	TOPdesk.....	24
3.	Kutatások összegzése.....	26
4.	Specifikáció	27
5.	Rendszerterv	28
5.1.	Táblák.....	28
5.2.	UML-diagram	28
5.3.	Csillag séma	29
5.4.	Munkaterv	29
6.	Megvalósítás	30
6.1.	Táblák definiálása	30
6.2.	Táblák szerkezete	30
6.3.	Adatmodell.....	31
6.4.	Lekérdezés optimalizálás	32
6.4.1.	Lekérdezés	32
6.5.	Adattárház specifikáció.....	34
6.6.	Dimenzionális modell	34
6.6.1.	Dimenzió táblák.....	34
6.6.2.	Ténytábla	34
6.6.3.	Mapping.....	34
6.7.	ETL folyamatok	36
6.7.1.	Extract réteg.....	36
6.7.2.	Transform réteg.....	37
6.7.3.	Load réteg	38

7.	Riportkészítés.....	40
7.1.	A dolgozók fizetésének összehasonlítása.....	40
7.2.	Országokon belüli üzletek.....	41
7.3.	A csomagok népszerűsége	42
7.4.	Városokon belüli ügyfelek száma	43
7.5.	Új ügyfelek eloszlása	44
8.	Továbbfejlesztési lehetőségek	45
9.	Összefoglaló.....	46
10.	Summary	47
11.	Irodalomjegyzék	48
12.	Ábrajegyzék	52

1. BEVEZETÉS

A mai világban már minden szakterületen használnak adatbázisokat. Gondoljunk csak arra, hogy valami információhoz szeretnénk hozzájutni az interneten. A Google keresés is egy adatbázisban megy végbe, és ez alapján adja ki a számunkra megfelelő oldalakat. Ha pedig már megtaláltuk a számunkra megfelelő weblapot, ott is tudunk keresni, vagy akár meg is rendelhetünk valamit. Ezek mögött is adatbázisok állnak.

Természetesen nem csak az interneten találkozhatunk ezzel. A bankok, a repterek, a telekommunikációs cégek, az orvostudomány és a boltok is ugyan úgy adatbázisokban tárolják az adataikat, és onnan nyerik ki az információt.

Majdnem mindenki használ valamilyen adatbázist, akár saját magának otthon, akár a saját cégének, vagy a cégnél, ahol dolgozik. Sokszor észre sem lehet venni, hogy éppen mi is működik a háttérben.

Én ebből a nagy halmazból egy telekommunikációs cég adattárolását szeretném bemutatni. Azért ezt választottam, mert manapság már majdnem mindenkinek van valamilyen kapcsolata ezekkel a cégekkel. Hozzájuk tartozik a mobil szolgáltatás, a tv szolgáltatás, az internet szolgáltatás, valamint a vezetékes telefon is. Ezek olyan eszközök, amik nélkül már nehéz lenne elképzelni a mindennapi életet.

Szakedolgozatom során bemutatok egy olyan alap adatkört, ami biztosan megtalálható ezeknél a cégeknél. Ebbe beletartoznak a cég adatai és az ügyfél adatai, valamint a kettő közötti kapcsolat. Ezeket az adatokat én fogom generálni a személyi jogok megsértését elkerülve.

Elsőként egy relációs adatbázisban fogom prezentálni ezeknek az adatoknak a kapcsolatát. Különböző lekérdezéseket fogok készíteni annak érdekében, hogy látható legyen az, hogy a cég például hány ügyféllel rendelkezik, vagy melyik szolgáltatás a legnépszerűbb.

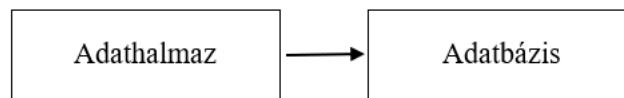
Másodjára pedig egy adattárházban fogom ezt szemléltetni, ami után különböző riportokat, kimutatásokat is készíteni fogok, annak érdekében, hogy vizuálisan is megjeleníthető legyen, és azonnal átlássa mindenki a cég helyzetét.

2. IRODALOMKUTATÁS

Az irodalomkutatásomban be szeretném mutatni a relációs adatbázisokat és az ezekhez tartozó szoftvereket. Ezután pedig az adattárházakat fogom bemutatni és a riportkészítő szoftvereket. Majd mindegyikből választok egyet amelyet én is használni fogok a szakdolgozatom során.

2.1. Adatbázis

„Adatbázison köznapiban értelemben valamely rendezett, valamilyen szisztéma szerint tárolt adatokat értünk, melyek nem feltétlenül számítógépen kerülnek tárolásra. Az adathalmaz csak akkor válik adatbázissá, ha az valamilyen rend szerint épül fel, mely lehetővé teszi az adatok értelmes kezelését.” [1]



2.1. ábra - Átalakítás

Egy adott adathalmazt többféleképpen is lehet rendezni, hogy abból adatbázis legyen, a 2.1. ábra mutatja ezt a kapcsolatot. [1] Például az ügyfeleket rendezhetem aszerint is, hogy hol laknak, de aszerint is, hogy melyik évben születtek, vagy az alapján, hogy melyik évben kötöttek szerződést a céggel.

Mindhárom rendezés megfelelő adatbázist ad vissza. A rendezés az igénytől függ.

2.1.1. Fogalma

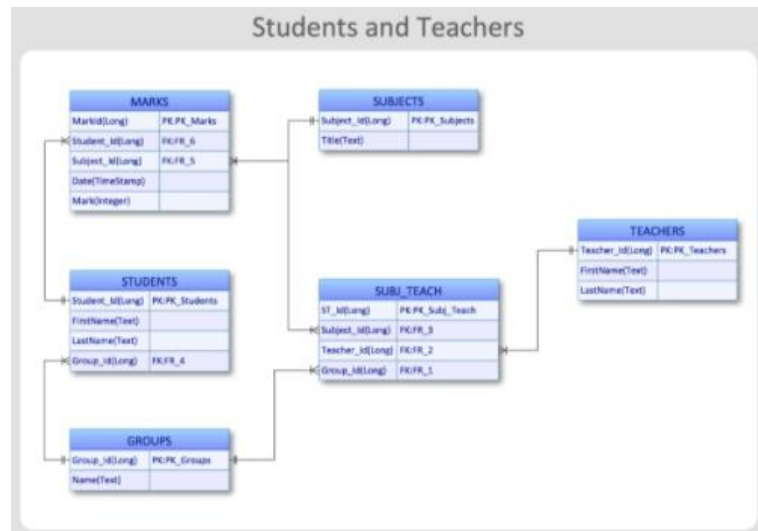
Az adatbázis olyan adatok gyűjteménye, amelyek valamilyen módon összekapcsolhatók egymással. Strukturálnak nevezhető, ha bizonyos lépésszám után minden adathoz el tudunk jutni. Az adatbázis egy olyan integrált adatszerkezet, amely metaadatokat is tartalmaz, azaz adatokat tárol az adatokról. [1]

Ezeket az adatbázisokat egy adatbázis-kezelő rendszer kezeli. Amelyektől elvárt, hogy bővíthető legyen még több adatbázissal, és ezzel nagy mennyiségű adatok tárolására legyen képes. Ezen kívül fontos, hogy biztonságos legyen, ne tudjon adat elveszni a rendszerből. [2]

2.2. Relációs adatbázis

2.2.1. Kialakulása, fogalma

A relációs adatbázisok az 1970-es években fejlődtek ki, miután Ted Codd kiadott egy cikket arról, hogy a felhasználó irányába az adatokat táblázatos formában kellene megjeleníteni, ezeket a táblákat nevezzük relációnak. [2]



2.2. ábra - Táblák és kapcsolataik [3]

Ezáltal már több táblát tudtak kezelni egyszerre, a kapcsolataikkal, összefüggéseikkel együtt, amit a 2.2. ábra be is mutat. A 2.3. ábra mutatja, hogy egy tábla sorokból és oszlopokból áll. Az adatokat pedig mezőértékeknek nevezik. Fontos, hogy nem lehet két ugyanolyan nevű oszlopunk. [1]

		OSZLÓP			
SOR					
				mezőérték	

2.3. ábra - Tábla szerkezet [1]

Annak ellenére, hogy a felhasználó egy sokkal egyszerűbb felületet lát, a háttérben bonyolult adatstruktúra is lehet. A fejlesztők az SQL nyelvet használták ahhoz, hogy minél nagyobb sebességgel tudjanak adatokat feldolgozni. [2]

2.2.2. Tárolt adatok

Egy adatbázisban nem csak magát az adatot tároljuk, amivel dolgozni szeretnénk, hanem metaadatokat is. Ez azt jelenti, hogy az adatokról tárolunk adatokat, hogy például milyen típusú az az adat. Ezen kívül még szükséges lehet napló rekordokra is, ami azt jelenti, hogy a változtatásokat tudjuk nyomon követni és ezáltal vissza is tudjuk vezetni őket. Fontos szerepe van még az indexeknek, ugyanis ez az adatszerkezet segít abban, hogy gyorsabban, hatékonyabban el lehessen érni az adatokat. [2]

2.2.3. Kapcsolatok

Három fajta kapcsolatot különböztetünk meg egymástól, az 1:1 fokút, az 1:N fokút és az M:N fokút. Az 1:1 fokú azokat a kapcsolatokat tartalmazza, amelyeknél egy egyed típus egy előforduláshoz tartozik. Ilyen például a személy igazolvány száma, egy valakihez csak egy személyi szám tartozhat. Az 1:N fokú kapcsolat azt jelenti, hogy egy egyed típus több előforduláshoz is tartozhat, de ez nem kötelező. Például egy embernek lehet több kutyája, de egy kutyának csak 1 gazdája lehet. A harmadik kapcsolat típus az M:N, aminél egy egyed típus több előforduláshoz is tartozhat. és ez visszafelé is igaz. Például egy embernek lehet több munkahelye és a munkahelyének is lehet több alkalmazottja. [1]

2.3. Relációs adatbázis-kezelő rendszerek

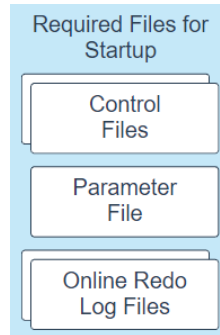
2.3.1. Oracle

Az Oracle rendszer egy kereskedelmi szoftver, ezért nincsen ingyenes verziója. Ez egy kliens-szerver felépítésű rendszer, amit egyidejűleg több felhasználó is tud használni. Több operációs rendszeren is futtatható. Nagy adathalmazokkal tud dolgozni, és magas szintű védelmet nyújt. Ezen kívül összetett struktúrák használata is lehetséges benne, ilyenek például az objektumok kezelése, vagy az eljárások kezelése. Az Oracle adatbázis két fő részből áll, a fizikai és a logikai felépítésből. Ezek egymástól különállnak és karbantarthatóak. Viszont kapcsolat is van közöttük, hogy az összefüggések értelmezhetőek legyenek. [4]

Az adatbázisunkat relációkra osztjuk és ezek adják a logikát. Egy relációba természetesen összetartozó elemeket rakunk. Vannak olyan rendszeradataink, amit mindenképpen el kell tárolni ezeket egy system nevű táblahelyre rakjuk. Ezen felül kötelező az undo (a változások naplózását tartalmazó) és a temp (átmenetelig létrejövő, tárolandó adatot tartalmazó) tábla. Különböző objektumokat tudunk létrehozni. Ilyen a tábla, a nézet, a szinonima, a klaszter és még sok más. [4]

A 2.4. ábra bemutatja, hogy a fizikai felépítést három különböző részre oszthatjuk. Az első a Data file, azaz az adatállomány, ami a táblahelyek adatait írja le. A második a Redo-log file,

amiből mindig legalább kettőnek kell lennie. Ezek arra szolgálnak, hogy az elvégzendő műveleteket tárolják, addig ameddig nem sikerült az adatkivitel. A harmadik pedig a Control file, ami az adatbázisunk alap adatait tartalmazza, ide tartozik az adatbázis neve, a fizikai elhelyezkedése vagy éppen az adatfájlok nevei. [4]



2.4. ábra - Oracle fizikai felépítése [5]

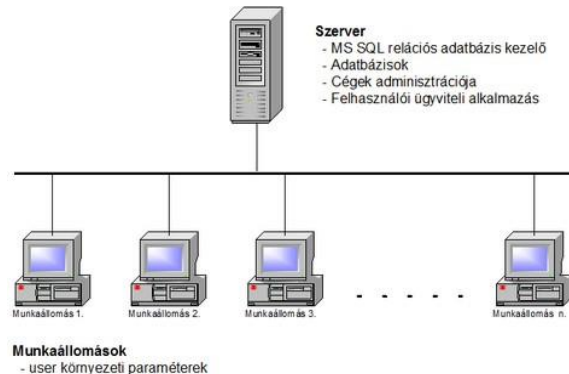
A fizikai rész logikai egységekből áll, ez adja a két rész közötti kapcsolatot. Ennek 3 szintje van. A legkisebb az adatblokk, a következő az extent, ami igazából adatblokkok halmaza, és az utolsó a szegmens, amit az összetartozó extentek alkotnak. [4]

2.3.2. MS-SQL

Az MS-SQL a Microsoft SQL szerverét takarja. Ami több platformon is használható (Linux, Windows), többfelhasználós és valós idejű adatokat szolgáltat. [6] Ki tudjuk választani, hogy milyen nyelven szeretnénk használni a szoftvert (angol, német, japán, spanyol, koreai, orosz, francia, kínai, portugál). Relációs, nem relációs, strukturált és nem strukturált adatokat is tudunk benne használni és elemzéseket végezni rajtuk. A legújabb, 2019-es verzió, gyorsabb, magasabb a teljesítménye és biztonságosabb, mint az előzőek. A továbbfejlesztett PolyBase lehetővé teszi, hogy más adatbázisból például Oracle, MongoDB, Teradata lekérdezéseket hajtsunk végre, anélkül, hogy az adatainkat másolni kellene. Ezen felül jól integrálható más szoftverekkel például a Power BI-al, így még gyorsabban tudunk adatokat elemezni, és megjeleníteni. [7]

A kliens nem tárol adatokat a saját számítógépén, hanem hálózati kommunikációval éri el a szervert, ahol megtalálja a szükséges adatokat, ezt ábrázolja a 2.5. ábra. Ez azt jelenti, hogy igazából a szerver végzi el az adatok tárolását és elemzését is, ezen kívül tranzakciót hajt végre és alkalmazásokat készít. A szerver OLTP és OLAP adatkockákat használ, azaz az adatok rendezve és összesítve vannak. Kétféle adatbázist tartalmaz a szerver. Az egyik a rendszeradatbázis, ami magáról a szerverről tárol adatokat, ilyen adat a működés vagy a

felépítés. A másik a felhasználói adatbázis, ahol a felhasználó által létrehozott adatok vannak eltárolva. A felhasználó csupán az utóbbival tud dolgozni. [8]



2.5. ábra - Szerver - kliens kapcsolat [9]

2.3.3. MySQL

A MySQL a világ egyik legnépszerűbb nyílt forráskódú adatbázisa. Több platformon futtatható (Windows, Linux, MacOS), alacsony a költsége, megbízható és stabil. Könnyű a használata, mert jól átlátható a felépítése. [10] Három lehetőség közül tudunk választani amikor MySQL adatbázis kezelő rendszert keresünk. Az első a Standard Edition, amivel elkészíthetők nagy teljesítményű, skálázható, OLTP alkalmazások. Könnyű kezelhetőséget biztosít, a szilárd teljesítmény és megbízhatóság mellett. Ezen felül tartalmazza az InnoDB-t így teljesen integrált tranzakció biztos. A MySQL Workbench egy integrált fejlesztési, tervezési és adminisztratív környezetet biztosít. [11]

A második lehetőség az Enterprise Edition, ami a legátfogóbb funkciókat, eszközöket tartalmazza, a legmagasabb szintű méretezhetőség, biztonság és üzemidő elérése érdekében. Csökkenti a kockázatot, a költséget, valamint a fejlesztés és az üzembe helyezés komplexitását. Sokféle különböző szolgáltatás kapcsolódik hozzá, ilyen például a TDE (Transparent Data Encryption), ami által lehetőség van az adatok titkosítására, akár valós időben, mielőtt a tárolóba íródnának. Van benne egy másik féle titkosítás, ami kulcsgenerálást, digitális aláírást és más kriptográfiai szolgáltatásokat biztosít. Ezek a bizalmas adatok megvédése és a hatósági előírások érdekében fontosak. [12]

A harmadik lehetőség a Cluster Carrier Grade Edition (Cluster CGE), amelyet a kritikus alkalmazásokhoz terveztek. Ez egy elosztott adatbázis, ami kombinálja a lineáris méretezhetőséget és a magas elérhetőséget. Valamint valós idejű hozzáférést nyújt, dinamikusan kezeli a memóriát, ami azt jelenti, hogy az erőforrásokat automatikusan elosztja és 4 adatmásolatot is támogat a rendszer. A memória optimalizált táblákból kifolyólag valós idejű válaszidőt és átviteli sebességet biztosít. Ezen felül automatikusan particionálja a

táblákat a csomópontokon, ami fokozza a hatékonyságot. Leginkább weboldalak, telekommunikációs cégek és vállalatok használják. [13]

2.3.4. PostgreSQL

A PostgreSQL egy objektum relációs adatbázis kezelő rendszer, amely a POSTGRES-en alapszik. Ez egy nyílt forráskódú rendszer, amit bárki ingyenesen használhat, átalakíthat és terjeszthet bármilyen célra, legyen az magán, kereskedelmi vagy tudományos cél. Számos modern funkciót kínál, mint például az összetett lekérdezés, az idegen kulcsok, a triggerek, a frissíthető nézetek. Ezen felül pedig a felhasználó készíthet magának akár új adattípusokat, funkciókat, index módszereket, aggregált függvényeket. [14]

2.3.5. SQLite

Az SQLite egy folyamaton belüli könyvtár, amely önálló, szerver nélküli, konfiguráció nélküli, tranzakciós SQL adatbázis-motort valósít meg. Az SQLite kódja nyilvános, és ingyenesen használható kereskedelmi vagy magán célra. A legtöbb más SQL adatbázissal ellentétben, neki nincsen külön szerver folyamata. Közvetlenül a lemezfájlokra ír és olvas. Többplatformos a fájlformátuma, nyugodtan másolható egy adatbázis a 32 és 64 bites verziók között. Az adatbázisfájlok az Egyesült Államok kongresszusi Könyvtára által ajánlott tárolási formátumúak. [15]

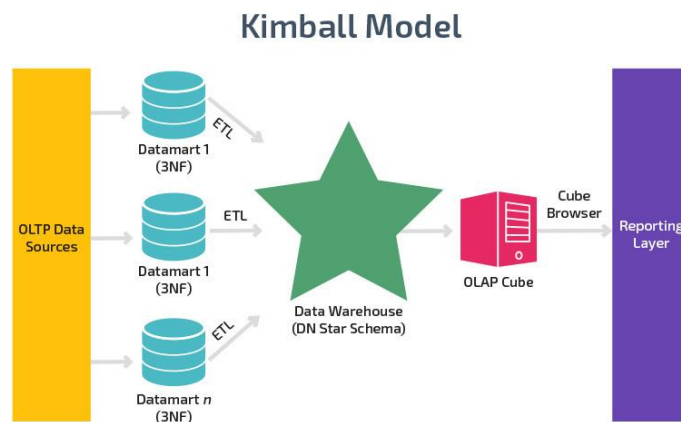
Azt SQLite egy kompakt könyvtár, ha az összes funkció engedélyezve van akkor a könyvtár mérete akár kevesebb lehet, mint 600KB. Kompromisszum van a memória használat és a sebesség között. Minél több memóriát adunk neki, annál gyorsabban fut le. Ennek ellenére kevés memória mellett is jó a teljesítmény. Az is előfordulhat, hogy gyorsabb, mint a közvetlen fájlrendszer írás, olvasása. A rendszert kiadás előtt gondosan tesztelik, ezáltal nagyon megbízható. [15]

2.4. Adattárház

Az adattárházaknak két nagy alapítója van, Ralph Kimball és Bill Inmon. Bill Inmon szerinti definíció: „*Data Warehouse (DW) as, a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision making process*” [16]. Magyarul ez azt jelenti, hogy az adattárház egy témakör-orientált, integrált, időfüggő, de időben nem változó adatok gyűjteménye, amelyet a cég stratégiai döntéshozatalának támogatására használnak. Amit még mindenképpen ismernünk kell az a Data Mart, ami egy témaorientált, szakterülethez köthető része az adattárháznak. Ezek konkrét feladatokat is el tudnak látni és egy kisebb adattárház funkcióit is elláthatja. [17]

2.4.1. Kimball

Az adattárház architektúrákból számos fajta létezik. A Kimball megközelítés szerint, amit a 2.6. ábra mutat, az ETL szoftvert arra használja, hogy összegyűjtse a különböző forrásból származó adatokat egy átmeneti területre. Innen kerülnek az adatok egy dimenziós modellbe. A modellezés alapkoncepciója a csillag séma, ami általában egy tény táblából áll és ehhez kapcsolódik több dimenzió tábla. A ténytábla tartalmazza azokat az adatokat, amik közvetlenül kapcsolódnak a tématerülethez, valamint az idegen kulcsokat, amivel a dimenziókat hozzá lehet kapcsolni. Létre lehet hozni olyan dimenziókat, amik állandóak lesznek, mint például a vásárlók dimenzió. Ebben az esetben nem kell minden alkalommal új dimenziót készíteni, hanem mindenki ezt az egyet használja, így csökkenthető a redundancia. A csillagsémának nagy előnye, hogy könnyen értelmezhető, gyorsan átlátható az üzleti felhasználók számára is. Teljesítménye nagyon hatékony. Hátránya, hogy a régi adatokat nehézkes integrálni, és a ténytábla oszlopainak bővítése teljesítményproblémához vezethet. [18]

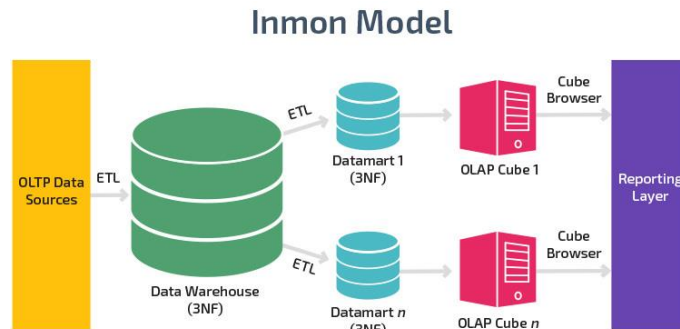


2.6. ábra - Kimball modell [19]

2.4.2. Inmon

Az Inmon megközelítés, amit a 2.7. ábra is mutat, meghatározza a legfontosabb tématerületeket, és az entitásokat, amelyekre szükség van, ilyen például az ügyfél. Ezekből egy részletes logikai modell készül, ami eltárolja az entitásokat, a kulcsokat, az attribútumokat, függőségeket. Az entitások struktúrája normalizált formában épül fel. Így a redundanciát nagyban lehet csökkenteni. Ezek után következik a fizikai modell, ami szintén normalizált. Itt tárolja az adatokat. Ebben a rendszerben az adatbetöltés könnyebb, viszont a lekérdezések nehezebbek, a sok adatkapcsolat miatt. Előnye, hogy az adatok ténylegesen egy helyen vannak tárolva, és kicsi a redundancia, így nehezebben akadnak össze az adatok. Nagyon rugalmas, akár a vállalaton belüli változásnál, akár az üzleti követelmények

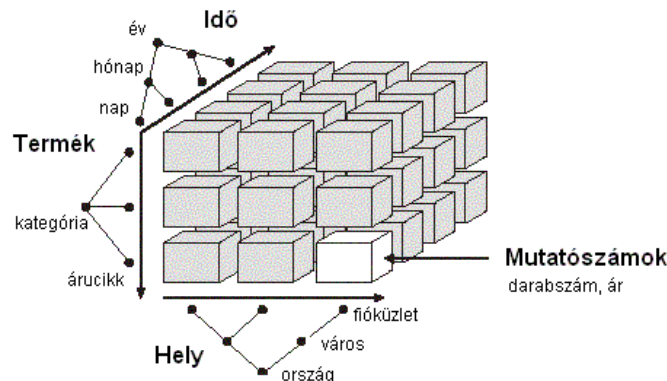
változásánál. Hátránya, hogy egy idő után nagyon összetetté és kissé értelmezhetetlenné válhat, a sok táblakapcsolat miatt. Több ETL folyamatra van szükség mert a Data Martok az adattárházból épülnek fel, ezáltal a megtervezése is nehezebb és időigényesebb. [18]



2.7. ábra - Inmon modell [19]

2.5. OLAP

A 2.8. ábra bemutatja az OLAP kockát (Online Analytical Processing - közvetlen analitikus feldolgozás), amit az adattárházból származó adatok elemzésére használunk. A rendszerben tárolhatunk múltbéli adatokat is az aktuális állapoton kívül, ez arra szolgál, hogy tudjunk jövőbe mutató elemzéseket készíteni. Az OLAP nem egy adatforrás, ő csak veszi az adatokat, például egy OLTP rendszerből. Az egyszerű lekérdezések helyett inkább az összetettebb lekérdezésekre van építve, ami hosszabb ideig tartana, ezért nagyon kell figyelni a hatékonyságra. Többfelhasználós, és felhasználó barát rendszer. [20]



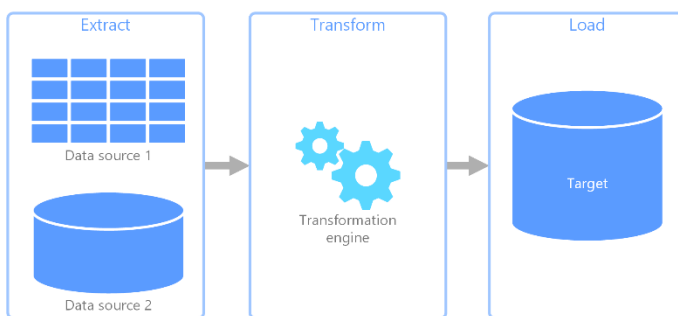
2.8. ábra - OLAP kocka [21]

Az OLAP kockának több csoportosítása van. Az egyik a MOLAP, azaz a multidimenzionális OLAP. Itt az adatok többdimenziós kockákban vannak tárolva, elszeparálva az adattárháztól, adatkonténerekben. Itt egy speciális multidimenzionális séma az adattárház séma. A következő a ROLAP, ami a kapcsolati OLAP. Itt az adatokat egy hagyományos relációs

adatbázisban tároljuk, ami a valóságban egy adattárház egy tábláját jelenti. Ide tarozik a csillagséma, vagy a hópehely séma. A harmadik pedig a HOLAP, vagyis a hibrid OLAP, ami az előző kettő lehetőséget ötvözi. Tárolhatjuk az adatokat hagyományosan, de multidimenzionálisan is. [17]

2.6. Extract Transform Load

Az ETL a különböző forrásból származó adatok összegyűjtésére, ezek átalakítására a megfelelő üzleti folyamatok szerint és a célhelyre való másolásra szolgál, ezt mutatja a 2.9. ábra is. Eleinte javarészt adattárházaknál alkalmazták, de manapság szélesebb körben használják. Ma már támogatja az adatmozgást az adattárak, a BI rendszerek, a felső és a Hadoop között is, valamint a BigData platformokat. Az ETL három fázisra bontható, az Extractra ami az adatok beolvasásáért felelő, a Transformra ami az adatok transzformálásra szolgál, és a Loadra, ami a már átalakított adatokat tölti be. Ezek a folyamatok párhuzamosan is történhetnek, a nagy adatmennyiség miatt. Tehát ha betöltött már egy adathalmazt azon elkezdődik a transzformálás, de közben töltődik a következő adathalmaz. [22]



2.9. ábra - ETL folyamat [23]

2.6.1. Extract

Ebben a rétegben az a cél, hogy az adatainkat sikeresen kinyerjük a különböző forrásokból és bekerüljenek az adattárházba úgy, hogy azok a többi réteg számára megfelelőek legyenek, ezért ez a legidőigényesebb, és a legkritikusabb rész. [22]

Az adatokat többféleképpen lehet importálni, akár CSV, relációs adatbázis, XML, JSON, de még a weboldalak beolvasásával is. Az adatokat vagy közös formára hozzuk és ideiglenesen eltároljuk, vagy rögtön a cél adattárba töltjük. Az előbbi szokták gyakrabban használni. Itt kell leellenőrizni az adataink minőségét, hogy a szabályoknak megfelelően érkezett meg és hogy valid értéket tartalmaz. Amennyiben van olyan sor, ami nem felel meg, azt egy külön táblába elrakhatjuk, de nem dolgozzuk fel. Emellett pedig jelzünk a forrásrendszer felé, hogy nem volt teljes a betöltés. [22]

Az, hogy az adatok milyen gyakran érkeznek a megrendelőtől függ, ez lehet napi, havi, de lehetséges, hogy csak egy esemény váltja ki, vagy hogy igény szerint van futtatva.

2.6.2. Transform

Ebben a fázisban állítjuk elő a végleges adatokat. Lehetséges, hogy semmi változtatást nem kell végrehajtani az adatokon, ezzel szemben az is lehet, hogy függvények sorozatával kell módosítani. Ez minden esetben attól függ, hogy milyen adatokat szeretnénk kapni. Leggyakrabban az adatok rendezése, a soroknak a kicserélése, az adatoszlopok szűrése, egy oszlop több oszlopra való bontása, új számított értéket tartalmazó oszlop beszúrása, több oszlop egy oszlopra való szűkítése, értékek helyettesítése vagy az értékek cseréje, rövidítése szükséges. [22]

2.6.3. Load

Ebben a lépésben töltjük be az adatainkat a cél adatforrásba, ami lehet tábla, adattárház, de akár egy sima fájl is. Mivel akár napi szinten is érkezhetnek adatok, el kell dönteni, hogy az előzőekkel mit szeretnénk csinálni. A legegyszerűbb, ha az új adatokat csak betöltjük az előzőek után vagy felülírjuk az előzőeket. Az utóbbi esetben elvesznek az adataink, ez előbbiben pedig egy idő után túl sok adatunk lehet. Egy jó megoldás, ha szükségünk van az összes adatra a historizáció. Ez akár abban a táblában akár egy másikat létrehozva az utolsó változatot el tudja menteni.

2.7. ETL eszközök

2.7.1. Talend Open Studio

A Talend egy nyílt forráskódú, Eclipse alapú ETL eszköz, adatintegrációhoz vagy Big Data-hoz. Egyszerűen drag and drop módszerrel működik, és utána már futtatható is az ETL. Adatforrás lehet egy relációs adatbázis, Excel, Big Data, vagy akár konkrét rendszerek, mint például az SAP vagy a CRM. Előnye, hogy ingyenes és minden szükséges funkciót tartalmaz. [24]

2.7.2. OracleData Integrator

Az ODI platform a komplex adattranszformációra és a gyors, tömeges adatmozgatásra épül. Az ODI funkciókban és képességekben gazdag, rugalmas eszköz. Az adatbehozatal történhet alkalmazásokból, Big Data-ból vagy adatbázisokból is. Nagy teljesítményt, méretezhetőséget és megbízhatóságot biztosít. [25]

2.7.3. SQL Server Integration Service

Az SSIS egy gyors és rugalmas adatadattárház eszköz, az adatok kinyerésére, transzformálására és betöltésére. Használják az adatok átalakítására, tisztítására, összegzésére. Az adatok kinyerhetők Excel fájlból, CSV fájlból, de Oracle adatbázisból is. Grafikus felületének köszönhetően könnyen átlátható és kezelhető, drag and drop segítségével ki is tudjuk alakítani a folyamatokat. [26]

2.8. Üzleti intelligencia

Az üzleti intelligencia (Business Intelligence – BI) fogalmára sokféle definíció létezik, de ami mindegyikben benne van, hogy egy döntéstámogató rendszer a különböző információk alapján. Tehát a BI eszközök nagy adatmennyiségeket dolgoznak fel különböző forrásokból, és ezeket vizuálisan megjelenítik, a könnyebb értelmezhetőség érdekében. Természetesen a megfelelő döntés meghozásához szükségesek a megfelelő adatok is. A BI rendszerek előtt ezeket az elemzéseket manuálisan végezték. Manapság már számos folyamat automatizálva van, ezzel is csökkentve az ebbe befektetett időt és energiát. [27]

Az üzleti intelligenciának 4 fő lépése van. Az első az adatok gyűjtése és átalakítása, annak érdekében, hogy a több forrásból érkező adatok egymással kompatibilisek legyenek. Ehhez a BI az ETL módszert használja, ezáltal könnyen elemezhető és lekérdezhetővé válik az adathalmaz. A második lépés a trendek és a következtetések megismerése. Ez a folyamat legtöbbször már automatizálva van. Többféle adatmodellezési típust tartalmaznak a BI eszközök, ezáltal előre tudják jelezni a trendeket, többféleképpen megvizsgálják az adatokat, és javaslatokat tesznek. A harmadik lépés az adatok megjelenítése. Ebben a fázisban a BI vizuálisan jeleníti meg az adatok grafikonokon, diagramokon keresztül, hogy mindenki számára egyértelmű és könnyen átlátható legyen. Az utolsó lépés pedig az, hogy a múlt és a jelen adatai alapján, ha szükséges javítson a stratégián, ehhez figyeli a piaci változásokat is. [27]

Az üzleti intelligenciával kapcsolatban négy különböző kategóriát tartanak számon. Az első a lekérdezések és a riportolások. Ebben a kategóriába tartoznak az adatok hagyományos elemzése és feldolgozása. Itt általános riportok vagy ad-hoc jellegű lekérdezések megjelenítése a cél, akár képernyőn akár nyomtatott formában. A második kategória az üzleti elemzések, ide tartozik amikor már figyeljük a piacot, és változtatunk is a stratégián. A harmadik kategória az adatbányászat, ami már elkülönül az előzőektől, ugyanis ez már szaktudást igényel. Az adatbányászati eszközöknél már mindenképp ismerni kell az adatbányászat algoritmusait. Ebben a kategóriában próbálnak mintákat találni az adatokban, annak érdekében, hogy következtetéseket tudjanak levonni a jövőre nézve. Nem tudja biztosan megmondani a jövőt, de egy sejtést nyújthat. Az adatbányászat alapja az, hogy

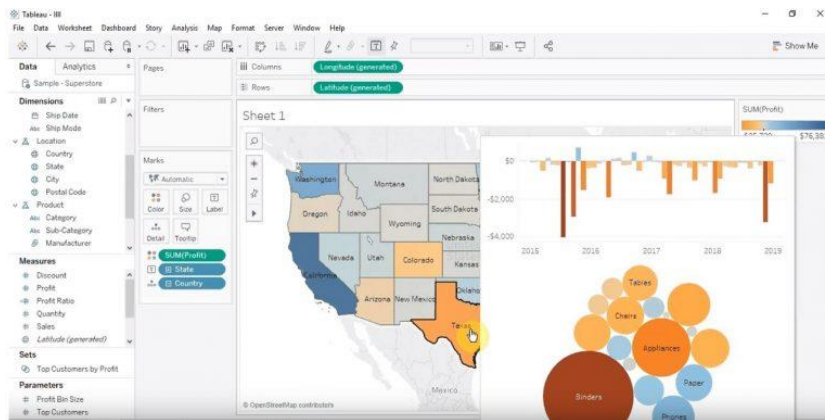
kérdésekre keressük a választ, ha nincs kérdés nem fogunk számunkra megfelelő választ találni. Az utolsó kategória a dashboard. Ez egy olyan eszköz, ami a grafikus megjelenítésre ad számunkra megoldást. [28]

A BI nagy előnye, hogy felgyorsítja az információszerzést és az elemzést is. Csökkenti a potenciális problémákat és növeli a hatékonyságot. A rendszer által betekintést nyernek az ügyfelek viselkedési és vásárlási szokásaiba, így sokkal jobban megismerik az ügyfélkörüket. Amint probléma lenne egy ügyféllel figyelmeztet a rendszer. Sokkal pontosabban nyomon lehet követni a pénzügy, az értékesítés és a marketing teljesítményét. Valamint azonnal meg tudják osztani egymással az elemzéseket, ezzel is hatékonyabbá téve a munkát. [27]

2.9. Riport készítő eszközök

2.9.1. Tableau

A Tableau egy intuitív és interaktív vizualizációs eszköz, ami a Salesforce tulajdonában van. 3 fő termékünk van, a Desktop, ami látható a 2.10. ábra képén, ami a szerkesztői eszköz, itt lehet létrehozni a riportokat, diagramokat, és ezt tudod megosztani másokkal. A Server az a megosztási platform, ami egy helyszíni megoldás. Ez a helyi hardveren vagy a felhőben vehető igénybe. Az Online együttműködési termék pedig teljesen felhő alapú, ezt a Tableau tartja fent és üzemelteti. A Tableau fő célja a vizuális adatelemzés, ezért analitikákat is használ, és szélesen skálázható. Előny, hogy van belőle ingyenes próbaverzió és hogy sok ismeretterjesztő anyag áll rendelkezésre. Drag and drop módszerrel tudunk vizualizációkat létrehozni, ezért nagyon könnyű és felhasználóbarát a kezelőfelülete.



2.10. ábra – Tableau [29]

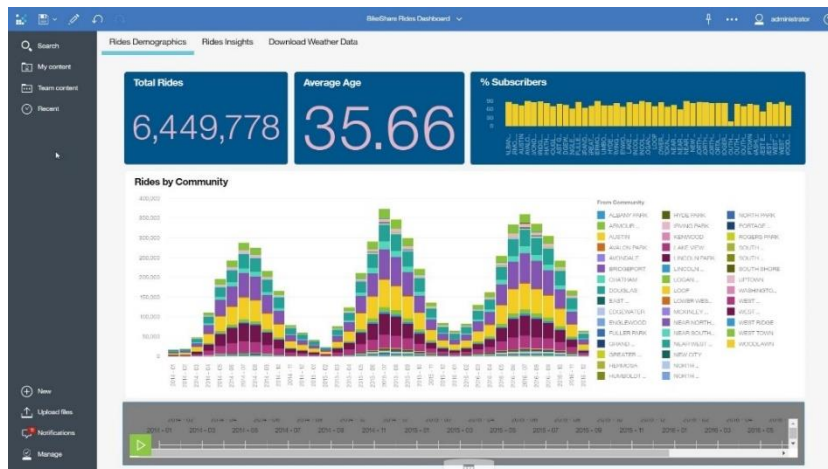
2.9.2. Cognos Analytics

A Cognos Analytics az IBM saját adatelemző és adatkezelő üzleti intelligencia eszköze. Ez egy olyan web alapú eszköz, ami olyan minták feldolgozására törekszik, amit egy ember nem biztos, hogy észrevesz. Számos önkiszolgáló része van, amit a 2.11. ábra is mutat, ezáltal a felhasználó könnyen feldolgozhatja, elemezheti és megoszthatja az adatokat és a vizualizációkat, ez azt jelenti, hogy nincsen szükség adatelemzői tudásra a riportok készítéséhez. [30]

Adatokat importálhatjuk CSV fájlokból vagy táblázatokból, de használhatunk felhőalapú vagy helyszíni adatforrást is, beleértve az SQL adatbázisokat. [31] A Cognos által használt mesterséges intelligencia segít abban, hogy az adatok egyetlen megbízható adatmodulba kerüljenek. Ezt percek alatt elkészíti és elvégzi az adatok tisztítását is, valamint ajánlásokat készít az adatok összekapcsolására és keverésére. Könnyen létre lehet hozni interaktív, dinamikus irányítópultokat. Csak be kell húzni az adatokat a vizualizációba, ha szükséges még lehet részletezni, de ezzel készen is van. A rendszer automatikusan a legjobb megjelenítést ajánlja fel az adatoknak megfelelően. Ezeket könnyedén meg tudja osztani másokkal, akár e-mailben is el tudja küldeni. [30]

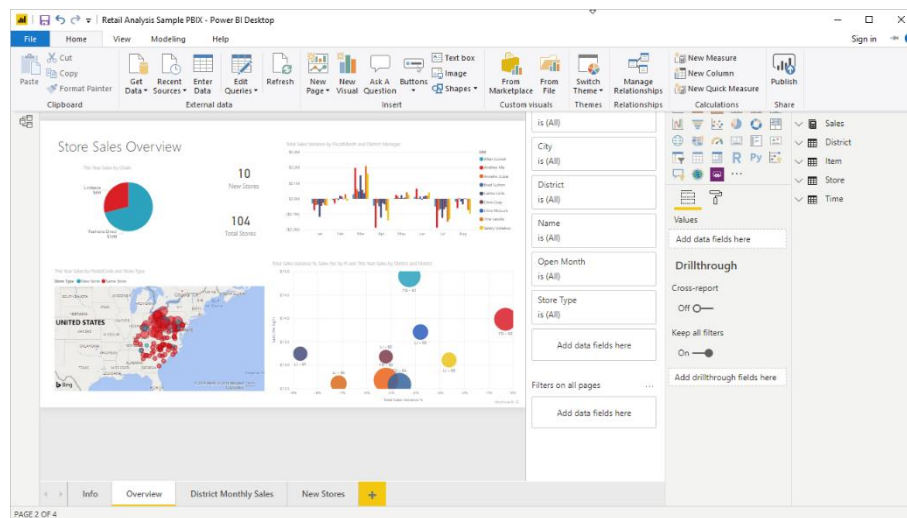
Rejtett minták felfedezésében is segít a mesterséges intelligencia, egyszerűen csak fel kell tenni neki angolul egy kérdést, és ő a választ egy vizualizációban megjeleníti. A szezonális trendek előrejelzése érdekében lehet idősoros modellezést alkalmazni. Ezen kívül személyre szabott megjelenítést is lehet készíteni, ahhoz, hogy napra kész legyen lehet automatikus frissítést is beállítani. [31]

Nagy előnye, hogy van egy 30 napos próba verziója és van mobil alkalmazása is, így mindig azonnal kéznél van. [31]

2.11. *ábra* – *Cognos* [32]

2.9.3. Power BI

A Power BI a Microsoft terméke, ezért könnyen összekapcsolható más Microsoft termékkel. Valamint könnyebb a használata, mert hasonló a felülete, mint a többi terméknek, ezt mutatja a 2.12. ábra is. Több változatot lehet belőle vásárolni, de van egy ingyenes próba verziója is. Tudunk az Office 365 eszközein kívül is importálni akár egy CSV fájlt is, de akár a Big Data-ból is. A különlegessége a Power BI-nak, hogy van egy platform, ahol meg lehet osztani a többiekkel a riportokat, és ők is tudják ezt szerkeszteni, ezáltal egyszerre többen tudják szerkeszteni. Ki tudjuk menteni PDF, JPG ként is. Mivel van telefonos applikációja is, ezért ott is megtekinthetők. A Microsoft AI segítségével sokkal gyorsabb az adatok tisztítása, előkészítése és az információk kinyerése. Ezen kívül gépi tanulási modelleket is létre lehet hozni. Valós idejű adatokat is lehet elemezni vele, idő függvényében is. [33]

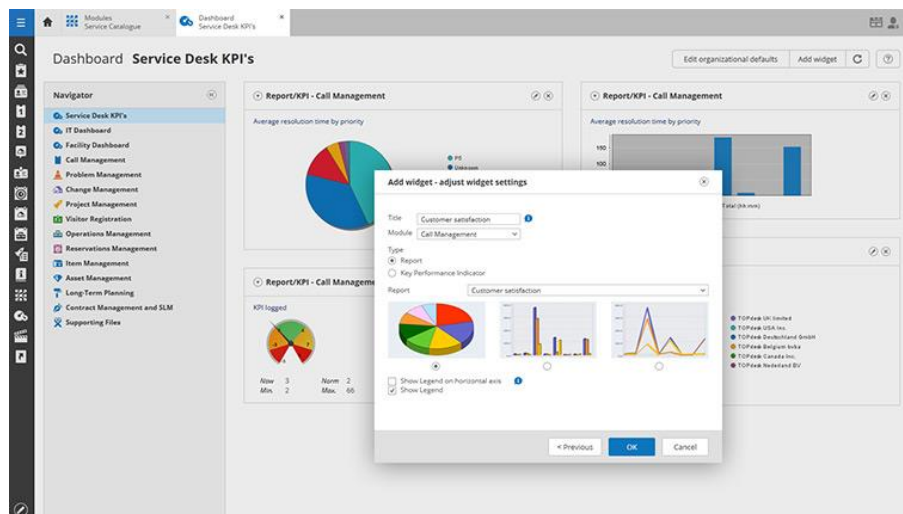


2.12. ábra - Power BI [34]

2.9.4. TOPdesk

A 2.13. ábra mutatja a TOPdesk riport készítő eszközt, ami felhasználóbarát és könnyen kezelhető, ugyanis akár pár kattintással lehet elemzést készíteni. Lehet választani a szűrési szempontok között például osztály, telephely, de készíthetünk vezetői riportokat is például az összegek, bevételek százalékos eloszlás függvényében. Segítségünkre lehet a riportvarázsló is, ami négy lépésben tudja elkészíteni a riportot. Bármelyik riport, beszámoló szerkeszthető azután is, hogy már egyszer megcsináltuk, így nem szükséges minden alkalommal újat csinálni, elég a régit módosítani. Lehetséges az is, hogy egy folyamatos frissítést állítunk be a riportra, hogy mindig a legaktuálisabb adatokat lássuk. Valamint automatikus küldést is tudunk beállítani, hogy az ügyfél is mindig értesüljön a változásról, vagy számára is adunk hozzáférési jogot. A riportokhoz tudunk KPI-ket (Key Performance

Indicator) csatolni, ennek köszönhetően valós időben láthatóak a változások. Be lehet állítani egyedi mutatókat, időintervallumokat vagy optimális értékeket is az összehasonlítás érdekében. A riportokat ki tudjuk exportálni Excel vagy PDF-ként is. [35]



2.13. ábra – TOPdesk [35]

3. KUTATÁSOK ÖSSZEGZÉSE

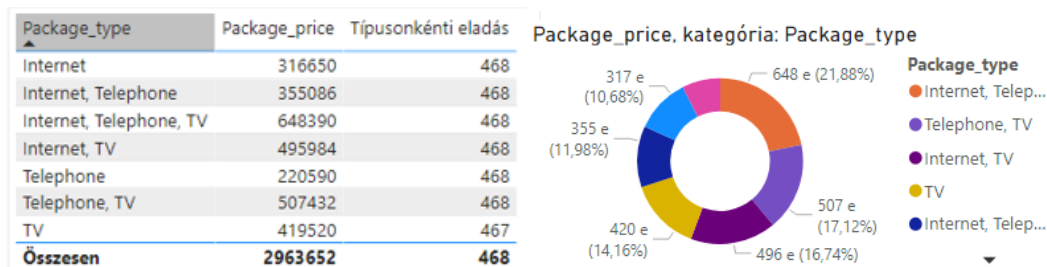
Az irodalomkutatás során megismert alkalmazások és rendszerekből ki kell választanom részenként egyet, amit én is fogok használni. Ez a relációs adatbáziskezelő rendszerek közül az Oracle lesz, mert ez a legismertebb az összes közül, sok oktató anyag van hozzá, könnyen kezelhető és felhasználó barát, valamint ezzel az alkalmazással van a legtöbb korábbi tapasztalatom. Az ETL eszközök közül a Visual Studiohoz integrálható SSIS-t választottam. Azon kívül, hogy már ezt az eszközt is használtam, ez az, ami ingyenes és a legtöbb funkcióval bír. Jól importálható bele Excel és CSV fájl is, ami számomra fontos volt, valamint lehet benne historikus adatokat is készíteni, ami szintén számított. A riport készítéshez a Power BI eszközt választottam. Ennek van egy ingyenes verziója, ami minden olyan funkciót tartalmaz amire nekem szükségem van. Könnyen kezelhető a már korábbi Office365 tapasztalatok miatt, így sokkal gyorsabb munkát eredményez, nem szükséges olyan hosszú tanulási folyamat.

4. SPECIFIKÁCIÓ

Az Oracle rendszerben meg kell valósítani az adatok megfelelő importálását, és ezeknek az adatoknak a tárolását, úgy, hogy ne vesszen el adat és ne is duplikálódjon. Tudni kell az adatokat módosítani, törölni, összevonni, vagy transzformálni. Szükséges az is, hogy le lehessen kérdezni azt, hogy hány olyan előfizető van, aki egy bizonyos városban lakik és Internet – Telefon vagy Internet – TV csomagja van. Kell egy olyan lekérdezést is biztosítani, ami azt mondja meg, hogy mennyien vásároltak olyan csomagot, ami 5000 forintnál drágább, ezt csomagokra csoportosítva. Szükséges, hogy ezt a lekérdezést optimalizálni lehessen, így növelve a hatékonyságot, és csökkentve a futásidőt. Tudni kell kiexportálni az adatokat, akár CSV fájlba, akár Excel fájlba.

Az SSIS rendszerben meg kell valósítani az ETL folyamat összes lépését. Be kell olvasni a fájlokat, úgy, hogy azok ne sérüljenek. Majd el kell végezni a transzformációkat az adatokon, például a típus módosítást. Tudni kell kezelni a historizálni kívánt adatokat is, olyan módon, hogy egy új táblában menti el a régieket, és az adott táblában csak az aktuális lesz, valamint, hogy az adott táblában mellé írja, hogy az mettől meddig érvényes. Majd a már kész adatainkat be kell tudni tölteni a csillag sémába, úgy, hogy először a dimenzió majd a tény tábla töltődjön. Innen is tudni kell exportálni a táblákat, dimenziókat.

A Power BI rendszerben meg kell valósítani az adatok beolvasását, úgy, hogy azok sértetlenül kerüljenek be. Tudni kell szűréseket végrehajtani, és ezek alapján vizualizálni. Szükséges megjeleníteni az eladott csomagok számát, csomagokra lebontva, olyan módon ahogy ezt a 4.1. ábra mutatja. Tudni kell a napi eladásokat összegezni, hogy ezáltal látható legyen, hogy melyik időszak a legforgalmasabb egy évben. Össze kell hasonlítani a városokon belüli eladásokat, figyelve ezzel azt, hogy hol kevésbé és hol jobban keresett a szolgáltatás. Ezen kívül meg kell jeleníteni a megyéken belüli üzletek számát, hogy tudjuk merre kell terjeszkedni. Tehát ki kell tudni választani, azt, hogy melyik táblákból, melyik oszlopokat szeretnénk látni, és milyen típusú diagramon. Valamint ezeket is tudni kell kiexportálni.



4.1. ábra - Eladott csomagok

5. RENDSZERTERV

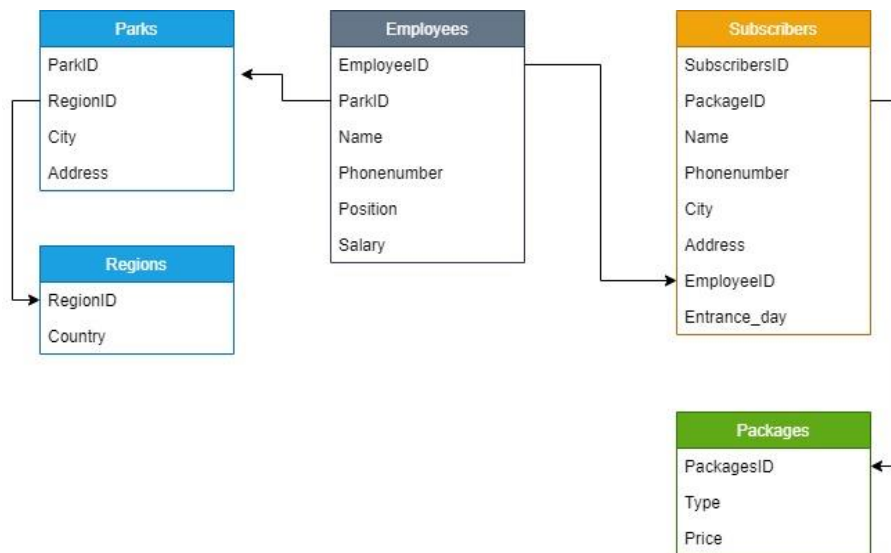
Azért választottam azt, hogy egy telekommunikációs cég adatbázisát reprezentálom, mert a mai világban elengedhetetlen a kommunikáció, akár online akár offline. Erre a kommunikációra pedig ezek a cégek nyújtanak megoldást. Ha arra gondolunk, hogy az internetek keresztül tájékozódunk, vagy csak tartjuk szeretteinkkel, barátainkkal a kapcsolatot, vagy éppen a tévében nézzük a híreket, vagy felhívjuk a nagymamát. Mindhez elengedhetetlen, hogy meglegyen ez a hálózat, és lehetőségünk is legyen használni.

5.1. Táblák

A következő táblákra lesz szükségem. Egy regions nevű táblára, ami a különböző megyéket fogja tartalmazni. Egy parks táblára, ami a megyéken belüli telephelyeket tárolja. Egy employees táblára, ami a dolgozók adatit tartalmazza. Egy subscribers táblára, ami az előfizetők adatait tartalmazza. Egy packages nevű táblára, ami az előfizethető csomagokat tartalmazza.

5.2. UML-diagram

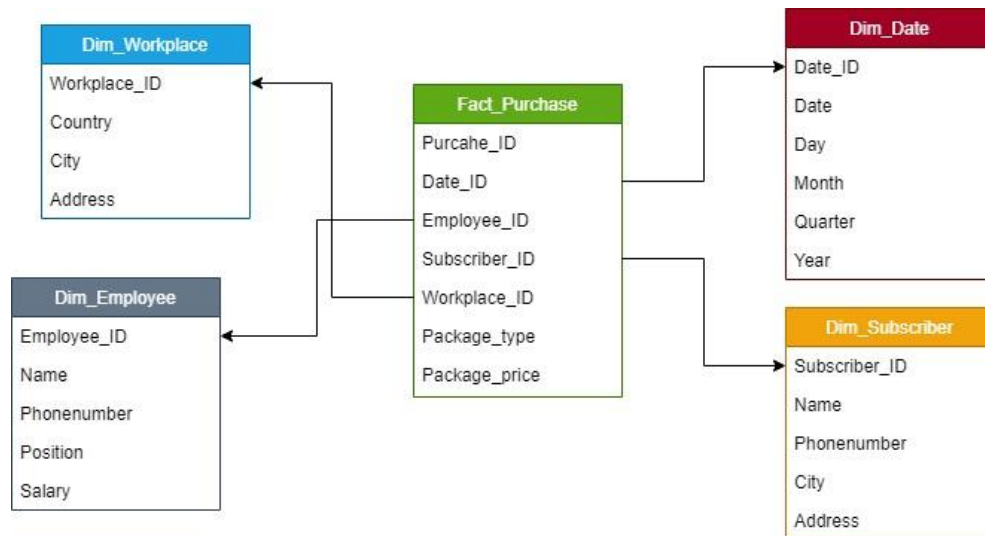
Ez a fajta diagramm a rendszer fizikai és logikai felépítését mutatja be. A felső részben látszik az osztály neve, a középső szakaszban pedig az osztály attribútumai. [36] A saját UML-diagramomat a 5.1. ábra mutatja.



5.1. ábra - UML-diagram

5.3. Csillag séma

A 5.2. ábra mutatja be, azt a csillag sémát, amit a szakdolgozatom során szeretnék kialakítani.



5.2. ábra - Csillag séma

5.4. Munkaterv

Először a megfelelő adatok, táblák összeállítása a cél, ugyanis enélkül nem fognak működni a további folyamatok. Ezután ezeknek az adatoknak a betöltése történik a kiválasztott relációs adatbázis kezelőbe, ahol lekérdezéseket fogok végezni és ezeket megpróbálom optimalizálni, a minél nagyobb hatékonyság érdekében. Majd ezeket az adatokat importálom az SSIS-be is, hogy el tudjam rajtuk végzeni az előkészítést. Ezután, ha szükséges transzformációt végzek, és elkészítem a historikus beállításokat, majd betöltöm a csillag sémát. Ha ez készen van kiimportálom az adatokat, és betöltöm a Power BI-ba ahol különböző riportokat készítek.

6. MEGVALÓSÍTÁS

6.1. Táblák definiálása

Az egész megvalósítás az adatok beszerzésével, generálásával kezdődik. Én most 4 táblát választottam ki. Ezek a Parks, Employees, Subscribers, Packages táblák. Minden táblában található egy ID, ami az egyedi azonosításra szolgál.

A Parks táblám egy olyan adathalmaz, amiben megtalálhatóak az üzletek adatai. Ilyen a megye, város, irányítószám és a pontos cím.

Az Employees táblában a dolgozókra vonatkozó személyes és céges adatok találhatóak. Ezek a név, telefonszám, üzlet ID, munkakör és fizetés.

A Subscribers tábla az ügyfelekhez tartozó adatokat írja le. Ilyen az ügyfél neve, születési dátuma, telefonszáma, belépésének dátuma, csomagjának az azonosítója, egy dolgozói azonosító, megye és város.

A Packages tábla pedig az eladható csomagjainkat tartalmazza. A csomagok nevével, típusával és árával.

6.2. Táblák szerkezete

Táblanév: Parks

- Hogyan állítottuk elő az adatokat?
Az adatokat egy cégnek a honlapjáról töltöttem le, majd a pontos címet anonimizáltam.
- Milyen módosításokat kellett eszközölni?
Anonimizálni kellett egy oszlopot.
- Milyen formátumban van az adatokat tartalmazó fájl?
.csv
- Hány rekordot tartalmaz a tábla?
193

Táblanév: Employees

- Hogyan állítottuk elő az adatokat?
Az adatokat véletlenszerűen generáltam.
- Milyen módosításokat kellett eszközölni?
Nem volt szükséges módosítást végrehajtani az adatokon.

- Milyen formátumban van az adatokat tartalmazó fájl?
.csv
- Hány rekordot tartalmaz a tábla?
7537

Táblanév: Subscribers

- Hogyan állítottuk elő az adatokat?
Az adatokat véletlenszerűen generáltam.
- Milyen módosításokat kellett eszközölni?
Nem volt szükséges módosítást végrehajtani az adatokon.
- Milyen formátumban van az adatokat tartalmazó fájl?
.csv
- Hány rekordot tartalmaz a tábla?
262649

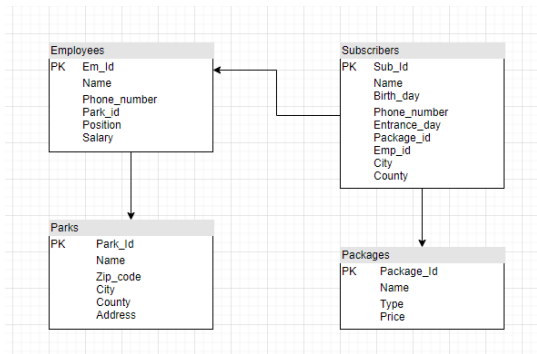
Táblanév: Packages

- Hogyan állítottuk elő az adatokat?
Az adatokat saját magam találtam ki.
- Milyen módosításokat kellett eszközölni?
Nem volt szükséges módosítást végrehajtani az adatokon.
- Milyen formátumban van az adatokat tartalmazó fájl?
.csv
- Hány rekordot tartalmaz a tábla?
26

6.3. Adatmodell

2.1. ábra bemutatja azt az adatmodellt, amit a draw.io eszköz segítségével készítettem el. A képen láthatóak a táblák és a kapcsolataik. Minden táblában található egy ID ez egy egyedi azonosító, ami a `primary_key`. Ennek az értéke csak egyszer szerepelhet a táblában, nem lehet duplikátum.

Vannak benne még `foreign_key`-ek, amiket arra használunk, hogy a táblákat össze tudjuk kapcsolni egymással. Ilyen idegen kulcs az `Employees.Park_Id`, ami a `Parks.Park_id`-ra mutat, és a `Subscribers.package_id`, ami a `packages.Package_id`-ra mutat. Ezen kívül még egy kapcsolat látható, az pedig az `Employees` és a `Subscribers` tábla között van, ahol a két `Emp_Id` van összekötve. Ezáltal bármelyik táblából el tudunk jutni a többibe.



6.1. ábra - Táblák

6.4. Lekérdezés optimalizálás

Amennyiben szeretnénk információhoz jutni, lekérdezéseket kell készítenünk. Ezeknek a futásideje nagyon változó és nagyban befolyásolható. Azzal, hogy egy állandó kapcsolatot alakítunk ki a táblák között már tudjuk gyorsítani a lekérdezésünket. Ezt minden esetben meg kell tenni. Ezen kívül tudunk indexeket készíteni és partícionálni is, ami szintén a segítségünkre lehet. Ezek olyan módszerek, amiket nem szükséges minden alkalommal megcsinálni. Elegendő egyszer beállítani és utána már nem is kell foglalkozni vele, mert automatikus figyelembe fogja venni az alkalmazásunk.

6.4.1. Lekérdezés

A lekérdezés során azt szeretném megtudni, hogy hány olyan ügyfél van, aki a 2-es Park_ID helyén lakik és internet-telefon vagy internet-TV csomagja van? A 6.2. ábra mutatja a pontos SQL utasítást.

```
SELECT COUNT(DISTINCT(subscribers.subscriber_id))
FROM subscribers
INNER JOIN parks ON (parks.city = subscribers.city)
INNER JOIN packages ON (subscribers.package_id = packages.package_id)
WHERE parks.park_id = 2
AND (packages.type = 'Internet, TV' or packages.type = 'Internet, Telephone');
```

6.2. ábra – SQL utasítás

OPERATION	OBJECT_NAME	CARDINA...	COST
SELECT STATEMENT		1	518
SORT (AGGREGATE)		1	
VIEW	VW_DAG_0	8	518
HASH (GROUP BY)		8	518
HASH JOIN (SEMI)		8	517
Access Predicates			
SUBSCRIBERS.PACKAGE_ID=PACKAGES.PACKAGE_ID			
NESTED LOOPS		28	514
TABLE ACCESS (BY INDEX ROWID)	PARKS	1	1
INDEX (UNIQUE SCAN)	PARKS_PK	1	0
Access Predicates			
PARKS.PARK_ID=2			
TABLE ACCESS (FULL)	SUBSCRIBERS	28	513
Filter Predicates			
PARKS.CITY=SUBSCRIBERS.CITY			
TABLE ACCESS (FULL)	PACKAGES	6	3
Filter Predicates			
OR			
PACKAGES.TYPE='Internet, TV'			
PACKAGES.TYPE='Internet, Telephone'			

6.3. ábra – Lekérdezés költsége

Lekérdezés elemzése a 6.3. ábra alapján:

1. A Filter Predicates megnézi az Internet, Tv vagy az Internet, Telefon csomagokat.
2. A következő lépést szintén a Filter Predicate végzi el, ahol összekapcsolja a két táblában a városokat.
3. Unique scan segítségével összehasonlítjuk a parknak az id-ját.
4. Az Access Predicates összekapcsolja a két táblában a csomagok id-ját.
5. Majd a HASH(GROUP BY) függvény ezeket összegzi, és visszaadja a végeredményt.

Lekérdezés optimalizálását a 6.4. ábra alapján:

Egy-egy non-unique index lett rakva a csomag típusára (Packages.type) és az előfizetők városaira is (Subscribers.city). Ezzel látható, hogy a költségünk nagy mértékben lecsökkent. Ez annak köszönhető, hogy range scan-t tud használni a rendszer. Így csupán 33 a költségünk az 518 helyett.

OPERATION	OBJECT_NAME	CARDINA...	COST
SELECT STATEMENT		1	33
SORT (AGGREGATE)		1	
VIEW	VW_DAG_0	8	33
HASH (GROUP BY)		8	33
HASH JOIN (SEMI)		8	32
Access Predicates			
SUBSCRIBERS.PACKAGE_ID=PACKAGES.PACKAGE_ID			
NESTED LOOPS		28	30
TABLE ACCESS (BY INDEX ROWID)	PARKS	1	1
INDEX (UNIQUE SCAN)	PARKS_PK	1	0
Access Predicates			
PARKS.PARK_ID=2			
TABLE ACCESS (BY INDEX ROWID BATCHED)	SUBSCRIBERS	28	29
INDEX (RANGE SCAN)	SUBSCRIBERCITYI	28	2
Access Predicates			
PARKS.CITY=SUBSCRIBERS.CITY			
INLIST ITERATOR			
TABLE ACCESS (BY INDEX ROWID BATCHED)	PACKAGES	6	2
INDEX (RANGE SCAN)	PACKAGETYPEPI	1	1
Access Predicates			
OR			
PACKAGES.TYPE='Internet, TV'			
PACKAGES.TYPE='Internet, Telephone'			

6.4. ábra – Optimalizált lekérdezés

6.5. Adattárház specifikáció

A cégen belüli leghatékonyabb fejlődéshez elengedhetetlen, hogy vizuálisan is meg legyenek jelenítve a legfontosabb adatok.

A fő kérdéskörök:

- A dolgozók fizetésének összehasonlítása
- Országokon belüli üzletek eloszlása
- Csomagok népszerűsége
- Városokon belüli ügyfelek száma
- Új ügyfelek eloszlása

6.6. Dimenzionális modell

6.6.1. Dimenzió táblák

Dim_Workplace

A cég üzletei, illetve ügyfél fogadására alkalmas telephelyeinek pontos címét tartalmazza.

Dim_Employee

A cég alkalmazottainak az adatait tartalmazza. Ilyen a személyes adatok és az is, hogy pontosan hol dolgozik.

Dim_Date

A dátumokat tartalmazza. Évre, negyedévre, hónapra, napra lebontva.

Dim_Subscriber

Az ügyfelek, felhasználók adatait tartalmazza. Ezek a személyes adatai.

6.6.2. Ténytábla

Itt találhatóak a kulcsok a dimenzió táblákhoz, valamint az, hogy az illető pontosan milyen csomagot vásárolt és mennyiért.

6.6.3. Mapping

Létrehoztam egy táblázatot Excelben annak érdekében, hogy az adatok változását megfelelően nyomon lehessen követni, és átlátható legyen. Ezt a táblázatot a 6.5. ábra mutatja be.

Ebben a fájlban megtalálható az össze oszlop és azoknak a tulajdonságai. Ez fel van tüntetve minden dimenzióban. Megtalálható a forrás adatbázis, a forrás adatbázisban lévő táblának a neve, az oszlop neve és a típusa. Ezzel egy sorban van a cél adatbázis, tábla, oszlop és típus.

A 6.6. ábra mutatja, hogy a sorok végén kell-e valami plusz műveletet végrehajtani az áttöltés során. Ez lehet transzformálás vagy kulcs beállítás. Transzformálás alatt érthetünk átlagot, összeadást, kivonást vagy más oszlopokkal való összevonást is.

Source				Target				Transformation	Description
Database Name	Table Name	Column Name	Data Type	Database Name	Table Name	Column name	Data Type		
STA	STA_Employees	Employee_ID	VARCHAR(256)	DW	DW_Employees	Employee_ID	INT		
STA	STA_Employees	Park_ID	VARCHAR(256)	DW	DW_Employees	Park_ID	INT		
STA	STA_Employees	Name	VARCHAR(256)	DW	DW_Employees	Name	VARCHAR(128)		
STA	STA_Employees	Phonenumber	VARCHAR(256)	DW	DW_Employees	Phonenumber	VARCHAR(128)		
STA	STA_Employees	Birthday	VARCHAR(256)	DW	DW_Employees	Birthday	DATE		
STA	STA_Employees	Position	VARCHAR(256)	DW	DW_Employees	Position	VARCHAR(128)		
STA	STA_Employees	Salary	VARCHAR(256)	DW	DW_Employees	Salary	INT		
STA	STA_Subscribers	Subscriber_ID	VARCHAR(256)	DW	DW_Subscribers	Subscriber_ID	INT		
STA	STA_Subscribers	Package_ID	VARCHAR(256)	DW	DW_Subscribers	Package_ID	INT		
STA	STA_Subscribers	Employee_ID	VARCHAR(256)	DW	DW_Subscribers	Employee_ID	INT		
STA	STA_Subscribers	Name	VARCHAR(256)	DW	DW_Subscribers	Name	VARCHAR(128)		
STA	STA_Subscribers	Phonenumber	VARCHAR(256)	DW	DW_Subscribers	Phonenumber	VARCHAR(128)		
STA	STA_Subscribers	City	VARCHAR(256)	DW	DW_Subscribers	City	VARCHAR(128)		
STA	STA_Subscribers	Address	VARCHAR(256)	DW	DW_Subscribers	Address	VARCHAR(128)		
STA	STA_Subscribers	Entrance_day	VARCHAR(256)	DW	DW_Subscribers	Entrance_day	DATE		
STA	STA_Regions	Region_ID	VARCHAR(256)	DW	DW_Regions	Region_ID	INT		
STA	STA_Regions	Country	VARCHAR(256)	DW	DW_Regions	Country	VARCHAR(128)		
STA	STA_Parks	Region_ID	VARCHAR(256)	DW	DW_Parks	Region_ID	INT		
STA	STA_Parks	City	VARCHAR(256)	DW	DW_Parks	City	VARCHAR(128)		
STA	STA_Parks	Address	VARCHAR(256)	DW	DW_Parks	Address	VARCHAR(128)		
STA	STA_Parks	Park_ID	VARCHAR(256)	DW	DW_Parks	Park_ID	INT		
STA	STA_Packages	Package_ID	VARCHAR(256)	DW	DW_Packages	Package_ID	INT		
STA	STA_Packages	Type	VARCHAR(256)	DW	DW_Packages	Type	VARCHAR(128)		
STA	STA_Packages	Price	VARCHAR(256)	DW	DW_Packages	Price	MONEY		

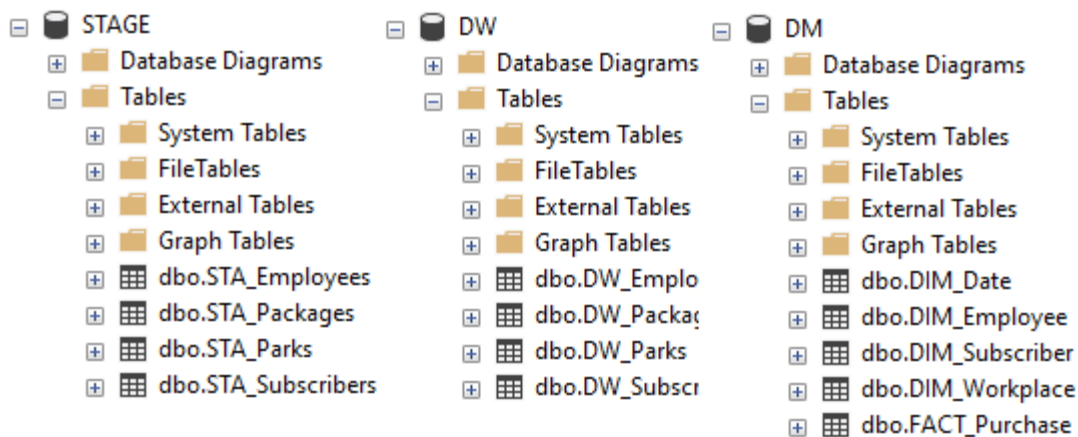
6.5. ábra - Mapping

Source				Target				Description
Database Name	Table Name	Column Name	Data Type	Database Name	Table Name	Column name	Data Type	
DW	DW_Employees	Employee_ID	INT	DM	DIM_Employee	Employee_ID	INT	Key
DW	DW_Employees	Name	VARCHAR(128)	DM	DIM_Employee	Name	VARCHAR(128)	
DW	DW_Employees	Phonenumber	VARCHAR(128)	DM	DIM_Employee	Phonenumber	VARCHAR(128)	
DW	DW_Employees	Birthday	DATE	DM	DIM_Employee	Birthday	DATE	
DW	DW_Employees	Position	VARCHAR(128)	DM	DIM_Employee	Position	VARCHAR(128)	
DW	DW_Employees	Salary	INT	DM	DIM_Employee	Salary	INT	
DW	DW_Subscribers	Subscriber_ID	INT	DM	DIM_Subscriber	Subscriber_ID	INT	Key
DW	DW_Subscribers	Name	VARCHAR(128)	DM	DIM_Subscriber	Name	VARCHAR(128)	
DW	DW_Subscribers	Phonenumber	VARCHAR(128)	DM	DIM_Subscriber	Phonenumber	VARCHAR(128)	
DW	DW_Subscribers	City	VARCHAR(128)	DM	DIM_Subscriber	City	VARCHAR(128)	
DW	DW_Subscribers	Address	VARCHAR(128)	DM	DIM_Subscriber	Address	VARCHAR(128)	
DW	DW_Subscribers	Entrance_day	DATE	DM	DIM_Subscriber	Entrance_day	DATE	
DW	DW_Regions	Country	VARCHAR(128)	DM	DIM_Workplace	Country	VARCHAR(128)	
DW	DW_Parks	City	VARCHAR(128)	DM	DIM_Workplace	City	VARCHAR(128)	
DW	DW_Parks	Address	VARCHAR(128)	DM	DIM_Workplace	Address	VARCHAR(128)	
DW	DW_Parks	Park_ID	INT	DM	DIM_Workplace	Workplace_ID	INT	Key
DW	DW_Packages	Package_ID	INT	DM	DIM_Purchase	Package_ID	INT	Key
DW	DW_Packages	Type	VARCHAR(128)	DM	DIM_Purchase	Package_Type	VARCHAR(128)	
DW	DW_Packages	Price	MONEY	DM	DIM_Purchase	Package_Price	MONEY	

6.6. ábra – Description

6.7. ETL folyamatok

Ahhoz, hogy egy adattárház fel tudjunk építeni szükséges az SSIS alkalmazás használata. Illetve 3 egymásra épülő réteg kialakítása, hogy minél hatékonyabb legyen az adat áttöltő folyamat. Így az első lépésként létrehoztam a 3 adatbázist a rétegekhez. Ezek lettek a STAGE, DW és a DM. A második lépés, hogy táblákat készítettem bele, ami a 6.7. ábra mutat, a megfelelő oszlopokkal és típusokkal, amit a mappingből ki lehetett olvasni.

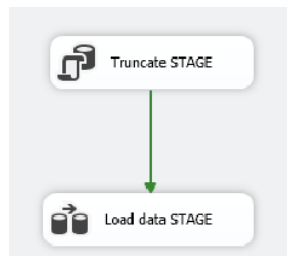


6.7. ábra - Adatbázisok

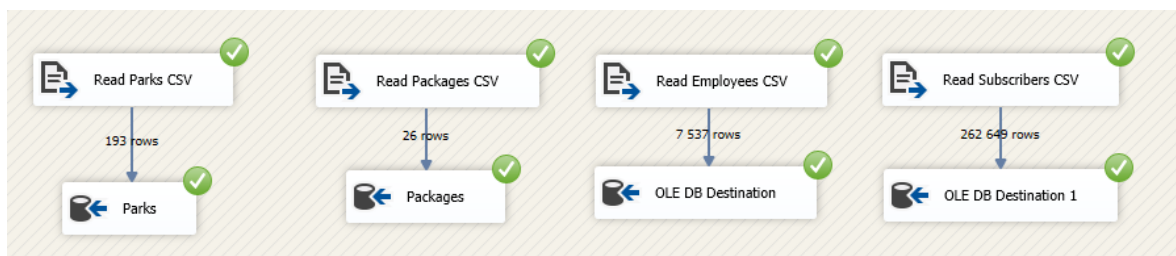
6.7.1. Extract réteg

Az extract az első réteg, aminek a lényege, hogy a forrás adatokat minden transzformáció nélkül betöltjük a STAGE adatbázisba. Ezt a lépés azért fontos mert így tudjuk a leghatékonyabban és a leggyorsabban feltölteni adatokkal a tábláinkat, ezt prezentálja a 6.8. ábra.

Ehhez szükséges, hogy a CSV fájlból kiolvassuk az adatokat és azokat betöltsük egy konkrét táblába, amihez az OLE DB Destination-t használtam, amit a 6.9. ábra mutat. Ezt minden táblánál meg kell tennünk, és minden adatnak VARCHAR(256) típusúnak kell lennie.



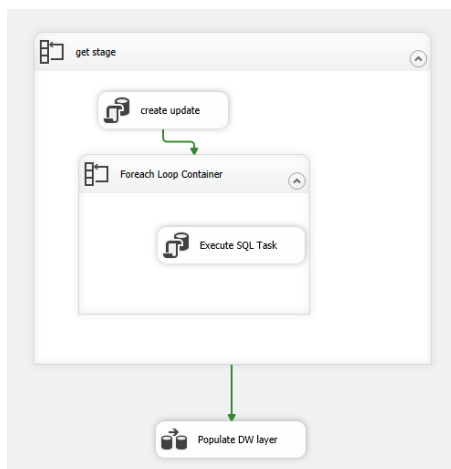
6.8. ábra - Extract



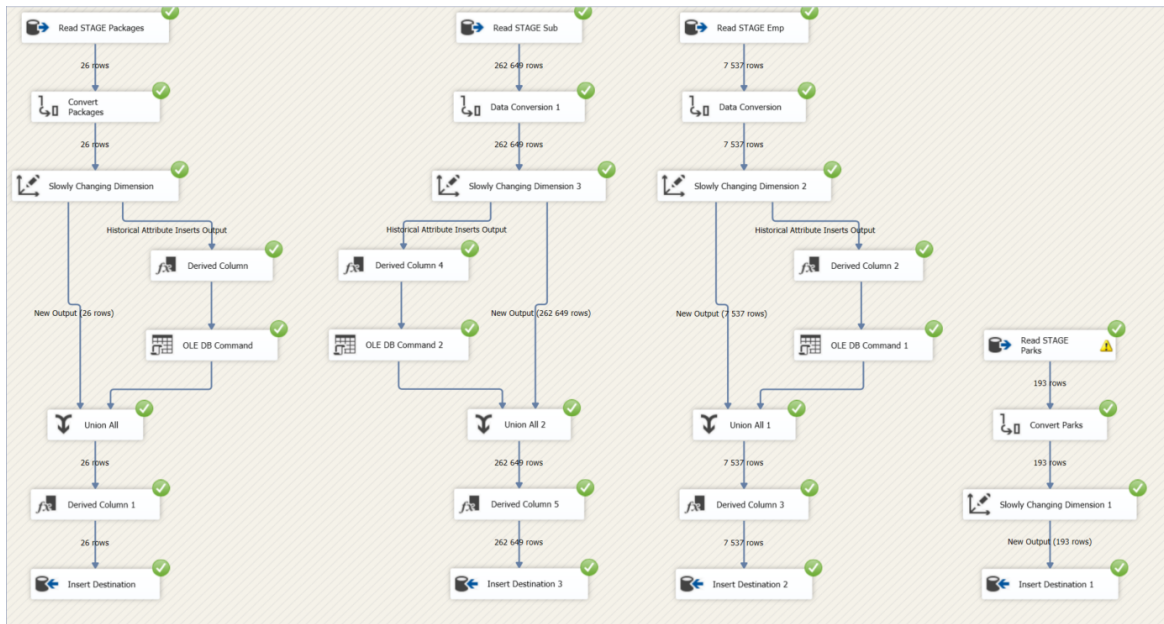
6.9. ábra - Load .CSV

6.7.2. Transform réteg

A második réteg a transform, amit a 6.10. ábra mutat be. Lényege, hogy a STAGE adatbázis tábláinak az adatait áttöltjük a DW adatbázis táblákba. Itt hajtjuk végre az adattípusok konvertálását, ha szükséges. Szükségem volt, VARCHAR, INT, MONEY és DATE típusra. Valamint itt tudjuk beállítani, hogy az adatokat nyomon szeretnék-e követni és hogy hogyan szeretnénk számontartani ezeket a változásokat. Erre a 6.11. ábra mutat példát.



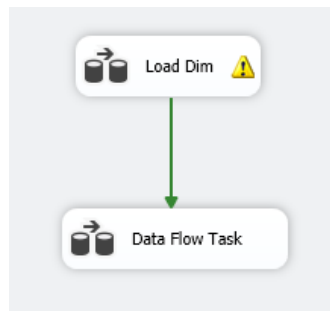
6.10. ábra - Transform



6.11. ábra – Read STAGE

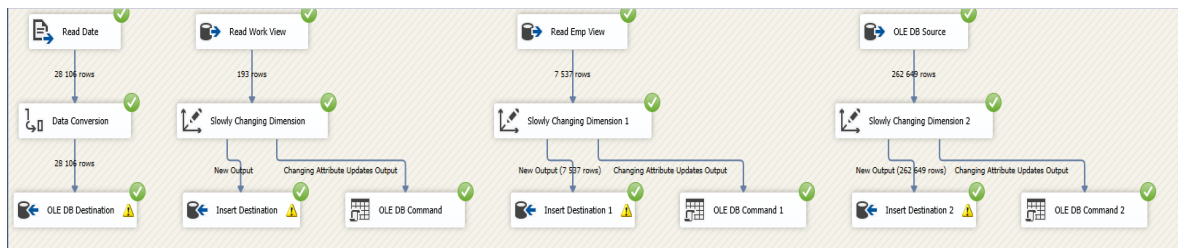
6.7.3. Load réteg

Az utolsó réteg a Load, ez arra szolgál, hogy a csillagsémát kialakítsuk és feltöltsük adatokkal, amit a 6.12. ábra prezentál. Ehhez elsőként szükséges volt létrehozni a nézeteket a dimenziókhoz és a ténytáblához.



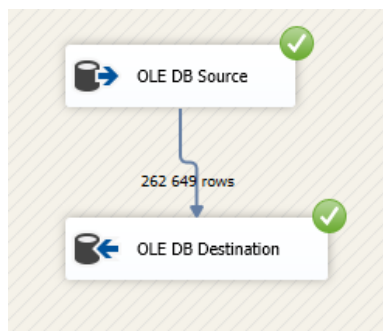
6.12. ábra - Load

A második lépés, amit a 6.13. ábra is mutat, hogy a nézetek alapján feltöltsük adatokkal a dimenzió táblákat. Itt kell eldönteni, hogy a historikus adatokkal mit szeretnénk csinálni, hogy megjelenjenek-e a dimenziókban vagy sem.



6.13. ábra - Load DIM

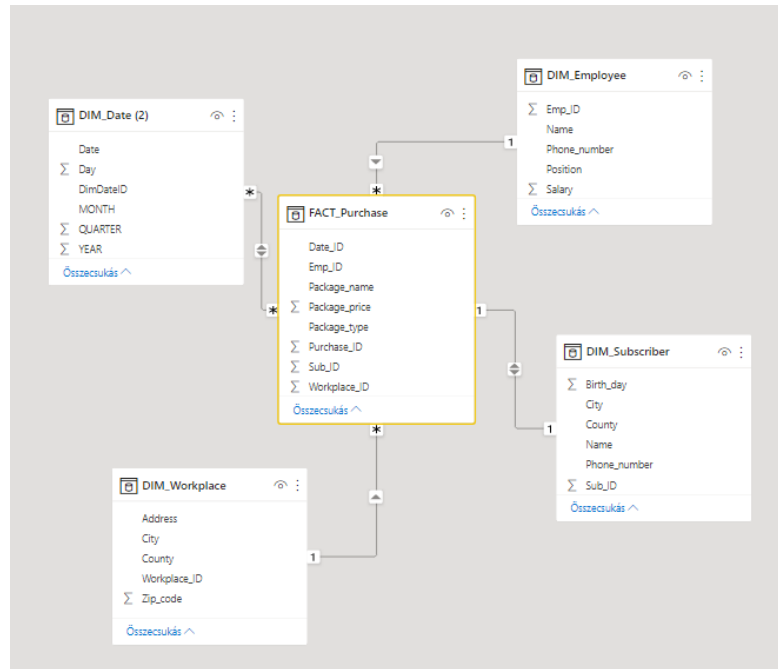
A dimenzió táblák feltöltése után következhet a ténytábla, ezt mutatja a 6.14. ábra. Ebbe is a nézet segítségével töltjük át az adatainkat.



6.14. ábra - Load FACT

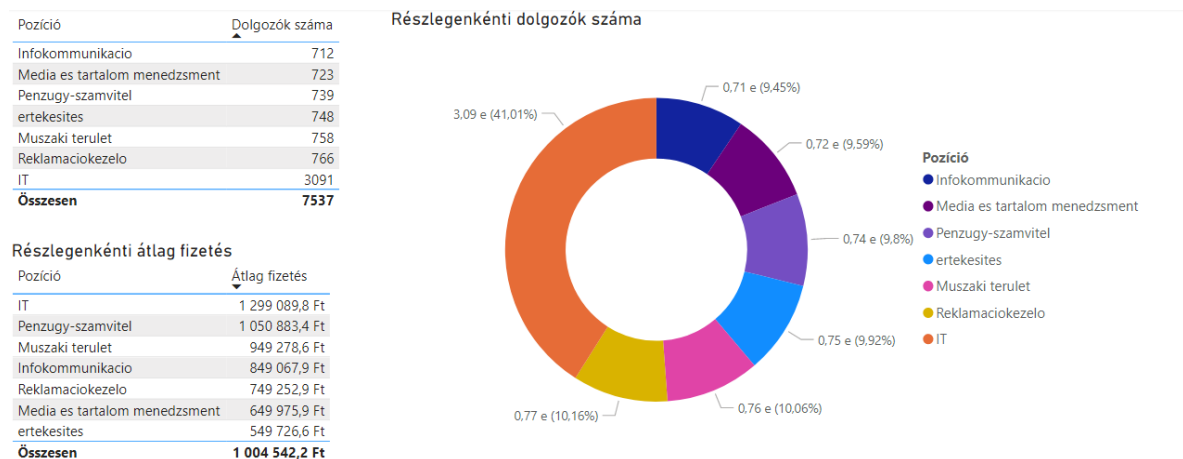
7. RIPORTKÉSZÍTÉS

Ebben a részben fogom bemutatni az adatok vizualizációját. Itt is szükséges volt betölteni a CSV fájlokat és kialakítani a kapcsolatokat, hogy megfelelő csillagsémát tudjak használni, amit a 7.1. ábra szemléltet.



7.1. ábra – Power Bi

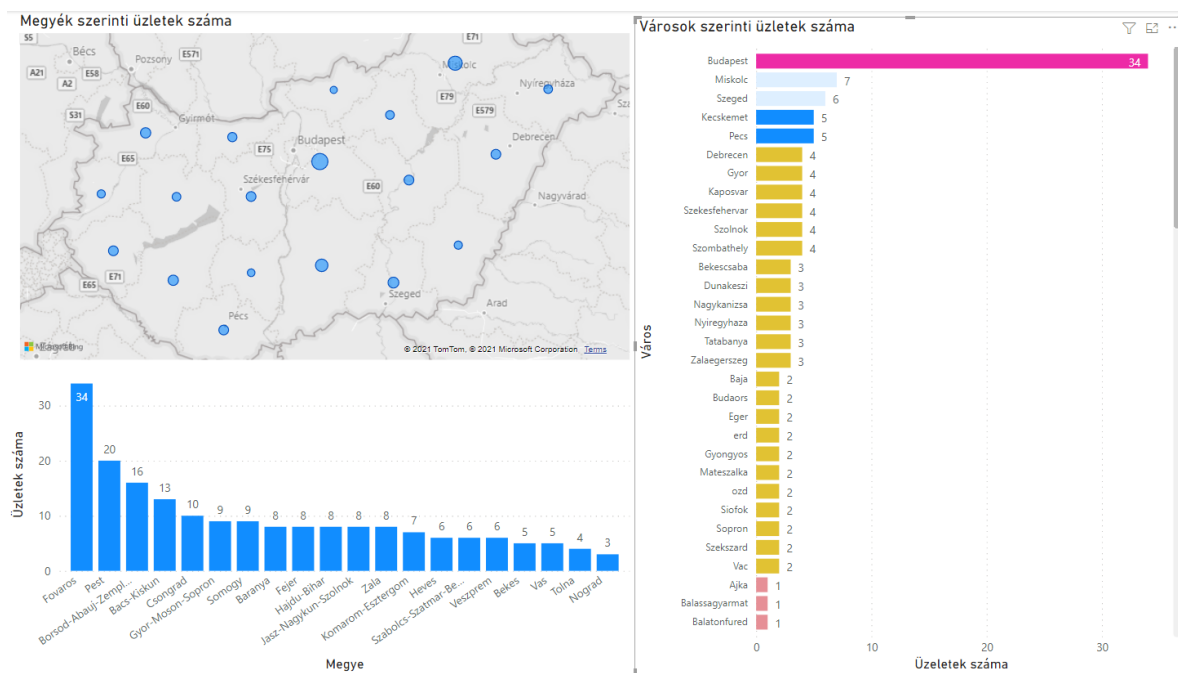
7.1. A dolgozók fizetésének összehasonlítása



7.2. ábra - Fizetések összehasonlítása

A 7.2. ábra mutatja, hogy egyes munkakörökben hány ember dolgozik. Emellett van számítva egy részlegenkénti átlagfizetés. Látható, hogy az átlagfizetés több mint a duplája is lehet az egyes munkakörök között. A fánk diagrammon láthatjuk, hogy hogyan oszlanak el a dolgozók a részlegek között. Mindegyik diagrammon tudunk szűrni, amivel a többen is változásokat érünk el, ezáltal sokkal könnyebb megtalálni az összetartozó részleteket.

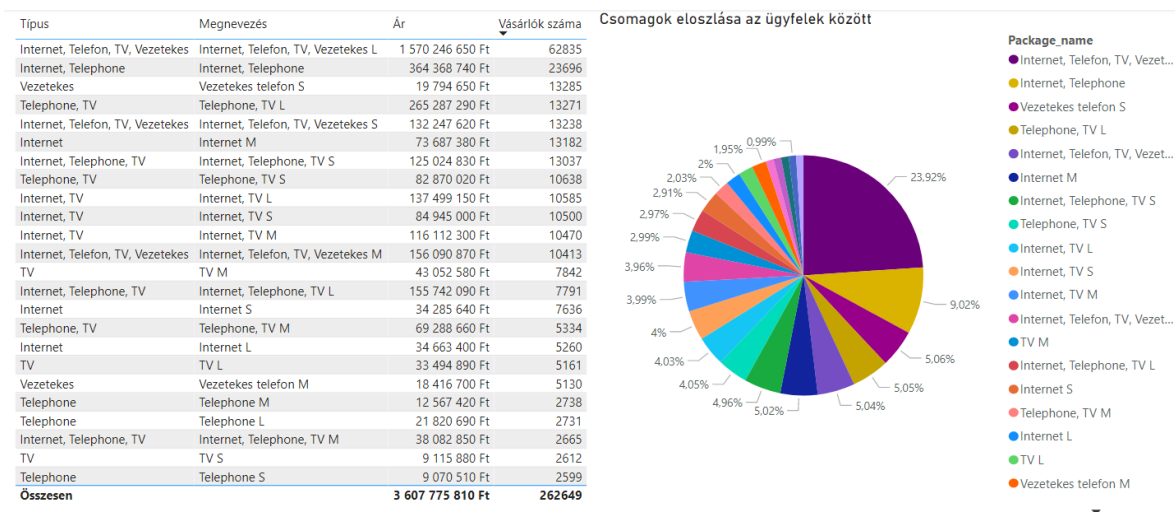
7.2. Országokon belüli üzletek



7.3. ábra - Üzletek csoportosítása

Ez az 7.3. ábra 7.2. ábra arra ad választ, hogy melyik megyében, illetve városban, hány üzlete van a cégnek. A térképen jól látható, hogy az egész országban vannak üzletek. A Főváros és Pest megyék külön kategóriába tartozik. Így látható, hogy Budapesten van a legtöbb üzlet, több mint másfélszer annyi, mint Pest megyében. A második város Miskolc, de még itt is csak ötöd annyi üzlet van, mint a Fővárosban.

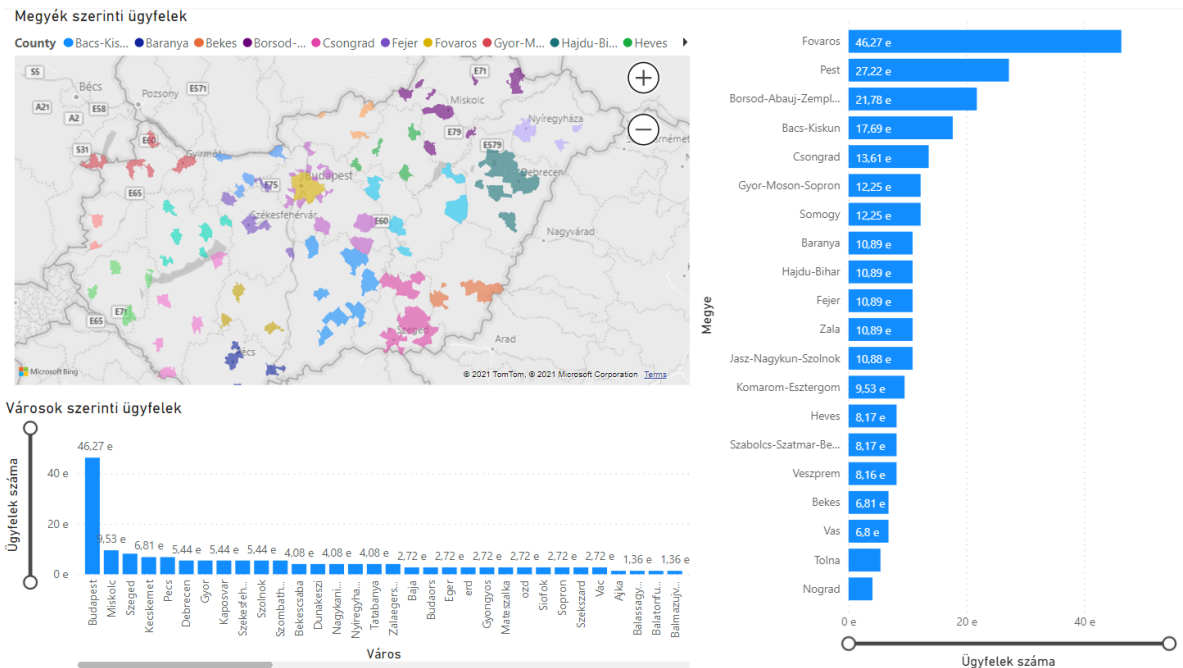
7.3. A csomagok népszerűsége



7.4. ábra - Csomagok népszerűsége

A 7.4. ábra alapján az látható, hogy a csomagok aránya eléggé eltérő. Minél komplexebb, nagyobb, jobb csomagot szeretnénk, annál drágább is lesz, de annál nagyobb csoportos kedvezményt adunk rá. Azt sikerült kimutatni, hogy a legnagyobb csomag a legnépszerűbb. Ezáltal azt mondhatjuk, hogy ha csoportos kedvezményt adunk az ügyfeleknek az kifizető. Ugyanis ebből a csomagból van a legnagyobb bevételünk is. Ezért látható a fánk diagrammon, hogy ekkora eltérés a cikkek arányában.

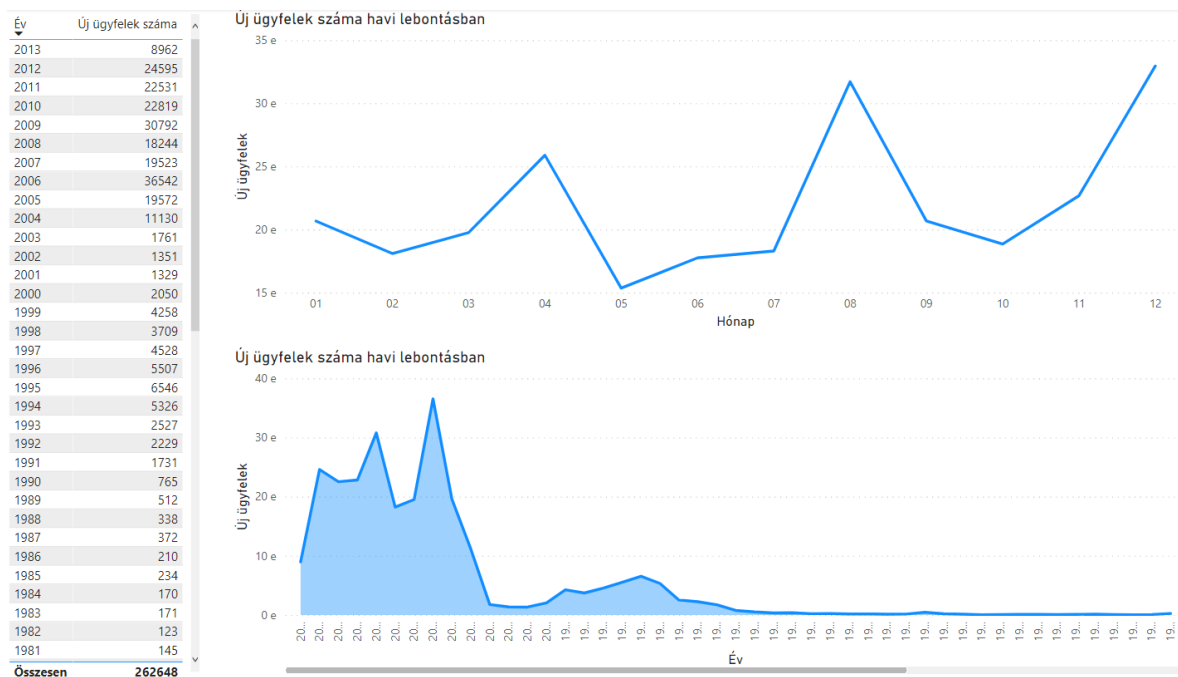
7.4. Városokon belüli ügyfelek száma



7.5. ábra – Ügyfelek csoportosítása

A 7.5. ábra mutatja, hogy minden nagyobb városban és körzetében jelen van a cég. Ennek ellenére Budapest kimagaslik az ügyfelek számában, míg itt több mint 45 ezer ügyfél van, a második helyen álló Miskolcon kevesebb mint 10 ezer. Ez természetesen amiatt is van, hogy a Fővárosban laknak a legtöbben, és itt a legelterjedtebb a telekommunikációs eszközök használata.

7.5. Új ügyfelek eloszlása



7.6. ábra - Új ügyfelek

A 7.6. ábra bemutatja, hogy 2013-as év végéig évente mennyit nőtt az ügyfeleink száma. A 2000-es évektől egy nagyobb ugrás látható a diagrammon. Valamint található egy vonaldiagramm azzal kapcsolatban, hogy melyik hónapban hányan csatlakoztak. Ebből leolvashatjuk, hogy inkább tavasszal és ősszel váltanak az emberek, valamint decemberben, a karácsony előtti időben.

8. TOVÁBBFEJLESZTÉSI LEHETŐSÉGEK

Az eddig kiépített adattárház jó alapot képez arra, hogy azt tovább lehessen bővíteni. A bővítése történhet mező szinten vagy tábla szinten is. Természetesen ez von maga után némi módosítást, de nem szükséges az egészet újra kezdeni.

Új táblának be lehetne a rakni a panaszokat, hogy kitől mikor és milyen panasz érkezett. Valamint, hogy hogyan orvosoltuk a problémáját, esetleg szerelő ment ki, vagy csak telefonos segítséget nyújtottunk, eszközt kellett-e cserélni.

Ezen kívül az ügyfelekhez lehetne hozzárendelni eszközöket, amiket megvásároltak, akár mobil telefont, tabletet, számítógépet, TV-t. Azt is számon lehetne tartani, hogy a megvásárolt készüléket kifizette-e vagy, hogy hányadik részletnél tart.

Ezen kívül lehetne egy táblánk a telefonálásokról, azokkal az adatokkal, hogy ki, kit és mikor hívott. Valamint, hogy mennyi ideig beszéltek.

Figyelni lehetne az interneten eltöltött idő mennyiségét is, annak érdekében, hogy lássuk, hogy milyen szociális felületeket használ. Ezáltal megtudnánk, hogy személyre szabottan hol érdemes számára hirdetni.

Ezekkel az új táblákkal még több riportot lehetne készíteni. Például a telefonálások összegzésével tudnánk, számára megfelelőbb csomagot ajánlani. Akár több telefonálással, akár több internettel.

9. ÖSSZEFOGLALÓ

A szakdolgozatom célja egy olyan adattárház kiépítése volt, amiből megfelelő mennyiségű és minőségű adatokat lehet lekérdezni és megjeleníteni, hogy az mindenki számára érthető legyen. Célom volt az is, hogy megmutassam az optimalizálást egy nagyobb adathalmazon, valamint ennek a szükségességét.

Az irodalomkutatásom alatt mélyrehatóan körbejártam a relációs adatbázis, adattárház és riport készítés lényeges részeit. Valamint ezekre kerestem alkalmazásokat és hasonlítottam össze egymással, hogy el tudjam dönteni számomra melyik lesz a megfelelő.

Miután kiválasztottam az eszközöket az első dolgom a táblák és kapcsolataik megtervezése volt. Amint ezzel készen lettem elkezdhettem adatokat gyűjteni, aminek a nagyobb részét én generáltam le, vagy találtam ki, de volt olyan is, amit az internetről szedtem le.

Mikor már a .CSV fájlok is készen voltak, amiben az adatok voltak, elkezdtem betölteni őket a relációs adatbázisba. Majd elkészítettem a kapcsolatokat közöttük. Ekkor volt szükségem egy lekérdezésre, ami több táblát is figyelembe vesz. Ennek a futtatása után kellett ezt optimalizálni, hogy jobb futásidőt kapjunk.

Ezek után kezdtem el az adattárházba való rétegek betöltését. Ha szükségesnek találtam akkor historikusan tároltam el az adataimat, hogy később is vissza lehessen nézni a változásokat. A rétegek betöltése után pedig kialakítottam a dimenziókat és a tény táblát, ami a csillagsémát alkotta.

A következő lépésem a Power BI használata volt, amibe könnyedén lehetett importálni az adatokat. Itt készítettem el a vizualizációkat, az üzleti igények alapján. Figyeltem arra, hogy könnyen érthető és átlátható legyen a megjelenítés, mivel ezeket már inkább az üzlet használja nem pedig a fejlesztő.

Végezetül kijelenthetem, hogy sikerült megvalósítanom a szakdolgozatom célját, és egy használható adattárházat építettem fel, könnyen értelmezhető vizualizációval.

10.SUMMARY

The aim in my thesis was to create a data warehouse that would provide enough quantity and quality of data, which can be visualized in a way that end users can also understand. My other goal was to show optimization on a bigger dataset and why it is useful.

During my literature research I studied the topic of relational database, data warehouses and visualizations of data. I looked for applications for these and compared them to each other so I could decide which one is the optimal choice for my work.

After I choose the tools and application the first step was to plan the tables and their relations. I started to collect data which bigger part was generated, and the other parts were either fictional or gathered from the internet.

When the .CSV files were ready, that contained the data, I imported them to the relational database. After that I created the connections between them. When these relations were set, I created a query which joined multiple tables. I analyzed the results and optimized it to have better run time.

I loaded the layers of the data warehouse. If I found it necessary, I stored my data historically so the changes can be reviewed later. After loading the layers, I created the dimensions and the fact table that formed the star schema.

My next step was to use Power BI, where I could easily import my data. I created the visualizations based on the business needs. I made sure that the display was easy to understand, because these are used by the business and not by developers.

Finally, I can state that I have achieved the goals of my thesis and built a data warehouse with self-explanatory visualization.

11.IRODALOMJEGYZÉK

- [1] M. Dr. Szepesné Stiftinger, „Informatika 9, Adatbázisok tervezése,” [Online]. Available: https://regi.tankonyvtar.hu/hu/tartalom/tamop425/0027_INF9/ch01s02.html. [Hozzáférés dátuma: 08 05 2021].
- [2] J. D. Ullman és J. Widom, Adatbázis rendszerek alapvetés, Panem, 2009.
- [3] „Horizontal Cross Functional Flowchart,” [Online]. Available: <https://www.conceptdraw.com/How-To-Guide/drawing-software-horizontal-cross-functional-flowchart>. [Hozzáférés dátuma: 13 05 2021].
- [4] Z. Kardkovács, F. Győr és J. Marton, ADATBÁZISOK LABORATÓRIUM, BME-TMIT, 2017.
- [5] „Oracle Database Architecture,” [Online]. Available: <https://www.oracletutorial.com/oracle-administration/oracle-database-architecture/>. [Hozzáférés dátuma: 13 05 2021].
- [6] Microsoft, [Online]. Available: <https://www.microsoft.com/en-us/sql-server/sql-server-2019>. [Hozzáférés dátuma: 09 05 2021].
- [7] Microsoft, [Online]. Available: <https://www.microsoft.com/en-us/evalcenter/evaluate-sql-server-2019>. [Hozzáférés dátuma: 09 05 2021].
- [8] T. Radványi, „Haladó DBMS,” [Online]. Available: https://regi.tankonyvtar.hu/hu/tartalom/tamop425/0038_informatika_HDBMS-HU/ch01s08.html. [Hozzáférés dátuma: 09 05 2021].
- [9] „Telepítési módok,” [Online]. Available: http://dokumentacio.infoteka.hu/hpdotnet/index.html?telepitesi_modok.html. [Hozzáférés dátuma: 13 05 2021].
- [10] S. Suehring, MySQL Bible.
]

- [11 „MySQL,” [Online]. Available: <https://www.mysql.com/products/standard/>.
] [Hozzáférés dátuma: 09 05 2021].
- [12 MySQL, [Online]. Available: <https://www.mysql.com/products/enterprise/>.
] [Hozzáférés dátuma: 09 05 2021].
- [13 MySQL, [Online]. Available: <https://www.mysql.com/products/cluster/>. [Hozzáférés
] dátuma: 09 05 2021].
- [14 PostgreSQL, [Online]. Available:
] <https://www.postgresql.org/files/documentation/pdf/13/postgresql-13-A4.pdf>.
[Hozzáférés dátuma: 09 05 2021].
- [15 SQLite, [Online]. Available: <https://www.sqlite.org/about.html>. [Hozzáférés dátuma:
] 09 05 2021].
- [16 A. L. Dennis, „The Data Warehouse: From the Past to the Present,” [Online]. Available:
] <https://www.dataversity.net/data-warehouse-past-present/>. [Hozzáférés dátuma: 10 05
2021].
- [17 C. Sidló, „Összefoglaló az adattárházak témaköréről,” [Online]. Available:
] <http://scs.web.elte.hu/Work/DW/adattarhazak.htm#irodalom%202>. [Hozzáférés
dátuma: 10 05 2021].
- [18 S. Rangarajan, „Data Warehouse Design – Inmon versus Kimball,” [Online]. Available:
] <https://tdan.com/data-warehouse-design-inmon-versus-kimball/20300>. [Hozzáférés
dátuma: 10 05 2021].
- [19 „Data Mart vs. Data Warehouse,” [Online]. Available: <https://panoply.io/data-warehouse-guide/data-mart-vs-data-warehouse/>. [Hozzáférés dátuma: 13 05 2021].
- [20 L. Kovács, „OLAP rendszerek alapjai,” [Online]. Available: <http://users.iit.unimiskolc.hu/~kovacs/hirdetes/olapdoc.pdf>. [Hozzáférés dátuma: 10 05 2021].
- [21 P. Szabó és A. Pálfi, „Értékesítési elemzés OLAP kockával,” [Online]. Available:
] <http://centaur.sch.bme.hu/~saatetyi/doc/onlab.pdf>. [Hozzáférés dátuma: 13 05 2021].

- [22 P. Dr. Hegedűs és R. Dr. Ferenc, „Nagyméretű adatbázisok,” [Online]. Available:
] http://eta.bibl.u-szeged.hu/5031/2/EFOP343_A6_2e_BigData-data-transform-SPOC.pdf. [Hozzáférés dátuma: 10 05 2021].
- [23 „Kinyerés, átalakítás és betöltés (ETL),” [Online]. Available:
] <https://docs.microsoft.com/hu-hu/azure/architecture/data-guide/relational-data/etl>.
[Hozzáférés dátuma: 13 05 2021].
- [24 „Talend Open Studio,” Tutorialspoint, [Online]. Available:
] https://www.tutorialspoint.com/talend/talend_talend_open_studio.htm. [Hozzáférés dátuma: 11 05 2021].
- [25 „Oracle Data Integrator Statement of Direction,” [Online]. Available:
] <https://www.oracle.com/uk/a/tech/docs/odi-statement-of-direction-20200501.pdf>.
[Hozzáférés dátuma: 11 05 2021].
- [26 „SSIS Tutorial for Beginners: What is, Architecture, Packages,” [Online]. Available:
] <https://www.guru99.com/ssis-tutorial.html>. [Hozzáférés dátuma: 11 05 2021].
- [27 „Mi az az üzleti intelligencia?,” Microsoft, [Online]. Available:
] <https://powerbi.microsoft.com/hu-hu/what-is-business-intelligence/>. [Hozzáférés dátuma: 11 05 2021].
- [28 C. Csepei, „Intelligens Adattárházak,” [Online]. Available:
] <https://dea.lib.unideb.hu/dea/bitstream/handle/2437/85131/IntelligensAdatt%E1rh%E1zak.pdf?sequence=1>. [Hozzáférés dátuma: 11 05 2021].
- [29 „How to create a Viz in Tooltip in Tableau Software,” [Online]. Available:
] <https://btprovider.com/how-to-create-a-viz-in-tooltip-in-tableau-software-video-friday-skill-pill/>. [Hozzáférés dátuma: 13 05 2021].
- [30 „IBM COGNOS ANALYTICS,” aramar, [Online]. Available:
] <https://www.aramar.co.uk/software/cognos-analytics>. [Hozzáférés dátuma: 11 05 2021].
- [31 IBM, [Online]. Available: <https://www.ibm.com/products/cognos-analytics/features>.
] [Hozzáférés dátuma: 11 05 2021].

- [32 „Resources,” [Online]. Available: <https://www.ibm.com/products/cognos-dashboard-embedded/resources>. [Hozzáférés dátuma: 13 05 2021].
- [33 Microfost, [Online]. Available: <https://powerbi.microsoft.com/hu-hu/why-power-bi/>. [Hozzáférés dátuma: 11 05 2021].
- [34 „Gyorsútmutató: Csatlakozás adatokhoz a Power BI Desktopban,” [Online]. Available: <https://docs.microsoft.com/hu-hu/power-bi/connect-data/desktop-quickstart-connect-to-data>. [Hozzáférés dátuma: 13 05 2021].
- [35 TOPdesk, [Online]. Available: https://page.topdesk.com/hubfs/1_E-books_and_Articles/Hungarian/Whitepapers/Riport%C3%A1l%C3%A1s.pdf. [Hozzáférés dátuma: 11 05 2021].
- [36 Microsoft, [Online]. Available: <https://www.microsoft.com/hu-hu/microsoft-365/business-insights-ideas/resources/guide-to-uml-diagramming-and-database-modeling>. [Hozzáférés dátuma: 12 05 2021].
- [37 Tableau, [Online]. Available: <https://visualdataanalytics.wordpress.com/2017/05/15/bi-eszkozok-tableau/>. [Hozzáférés dátuma: 11 05 2021].
- [38 TOPdesk, [Online]. Available: <https://www.topdesk.com/hu/modulok/dashboard-es-riportalas/>. [Hozzáférés dátuma: 13 05 2021].

12.ÁBRAJEGYZÉK

2.1. ábra - Átalakítás	11
2.2. ábra - Táblák és kapcsolataik [3]	12
2.3. ábra - Tábla szerkezet [1].....	12
2.4. ábra - Oracle fizikai felépítése [5]	14
2.5. ábra - Szerver - kliens kapcsolat [9]	15
2.6. ábra - Kimball modell [19]	17
2.7. ábra - Inmon modell [19].....	18
2.8. ábra - OLAP kocka [21].....	18
2.9. ábra - ETL folyamat [23].....	19
2.10. ábra – Tableau [29]	22
2.11. ábra – Cognos [32].....	23
2.12. ábra - Power BI [34]	24
2.13. ábra – TOPdesk [35]	25
4.1. ábra - Eladott csomagok	27
5.1. ábra - UML-diagram	28
5.2. ábra - Csillag séma.....	29
6.1. ábra - Táblák	32
6.2. ábra – SQL utasítás	32
6.3. ábra – Lekérdezés költsége	33
6.4. ábra – Optimalizált lekérdezés.....	33
6.5. ábra - Mapping.....	35
6.6. ábra – Description	35
6.7. ábra - Adatbázisok	36
6.8. ábra - Extract.....	36
6.9. ábra - Load .CSV	37
6.10. ábra - Transform	37

6.11. ábra – Read STAGE	38
6.12. ábra - Load	38
6.13. ábra - Load DIM	39
6.14. ábra - Load FACT	39
7.1. ábra – Power Bi.....	40
7.2. ábra - Fizetések összehasonlítása.....	40
7.3. ábra - Üzletek csoportosítása	41
7.4. ábra - Csomagok népszerűsége	42
7.5. ábra – Ügyfelek csoportosítása	43
7.6. ábra - Új ügyfelek	44