



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

NEUMANN JÁNOS
INFORMATIKAI KAR



SZAKDOLGOZAT

**OE-NIK
2021**

Hallgató neve:
Hallgató törzskönyvi száma:

**Varga Dániel András
T/006429/FI12904/N**

Óbudai Egyetem

Neumann János Informatikai Kar
Kiberfizikai Rendszerek Intézet

SZAKDOLGOZAT FELADATLAP

Hallgató neve: **Varga Dániel András**

Törzskönyvi száma: T/006429/FI12904/N

Neptun kódja:

A dolgozat címe:

BI rendszer kialakítása adattárház segítségével egy könyvesbolt-hálózatnál
BI system development with the help of a data warehouse at bookstore network

Intézményi konzulens: Rusznák Attila

Külső konzulens:

Beadási határidő: 2021. december 15.

A záróvizsga tárgyai: Számítógép architektúrák
Big Data és üzleti intelligencia specializáció

A feladat

A szakdolgozat keretében a hallgató feladata egy olyan webes alkalmazás megvalósítása, ami egy könyvesbolt-hálózat kereskedelmi adatait teszi megfigyelhetővé kimutatások formájában. A cél a már meglévő adatok elérhetőségének javítása úgy, hogy a kimutatások elemezhetőek legyenek a webalkalmazáson keresztül. A hallgató feladata a projekt kapcsán meghatározott adatforrásokból az adatok átvitelének megtervezése és megvalósítása ETL réteg segítségével adattárházba. Az adattárházból automatikusan frissülő kimutatások lesznek elérhetőek a megfelelő szervezeti egységek számára a webalkalmazáson keresztül.

A dolgozatnak tartalmaznia kell:

- a feladat és a vállalat részletes ismertetése,
- adatforrás réteg meghatározása, illetve a rendszerek és a kívánt adatok kiválasztása,
- ETL réteg megtervezése és megalkotása,
- adattárház tervezése és megvalósítása,
- kimutatások és adat vizualizáció megvalósítása,
- webalkalmazás készítése a vizualizációk megjelenítéséhez.



Flc Ne

Dr. Fleiner Rita
intézetigazgató

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

A dolgozatot beadásra alkalmasnak tartom:

.....
külső konzulens

Russek AA
.....
intézményi konzulens



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. december 06.


.....
hallgató aláírása



KONZULTÁCIÓS NAPLÓ

Hallgató neve: Neptun kód: Tagozat:

Varga Dániel András [REDACTED] Nappali

Telefon: Levelezési cím (pl: lakcím):

Szakdolgozat / Diplomamunka¹ címe magyarul:

BI rendszer kialakítása adattárház segítségével egy könyvesbolt-hálózatnál

Szakdolgozat / Diplomamunka² címe angolul:

BI system development with the help of a data warehouse at bookstore network

Intézményi konzulens: Külső konzulens:

Rusznák Attila

Kérjük, hogy az adatokat nyomtatott nagybetűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	2021.02.20.	Téma egyeztetése	Rusznák AA/ik
2.	2021.03.26.	Felmerülő kérdések tisztázása	Rusznák AA/ik
3.	2021.04.19.	Szakdolgozat tartalmának megbeszélése	Rusznák AA/ik
4.	2021.05.12.	Leadás előtti pontosítások	Rusznák AA/ik

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

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

Budapest, 2021.május 13.

Rusznák AA/ik
intézményi konzulens

¹ Megfelelő aláhúzendő!

² Megfelelő aláhúzendő!

³ Megfelelő aláhúzendő!

⁴ Megfelelő aláhúzendő!



KONZULTÁCIÓS NAPLÓ

Hallgató neve: Varga Dániel András
Neptun kód: [REDACTED]
Tagozat: Nappali
Telefon: [REDACTED]
Levelezési cím (pl: lakcím): [REDACTED]

Szakdolgozat / Diplomamunka⁵ címe magyarul:

BI rendszer kialakítása adattárház segítségével egy könyvesbolt-hálózatnál

Szakdolgozat / Diplomamunka⁶ címe angolul:

BI system development with the help of a data warehouse at bookstore network

Intézményi konzulens: Ruzsnák Attila
Külső konzulens:

Ruzsnák Attila

Kérjük, hogy az adatokat nyomtatott nagybetűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	2021.09.17.	Felmerülő kérdések megbeszélése	Ruzsnák Attila
2.	2021.10.05.	Formai hibák javítása	Ruzsnák Attila
3.	2021.11.30.	Kész szakdolgozat közös áttekintése	Ruzsnák Attila
4.	2021.12.08.	Utolsó simítások elvégzése	Ruzsnák Attila

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

A hallgató a Szakdolgozat I. / Szakdolgozat II. (BSc) vagy Diplomamunka 1 / Diplomamunka 2 / Diplomamunka 3 / Diplomamunka 4⁷ tantárgy követelményét teljesítette, beszámolóra / védésre⁸ bocsátható.

Budapest, 2021. december 06.

Ruzsnák Attila
intézményi konzulens

⁵ Megfelelő aláhúzendő!

⁶ Megfelelő aláhúzendő!

⁷ Megfelelő aláhúzendő!

⁸ Megfelelő aláhúzendő!

Tartalomjegyzék

1.	BEVEZETÉS	9
2.	A VÁLLALAT ÉS AZ ADATBÁZIS BEMUTATÁSA	10
2.1.	A VÁLLALAT ISMERTETÉSE	10
2.2.	AZ ADATBÁZIS ISMERTETÉSE	10
2.2.1.	A tárolt adatok	10
2.2.2.	Tábla diagram	13
2.2.3.	Egyed-kapcsolat (E/K) diagram	14
3.	ADATBÁZIS-KEZELŐ RENDSZEREK BEMUTATÁSA	15
3.1.	RELÁCIÓS ADATBÁZIS-KEZELŐ RENDSZEREK ÖSSZEHOSONLÍTÁSA	15
3.1.1.	MSSQL ismertetése	16
3.1.2.	MySQL ismertetése	16
3.1.3.	PostgreSQL ismertetése	17
3.2.	ÖSSZEHOSONLÍTÓ TÁBLÁZAT	19
4.	EXTRACT TRANSFORM LOAD (ETL)	20
4.1.	ETL ISMERTETÉSE	20
4.2.	AZ ETL RÉSZEI	21
4.2.1.	Extract fázis	21
4.2.2.	Transform fázis	21
4.2.3.	Load fázis	21
4.3.	ETL ESZKÖZÖK ÖSSZEHOSONLÍTÁSA	22
4.3.1.	SQL Server Integration Services (SSIS) ismertetése	22
4.3.2.	Informatica PowerCenter ismertetése	22
4.3.3.	Talend Open Studio Data Integration ismertetése	23
4.4.	ÖSSZEHOSONLÍTÓ TÁBLÁZAT	24
5.	RENDSZEREK KIVÁLASZTÁSA ÉS INDOKLÁS	25
6.	ADATTÁRHÁZ MEGTERVEZÉSE	26
6.1.	ADATTÁRHÁZ FOGALMA	26
6.2.	ÜZLETI IGÉNY ISMERTETÉSE	26
6.3.	FORRÁSDATOK KIVÁLASZTÁSA	27
6.4.	KIALAKÍTOTT CSILLAGSÉMA	27
6.5.	A CSILLAGSÉMÁRÓL RÖVIDEN	28
6.6.	TÉNYTÁBLA RÉSZLETEZÉSE	28
6.6.1.	Vásárlások ténytábla (FACT_Purchases)	28
6.7.	DIMENZIÓ TÁBLÁK RÉSZLETEZÉSE	28
6.7.1.	Dátum dimenzió (DIM_Date)	28
6.7.2.	Könyv dimenzió (DIM_Book)	28
6.7.3.	Bolt dimenzió (DIM_Store)	29
6.7.4.	Vásárló dimenzió (DIM_Customer)	29
7.	ADATVIZUALIZÁCIÓS MÓDSZEREK ÉS ESZKÖZÖK BEMUTATÁSA	30
7.1.	ADATVIZUALIZÁCIÓ ISMERTETÉSE	30

7.2.	A LEGNÉPSZERŰBB DIAGRAM ÉS GRAFIKON FAJTÁK ISMERTETÉSE.....	33
7.2.1.	Táblázat és mutatószám	33
7.2.2.	Oszlopdiagram és hisztogram	33
7.2.3.	Sávdiaagram.....	34
7.2.4.	Szórásdiagram.....	35
7.2.5.	Vonaldiagram.....	35
7.2.6.	Területdiagram.....	36
7.2.7.	Kör- és fánkdiagram	36
7.2.8.	Fatérkép diagram	37
7.2.9.	Térképdiaagram	37
7.2.10.	Teljesítmény diagram.....	38
7.3.	ANALITIKA ÉS ÜZLETI INTELLIGENCIA ESZKÖZÖK ÖSSZEHASONLÍTÁSA.....	39
7.3.1.	Microsoft Power BI	39
7.3.2.	Tableau.....	40
7.3.3.	QlikView.....	41
7.4.	ADATVIZUALIZÁCIÓS ESZKÖZ KIVÁLASZTÁSA ÉS INDOKLÁS.....	42
8.	TECHNOLÓGIÁK A WEBALKALMAZÁSSAL TÖRTÉNŐ MEGVALÓSÍTÁSHOZ	43
8.1.	HTML ISMERTETÉSE.....	43
8.2.	CSS ISMERTETÉSE.....	43
8.3.	JAVASCRIPT ISMERTETÉSE	43
8.4.	BLAZOR KERETRENDSZER ISMERTETÉSE	43
8.5.	WEBALKALMAZÁS GRAFIKUS TERVE.....	44
9.	ADATINTEGRÁCIÓS FOLYAMAT MEGVALÓSÍTÁSA (ETL)	45
9.1.	STAGE RÉTEG.....	45
9.2.	DATA WAREHOUSE RÉTEG	46
9.3.	DATA MART RÉTEG.....	48
10.	ADATVIZUALIZÁCIÓKAT MEGJELENÍTŐ WEBALKALMAZÁS MEGVALÓSÍTÁSA	51
11.	RENDSZER KOMPONENSEINEK TESZTELÉSE.....	55
11.1.	ADATBÁZIS MEGFELELŐ MŰKÖDÉSÉNEK (KAPCSOLATAINAK) TESZTELÉSE	55
11.1.1.	Táblák közötti szimpla kapcsolatok.....	56
11.1.2.	Táblák közötti összetettebb kapcsolatok.....	56
11.2.	ADATINTEGRÁCIÓS FOLYAMATOK- ÉS AZ ADATTÁRHÁZ KAPCSOLATAINAK TESZTELÉSE	57
11.3.	WEBALKALMAZÁS TESZTELÉSE.....	57
12.	A PROJEKT TOVÁBBFEJLESZTÉSI LEHETŐSÉGEI	59
13.	ÖSSZEFOGLALÁS	60
14.	SUMMARY	61
15.	IRODALOMJEGYZÉK	62
16.	ÁBRAJEGYZÉK	67

1. Bevezetés

A szakdolgozatom témája egy webalkalmazás elkészítése, amely az általa megjelenített adatvizualizációk segítségével megkönnyíti a vállalat üzleti döntéseit. A vállalat egy fiktív könyvesbolt-hálózat, melynek három magyarországi telephelye van. Ezeknek a boltoknak az ügyfél-, vásárlási- és raktározási adatai egy köztes adatbázisban vannak eltárolva. A projektem alapja ez a teljes mértékben sajátkezűleg létrehozott és nem valós adatokat tartalmazó adatbázis.

A feladat megvalósításához létre kell hozni az adatbázisban tárolt vásárlási adatok alapján egy adattárházat. Az adattárház feltöltéséért és a vizualizációk létrehozásához szükséges forrásfájlok exportálásáért felel majd a kiválasztott ETL eszköz, és ezeknek az exportált állományoknak a megjelenítéséért pedig a webalkalmazás.

A vállalat fejlődése érdekében szükség van egy felületre, ahol a vásárlási adatokat vizualizációk formájában tudják a cég vezetői elemezni, és ezek alapján meghozni az üzleti döntéseiket. Ezt az igényt elégíti ki egy olyan webalkalmazás, amelyhez akár egy fontos üzleti megbeszélés előtt is bárhonnán hozzáfér a felsővezető, és látja az előző hetek vagy hónapok eladási statisztikáit különböző szempontok szerint.

A szakdolgozatomban kitérek a vállalat és a forrás adatbázis részletes ismertetésére. Emellett bemutatom a projekt során nélkülözhetetlen eszközöket, mint az adatbázis-, adatintegrációs- és üzleti intelligencia eszközök, illetve ezeknek a szerepét a szakdolgozat megvalósításában és a különböző komponenseknél szóba jövő szoftverek összehasonlítását. Az üzleti intelligencia eszközöknél is elvégzem az összehasonlítást, viszont a vizualizációkat és ezek megjelenítését egy Blazor technológiájú webalkalmazás fejlesztésével oldom meg. A szakdolgozatom záró részében a fejlesztést, annak tesztelését, és a jövőbeni továbbfejlesztési lehetőségeket mutatom be.

2. A vállalat és az adatbázis bemutatása

2.1. A vállalat ismertetése

A szakdolgozatom vállalati háttere egy kitalált könyvesbolt-hálózat, melynek a Varga Books nevet adtam. A vállalat 2018-ban lett bejegyezve, a cégforma egyszemélyes korlátolt felelősségű társaság, és a fő tevékenysége a könyv kereskedelem. Három boltból áll, amelyek Magyarország területén helyezkednek el. Ezekből kettő Budapesten és egy Debrecenben. Minden boltban van egy boltvezető és két dolgozó. A dolgozókra a szakdolgozatomban nem térek ki részletesebben, ebből kifolyólag az adatbázis szerkezetben sem jelenik meg olyan tábla, hogy „Workers”, mert a webalkalmazás és a kimutatások szempontjából nem relevánsak. Kivételt képeznek a boltvezetők, hiszen ők irányítják az egyes boltokat. Ennek okán nekik külön táblájuk van az adatbázisban és később a kimutatásoknál is nagy szerepük lesz.

2.2. Az adatbázis ismertetése

Napjainkban az adatbázisok használata széles körben elterjedt. A különböző szolgáltatások mind adatbázist használnak adataik tárolására, kezelésére és lekérdezésére, a könnyebb átláthatóság és nagyobb hatékonyság elérése érdekében. Manapság az adatbázisokon az adatok célszerűen rendezett tárolását értjük. A hangsúly nem az adatok mennyiségén, hanem a célszerű rendezettségen van. Ennek a célszerű rendezettségnek lehetővé kell tennie az adatok értelmes és gyors kezelését. Az adatbázisokkal kéz a kézben jár egy adatbáziskezelő rendszer (DBMS – Database Management System), amely az adatbázisok üzemeltetését biztosítja.

Az általam tervezett és létrehozott adatbázis egy könyvesbolt hálózat adatait tárolja.

2.2.1. A tárolt adatok

Az adatbázisom kilenc táblából és egy kapcsolótáblából áll. Ezek mindegyike hasznos és szükséges adatokat tartalmaz a könyvesbolt-hálózatról.

A táblák név szerint: Addresses, Authors, Books, Categories, Customers, Managers, Publishers, Purchases, Stocks, Stores.

Addresses tábla

address_id	A(z) Addresses tábla elsődleges kulcsa.
country	Az ország, ahol a cím található.
county	A megye, ahol a cím található.
zip_code	Az irányítószám, ahol a cím található.
city	A város, ahol a cím található.
street_name	A címhez tartozó közterület neve.
house_number	A címhez tartozó házszám.

2.1. táblázat: Addresses tábla elemei

Authors tábla

author_id	A(z) Authors tábla elsődleges kulcsa.
first_name	A szerző keresztnéve.
middle_name	A szerző középső nevének(második keresztnévének) rövidítése.
last_name	A szerző vezetéknéve.
gender	A szerző neme.
birth_date	A szerző születési dátuma.
country_of_birth	A szerző születési országa.

2.2. táblázat: Authors tábla elemei

Books tábla

isbn	A(z) Books tábla elsődleges kulcsa.
book_title	A könyv címe.
date_of_publication	A könyv publikálásának dátuma.
length	A könyv hossza.
list_price	A könyv lista ára.
category_id	A könyv kategóriájának azonosítója. Idegen kulcs, ami a(z) Categories tábla egyik rekordjára hivatkozik.
author_id	A könyv szerzőjének azonosítója. Idegen kulcs, ami a(z) Authors tábla egyik rekordjára hivatkozik.
publisher_id	A könyv kiadójának azonosítója. Idegen kulcs, ami a(z) Publishers tábla egyik rekordjára hivatkozik.

2.3. táblázat: Books tábla elemei

Categories tábla

category_id	A(z) Categories tábla elsődleges kulcsa.
category_name	A kategória megnevezése.

2.4. táblázat: Categories tábla elemei

Customers tábla

customer_id	A(z) Customers tábla elsődleges kulcsa.
first_name	A vásárló keresztnéve.
middle_name	A vásárló középső nevének(második keresztnévének) rövidítése.
last_name	A vásárló vezetéknéve.
gender	A vásárló neme.
address_id	A vásárló lakhelyének azonosítója. Idegen kulcs, ami a(z) Addresses tábla egyik rekordjára hivatkozik.
email_address	A vásárló e-mail címe.
phone_number	A vásárló telefonszáma.

2.5. táblázat: Customers tábla elemei

Managers tábla

manager_id	A(z) Managers tábla elsődleges kulcsa.
first_name	A boltvezető keresztnéve.
middle_name	A boltvezető középső nevének(második keresztnevének) rövidítése.
last_name	A boltvezető vezetéknéve.
gender	A boltvezető neme.
address_id	A boltvezető lakhelyének azonosítója. Idegen kulcs, ami a(z) Addresses tábla egyik rekordjára hivatkozik.
email_address	A boltvezető e-mail címe.
phone_number	A boltvezető telefonszáma.

2.6. táblázat: Managers tábla elemei

Publishers tábla

publisher_id	A(z) Publishers tábla elsődleges kulcsa.
company_name	A kiadó megnevezése.
phone_number	A kiadó telefonszáma.
email_address	A kiadó e-mail címe.
website	A kiadó weboldalának linkje.
address_id	A kiadó székhelyének azonosítója. Idegen kulcs, ami a(z) Addresses tábla egyik rekordjára hivatkozik.

2.7. táblázat: Publishers tábla elemei

Purchases tábla

purch_row_id	A(z) Purchases tábla elsődleges kulcsa. (rekordot azonosít)
purch_id	A vásárlás azonosítója. (vásárlást azonosít)
store_id	A vásárlás helyének azonosítója. Idegen kulcs, ami a(z) Stores tábla egyik rekordjára hivatkozik.
isbn	A vásárolt könyv ISBN száma. Idegen kulcs, ami a(z) Books tábla egyik rekordjára hivatkozik.
customer_id	A vásárló azonosítója. Idegen kulcs, ami a(z) Customers tábla egyik rekordjára hivatkozik.
purch_quantity	A vásárolt mennyiség a könyvből.
purch_price_per_piece	A vásárolt könyv egységára.
purch_date	A vásárlás dátuma.

2.8. táblázat: Purchases tábla elemei

Stocks tábla

stock_id	A(z) Stocks tábla elsődleges kulcsa.
store_id	A bolt azonosítója. Idegen kulcs, ami a(z) Stores tábla egyik rekordjára hivatkozik.
isbn	A könyv ISBN száma. Idegen kulcs, ami a(z) Books tábla egyik rekordjára hivatkozik.
quantity	Az eltárolt mennyiség.

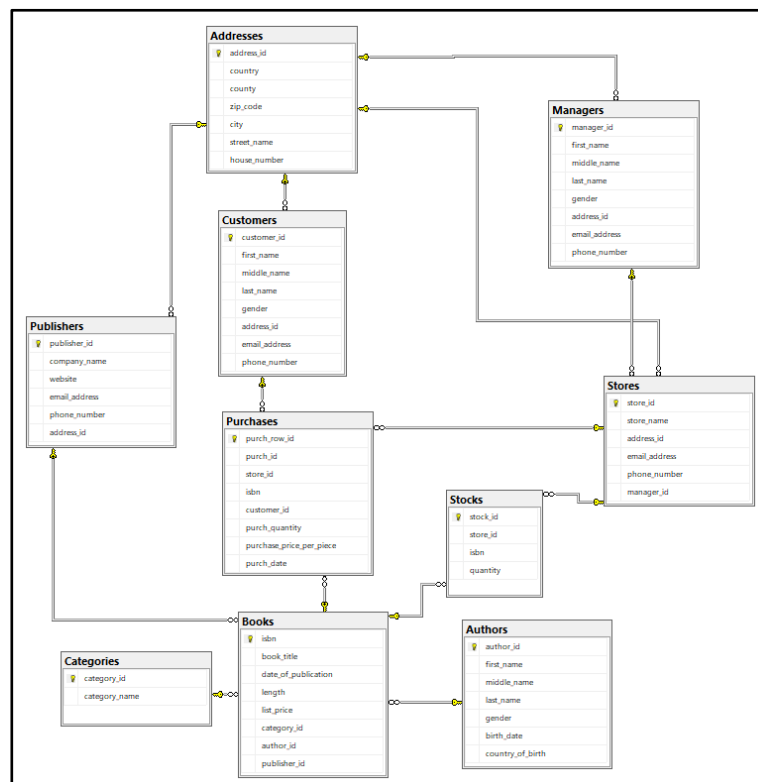
2.9. táblázat: Stocks tábla elemei

Stores tábla

store_id	A(z) Stores tábla elsődleges kulcsa.
store_name	A bolt neve.
address_id	A bolt székhelyének azonosítója. Idegen kulcs, ami a(z) Addresses tábla egyik rekordjára hivatkozik.
email_address	A bolt e-mail címe.
phone_number	A bolt telefonszáma.
manager_id	A bolt vezetőjének azonosítója. Idegen kulcs, ami a(z) Managers tábla egyik rekordjára hivatkozik.

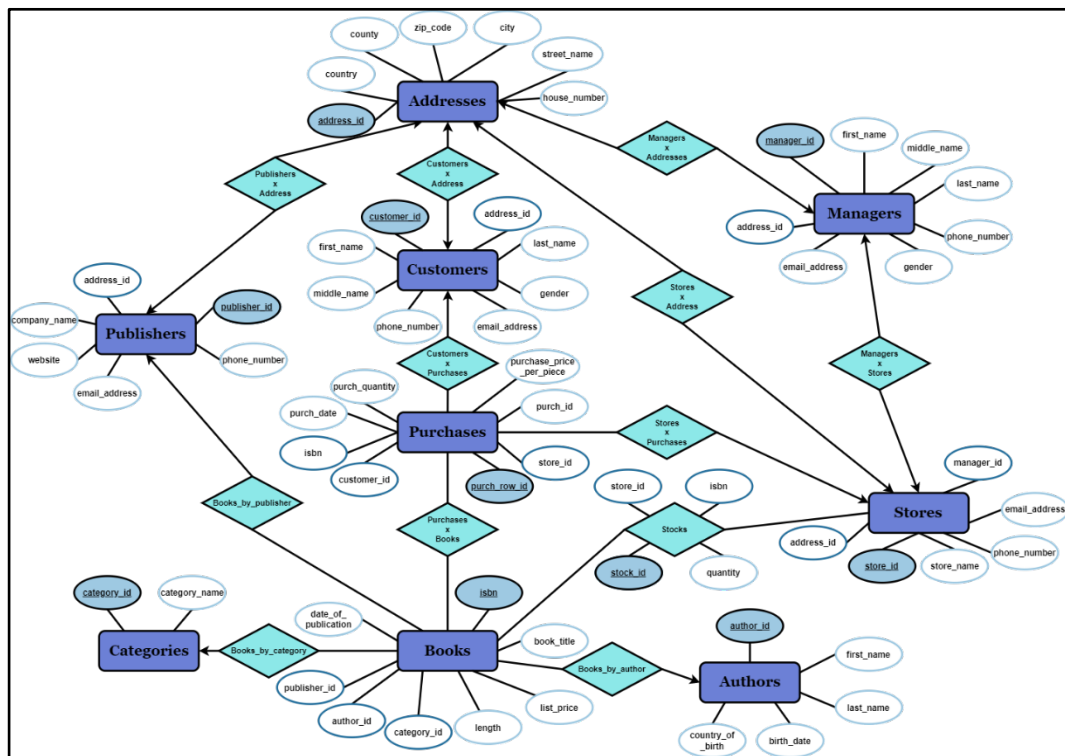
2.10. táblázat: Stores tábla elemei

2.2.2. Tábla diagram



2.1. ábra: Az adatbázis tábla diagramja

2.2.3. Egyed-kapcsolat (E/K) diagram



2.2. ábra: Az adatbázis egyed-kapcsolat diagramja

3. Adatbázis-kezelő rendszerek bemutatása

3.1. Relációs adatbázis-kezelő rendszerek összehasonlítása

Az adatbázis-kezelő rendszereket DBMS-nek nevezzük, ami a „Database Management System” rövidítése. Gyakorlatilag ez egy olyan szoftver rendszer, amelyet adatok tárolására, visszakeresésére és lekérdezésére használunk. Ez a rendszer kezeli a több forrásból is érkező adatokat, rendszerezi őket a feladat vagy modellnek megfelelően, és biztosítja a felhasználónak vagy más programoknak az adatok módosítását vagy törlését.

Manapság rengeteg adatbázis-kezelő rendszer van a piacon. Ezek közül egy pár: MSSQL, MySQL, PostgreSQL, Oracle, MariaDB. Ennek vannak bizonyos előnyei és hátrányai is. Az előnye, hogy mindenki megtalálja a saját projektjéhez a legmegfelelőbb rendszert. A hátrány pedig, hogy érdemes lehet végig böngészni a piacot, hogy megismerjük a legtöbbet ezek közül, annak érdekében, hogy meghozzuk a lehető legjobb döntést. Ha több különböző feladathoz keresünk több különböző DBMS-t, akkor az is felmerül, hogy ezek a rendszerek tudjanak egymással kommunikálni, azaz kompatibilisek legyenek egymással. Az is elképzelhető, hogy az évek során a vezetőség vagy technikai okok miatt át kell állni egy másik rendszerre. Ekkor nagyon jól tud jönni, ha a projekt elején jól körbe jártuk a piacot és tudjuk milyen alternatíváink vannak.

Az alábbi összehasonlításnál az RDBMS, azaz „Relational Database Management System”-eket fogom részletezni. Ezek olyan DBMS-ek, amelyek biztosítják, hogy a felhasználó képes legyen létrehozni, frissíteni, és adminisztrálni relációs adatbázisokat. A relációs adatbázisokra jellemző, hogy képesek vagyunk azonosítani és elérni adatokat egy másik adat vonatkozásában. Leggyakrabban ezek az adatbázisok táblákba rendezve tárolják az adatokat. Illetve a legtöbb az ilyen rendszerek közül egy strukturált lekérdezési nyelvet használ, hogy hozzáférjen az adatbázishoz. Ez legtöbbször az SQL (Structured Query Language), ami egy programozási nyelv annak érdekében, hogy kommunikáljunk az adatbázisokkal. Az SQL szintaxisa nagyon hasonlít az angol nyelvhez, így könnyen olvasható, tanulható és használható. A lentebb bemutatott adatbázis-kezelő rendszerek mind ezt a lekérdező nyelvet, vagy annak valamilyen variációját használják, hogy hozzáférjenek az adatokhoz.

Az RDBMS-ek közül is arra a háromra fogok fókuszálni, amelyeket a legmegfelelőbbnek találtam a szakdolgozatomban megfogalmazott feladat megvalósításához. A bemutatott rendszerek eléggé hasonlítanak egymásra, viszont mindegyiknek vannak bizonyos előnyei, illetve hátrányai is a többivel szemben. Ezeket fogom ismertetni a következőkben. [1] [2] [3]

3.1.1. MSSQL ismertetése

A Microsoft SQL Server a Microsoft zászlós hajója a relációs adatbázis-kezelő rendszerek piacán. A DB-Engines és a Stack Overflow statisztikái alapján is ez a világon a harmadik legnépszerűbb RDBMS. Népszerűsége a hatékony adattárolásnak és az adatok hatékonyan és egyszerűen való megváltoztatásában, illetve kezelésében rejlik. Érdemes még megemlíteni az SQL kibővítéseként ismert Transact-SQL-t (T-SQL), amit arra lehet használni, hogy kapcsolatot létesítsünk az adatbázisainkkal. [4] [5] [6]

Előnyök:

- **Verziók sokasága:** a Microsoft egy széleskörű választási lehetőséget nyújt a különböző verziói között, amik különböző funkcionálisokkal rendelkeznek.
- **Részletes dokumentáció:** a dokumentációk egyik célja átfogó adatbázis-karbantartás, erre mutat rá a teljes online dokumentációja a Microsoft SQL Servernek. Emellett egy hatalmas előny a fejlesztőknek, mérnököknek, hogy a Microsoft Premier hozzáférést enged a Microsoft dedikált közösségi támogatáshoz is.
- **Felhő szolgáltatás támogatása:** a Microsoft ökoszisztéma részeként az MSSQL könnyen megvalósítható a felhőben.
- **Gyorsaság és stabilitás:** hírnevének az egyik alapköve, hogy gyors és stabil megoldás. Emellett fontos megemlíteni a Microsoft többi termékével való kompatibilitását, ami az egyik oka a gyorsaságnak és a stabilitásnak is.

Hátrányok:

- **Drága:** ha figyelembe vesszük, hogy mennyi nyílt forráskódú és ingyenes megoldás van a piacon, akkor a Microsoft SQL Server Enterprise több, mint 3 000 000 Ft-os ára per két mag, egy igen komoly hátrány lehet kisebb vállalatok számára.
- **Erőforrásigény:** ha a teljesítményt nézzük, akkor az MSSQL elég szépen teljesít az International Journal of Advanced Research in Computer and Communication Engineering 2015 márciusi cikke alapján a MySQL-lel szemben, viszont ennek a teljesítménynek megvan a hardware követelménye is.

3.1.2. MySQL ismertetése

MySQL egy ingyenes, eredetileg nyílt forráskódú relációs adatbázis-kezelő rendszer, amit az Oracle Corporation cég birtokol és kezel 2010 óta. Több statisztika, köztük a DB-Engines rangsora is alátámasztja, hogy a MySQL egyike a három legnépszerűbb adatbázis-kezelő rendszernek, DBMS és RDBMS kategóriában is. Ezen kívül a Stack Overflow 2020-as Databases kategóriájában is első helyet ért el. Ennek az oka többek

közt az nyílt forráskódú licenz, ami rendkívül népszerűvé és széles körben használtá tette a piacon. [7] [5] [8]

Előnyök:

- **Ingyenes:** nyílt forráskódú relációs adatbázis-kezelő rendszer megoldás, szabadon használható, amire csak kedvünk tartja. Emellett van magasabb funkcionalitással rendelkező verziója is, amiért viszont fizetni kell. Ez akkor lehet érdemes, amikor egy nagyobb cég rendszeréhez keresünk adatbázist.
- **Felhő-kompatibilis:** manapság már támogatva van a legnagyobb felhőszolgáltatóknál. Elérhető a vezető platformok felhőszolgáltatásaiban, mint az Amazon, Microsoft és a Google.
- **Népszerűség és egyszerű használat:** az egyik legnépszerűbb adatbázis rendszer, ami többek között annak is köszönhető, hogy rengeteg dokumentáció lelhető fel az interneten, ezen kívül pedig hasonló mennyiségű harmadik féltől származó eszköz, ami azt a célt szolgálja, hogy egyszerűsítse a munkát az adatbázissal, ilyen például a phpMyAdmin.
- **Biztonság:** a MySQL lehetőséget nyújt arra is, hogy eltávolítsuk az ismeretlen fiókokat, töröljük a teszt adatbázisokat, amelyekhez minden vendég felhasználónak hozzáférése van és azt is, hogy felhasználónként testre szabjuk a hozzáférési jogaikat.
- **Sebesség:** ha eltekintünk az extra funkcióktól és megpróbáljuk minimalizálni a telepített szolgáltatások számát, akkor sokat tudunk javítani a sebességen.

Hátrányok:

- **Funkcionális korlátok:** mivel a fókusz a könnyű használhatóságon és a sebességen van, ez számos funkcionális korláttal jár.
- **Fejlesztés lassulása:** mióta vállalati kézben van a rendszer a közösség nem tud olyan gyorsan reagálni problémákra és változtatásokat eszközölni.
- **Licenz és saját tulajdon:** egy kettős engedéllyel rendelkező rendszerről beszélünk, aminek van egy ingyenes, nyílt forráskódú közösség által fejlesztett verziója és több fizetős az Oracle Corporation tulajdonában lévő verziója is.
- **Skálázhatóság:** a MySQL nem a skálázhatóságot szem előtt tartva készült, így az sem véletlen, hogy ez nagyon nehezen megvalósítható.

3.1.3. PostgreSQL ismertetése

Másnéven Postges, magát úgy jellemzi, hogy a legfejlettebb nyílt forráskódú relációs adatbázis a földön. A PostgreSQL egy objektum-relációs adatbázis, ami azt jelenti, hogy elsődlegesen a relációs modellű, de ezen kívül rendelkezik olyan jellemzőkkel,

amik az objektum-orientált modellnek felelnek meg. Azzal a céllal lett létrehozva, hogy egy olyan rendszert nyújtson a felhasználóknak, ami bővíthető és a szabványoknak minden téren megfelel. [4] [6]

Előnyök:

- **Megfelelés a standard SQL-nek:** a PostgreSQL-nek az egyik célja, hogy minél jobban megközelítse a standard SQL-t. Az eredeti PostgreSQL dokumentációra hivatkozva meg is valósít a 179 jellemzőből 160-at.
- **Nyílt forráskód:** a teljesen nyílt forráskódnak és a hatalmas fejlesztői közösségnek köszönhetően rengeteg forrás található az interneten, amik megkönnyítik a munkát a rendszerrel. Ezek közül a leghasznosabbak: hivatalos PostgreSQL dokumentáció, PostgreSQL wiki és a rengeteg online fórum.
- **Skálázhatóság:** ellentétben a MySQL-lel a PostgreSQL vertikálisan skálázható. Figyelembe véve a tényt, hogy szinte minden szoftveres projekt növekszik, ami magával hozza azt is, hogy az adatbázis is növekedni fog ez egy nagyon fontos szempont az adatbázis választásakor.
- **Egyedi adattípusok támogatása:** amellett, hogy támogatja az alap JSON, XML, CSV és egyéb más adattípusokat, lehetőséget nyújt, hogy létrehozzuk a saját típusunkat.

Hátrányok:

- **Inkonzisztens dokumentáció:** ami egyik szempontból az előnye, az egy másiktól a hátránya. A nagy fejlesztői közösség következtében a dokumentáció nem mindig konzisztens és teljes, illetve nem mindig követi egyenlő mértékben a standardek.
- **Monitorozás hiányosságai:** érdemes folyamatosan figyelni, ellenőrizni az adatbázis állapotát, mert az ezt végző eszközök nem mindig végzik jól a dolgukat, így könnyen előfordulhat, hogy későn veszünk belőlük észre egy problémát.
- **Teljesítmény:** minden újonnan csatlakozó kliens után indul egy új folyamat, és minden ilyen folyamat 10MB-ot kap a memóriából. Kliensenként 10MB nem tűnik soknak, de egy olyan adatbázisnál, ahol rengeteg kapcsolódó kliens van, ott nagyon hamar jelentősége lesz ezeknek a számoknak. Ide kapcsolódik még, hogy a nagy olvasás-igényű műveleteknél kevésbé hatékony a rendszer, mint például a MySQL.

3.2. Összehasonlító táblázat

			
Rövid leírás	Microsoft legnépszerűbb relációs adatbázisa	Széles körben használt, nyílt forráskódú RDBMS	Népszerű nyílt forráskódú RDBMS
Elsődleges adatbázis modell	RDBMS	RDBMS	RDBMS
Fejlesztő	Microsoft	Oracle	PostgreSQL Global Development Group
Licensz	Kereskedelmi, de van korlátozott ingyenes verzió	Nyílt forráskódú, de van az Oracle Corporation által birtokolt kereskedelmi verzió	Nyílt forráskódú, közösségileg fejlesztett
Implementációs nyelv	C++	C és C++	C
Szerver operációs rendszer	Linux Windows	Linux Windows FreeBSD OS X Solaris	Linux Windows Unix NetBSD OpenBSD Solaris OS X

11. táblázat: Adatbázis-kezelő rendszerek összehasonlítása

[7] [5] [8] [9] [6] [10]

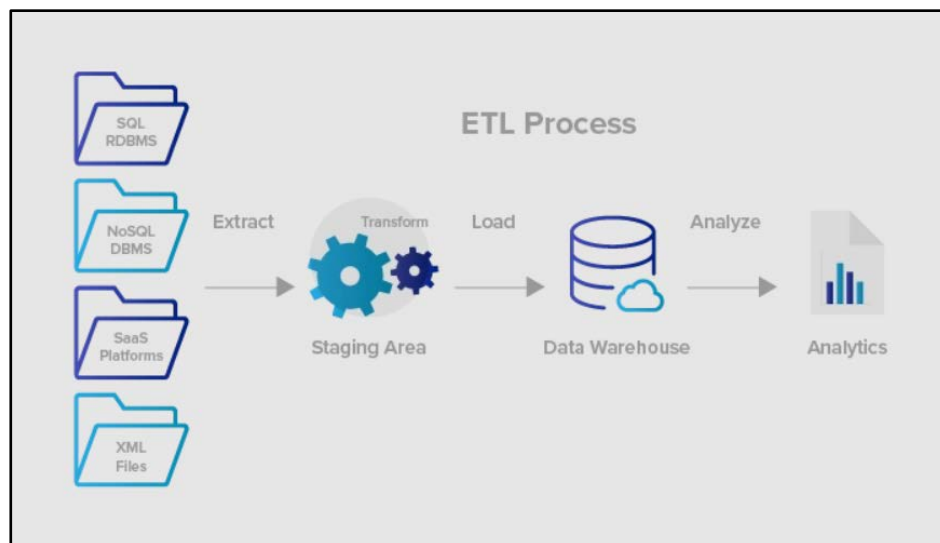
4. Extract Transform Load (ETL)

4.1. ETL ismertetése

Az ETL betűszó az Extract, Transform és Load angol kifejezésekből tevődik össze. Másnéven egy adatintegrációs folyamat, aminek a célja, hogy az említett lépések segítségével több különböző forrásból egy konzisztens adatállományt kapjunk, amit egy adattárházba vagy más cél rendszerbe töltünk be.

Az ETL-t az 1970-es évektől használják, bár akkor még szuperszámítógépekbe töltötték az adatokat számolás és analízis céljából. Az ezredforduló után az elsődleges feladat az volt, hogy adattárházakat hozzanak létre az ETL segítségével, amik később nagy segítséget nyújtottak az üzleti intelligenciával kapcsolatos alkalmazásoknak.

Azért fontos az ETL, mert a több forrásból származó adatok legtöbbször heterogének, vagyis teljesen más formátumban vannak az egyik rendszerben, mint a másikban. Ez nem éppen előnyös az elemzők szempontjából. Ők azt akarják, hogy az adatok amiket megkapnak konzisztensek legyenek és, hogy az üzleti igényeknek megfelelő formázással kerüljenek hozzájuk. Sokkal könnyebb úgy dolgozni, hogy nincsenek duplikált adatok, nincsenek üres mezők, megtörténnek bizonyos előkalkulációk az üzleti igényeknek megfelelően és még számos probléma adódhat, amit az ETL „transform” rétegében le tudunk kezelni, ezzel is könnyítve és gyorsítva az elemzők munkáját. [11]



4.1. ábra: Az ETL folyamat

4.2. Az ETL részei

4.2.1. Extract fázis

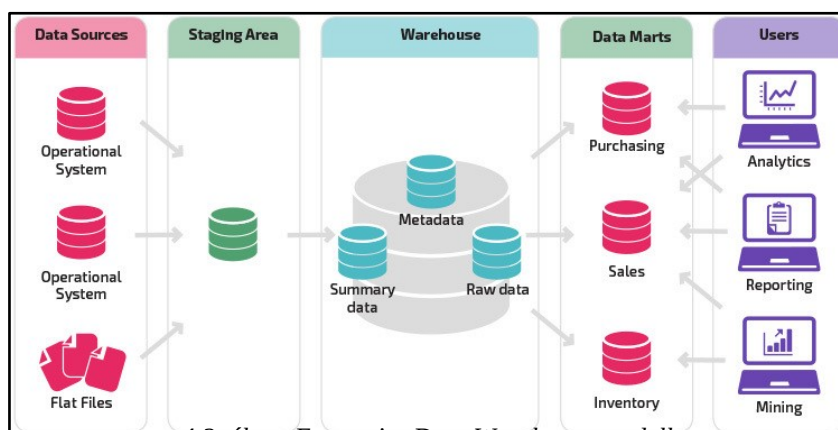
Ez az első lépése egy ETL folyamatnak, itt történik meg az adatok másolása és exportálása a forrásrendszerekből a „Stage” rétegbe. Ez a réteg már az adattárház része, itt tárolódnak el az átmeneti adatok a forrásokból. Bárhonnan érkezhethet ide adat, adatbázisokból, ERP rendszerekből, szolgáltató rendszerekből, de akár egyszerű adatsere formátumú fájllokból is, mint a CSV, XML, EBCDIC.

4.2.2. Transform fázis

Második lépésként jön az adatok transzformálása. Ebben a fázisban a „Stage” rétegben tárolt nyers adatokat az üzleti igénynek megfelelően formázzuk és elvégezzük a szükséges teendőket, hogy az adattárháznak megfelelő formátumú adatokat kapjunk. Ezek a teendők a következők lehetnek: adatok kiszűrése, adattisztítás, duplikált adatok eltávolítása, adat validálás, különböző számítások elvégzése, fordítás, átváltás, szöveges mezők átalakítása, és még sok egyéb folyamat. Ide tartoznak még az adatok védelmét elősegítő műveletek is, mint bizonyos adatok eltávolítása, titkosítása vagy elrejtése. Az adatmennyiségtől és a transzformálások komplexitásától is függ, de összességében ki lehet azt jelenteni, hogy ez az ETL folyamat legidőigényesebb és legerőforrásigényesebb része.

4.2.3. Load fázis

Harmadik és egyben utolsó lépése az ETL folyamatnak a „Load” fázis. Itt történik az adatok betöltése a „Stage” rétegből a „Storage” rétegbe. Itt történik azoknak az adatoknak a tárolása, amikből a „Presentation” réteg kimutatásokat, dashboardokat, riportokat tud készíteni. Innen nem törölünk adatot, hanem ha változás történik egy rekordnál, akkor az ETL folyamatnál be tudjuk állítani, hogy mi történjen olyankor. Ezt is, ahogy rengeteg mindent ennél a témánál, az üzleti igény határozza meg. A legtöbb helyen, ahol ETL-t használnak ez egy teljesen automatizált folyamat. [12] [13]



4.2. ábra: Enterprise Data Warehouse modell

4.3. ETL eszközök összehasonlítása

4.3.1. SQL Server Integration Services (SSIS) ismertetése

A Microsoft, ahogy az informatika összes területén, úgy az adatintegrációs megoldások között is megkerülhetetlen. Az SSIS egy adatmigrációs és integrációs eszköz a Microsoft SQL Server és sok más adatbázis rendszer transzformálási és integrálási folyamatainak ellátására. Az SSIS-t 2005-ben adták ki a Microsoft SQL Server 7-be épített „Data Transformation Services” (DTS) komponensének a helyébe egy külön platformként. A termék népszerűségét az is jól mutatja, hogy a Gartner 2020-as „Magic Quadrant for Data Integration Tools” összehasonlításában a „Challengers” kategória egyik vezető pozícióját ítélték neki. Ebben a kutatásban négy szempont szerint osztja fel az adatintegrációs piacot a Gartner: Leaders (vezetők), Challengers (kihívók), Visionaries (látnokok) és Niche Players, amit az olyan vállalatokra értenek, akiknek a fő üzleti tevékenységük a kisebb piaci szegmensek lefedése. Az SSIS jó választás lehet kisebb és nagyobb vállalatoknak egyaránt, hiszen remekül leköveti a piaci trendeket és remekül ellátja az adatintegrációs feladatot. Legcélszerűbb azoknak a vállalatoknak az SSIS mellett dönteni, akik alapból is SQL Server-rel dolgoznak, hiszen biztosak lehetnek abban, hogy a lehető legkevesebbszer ütköznek majd kommunikációs problémába a két eszköz között. [13] [14]

Előnyök:

- Átfogó dokumentáció és széleskörű támogatás
- Bevált adattárház módszerek támogatása
- Szabványosított adatintegráció
- Könnyű használat (GUI)
- Gyors implementálás
- Szoros együttműködés a Microsoft SQL család többi termékével (Visual Studio és SQL Server)
- Közösségi támogatás
- Ingyenes verzió elérhető

Hátrányok:

- Alternatív adatintegrációs stílusok támogatásának hiánya
- Problémás integráció más termék csoportokkal
- Erős kötődés a Microsoft Windows környezettel

4.3.2. Informatica PowerCenter ismertetése

Az Informatica PowerCenter nagyvállalati adatintegrációs platform az adatok elérésére, körüljárására és integrálására különböző forrásrendszerekből, ahol más és más formában lelhetők fel az adatok. Az Informatica, ahogy a Microsoft is az egyik legnagyobb szoftveripari vállalatok egyike, így az sem véletlen, hogy rendkívül sok felhasználó, akiknek egy dedikált és megbízható adatintegrációs platformra van

szüksége, az az ő megoldásukat választja. A választást az is megkönnyítheti, hogy a Gartner fentebb említett kutatásában az Informatica PowerCenter az legjobb helyet kapta meg a „Leaders” kategóriában. [15]

Előnyök:

- Különböző adatintegrációs stílusok támogatása: adat konszolidáció, adat propagálás, adattárház
- Széles funkcionális tartomány
- Könnyű váltás a különböző adatintegrációs stílusok között
- Követése az új piaci trendeknek
- Skálázhatóság
- Aktív közösségi és hivatalos támogatás
- Magas fókusz az ETL-en és az adatintegráción
- Vezető megoldás a nagyobb vállalatoknál
- Naprakészség a termékek fejlesztésében

Hátrányok:

- Túlságosan próbál igazodni az új trendekhez
- Viszonylag magas ár, főleg kisebb vállalatok számára nem éri meg
- Kevés integráció a különböző Informatica termékek között
- Kevésbé fókuszál a Business Intelligence-re

4.3.3. Talend Open Studio Data Integration ismertetése

A Talend Open Studio egy nyílt forráskódú integrációs szoftver, amely segít a nyers, forrásrendszerekből érkező adatokat az üzleti elemzéseknek megfelelő formátumba alakítani. Egyike a legnépszerűbb ETL eszközöknek a piacon, amit a Gartner 2020-as „Magic Quadrant for Data Integration Tools” kutatása is megerősít, hiszen itt a „Leaders” kategóriába lett besorolva. Mivel az Open Studio a Talend kereskedelmi eszközei közül az ingyenes verzió, így nincs lehetőség felhőszolgáltatásként futtatni, de így is rengeteg feladatot meg lehet vele gond nélkül oldani. [16]

Előnyök:

- Széles funkcionalitás,
- Könnyű használat tapasztaltabb fejlesztőknek,
- Csökkenti a megvalósítás komplexitását,
- Alacsony ár.

Hátrányok:

- Fókusz a kötegorientált adatszállításon,
- Problémás támogatás,
- Problémás nagy adatmennyiségek kezelésénél,
- Kevésbé felhasználóbarát a kezdő fejlesztőknek,
- Kevésbé stabil, mint a másik két termék.

4.4. Összehasonlító táblázat

			
Fejlesztő	Microsoft	Informatica	Talend
Licensz	Ingyenes, de van belőle nagyvállalati verzió is, ami \$10000 feletti áron kapható	Fizetős, de van 30 napos kipróbálási lehetőség	Talend Open Studio ingyenesen letölthető, nyílt forráskódú
Felhasználói felület	Könnyű használat	Könnyű használat	Könnyű használat
Operációs rendszer támogatás	Windows, Windows Server	Windows, IBM AIX, Linux Red Hat	Linux Ubuntu, Linux Red Hat, Windows 10, Apple macOS
Felhő kompatibilitás	Igen	Igen	Igen
Közösségi támogatás	Aktív közösségi fórum és részletes dokumentáció	Közösségi és hivatalos támogatás	Hiányos támogatás az ingyenes verzió esetén, de nagy és aktív közösség a nagyvállalati verziónál

4.112. táblázat: Adatintegrációs eszközök összehasonlítása

[15] [13] [14] [17] [12] [16] [11]

5. Rendszerek kiválasztása és indoklás

Az következőkben azt fogom kifejteni, hogy melyik rendszereket választottam a szakdolgozatomban megvalósítandó probléma megoldására, illetve, hogy miért ezeket.

A választott adatbázis kezelő rendszer a Microsoft SQL Server, és ennek megfelelően az adatintegrációs eszköz pedig a Microsoft SQL Server Integration Services. Ennek három főbb oka van.

Az első és szerintem a legfontosabb, ahogy már említettem az eszközök ismertetésénél a kompatibilitás. Főleg egy adattárház - más területeken is - kialakításánál és karbantartásánál elengedhetetlen az a megspórolt munka, idő, energia és idegeskedés, amit egy ilyen jól integrált, könnyen használható és jól támogatott szoftver család tud nyújtani. Ennél a pontnál az is szerepet játszott, hogy az adatbázis kezelő rendszerek elég hasonló specifikációkkal rendelkeznek. Vannak persze különbségek köztük, amiket mindenképpen érdemes megnézni egy ilyen döntésnél. A végleges döntés előtt utána jártam a teljesítménybeli különbségnek is a két esélyes jelölt között, ami a MySQL és az MSSQL volt. A döntésemet segítette az „International Journal of Advanced Research in Computer and Communication Engineering” 2015. márciusi kiadványa, ahol Windows-os környezetben hasonlították össze a két eszközt több kategóriában is. Ezekből azt az eredményt kapták, hogy egyetlen dolog, amiben legyőzi a MySQL az MSSQL-t, az a 100 rekord beszúrásának az átlagos ideje. Ezt a kategóriát is csak 0.0006-dal nyerte meg, miközben az összes többiben alulmaradt.

A második a szoftverek hozzáférhetősége. A Microsoft SQL Server Management Studio teljesen ingyenesen elérhető és pár kattintással már fent is van egy Windows operációs rendszerű gépen, amivel én is dolgozni fogok. Ez az SQL Server Integration Services esetén is igaz.

A harmadik szempont a saját tudásom és gyakorlatom volt. Mindkét szoftvert jól megismertem a tanulmányaim során és szabadidőmben is foglalkoztam már velük, így az előző két szemponttal együtt könnyű volt meghozni a döntést, hogy melyik eszközöket fogom használni a projekt során.

6. Adattárház megtervezése

6.1. Adattárház fogalma

Az adattárháznak sokféle meghatározása létezik, de ami mindegyikben közös, hogy adattárháznak olyan nagy mennyiségű, megtisztított, historizált, csak olvasható adatok egy több dimenziós modellbe konvertált összességét nevezzük, ami segít a cég üzleti döntéseinek meghozásában. Emellett még fontos, hogy az akár több forrásból is érkező működési adatoktól elkülönítve tároljuk. [18]

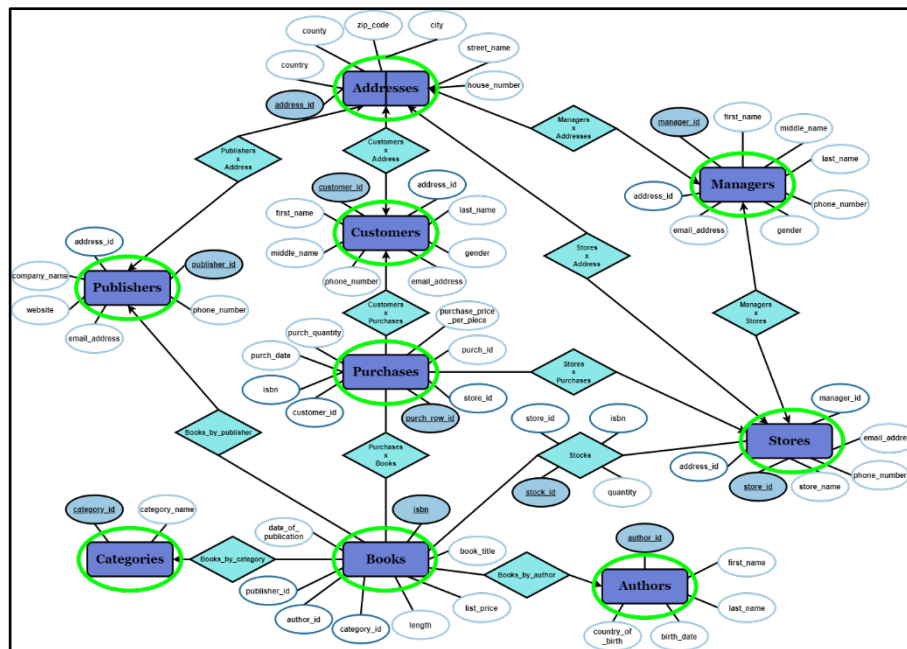
6.2. Üzleti igény ismertetése

Az üzleti szempont az volt, hogy kiderítsem, mekkora a boltok bevétele és hogy mennyi példányt adtak el a vizsgált időszakban. Ezeket lehet tovább pontosítani például úgy, hogy kategóriánként vagy kiadónként mennyi példányt sikerült eladni, illetve boltonként mekkora a bevétel. Ezekből olyan következtetéseket lehet majd tenni, hogy milyen kategóriájú könyveket éri meg forgalmazni, melyik kiadónak a könyvei fogynak a legjobban, esetleg ezekkel a kiadókkal erősíteni a kapcsolatot és megpróbálni olyan megegyezéseket kötni, amivel olcsóbban tudja a vállalat beszerezni a könyveket, így még nagyobb profitot tud később realizálni.

A másik fontos szempont lehet a vásárlók elemzése. Például, hogy honnan, melyik megyéből származnak a vevők. Ez alapján, ha kiderül, hogy egy messzebb lévő megyéből meglehetősen sok vevő érkezik, el lehet gondolkodni azon, hogy megérthető-e egy webshop létrehozása vagy további boltok nyitása az érintett megyékben. Akár azt is meg lehet vizsgálni, hogy a vásárlók milyen neműek, és a két nem milyen megoszlásban vásárol a különböző kategóriájú könyvek közül. Vagy akár azt is, hogy melyik nem költ többet a boltjainkban és ez alapján változtatni a boltok megjelenését, hogy a másik nemet is jobban megszólítsa a bolt. A feltehető kérdések száma közel végtelen és a kinyerhető tudás az adattárházból rengeteget ér, ha jól használjuk fel.

A harmadik szempont erősen kötődik az elsőhöz, csak más oldalról közelíti meg azt. Következtetni lehet a boltok bevételeiből olyasmire is, hogy melyik boltvezető végzi a legjobban a munkáját, melyik végzi kevésbé jól, és ezeknek mik lehetnek az okai. Ha feltárjuk ezeket a problémákat és kiderítjük az okaikat azzal is rendkívül sokat fejlődhet a vállalat és mi magunk is.

6.3. Forrásadatok kiválasztása



6.1. ábra: Kiválasztott táblák

6.4. Kialakított csillagséma



6.2. ábra: Kialakított csillagséma

6.5. A csillagsémáról röviden

A csillagséma az adattárházaknál használt többdimenziós modell, aminek két fő eleme van, a ténytábla és az azt körülvevő és vele összefüggésben lévő dimenzió táblák.. Elnevezését az alakjáról kapta, mert a ténytábla és a körülötte lévő dimenzió táblák egy csillag alakzatot formálnak.

A ténytáblában tároljuk a tényadatokat. Ezek olyan mérhető, numerikus adatok vagy mutatószámok, amelyeket elemezni szeretnénk. Általában ezek az értékek additívak (pl.: eladási ár), de lehetnek félig additívak is, mint például kedvezménykulcs százalékban kifejezve.

A dimenzió táblákban tároljuk azokat a tulajdonságokat, jellemzőket, amik segítségével a tényadatokat, mutatószámokat csoportosítani, illetve jellemezni tudjuk. Ezeket a jellemzőket egy dimenzió táblán belül hierarchiákba rendezhetjük. Ilyenek például a dátum dimenzióban a dátum, év, negyedév, hónap, nap vagy a bolt dimenzióban a város, utcanév felbontások.

6.6. Ténytábla részletezése

6.6.1. Vásárlások ténytábla (FACT_Purchases)

A táblában szereplő tényadatok:

- purch_quantity
- purch_price_per_piece
- purch_date

Ezekon kívül megtalálható a táblában az eredeti elsődleges kulcs, a „business key”, ami a ténytábla elsődleges kulcsa és a „surrogate key”-ek, amik a dimenzió táblákkal kötik össze a ténytáblát.

6.7. Dimenzió táblák részletezése

6.7.1. Dátum dimenzió (DIM_Date)

Ez egy előre legenerált dimenzió tábla, ami az idő-hierarchiára épül.

A táblában megtalálható adatok: date, year, quarter, month, day

6.7.2. Könyv dimenzió (DIM_Book)

Ez a dimenzió tábla a könyvekkel kapcsolatos adatokat tartalmazza a tények ebből a szempontból való megvizsgálásához, illetve itt van még eltárolva a listaár is, ami a későbbi számítások során még hasznos lehet.

A táblában megtalálható adatok: isbn, book_title, date_of_publication, length, list_price, category, author_name, publisher

6.7.3. Bolt dimenzió (DIM_Store)

Ez a dimenzió tábla a boltokkal kapcsolatos adatokat tartalmazza a tények ebből a szempontból való megvizsgálásához.

A táblában megtalálható adatok: store_id, store_name, manager_name, address_country, address_county, address_zip_code, address_city, address_street_name

6.7.4. Vásárló dimenzió (DIM_Customer)

Ez a dimenzió tábla a boltokkal kapcsolatos adatokat tartalmazza a tények ebből a szempontból való megvizsgálásához.

A táblában megtalálható adatok: customer_id, first_name, middle_name, last_name, gender, address_country, address_county, address_zip_code, address_city.

[19] [20] [18] [21] [22] [23] [24]

7. Adatvizualizációs módszerek és eszközök bemutatása

7.1. Adatvizualizáció ismertetése

Az információk átadásának egyik, ha nem a legfontosabb módja az adatok vagy információk vizuális megjelenítése képek, diagramok és grafikák segítségével. Röviden az információk megjelenítése grafikus formában. Ezeknek a vizualizációknak a fő célja, hogy értelmezhető formát adjanak az adatoknak és azokat érthetően kommunikálják az emberek felé. Ezeket az adatokat minél gyorsabban fel is kell dolgozni és analizálni, hiszen a mai világban bizonyos gazdasági, technikai vagy szervezeti döntések meghozásánál nem csak maga a döntés a lényeg, hanem az is, hogy azt a lehető leghamarabb meg tudjuk hozni. Itt jön elő az is, hogy az adatok vizuális megjelenítése és ennek használata nem csak az adattudósok és adatelemző szakemberek feladata, hanem bármilyen szakmában elengedhetetlen lehet, hogy tudjuk vizualizálni az adatainkat és ezeket értelmezni.

Az emberek alapvetően vizuális lények. Erre hagyatkozik az adatvizualizáció is, mégpedig úgy, hogy az emberi felfogás és megismerés képességét használja a megértés eléréséhez. Az ember képes észlelni a változásokat a mennyiségben, méretben, formában, alakban, színben és még sok másban is. Ezeket nevezzük preattentív tulajdonságoknak, amiket felhasznál a vizualizáció a céljai eléréséhez.

A vizualizációknak több fajtájuk van az egyszerű oszlop diagramtól kezdve az összetett, úgynevezett „dashboard”-okig (magyarul: műszerfal). A vizualizációk különböző fajtáit különböző esetekben érdemes használni. Egy közösségi média felületen érdemes egy-két látványosabb grafikont vagy infógrafikát használni a felhasználók figyelmének a megragadása céljából. Ezzel szemben egy dashboard-on minél egyszerűbb, letisztultabb és minimalistább diagramok az előnyösek. Az én projektemben a dashboard kapja a fő szerepet, így ezt fogom részletezni az alábbiakban.

A dashboardok három főbb típusát különböztetjük meg. Ezek a stratégiai, taktikai és az operatív dashboardok. Ez a felsorolás a vállalati hierarchia szinteknek megfelelően van csökkenő sorrendben.

A stratégiai típusra jellemző, hogy egyszerűen áttekinthető, magas az aggregáltsági szintje, a legfontosabb üzleti mutatókat tartalmazza és horizontális áttekintést biztosít. Ez a típus a felsővezetők igényeit fedi le.

A taktikai dashboard is egyszerűen és gyorsan áttekinthető, magas és közepes az aggregáltsági szintje, emellett egy adott üzletág adatait tartalmazza és vertikális áttekintést biztosít. Gyakori a lefűrés (drill-down) és a szűrés ennél a típusnál, ami azt jelenti, hogy meg tudja nézni az elemző az általa kiválasztott adatot egy alacsonyabb szinten és tud bizonyos szűrési feltételeket állítani. A taktikai típus a középvezetők, elemzők igényeit fedi le.

Az operatív a dashboardok legalacsonyabb szintje. Itt már jellemző az alacsony aggregáltsági szint, a vertikális áttekintés, lehetőség a széleskörű lefűrészesre és részletes szűrésre egy adott üzletág adataiban, aminek ebben az esetben van egy felső korlátja is. Ez a dashboard típus nagyon részletes betekintést biztosít az alacsony szintű adatokhoz.

Nagyon sok tényezőt kell figyelembe venni egy dashboard készítésekor. Ilyen például a preattentív tulajdonságok kihasználása, a megfelelő diagramtípusok használata, az elrendezés, a színezés és a színpaletta, a kimutatásokhoz nem kapcsolódó grafikai elemek minimalizálása és az általános dizájn alapelvek betartása. Ezek a gyors megértést és az átláthatóságot segítik.

A megfelelő diagramok alatt itt a preattentív diagramokra kell gondolni. Kerülni kell a kör alakú diagramokat, ilyen például a torta, a fánk és ezeknek a különböző változatai. Ezeket azért kell kerülni, mert az emberek nagyon lassan és nem is teljesen tudják értelmezni őket, főleg, ha négy-öt cikk is van egy kördiagramon belül. Szinte lehetetlen pontosan megmondani, hogy egy cikk körülbelül mekkora része a körnek. Ennek elkerülésére külön jelölni kell a körcikkeken a százalékos értéküket, ami rengeteg felesleges feliratot jelent egy diagramon. Érdemes a kördiagramokat helyettesíteni oszlopdiagramokkal, amin sokkal könnyebben fel tudja mérni az ember, hogy melyik a legnagyobb érték és hogy milyen arányuk van egymáshoz képest. A cél egy dashboard elkészítésénél, hogy maradjon egyszerű és átlátható, amit a kördiagramok nem könnyítenek meg.

Az elrendezésnél fontos megemlíteni, hogy az emberek a világ legtöbb részén balról jobbra olvasnak. Ez a dashboardoknál is így van. A legnagyobb hangsúly a bal felső sarokban és középen van, a legkisebb pedig a jobb alsó részen. Érdemes a legfontosabb adatokat, mutatókat a felső részen elhelyezni, ezekből is a legfontosabbakat a bal oldalon. Például egy bolt esetében az éves bevétel mutatóját a bal felső sarokba érdemes rakni, ezzel szemben a cég logója vagy egyáltalán ne szerepeljen vagy pedig a jobb alsó sarokban az összes dashboardon. Ugyan ez igaz a különböző szűrőkre is, hogy lehetőség szerint az összes nézetben ugyanott legyenek elhelyezve.

Színezésnél érdemes figyelembe venni, hogy az emberek elég nagy számban színtévesztők. Ilyenkor előnyös lehet ezzel is kalkulálni egy színpaletta összerakásakor és különböző formákkal is megkülönböztetni bizonyos eseteket. Kellemes hatást nyújt az is, ha minél kevesebb színt használunk. Így az eredmény sokkal átláthatóbb lesz.

Minimalizálni kell a logók, zavaró képek, háttérképek megjelenését a dashboardon. Ide tartozik még a színétmenetes diagram háttér, a diagramok külön dobozokba helyezése és a diagram rácsok kerülése is. Előnyösebb az üres helyek használata két diagram között, mint elválasztani őket egy vonallal, mivel így sokkal áttekinthetőbb képet kapunk.

A diagramok ajánlott és kevésbé ajánlott típusai dashboard készítés esetén:

Ajánlott diagram típusok:

- oszlopdiagram,
- vonaldiagram,
- szórásdiagram,
- pontdiagram,
- árnyékolt és szimbólum térkép
- és a teljesítmény diagram.

Kevésbé ajánlott diagram típusok:

- fánkdiagram,
- kördiagram,
- koncentrikus fánk diagram,
- buborék diagram
- és a szófelhők.

Az alábbi alponban a leggyakoribb diagram és grafikon fajtákat fogom bemutatni, amelyekkel az adatvizualizáció témakörében összefuthat az ember. Ezeken kívül rengeteg különböző fajta van, de ezek kerülnek leggyakrabban elő, akár egy egyszerű Excel munkafüzetben vagy egy összetettebb üzleti intelligencia eszköz használatakor.

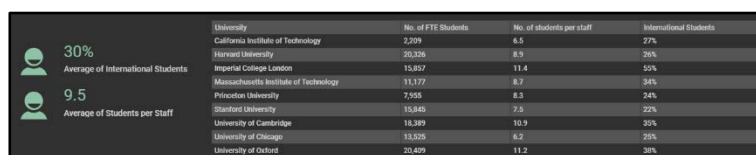
[25] [26] [27] [28] [29]

7.2. A legnépszerűbb diagram és grafikon fajták ismertetése

7.2.1. Táblázat és mutatószám

Az első és egyben az egyik legegyszerűbb, illetve a leggyakoribb vizualizációk egyike, amikor csak egy értéket vagy mutatószámot jelenítünk meg. Ez bármelyik dashboard fajtánál megjelenhet, viszont a stratégiainál és taktikainál a leggyakoribb, hiszen itt a közép- és felsővezetőknek gyakran van szükségük szimpla mutatószámokra, hogy lássák például, hogy a hónapban mekkora bevételre tett szert a vállalat.

Ide vettem a táblázatokat is, mert ez a kettő úgy mond kakukktojás a diagramok és grafikonok között. A táblázat nem túl gyakori, de előfordul főleg operatív dashboardok esetén. Ilyenkor tényleges adatokra vagyunk kíváncsiak táblázatos formában, mint például egy hő térkép táblázat. Így pontosan meg tudjuk nézni, hogy a sorokban milyen értékek szerepelnek egymás mellett.



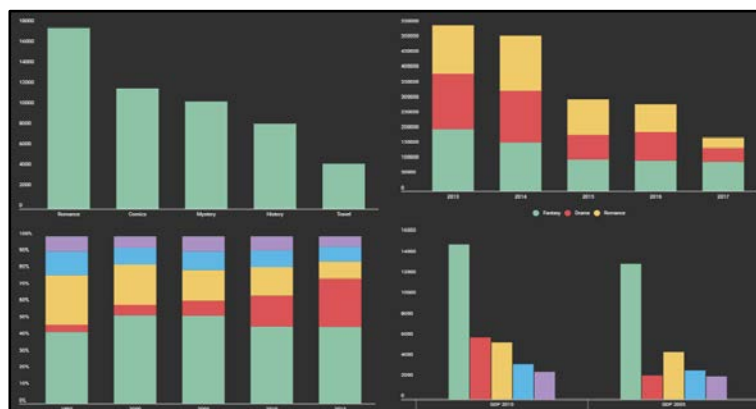
The dashboard displays two KPIs on the left: 'Average of International Students' at 30% and 'Average of Students per Staff' at 9.5. To the right is a table with four columns: University, No. of FTE Students, No. of students per staff, and International Students.

University	No. of FTE Students	No. of students per staff	International Students
California Institute of Technology	2,299	6.5	27%
Harvard University	20,326	8.5	26%
Imperial College London	15,857	11.4	55%
Massachusetts Institute of Technology	11,177	8.7	34%
Princeton University	7,565	8.3	24%
Stanford University	15,846	7.5	25%
University of Cambridge	18,389	10.2	32%
University of Chicago	13,525	8.2	25%
University of Oxford	20,409	11.2	38%

7.1. ábra: Táblázat és mutatószám példa

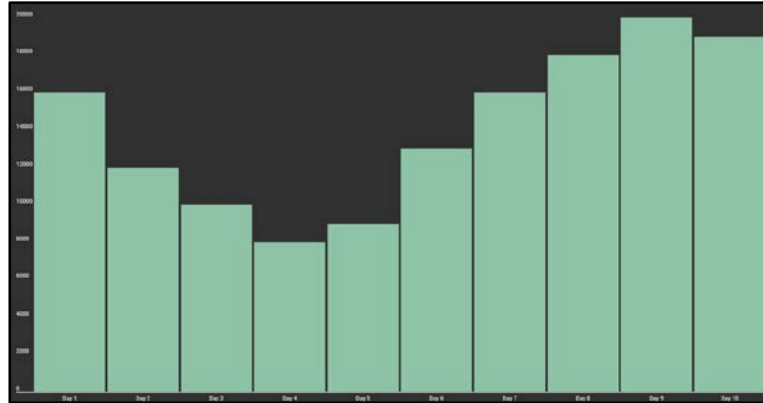
7.2.2. Oszlopdiagram és hisztogram

Az oszlopdiagramokat akkor használjuk, amikor két vagy több csoportot akarunk összehasonlítani. Ekkor az összehasonlítást az oszlopok magassága alapján végezzük, hiszen ez jelzi a csoporthoz társított értéket. Az oszlopdiagramok főbb fajtái: a halmozott-, 100%-ig halmozott- és a csoportosított oszlopdiagramok.



7.2. ábra: Példa az oszlopdiagram típusokra

A hisztogramokat is ide vettem a formai hasonlóság miatt. Jelentésben és tartalomban viszont a kettőnek semmi köze egymáshoz. A hisztogram az adatok eloszlását szemlélteti egy folyamatos intervallumon vagy időtartamon keresztül. Így jeleníti meg például a mobiltelefonokban lévő egészség applikáció, hogy a héten melyik nap hány lépést tettünk meg.



7.3. ábra: Példa hisztogramra

7.2.3. Sávdiaqram

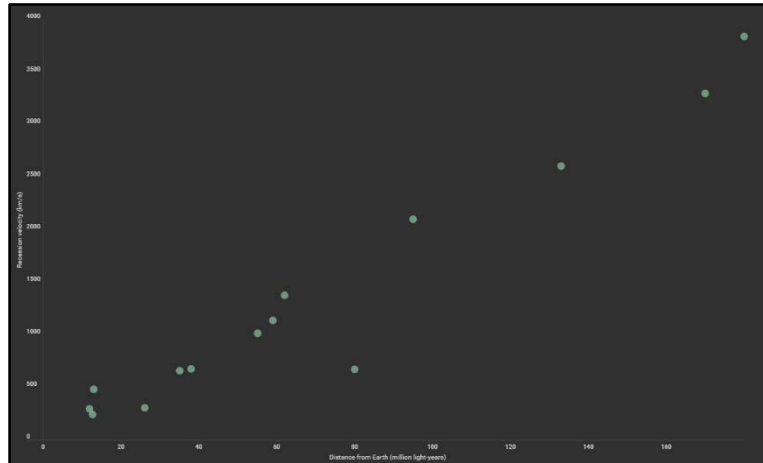
Az oszlopdiagramot és a sávdiaqramot lehetne egyként kezelni, hiszen ugyanarra a célra használjuk őket, viszont az oszlopdiagramnak van egy korlátja. Az emberi szemnek sokszor megterhelő, ha egymás mellett lát kilenc-tíz oszlopot ugyanazon a diagramon. Erre ad megoldást a sávdiaqram, ahol egymás alatt vannak sávok és ezeknek a hossza szolgál az összehasonlítás alapjaként. Ez sokkal kellemesebb látványt nyújt az emberi szem számára. Hasonlóan az oszlopdiagramhoz a sávdiaqramnak is vannak különböző típusai, mint például a halmozott-, 100%-ig halmozott-, és a csoportosított sávdiaqramok. Ezek közül a csoportosított arra szolgál, ha a különböző kategóriákat egy csoportosítás szerint akarjuk nézni. Például a különböző boltok bevételeit nézzük éves csoportosításokban.



7.4. ábra: Példa sávdiaqram típusokra

7.2.4. Szórásdiagram

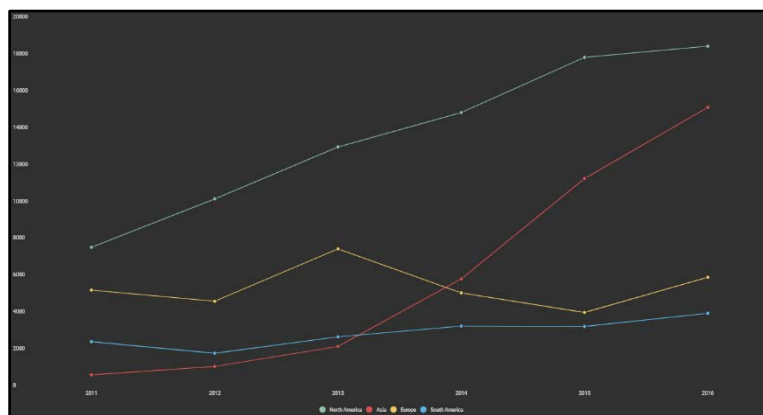
A szórásdiagram hatékony módja, hogy megvizsgáljuk két változó kapcsolatát. Az alapján tesszük ezt, hogy az egyik változó mennyire jelzi jól előre a másik változását vagy hajlamosak-e egymástól függetlenül változni. A diagram javítható később olyan módszerekkel, mint a klaszterelemzés, vagy trendvonalak segítségével.



7.5. ábra: Példa szórásdiagramra

7.2.5. Vonaldiagram

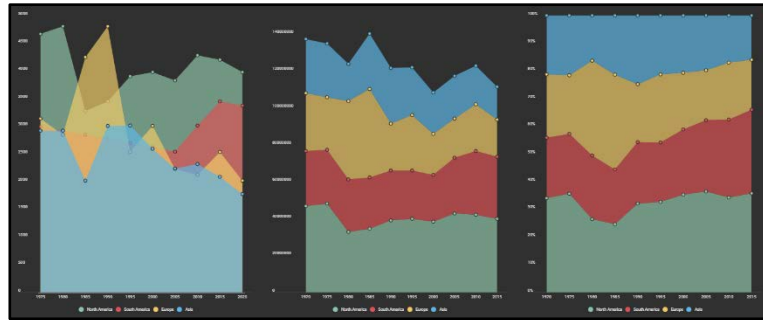
A vonaldiagram jó módja, hogy nyomon kövessük kvantitatív értékek változását egy intervallumon vagy időszakon belül. Jól meg lehet rajta figyelni, hogy az adatok hogyan változtak az idő előrehaladtával. Általában az X tengely jelenti az intervallumot vagy időszakot, az Y tengely pedig a kvantitatív értékeket. Érdeemes odafigyelni, hogy egy diagramon ne legyen több mint három-négy vonal, mert nagyon könnyen átláthatatlanná válik a diagramunk.



7.6. ábra: Példa vonaldiagramra

7.2.6. Területdiagram

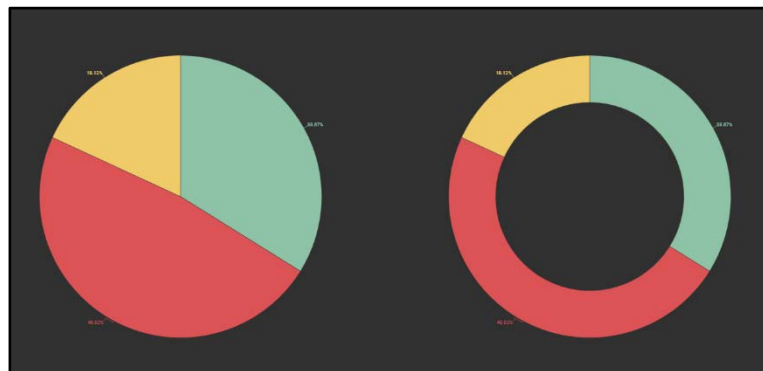
A területdiagram pont ugyan az, mint a vonaldiagram csak itt a vonal alatti terület ki van töltve. Ugyanúgy kvantitatív értékeket követhetünk nyomon egy intervallumon vagy időszakon keresztül. Három fő típusa van, az egyik a csoportosított területdiagram, aminél a különböző adatsorok mind a nulla értéktől kezdődően vannak ábrázolva. A második a halmozott, ahol az előző adatsortól kezdődően ábrázoljuk a következőt. A harmadik pedig a 100%-ig halmozott területdiagram.



7.7. ábra: Példa a területdiagram típusokra

7.2.7. Kör- és fánkdiagram

A kör- és fánkdiagramok lényegében ugyan azt a célt szolgálják, össze tudunk velük hasonlítani különböző kategóriákat úgy, hogy a kört a kategóriák százalékos megoszlása szerint felosztjuk körcikkekre. Ennek ellenére a kördiagramok nem a legmegfelelőbbek ilyen összehasonlításokra. Érdekes lehet egy kördiagram helyett inkább egy oszlopdiagramot vagy sávdiaagramot használni, mivel sokkal átláthatóbb és könnyebb összehasonlítani kategóriákat, ha oszlopokat látunk egymás mellett vagy alatt.



7.8. ábra: Példa kör- és fánkdiagramra

7.2.8. Fatérkép diagram

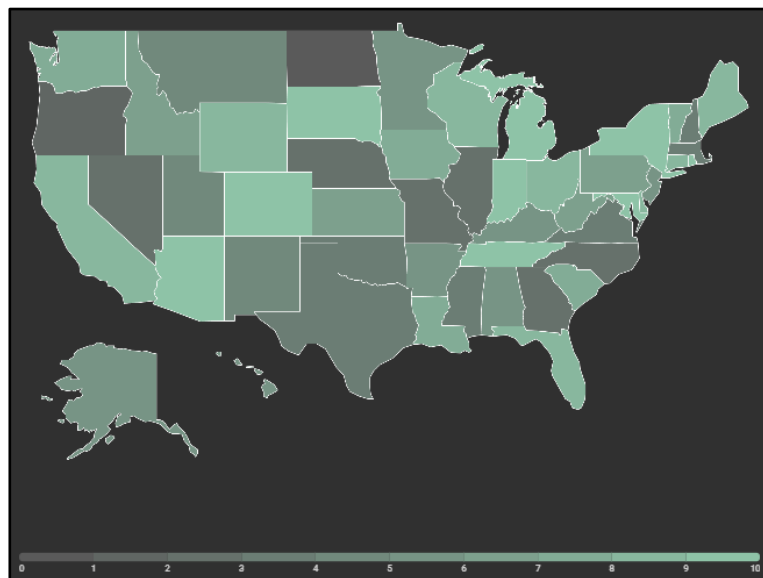
A fatérkép diagram egyszerre alkalmas a hierarchikus struktúra megjelenítésére, mint a fadiagram és ezen belül a kategóriák mennyiségeinek megjelenítésére is. Ebből adódóan a kategóriák értéke megegyezik az alkategóriái értékének összegével. Ez a példán nem feltétlen látszik jól, hiszen itt két hierarchia szint van csak feltüntetve, az egyik a Föld területe, a másik a kontinensek területei.



7.9. ábra: Példa fatérkép diagramra

7.2.9. Térképdiaagram

A térképdiaagramoknak is több fajtája van, viszont itt csak az úgynevezett choropleth térképről lesz szó. Ez is az egyike a legnépszerűbb diagram típusoknak, hiszen remekül lehet vele ábrázolni, hogy például a megrendelések a vállalatnak milyen területekről érkeztek, honnan a legtöbb és honnan a legkevesebb.



7.10. ábra: Példa térképdiaagramra

7.2.10. Teljesítmény diagram

A teljesítmény diagramok közül is kettőről lesz szó, az egyik a mérő műszerekre hasonlító úgynevezett „Gauge chart” és a „Bullet chart”. Mindkettő arra szolgál, hogy nyomon tudjunk követni folyamatokat. A gyakorlatban érdekesebb „Bullet chart”-ot használni „Gauge chart” helyett. Ugyanabból az okból, mint amiért a kördiagramok helyett érdemes oszlopdiagramokat használni. Ezt a két diagram típust leginkább dashboardokon használják.



7.11. ábra: Példa teljesítmény diagramokra

[25] [26] [27] [28] [29] [30] [31] [32]

7.3. Analitika és üzleti intelligencia eszközök összehasonlítása

7.3.1. Microsoft Power BI

A Power BI a Microsoft felhő-alapú üzleti intelligencia és adatelemző szolgáltatása. Röviden a nyersadatok hasznos és érthető információvá alakításáért felelős eszköz vizualizációk segítségével. A Microsoft termékeként rendkívül széles körben használt többek között azért is, mert egy közzétett riportot vagy dashboardot bárhol el tudunk érni, mivel a szolgáltatás elérhető a gépünkön, a telefonunkon applikáció formájában és a weben is. [33]

Előnyök:

- **Megfizethetőség:** az egyik nagy előnye, hogy ingyenesen elérhető bárki számára. A legnagyobb profit mégis a fizetős verzióval jár, hiszen itt megkapjuk a dashboardok és riportok megosztásának lehetőségét, emellett rendkívül széles eszközkészletből tudunk válogatni mindössze tíz dollárért. Ha van Office 365 E5 csomagunk (nagyvállalati csomag) akkor külön költség nélkül is hozzáférünk ezekhez a fizetős funkciókhoz sok más mellett.
- **Integráció:** a Microsoft termékek egyik része, hogy remekül használhatók egymással. Ezen kívül viszont rengeteg adatforrás eszközhöz tudunk egyszerűen csatlakozni, mint az MS Excel, Microsoft adatbázisok, XML és JSON formátumok és a Google-nek is a különböző szoftvereit akadálymentesen tudjuk használni a Power BI-al.
- **Hozzáférhetőség:** könnyen hozzáférhető bármilyen eszközről és bárhonnan, legyen az iOS, Android, Windows vagy web. A legszebb ebben az, hogy nem veszít egy vizualizáció az értékéből és használhatóságából attól, hogy éppen egy telefonon vagy a gépünkön nyitjuk-e meg.
- **Aktív közösség:** rengeteg fórumot és segítséget találni az interneten, amivel fejleszteni tudjuk magunkat és segítséget tudunk kérni, ha elakadunk egy folyamat során.

Hátrányok:

- **Komplexitás:** egy kezdőnek nehéz lehet átlátni a funkciókat és szolgáltatásokat elsőre.
- **Nagy adatmennyiségek:** az egyszer importálható adatmennyiség a Power BI-ban 1 GB, ami a fizetős verzióval bővíthető. Ha feldaraboljuk a feldolgozandó adatkészletet az is egy megoldás lehet. Ez a limitáció csökkentheti a riportok és dashboardok komplexitását.

- **Felhasználói élmény:** a felhasználói felülete túl nagy helyet foglal el a vizualizációkat sokszor eltakarva és nem enged sok testreszabhatósági lehetőséget. [34] [33] [35]

7.3.2. Tableau

Hasonlóan a Power BI-hoz ez is az egyik legnépszerűbb és leghasználhatóbb üzleti intelligencia eszközök egyike. A Tableau egy gyorsan és erősen növekedő adatvizualizációs szolgáltatás csomag, ami a következőket tartalmazza: Tableau Desktop, Tableau Public, Tableau Online, Tableau Server, és Tableau Reader. Ez a csomag elősegíti a könnyű és gyors megértését az adatoknak egy vállalaton belül akármelyik szintről is legyen szó. [36]

Előnyök:

- **Vizuális megjelenítési képességek:** szembe állítva a nagyobb vállalatok, mint az Oracle vagy az IBM eszközeivel a Tableau olyan minőségi stílust és szabadságot nyújt a felhasználóknak, ami miatt nem kérdés, hogy miért az egyik legnépszerűbb megoldás a piacon.
- **Könnyű használat:** a kezdő felhasználók is könnyen tudnak boldogulni a használatakor az egyszerű „húzd és ejtsd” módszernek és az átlátható felhasználói felületnek köszönhetően. Az eszköz próbálja redukálni azon esetek számát, amikor a felhasználó az informatikai osztály segítségét kell kérje, ezzel is javítva a felhasználói élményt.
- **Magas teljesítmény:** gyorsan kezel nagyobb adat és információ mennyiségeket is. Akár több millió rekordot is tud kezelni anélkül, hogy ez a dashboardok teljesítményén és dinamikus működésén rontana.
- **Kapcsolatok külső forrásokkal:** az eszköz könnyű elérést biztosít számos adatforráshoz, az egyszerű és általános adatforrásoktól kezdve a nagyobb forrásokhoz, mint a Hadoop vagy SAP rendszerek.
- **Riportok prezentálása:** a Tableau rendelkezik egy olyan tulajdonsággal is, hogy a dashboardokat a különböző eszközöknek megfelelően testre lehet szabni, hogyha valaki tableten vagy mobilon nézi őket, akkor is ugyanolyan élménye legyen, mintha számítógépen tenné.

Hátrányok:

- **Magas ár:** a nagyobb vállalatok árait nem éri el, de így is sok kisebb cégnek túl költséges lehet beruházni a Tableau szolgáltatására.
- **Rugalmatlan árazás:** az értékesítés kevésbé veszi figyelembe azt a tény, hogy különböző vállalatoknak különböző funkciók kellenek csak a szolgáltatásból és vannak, amit egyáltalán nem használnak belőlük.
- **Importálás:** más eszközökkel összehasonlítva a Tableau kevésbé támogatja az egyedi vizualizációk importálását más eszközökből.

- **Információs osztály segítsége:** igaz a használatban kevés segítséget kell kérnie egy átlagos felhasználónak a Tableau használatakor, viszont a konfigurációnál és bizonyos esetekben mikor egy bonyolultabb SQL lekérdezést kell megvalósítani, akkor szükség lehet egy hozzáértő szakember segítségére.
- **Előzmények:** ugyan már támogatja az előzmények visszaállítását, de ezt csak a legutolsó verziók tartalmazzák, ha korábbi verziót használ a felhasználó, akkor ez nem lehetséges. [37] [34] [36]

7.3.3. QlikView

Az előzőekhez hasonlóan a QlikView is egy üzleti intelligencia eszköz, ami a forrásadatainkból információt és tudást gyárt a felhasználók számára. Legfontosabb szolgáltatásai közé tartozik az eseti lekérdezések lehetősége és a gyors döntést támogató könnyű hozzáférés az adatokhoz. Ezen kívül a QlikView egy új megközelítésű memória alapú architektúrát használ az adatok tárolására. Ez lehetővé teszi a tárolás és a lekérdezések hatékonyságának növelését. Ezt úgy teszi, hogy a forrásokból az adatok a RAM-ba töltődnek és innen keresi vissza őket az eszköz, ha szükség van rájuk. Ha nagyobb adathalmaz forrásokkal dolgozunk, mint a Teradata vagy Cloudera esetében, akkor a teljes adatmennyiség betöltése nélkül is tudunk böngészni bennük ennek a technológiának köszönhetően. A QlikView nem véletlenül került a korábban már említett Gartner üzleti intelligencia eszközökkel foglalkozó összehasonlításában a „Vezetők” közé, de mégis közel az „Újítók”-hoz, hiszen a memórián belüli számítások és tárolás egyik úttörője. [38]

Előnyök:

- **Adatfeltárás:** az egyik legfontosabb szolgáltatása a Qlik-nek a képesség a hatékony adatfeltárára. Ez azt jelenti, hogy egy természetes kereső folyamat van beépítve a szoftverbe, ami nem követ semmilyen előre definiált szabályt vagy mintát. Ez megkönnyíti és felgyorsítja az adatfeltárást és az adatok analizálását.
- **Memórián belüli technológia:** a fentebb említett módszer, amivel a forrásokból nyert adatok a RAM-ba töltődnek és innen használja őket a szoftver. Ez azért úttörő, mert az átlagos üzleti intelligencia eszközök egy jelentősen lassabb módszert használnak, amivel az adatokat merevlemez alapúan tárolják egy köztes állapotban. Adattárházak esetén úgynevezett „data cubes”-okban, adatkockákban.
- **Mobil támogatás:** akárcsak az előző kettő, úgy ez is támogatja a telefonos interaktív adatfeltárást és felfedezését az adatinknak. Támogatja az iOS, Android és Windows-os mobiltelefon operációs rendszereket is.
- **Ár és rendszerkövetelmény:** a szoftvernek van ingyenesen hozzáférhető próbaidőszaka, de a termék ára kevesebb mint fele a

Tableau-nek, így jó választás lehet kisebb vállalkozásoknak. Emellett a rendszerkövetelménye is alacsonynak mondható a hivatalos oldalon feltüntetett alapján.

- **Tömörítés:** rendszerbetöltéskor csak a hasznos adatbitek töltődnek be és az összes feleslegeset eltávolítja a rendszer. A QlikView szoftver a teljes adatmennyiség méretét a 10%-ra tömöríti az eredeti méretnek. A tömörítés után csak az elemzés során hasznos adatok maradnak meg.

Hátrányok:

- **Valós idejű adatelemzés:** sok esetben nem túl hatékony.
- **Memória, mint korlát:** a memórián belüli tárolásnak a számítógép memória mérete szab határt. Az hogy mennyi mezőt, táblát, cellát és sort tudunk tárolni az mind a memória méretétől függ.
- **Fejlesztői ismeret szükséges:** mivel a Qlik egy saját SQL lekérdezési nyelvet használ, így a vállalatnak egy Qlik-ben képzett szakemberre van szüksége.
- **ETL hiánya:** nem kínál semmilyen ETL eszközt a komplexebb adatintegrációs feladatokhoz.
- **Egyszerűség:** egyszerű riport és analizáló eszközként nem ad lehetőséget az adatbázis írására. [34] [37] [38] [39]

7.4. Adatvizualizációs eszköz kiválasztása és indoklás

Az előzőleg kiválasztott adatbázis- és ETL rendszerekhez legjobban a Microsoft Power BI megoldása illik. Ennek ellenére az én választásom nem egy üzleti intelligencia eszköz lesz, hanem egy különálló webalkalmazás fejlesztése a szakdolgozat vizualizációs feladatainak ellátására. A webalkalmazásban az elemző szakemberek hozzáférnek az előre létrehozott vizualizációkhoz, amik alapján nyomon tudják követni az üzleti folyamatokat és meg tudják hozni döntéseiket. A webalkalmazásról részletesebben a következő fejezben lesz szó. Ott fogom részletezni a megvalósításhoz szükséges fejlesztői eszközöket és bemutatni a látványtervet.

8. Technológiák a webalkalmazással történő megvalósításhoz

8.1. HTML ismertetése

A HTML a „Hypertext Markup Language” rövidítése és a web legalapvetőbb építőeleme. Meghatározza a weboldalon található tartalom strukturális felépítését és lehetőséget ad cím, szekciók, paragrafusok, linkek, képek, videók és számos más tartalom létrehozására és elhelyezésére az oldalon.

„Hypertext” jelentése: Olyan szöveg, ami egy linket tartalmaz. Ez teszi lehetővé, hogy több weboldalt vagy HTML állományt egymással összekapcsoljunk.

„Markup Language” jelentése: Magyarul jelelőnyelvnek szokás nevezni. Ez egy olyan számítógépes nyelv, amivel egy szöveges dokumentum elrendezését és formázási szabályait tudjuk megszabni. Ezt különböző címkék és attribútumok segítségével érjük el. Ezekkel a címkékkel tudunk képeket és egyéb tartalmakat megjeleníteni egy weboldalon. Emellett sokkal interaktívabbá és dinamikusabbá tudja tenni a szöveg olvasását.

A legnagyobb fejlődés a nyelv élete során a HTML5 bevezetése volt 2014-ben. Röviden ez rengeteg új címkét adott a nyelvhez, amivel sokkal jobban ki tudjuk fejezni a weboldalaink tartalmát. A HTML-en kívül számos egyéb technológiát és nyelvet használnak egy weboldal vagy webalkalmazás létrehozásakor. [40] [41] [42]

8.2. CSS ismertetése

A CSS a „Cascading Style Sheets” kifejezés rövidítése és a weboldalak stílusát és kinézetét lehet vele beállítani. Három fajtája van, az „Inline”, az „Internal” vagy „Embedded” és az „External” CSS. Ebből az első, amikor a törzs címkén belül definiáljuk a beállításokat. A második, mikor a fejléc címkén belül állítjuk be ezeket. És a harmadik, mikor egy külön CSS dokumentumot hozunk létre ezeknek a beállításoknak. Manapság a harmadik variáció az elterjedt, mivel sokkal átláthatóbb és lehetőséget nyújt a különböző stílus csomagok importálására más HTML dokumentumok létrehozásakor is. [43] [44]

8.3. JavaScript ismertetése

A JavaScript egy általánosan és széles körben használt programozási nyelv a webfejlesztés világában. Célja, hogy dinamikus és interaktív dolgokat, funkcionalitást adjon a weboldalakhoz. Ezen kívül a JavaScript egy kliens-oldali szkript nyelv, ami azt jelenti, hogy a forráskód a kliens böngészőjében kerül feldolgozásra és nem a webszerveren. Legtöbbször a weboldal tartalmának módosítására és a különböző felhasználói cselekedetekre adandó válaszok megvalósítására használják. [45] [46]

8.4. Blazor keretrendszer ismertetése

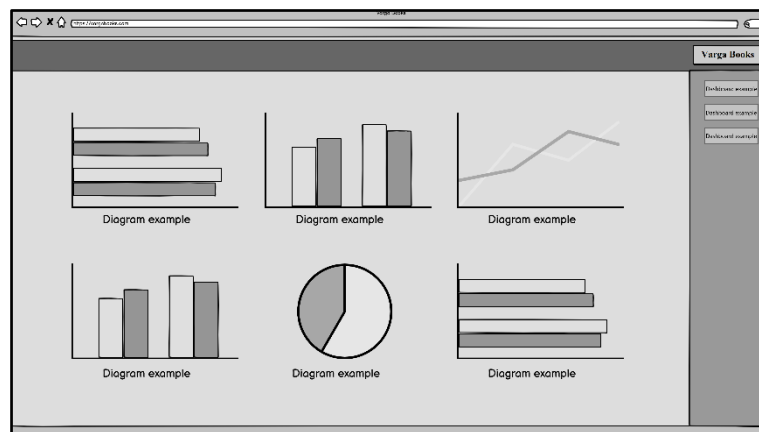
A Blazor a Microsoft által fejlesztett webes keretrendszer, mely felveszi a versenyt a mezőny vezető rendszerekkel, mint a React vagy az Angular. A különbség a Blazor

és az előbb említett keretrendszerek között, hogy míg a másik kettő JavaScript-et használ, addig a Blazor .NET alapokra épül és biztosítja a fejlesztőknek, hogy C# és HTML segítségével írjanak webalkalmazásokat. Így lehetségessé válik az, hogy kizárólag C# nyelv használatával fejlesszük le a háttérfolyamatokat és magát a webalkalmazást is.

A Blazornak jelenleg két megvalósítási lehetősége van. Az egyik a kliens-oldali a másik pedig a szerver-oldali. A kettő között a fő különbség, hogy míg a kliens-oldali verzió a WebAssembly technológia segítségével a .NET kódot közvetlenül a kliens böngészőben futtatja, addig a szerver-oldali variációban a munka nagyrészt a szerver végzi és egy úgynevezett SignalR kapcsolat segítségével kommunikál a kliens böngészővel.

A dolgozatomban én a szerver-oldali verziót fogom használni, mivel a legfőbb szempont a gyorsaság és az egyszerűség. Azaz, hogy a felhasználó mikor megnyitja az oldalt, akkor az minél gyorsabban betöltsön és a kliens böngészője ne legyen leterhelve letöltendő adatokkal. Mivel egy kisebb cégről beszélünk, így a kliens-oldali megoldásnak azt az előnyét sem tudom kihasználni, hogy sok felhasználó esetén mentesíti a szerveret a rengeteg számítás elvégzésétől. [47] [48] [49]

8.5. Webalkalmazás grafikus terve

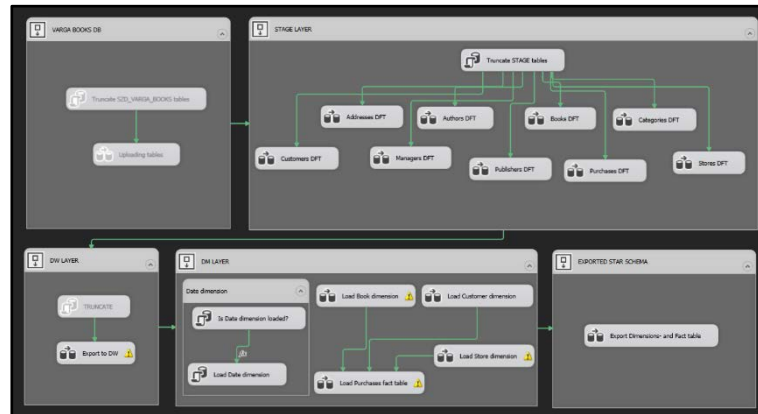


8.1. ábra: Webalkalmazás grafikus terve számítógépes nézetben

9. Adatintegrációs folyamat megvalósítása (ETL)

Első lépésként -mint mindegyik rétegemnél- létrehoztam egy „Sequence Container”-t (szekvenciális konténer). Ez a komponens képviseli majd a különböző rétegeket (Stage, Data Warehouse és a Data Mart) és ezek segítségével kapcsolom össze a fázisokat, hogy megadjam az integrációs eszköznek a folyamatok sorrendjét.

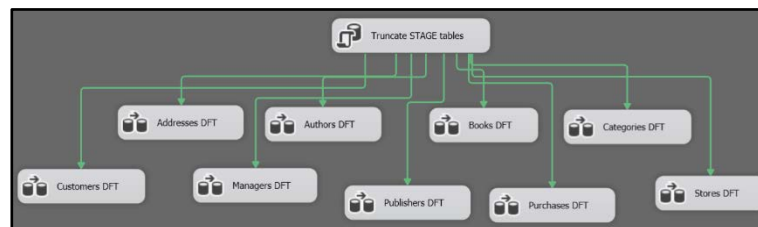
A következő lépés, hogy a „Connection Manager”-hez (kapcsolatkezelő) hozzáadtam a négy lokális adatbázist, hiszen innen, illetve ide fognak érkezni az adatok.



9.1. ábra: Teljes ETL folyamat

9.1. Stage réteg

Ebben a rétegben történik az adattárházba érkező adatok átmeneti tárolása, így itt minden adat szöveges formátumban tárolódik, és minden alkalommal kiürítésre, illetve újra feltöltésre kerül.



9.2. ábra: Stage réteg folyamatai

A „Stage Layer”-en belül az első lépésként egy „Execute SQL Task”-ot futtattam, ami lényegében annyit csinál, hogy végrehajt bármilyen SQL parancsot, amit megadunk neki.

Az itt szereplő komponensnek az a feladata, hogy kiüríti a táblákat, mindezt úgy, hogy a kapcsolatkezelőben megadott forrásadatbázisban lefuttat minden táblára egy „TRUNCATE” parancsot.

A táblák ürítése után minden táblához létrehoztam egy-egy „Data Flow Task”-ot, amire azért volt szükség, hogy az eredeti adatbázisomból áttöltssem a tábláim tartalmát a „Stage” rétegembe.

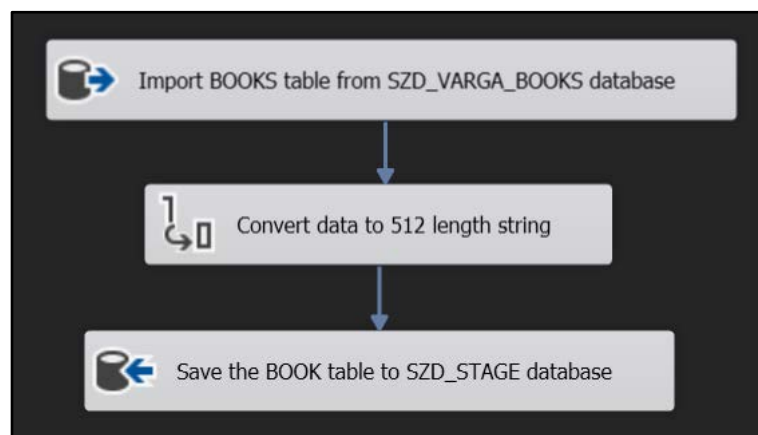
A „Data Flow Task”-okon belül három dologra volt szükség, egy „OLE DB Source” (forrás), egy Data Conversion (adatátalakítás) és egy „OLE DB Destination”(cél) komponensre.

Az „OLE DB Source” objektumon beállítottam, hogy melyik adatbázisból, és azon belül melyik táblából és melyik oszlopokat szeretném beolvasni. Ebben az esetben a forrás adatbázisomon belül az összes táblából az összes oszlopot.

Ezután a „Data Conversion” komponensnél azt konfiguráltam be, hogy milyen adattípusokat tartalmaznak a beolvasott oszlopok. A Stage réteg esetén mindegyik táblánál minden oszlop 512 karakter hosszúságú, „string[DT_STR]” típusú lesz, mivel a könyvek címei néhány helyen akár 350-400 karaktert is elérnek.

Az utolsó lépés pedig az, hogy kivezetjük az átalakított adatainkat egy OLE DB Destination segítségével, viszont figyelni kell arra, hogy a Data Conversion-ből származó értékeket párosítjuk össze a nekik megfelelő oszloppal a STA_Addresses táblában és nem az eredeti adatokat, amik még a forrásadatbázisból érkeztek.

A folyamatot ezen az ábrán szemléltetem:



9.3. ábra: Stage rétegben történő folyamatok egy táblán szemléltetve

9.2. Data Warehouse réteg

Az adattárház rétegben több minden is változik az előző réteghez képest. Itt a táblák adatai már típusosan szerepelnek, ezen felül pedig minden tábla kap egy „Surrogate Key” oszlopot, ami egyedileg azonosítja az adattárház sorait.

Ebben a rétegben fontos változás, hogy nem szabad kiüríteni a táblákat, mint az előző rétegben, hiszen itt halmozódik fel az üzleti kérdések megválaszolását segítő hatalmas adatmennyiség. A képen ugyan szerepel egy „TRUNCATE” feladatot ellátó „Execute SQL Task” komponens, de csak teszt jelleggel, és az éles futtatások során már végig szigorúan kikapcsolt állapotban.

A bemutatás során a „Books” tábla folyamatát fogom részletezni, mert ez a legösszetettebb és legérdekesebb a folyamatok közül. Itt megfigyelhető, hogy hogyan lehet a historikusan tárolandó adatokkal dolgozni.



9.4. ábra: Transform fázis a „Books” táblán bemutatva

A forrás ebben az esetben már Stage rétegben található „STA_Books” táblám, amin a betöltést követően a „Data Conversion” komponens segítségével minden oszlopnak beállítjuk a végleges, „Mapping” fájlban felvázolt típusát.

A következő komponens ennek a rétegnek a legfontosabb része, hiszen ez dönti el, hogy új adat érkezett-e a táblákba, illetve azt, hogy a régi adatokon történt-e bármilyen változás. Ha igen, akkor vagy végrehajtja a változtatást vagy elvégzi a módosításokat a historikus tároláshoz és beszúrja új sorként három új oszloppal kiegészítve: mikortól érvényes a sor, meddig érvényes és hogy éppen aktív-e az aktuális rekord. Ez a komponens a „Slowly Changing Dimension”, röviden SCD.

Az SCD-ben az első lépés beállítani a bemenetek és a kimenetek párosításait, és kiválasztani a táblák üzleti kulcsait, ami esetünkben a könyvek ISBN azonosítója. Ezek után be kell állítani az oszlopoknak a megfelelő változási típusát, ez lehet „Changing Attribute” (változó attribútum), „Historical Attribute”(historikus attribútum) és „Fixed Attribute” (rögzített attribútum).

Ezen tábla esetén a változó- és a historikus változási típusokat használtam, hiszen a könyvek lista árának változását érdemes lehet historikusan tárolni, hogy később az elemzők észrevehessenek olyan összefüggéseket, amik a változás megtörténtekor nem biztos, hogy bárkinek is feltűnnek, de a jövőben lehet azok alapján következtetni más árváltozásokra.

Az SCD-t követően három irányba megy tovább a folyamat. Az első eset, mikor az adat már szerepel a táblában, de megváltozik a „list_price” oszlopában szereplő érték. ez a historikus ág. A „VALID_TO” függvény komponens a megváltozott adatnak beállítja a lejárat dátumát a folyamat lefutásának időpontjára. Ez követően az „OLE DB Command” komponens végrehajtja a megfelelő rekordokon a változtatást.

A következő ágon mennek tovább a teljesen új rekordok. A „valid_to” oszlop ezen az ágon alapértelmezetten „NULL” értékkel kerül a táblába, hiszen az új rekordoknál nem ismert az érvényesség vége.

Az utolsó ág a „Changing Attribute”-nak jelölt oszlopok változása esetén felülírja azokat a rekordokat, ahol történt változás az így jelölt oszlopokon.

A „Union All” résznél egyesítjük a három ágról érkező adatokat.

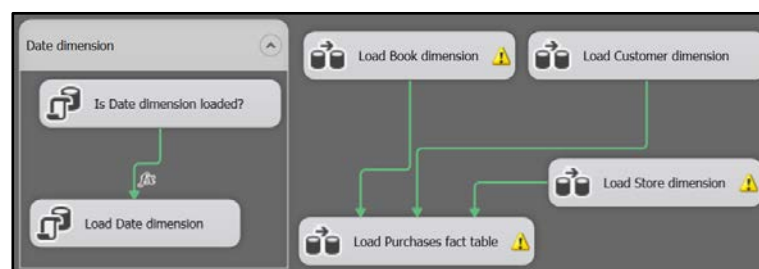
Ezt követően két „Derived Column” függvény található, amiből az első hozzáadja és beállítja az érkező adatoknál a VALID_FROM mezőt a futtatás napjának dátumára. A második pedig hozzáadja és beállítja az ACTIVE_FG mező értéket 1-re, ha az adat aktív, 0-ra pedig, ha már inaktív. Ezt a „VALID_TO” mező értéke alapján dönti el. Ha „NULL” érték van benne, akkor még aktív, ha a benne lévő dátum érték kisebb, mint a futtatás dátuma, akkor pedig inaktívvá állítja.

Az utolsó lépés betölteni az elkészült rekordokat a „DW_Books” táblába. A többi tábla esetén csak legfeljebb két ág szerepel, mivel csak ebben az egy táblában van historikus adattárolás.

Az egyetlen táblám, ahol csak egy ág szükséges, az a vásárlások tábla, hiszen itt nem történik semmiféle adatváltozás, csak új rekordokat töltünk bele minden alkalommal.

9.3. Data Mart réteg

A „Data Mart” rétegben először a dimenzió táblák kerülnek feltöltésre, csak utána történik a ténytábla feltöltése adatokkal.



9.5. ábra: DM réteg folyamatai

A dátum dimenzió egy egyedi dimenzió, mert itt nem a forrásadatbázisból származó adatok szerepelnek, hanem két szélsőérték között található dátumok egy tárolt eljárás által generálva. Emiatt a különbség miatt itt először ellenőrizni kell, hogy fel van-e töltve a dátum dimenzió, ezt a feladatot végzi el az „Execute SQL Task” komponens. Ha fel van töltve, akkor nem történik semmi, ha nincs, akkor feltöltésre kerül az alatta található „Load Date Dimension” komponens segítségével.

Azt, hogy a dátum dimenzió fel van-e töltve egy egyszerű „COUNT” segítségével nézem meg, aminek az eredményét kimentem egy egész szám típusú változóba, majd

ezt a változót egy függvény segítségével ellenőrzöm, hogy egyenlő-e nullával. Ha egyenlő akkor lefut a tábla feltöltése.

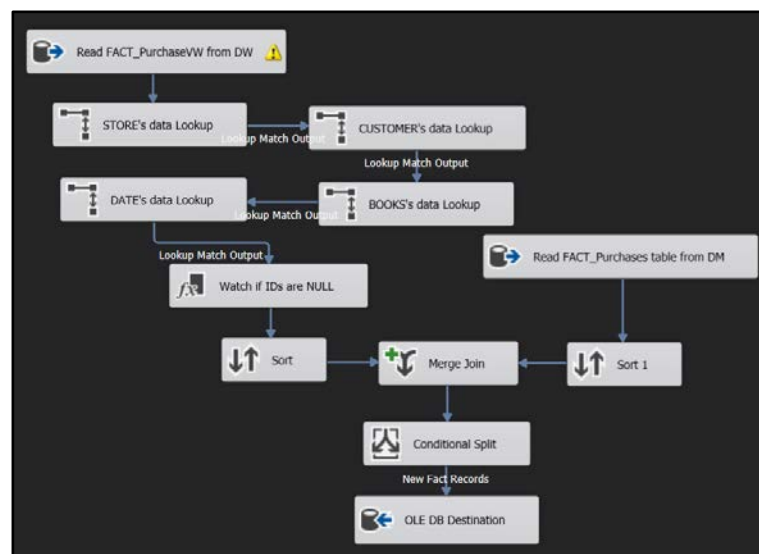
A többi dimenzió és a ténytábla feltöltésekor nézetek lesznek a források, hiszen itt az alap kilenc táblából kell három dimenzió táblát és egy ténytáblát csinálnom. Itt annyi történik, hogy a nézetekben kigyűjtöm az adott dimenzió táblához kellő oszlopokat a többi táblából és az „OLE DB Source”-nál ezek a nézetek lesznek a források.

Ennél a rétegnél a nézetek nem tartalmaznak semmiféle „Surrogate Key”-t, csak magát az eredeti tábla kulcsát, azaz a „Business Key”-eket. Mivel itt már megfelelő típusban vannak a rekordok a „Data Conversion” részt el is lehetne hagyni, viszont néhány oszlopnál történtek összefüszések, így ezen oszlopok hosszát itt korrigálni kell a betöltéshez.

Az SCD komponens itt csak annyit csinál a dimenzióknál, hogy ha változott bármelyik rekordon belül egy érték, akkor azt megváltoztatja az aktív értékre, ha pedig új rekord érkezik, azt hozzáadja az adott dimenzióhoz.

Mindegyik dimenzió táblánál lényegében ugyanaz történik, néhány kisebb adatkonvertálástól eltekintve.

A lényeges rész a ténytáblánál van, hiszen ott nem csak egy nézet a forrás, hanem maga a ténytábla is.



9.6. ábra: Ténytáblán elvégzett műveletek

A ténytábla feltöltése két ágon indul el. A baloldali ágon a ténytábla nézete a forrás. A jobboldali ágon pedig maga a „Data Mart” rétegben található ténytábla.

A baloldali ágon szerepel négy darab „Lookup” elem, amik a továbbhozott „Business Key” alapján összepárosítják a nézetben található azonosítóval a dimenzió táblában található „Business Key”-eket. Ezután beszúrom a rekordhoz tartozó dimenzió tábla elsődleges kulcs értékét a ténytábla nézetének megfelelő mezőjébe .

Ezt az összes dimenzió táblával meg kell csinálni, majd rendezni a jobb és baloldalt is a ténytábla, illetve a nézet elsődleges kulcsa, azaz a „Surrogate Key” alapján.

A két rendezést összevonjuk egy „Merge Join” komponens segítségével, amelynek feltétele, hogy a két bemeneti ág rendezve legyen, és a két érték, ami alapján rendezve van, azonos típusú legyen.

A „Merge Join”-ban kiválasztom a két ágról érkező adatokból, hogy miket kell tovább vinni, viszont fontos, hogy az összekapcsolás típusánál a „Left outer join”-t kell kiválasztani, mert a nézet a baloldali bemenetére van kötve.

Az utolsó előtti lépés egy „Conditional Split”, amiben két eset közül csak akkor foglalkozunk tovább a rekorddal, ha a ténytábla elsődleges kulcsa „NULL” értékű, ekkor bekerül új rekordként a ténytáblába.

Ezzel elkészült a csillagséma és annak feltöltése az adattárház alapján.

10. Adatvizualizációkat megjelenítő webalkalmazás megvalósítása

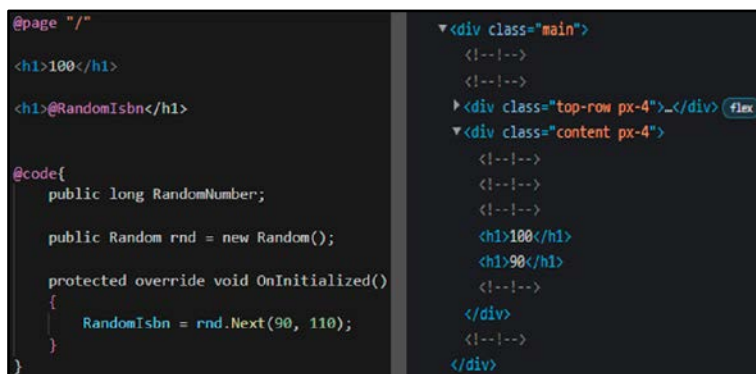
A dolgozatom utolsó és a cég elemzői számára legfontosabb része maga a vizualizációkat megjelenítő webalkalmazás. Lehet bármilyen részletes és jól felépített egy adatbázis, lehet bármilyen jól összerakott egy adatintegrációs folyamat és lehet akármilyen tökéletes a létrehozott adattárház ezek mind majdhogynem másodlagos dolgok a cég üzleti fejlődését tekintve. Nagyon fontos dolgok ezek is, de a legfontosabb része az egész folyamatnak az információk kinyerése és ezek felhasználása, hogy a vállalat megfelelő ütemben tudjon fejlődni és alkalmazkodni a piaci változásokhoz.

Korábban már említésre került, hogy a webalkalmazás Blazor technológiával készült. Ennek sajátossága, hogy az oldal felépítése megosztott és egyszerű oldal komponensekből áll. Megosztott komponens például a navigációs oldalsó menü, vagy az oldal alap struktúráját adó „MainLayout.razor” fájl. A második adja az oldal felépítését és ide kerülnek beillesztésre az egyszerű oldal komponensek, így maga a keret nem változik, ha törölni vagy új oldalt akarunk hozzáadni az projekthez. Ez nagyon rugalmassá teszi az alkalmazást, mivel különösebb strukturális változtatás nélkül tudunk hozzáadni és elvenni oldalakat, és ebből a felhasználó semmit sem lát.

Azért is választottam ezt a megoldást, mert ez az egyik legegyszerűbb módja annak, ha egy új szempont szerint akarunk vizualizációkat összeállítani, mert így csak létre kell hoznunk egy oldal komponens a megfelelő üzleti logikával és kinézettel, hozzáadni egy hivatkozást a navigációs menühöz és máris bővült a webalkalmazás egy új felülettel. Például, ha később felmerül üzleti igényként vagy fejlesztési célként, hogy legyen egy olyan felület, ahol boltvezetők különböző statisztikáit lehet megfigyelni, ami mondjuk a felsőbb vezetőknek adhat egy képet arról, hogy ezek boltvezetők hogyan végzik a munkájukat. Ezt az új és a jelenlegi tematikáktól merőben eltérő felületet is egyszerűen hozzá lehet adni a menühöz, és máris elérhető ez a felület is egy kattintással.

Az is nagy előny, hogy ezeket az oldalakat más ASP.NET-es projektben is fel lehet használni, így könnyen újrahasznosíthatóak. Ez a „razor” kiterjesztésű komponenseknek köszönhető. Ezek a „Razor-szintaxis” szerint épülnek fel, ami azt jelenti, hogy tartalmazznak HTML és a C# kódot is, ezeknek a kombinációját nevezzük „Razor Pages” komponenseknek. Az oldal dinamikus elemeit határozza meg a C# kód. Két féle tartalmat lehet megjeleníteni ezeken az oldalakon: ügyfeleknek szóló tartalmat és kiszolgálói kódot. Az ügyfeleknek szóló tartalom az, amit ha a böngészőben a „Fejlesztői módban” vizsgáljuk az oldalt, akkor szabad szemmel is látunk. Ilyen a HTML kód, ami oldal felépítését biztosítja, CSS kód, ami a formázást valósítja meg és például a JavaScript, ami egyéb funkcionalitásokért felel. A kiszolgálói kód kerül először futtatásra a felhasználó, vagyis a böngésző által, és ennek segítségével tudunk dinamikusan változó ügyfeleknek szóló tartalmakat hozzáadni az oldalhoz. Ilyenek például a HTML elemek, amiket egy változó elemszámú lista alapján hozunk létre C#

segítségével. Ezek komponensek a felhasználó számára ugyanúgy jelennek meg, mint a nem generált statikus elemek.

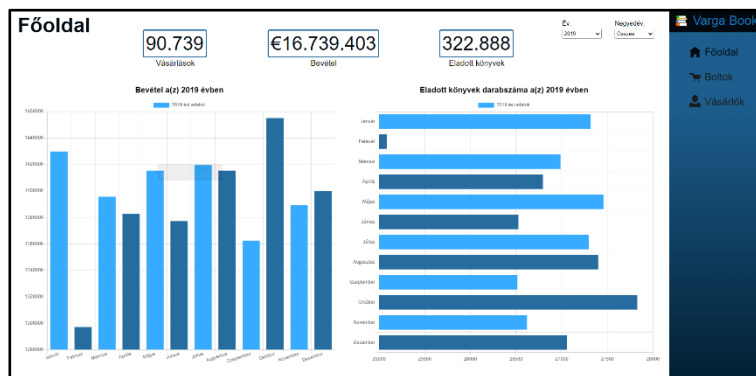


10.1. ábra: Példa az adatok megjelenésére a böngészőben

A korábbi fejezetekben már említésre került, hogy a webalkalmazás célja az, hogy egy olyan felületet biztosítson az elemzők, menedzserek és cég vezetői számára, ahol könnyen hozzáférnek az egyszerűen felépített és könnyen értelmezhető vizualizációkhoz, és akár egy üzleti megbeszélés vagy a reggeli kávé mellett tudják ezeket elemezni.

A felületnek viszonylag egyszerű felépítése van, ahogy ez már a terveken is látszódott. Ez viszont nem hiányosság, hanem előny, hiszen éppen ez volt az egyik megvalósítási cél, hogy a felületet vizsgálva ne az legyen az első gondolata a felhasználónak, hogy nem tudja mit és hol talál, hanem, hogy rögtön átlássa az oldal felépítését, és, hogy az információ, amit keres az hol található meg rajta.

Az oldalt megnyitva ez a kép fogadja a szakembereket:

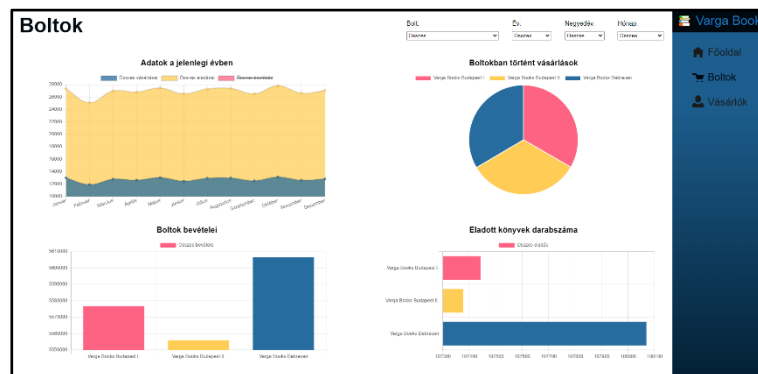


10.2. ábra: A webalkalmazás főoldala

A felület egészét vizsgálva elég gyorsan látja a felhasználó, hogy az információ, amit keres az biztosan a felület bal oldalán (nagyobbik tartalmi részén) fogja találni. Ha speciálisabb dolgot keres, akkor nagy valószínűséggel az egyik részletesebb irányítópulton fogja megtalálni azt, amit a felület jobb oldalán található menüvel ér el.

A felület jobb oldalán található menü, aminek segítségével tudnak az elemzők váltani a különböző irányítópultok között. Itt jelenleg három felület érhető el.

A főoldal, ahol egy általános képet kapunk a cég üzleti folyamatairól. A boltokkal kapcsolatos felület, ahol részletesebben látjuk, hogy az egyes boltok hogyan teljesítenek a megfigyelt időszakban. Illetve itt található még a vásárlókat részletező felület is, ami segít megérteni, hogy a vásárlóknak milyen az összetétele nemek, lakhely és az ízlés szempontjából a könyvek műfaját tekintve. Ez a három felület segít felhasználóknak tisztább képet kapni a vállalat helyzetéről, azzal hogy több különböző részről tudjuk megnézni, hogy hogyan épülnek fel a bevételek, az eladások és a vevők. De rálátást kapunk a kiadókkal, a boltokkal, az írókkal, a műfajokkal és a lokációkkal kapcsolatos kérdéskörökre is.



10.3. ábra: Példa a további felületek kinézetére

A felület, amin bemutatom a webalkalmazás felépítését a főoldal, ahol az összefoglaló felület található, amin azokat a vizualizációkat és kártyákat találjuk, amelyek segítségével egy általános képet tud az elemző kapni a bevétel, az eladások és a vásárlások szempontjából. Ezt az alkalmazáson belül dátum szűrés biztosítja, amellyel teljesen az igényekre lehet szabni, hogy mit lásson a felhasználó mikor az oldalra navigál. Ezeken a szűrőkön tudjuk állítani az évet és a kiválasztott érték alapján frissülnek a kártyákon található adatok. Ha kiválasztunk a negyedéves szűrőn az alapértelmezett „Összes” érték helyett egyet az „1, 2, 3, 4” értékek közül, akkor szintén frissülnek az adatok a kiválasztott érték alapján és a felhasználó a felületen rögtön az általa kiválasztott negyedév szerint szűrt értékeket látja.

Ezen a felületen öt információt tartalmazó elem található. A felső részén a felületnek a három kártyán olyan adatok szerepelnek, amelyek segítségével tisztább képet kaphatunk a cég gazdasági helyzetéről. Ezek jelen esetben az összbevétel, összes eladott könyv darabszáma és az összes vásárlás. Ezeket az értékeket az előbb említett szűrő funkcióval tudjuk szűkíteni az év és a negyedév szempontjából, amiket egy listából tud kiválasztani a felhasználó és ennek segítségével tudja lekérdezni a neki szükséges adatokat az felületen. A három kártyán kívül két oszlopdiagram is található az oldal alján. A baloldalin a bevételeket, a jobboldalin pedig a könyv eladásokat látjuk az adott év hónapos bontásában. Azért oszlopdiagramot választottam ezeknek az információknak a prezentálásához, mert ez az egyik legjobb formája annak, hogy több értéket több különböző csoport szerint (esetemben a hónapok szerint) össze tudjunk

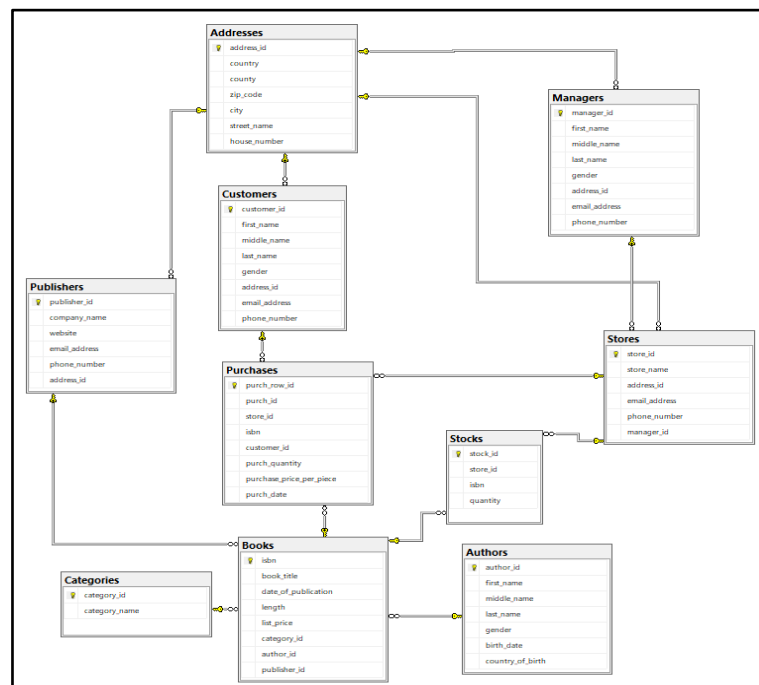
használni és rögtön lássa a felhasználó, hogy melyik hónapban voltak kiemelkedően rosszak vagy kiemelkedően jók a vizsgált értékek. Fontosak még a diagramoknál használt színek is, mivel a felhasználó ezeket veszi észre szinte először. A cég alapszíne a kék, így a webalkalmazásnál is ez és ennek az árnyalatai voltak a domináns színek.

11. Rendszer komponenseinek tesztelése

Minden létező projektnek vagy folyamatnak az egyik, ha nem a legfontosabb része a tesztelés. Ahhoz, hogy megfelelő minőségű terméket tudjon előállítani egy cég vagy fejlesztő a hozzáértésen és kemény munkán kívül rendkívül fontos az alapos tesztelési folyamatok elvégzése is. Ahány féle és fajta projekt vagy termék, annyi fajta tesztelési folyamat létezik. Csak hogy pár tesztelési fajtát említsek a teljesség igénye nélkül, beszélhetünk komponens-, interfész-, integrációs-, rendszer-, használati eset alapú-, teljesítmény- és stressz tesztelésről. Iparáganként lehetnek ezeknél speciálisabb/különbözőbb tesztelési folyamatok, de általánosan kijelenthető, hogy a termékek tesztelése manapság nem csak mindennapossá, de rendkívül fontossá is vált, hiszen ez is hozzájárul a magas minőség szinten tartásához.

11.1. Adatbázis megfelelő működésének (kapcsolatainak) tesztelése

Kicsit specifikusabban a forrásadatbázison adatintegritási tesztelést fogok végezni, ami azt jelenti, hogy megvizsgálom a benne található táblák közötti kapcsolatot, illetve azt, hogy ellenőrzöm, hogy ez megfelel-e a dolgozatom elején felvázolt üzleti modellnek és felépítésnek. Ehhez először kettesével ellenőrzöm a táblák közötti kapcsolatot, aminek az eredményét egy táblázatban összesítve fogom prezentálni. Ezen felül pedig pár több táblát is érintő lekérdezés segítségével megvizsgálom, hogy a később gyakrabban felmerülő táblakapcsolatok is terv szerint működnek-e. Az adatintegritási teszteléshez a Microsoft SQL Server Management Studio is nyújt egy kis segítséget, hiszen viszonylag egyszerűen lehet benne tábla diagramot létrehozni, amiben jelölve vannak a táblák közötti kapcsolatok is.



11.1. ábra: Forrásadatbázis exportált tábladiagramja

11.1.1. Táblák közötti szimpla kapcsolatok

Egyik tábla	Másik tábla	Összekapcsolás
Addresses	Managers	Sikeres
Addresses	Customers	Sikeres
Addresses	Stores	Sikeres
Addresses	Publishers	Sikeres
Managers	Stores	Sikeres
Customers	Purchases	Sikeres
Stores	Purchases	Sikeres
Stores	Stocks	Sikeres
Publishers	Books	Sikeres
Purchases	Books	Sikeres
Books	Categories	Sikeres
Books	Authors	Sikeres
Books	Stocks	Sikeres

11.1. táblázat: Táblák közötti szimpla kapcsolatok tesztelése

11.1.2. Táblák közötti összetettebb kapcsolatok

Az első összetett lekérdezésben arra voltam kíváncsi, hogy a „Customers – Purchases -Books – Categories – Authors” táblák közötti kapcsolatok rendesen létrejöttek-e. Ebből a célból írtam meg azt a lekérdezést, ami kilistázza azokat a személyeket és az általuk vásárolt könyvek nevét, illetve még néhány adatot, ahol a vásárlásnál a termékek összköltsége egyenlő volt a vásárlások táblában található maximális kifizetett árral.

A második lekérdezésben arra voltam kíváncsi, hogy a „Publishers – Books – Purchases - Stores” táblák közötti kapcsolatok rendesen létrejöttek-e. Ezen kapcsolatok tesztelését egy viszonylag egyszerű lekérdezéssel meg tudtam nézni, ami azt vizsgálja, hogy a Debreceni boltban mennyi eladás történt, és az kiadónként hogyan oszlik meg.

11.2. Adatintegrációs folyamatok- és az adattárház kapcsolatainak tesztelése

Az adatintegrációs eszközön (SQL Server Integration Services) létrehozott folyamatokon, aminek megvalósítását pár fejezettel ezelőtt mutattam be, úgynevezett funkcionális tesztelést végeztem. Ez jelen esetben annyit jelent, hogy megvizsgálom azt, hogy adatok beszúrásakor, illetve módosításakor a forrásadatbázisban a táblák megfelelően töltődnek-e át az adattárházamba. Azaz rendesen lefut-e minden folyamat és az új, a változó, illetve historikusan tárolandó adatok esetén csak az érintett rekordok kerülnek be, vagy módosulnak az adattárházban.

Tábla	Új rekord beszúrás	Módosítás (Változó)	Módosítás (Historikus)
Addresses	Sikeres	Sikeres	-
Authors	Sikeres	Sikeres	-
Books	Sikeres	Sikeres	Sikeres
Categories	Sikeres	Sikeres	-
Customers	Sikeres	Sikeres	-
Managers	Sikeres	Sikeres	-
Publishers	Sikeres	Sikeres	-
Purchases	Sikeres	-	-
Stores	Sikeres	Sikeres	-

11.2. táblázat: Adatintegrációs folyamatok- és az adattárház kapcsolatainak tesztelése

11.3. Webalkalmazás tesztelése

A webalkalmazás tesztelésénél elsősorban megnéztem, hogy a létrehozott objektumok rendesen illeszkednek-e az importálandó adatok struktúrájához. Miután ezt leellenőriztem mindegyik objektumnál, leteszteltem, hogy az adatokat importáló metódusok jól működnek-e. Az importáló metódusoknál fontos volt figyelnem azt, hogy ne olvassak be felesleges sorokat vagy adatokat, mivel így torzulhatnak a vizualizációk és a belőlük kinyerhető információk. Így minden beolvasott sornál le ellenőrzöm, hogy maga a sor üres-e, vagyis azt, hogy üres szöveget tartalmaz-e. Ez leggyakrabban a beolvasott fájl végén probléma, mivel az adattárházból való kimentéskor az utolsó sor után még beíródik egy üres sor.

Az alkalmazással kapcsolatban főleg felületi teszteléseket végeztem. Olyan teszteléseket, hogy a különböző elemek a terveknek megfelelően jelennek-e meg és minden komponens jól látható-e a felületen.

A felületi tesztelésen kívül az év választó komponenst is teszteltem. Mivel ez egy olyan lista, amiből elemeket tud kiválasztani a felhasználó, így azt tudtam tesztelni,

hogy egy másik év kiválasztásakor a különböző elemek megfelelően frissülnek-e a felületen az oldal újra töltése nélkül. Ezt leteszteltem úgy, hogy a megadott évben egyáltalán nem volt adatom, és úgyis, hogy volt. Mindkét esetben jól működött a szűrés funkció és minden kapcsolódó komponens rögtön frissült.

12. A projekt továbbfejlesztési lehetőségei

Akármilyen rendszerről is legyen szó a továbbfejlesztés mindig egy fontos kérdés, hiszen tökéletes rendszer nem létezik. Ez persze erre a projektre is igaz, hiszen ahogy az emberek is, úgy a technológiák és a rendszerek is folyamatosan változnak, illetve fejlődnek is.

Ha egy rendszer továbbfejlesztéséről beszélünk, akkor ahány ember, fejlesztő, analitikus, döntéshozó, cégvezető, annyi ötlet és megvalósítási terv. Ezek közül csak a leghasznosabbakat, leginnovatívabbakat és az üzletileg leglényegesebbeket emelném ki.

A cég jövőbeni méretétől is függ, de valamilyen hitelesítési megoldás elengedhetetlen ahhoz, hogy a privát üzleti folyamatokkal kapcsolatos adatokhoz, még ha csak vizualizációk formájában is, de csak a cég dolgozói férhessenek hozzá. Ezt elérhetjük egy egyszerűbb bejelentkezési rendszerrel is, de ha még jobban el akarjuk zárni az adatokat a külvilágtól, akkor a bejelentkezési rendszert egy privát VPN kapcsolat mögé is rakhatjuk. Így tényleg csak azok tudják majd elérni az adatokat, akiknek el is kell érniük.

A bejelentkezés után lehet beszélni a különböző jogosultságok kezeléséről is, hiszen egy ilyen felületet nem csak egy hierarchia szinten lévő felhasználók fognak használni. A különböző jogosultsági és szerepkörökkel el lehet azt is érni, hogy ugyanazt a webalkalmazást nézve két különböző jogkörrel rendelkező felhasználó teljesen más lásson a felületen. De akár olyasmit is, hogy az egyik szerepkörű felhasználó az oldalon nézze és elemezze a különböző dashboardokat, amikhez jogosultsága van, amíg a másik felhasználó a forrásadatbázisban lévő boltok vagy vevők adatainak értékét tudja módosítani egy adminisztrátori felületen.

Egy következő lépcsőfok lehet a további platformok felé való nyitás. Igaz egy webalkalmazást könnyen el tud érni a felhasználó az okostelefonjáról is, de manapság szinte elengedhetetlen, hogy a telefonon egy pöccintéssel el tudja érni azokat a dolgokat, amik megkönnyítik a mindennapokat, legyen az szabadidős tevékenység vagy munka egy mobilalkalmazás formájában.

A továbbfejlesztési lehetőségek rövid összefoglalója:

- Hitelesítés lehetősége bejelentkezéssel vagy privát VPN kapcsolattal,
- jogosultságok kezelése, hierarchiája, különböző jog- és szerepkörök,
- a webalkalmazás funkcionális bővítése,
- mobil platformok felé nyitás (Android és/vagy iOS).

13. Összefoglalás

A szakdolgozatom célja egy adatbázis megtervezése, a hozzá kialakított csillagséma alapján egy adattárház létrehozása, az ezeket összekötő adatintegrációs folyamat megvalósítása és egy olyan felület fejlesztése volt, ahol a fiktív könyvesbolt-hálózat elemzői láthatják a legfontosabb eladási adatokat vizualizációk és mutatószámok formájában. Ennek a felületnek az előzetes tudásom, érdeklődési köröm és az aktuális trendek alapján egy Blazor alapú webalkalmazást választottam.

Legelső lépésként minden szükséges eszköznél kutatómunkát végeztem, mindegyiknél kiválasztottam hármat, majd ezeket elemezve és összehasonlítva meghatároztam a számomra, és a cél megvalósítására legalkalmasabb összeállítást.

A projekt megvalósításához egy forrásadatbázis létrehozására volt szükség. Az adatbázist úgy próbáltam kialakítani, hogy az adatok a lehető legvalóságosabbnak tűnjenek, annak ellenére, hogy a lokáció adatokat leszámítva minden adat általam létrehozott vagy generált. A forrásadatbázisom a könyvesbolt-hálózat vevőit, vásárlásait, könyveit, kapcsolatban álló kiadóit, illetve raktárkészleteit tartalmazza. A forrásadatbázis alapján létrehoztam egy csillagsémát, aminek középpontjában a vásárlások állnak, mint tényadatok. A tényadatok köré pedig a bolt, vásárló, könyv és dátum dimenziók kerülnek, hogy megfelelően lehessen elemezni az információkat. A csillagséma alapján épül fel az adattárház, ahol az elemzéshez gyűjtött adatok tárolása történik. Az adattárház feltöltéséhez szükséges egy adatintegrációs folyamat megvalósítása is, amely a forrásadatbázisba érkező adatok alapján feltölti a különböző rétegeket úgy, hogy az utolsó rétegben a csillagsémának megfelelően már csak öt tábla szerepeljen a kimutatások elkészítéséhez. Ez a folyamat úgy működik, hogy a vásárlásokat csak folyamatosan betölti az adattárházba, mivel azokon nem hajtunk végre semmiféle változtatást. A dimenzió tábláknál viszont figyeli a változásokat és ahol szükséges ott felülírja, vagy historikusan eltárolja azokat az adattárházban. A kész forrásadatbázist, adatintegrációs folyamatot és adattárházat követően elkészítettem a webalkalmazást, amely lehetővé teszi a kinyert információk értelmezését, a bevételek alakulását és a boltok teljesítményeinek megfigyelését vizualizációk segítségével. Mikor a rendszer összes része elkészült, akkor elvégeztem az egyes komponensek tesztelését, a tesztelés alapján pedig a felmerült hibák javításra kerültek.

A szakdolgozat megvalósítása közben sikerült jobban bele látnom az adatbázisok világába és gyakorlatot szerezni egy adatintegrációs eszköz használatában. Megismertem az adattárházak működését, a különböző típusú adatok tárolásának speciális igényeit. Végül, de nem utolsó sorban pedig tapasztalatot szereztem egy webalkalmazás fejlesztésében és az adatvizualizációk elkészítésében.

14. Summary

The aim of my dissertation was to design a database, to make a data warehouse based on the star scheme created for said database, to connect these with a data integration process and to develop an interface where analysts of the fictitious bookstore-network can see the most important sales data in the form of visualizations and indicators. For this interface, derived from my prior knowledge, interest and current trends, I chose a Blazor based web application.

As a first step, I did research work with all the necessary tools, I selected three from each, then analysing and comparing these, I defined the most appropriate assembly for me and the execution of the goal.

It was essential to create a source database in order to implement this project. I tried to design the database in a way which the data appeared to be the most real as much as possible, despite the fact that all data, except location, was either generated or provided by me. My source database consists of customers, purchases, books, related publishers, and inventories of the bookstore network. Based on the source database, I created a star scheme which is centred around the purchases as fact data. Around this fact data there are the store, customer, book, and date dimensions in order to properly analyse the information. I built the data warehouse based on the star scheme, where data collected for analysis is stored. To fill up the data warehouse, the execution of a data integration process is also required, which first fills the different layers with data coming to the source database, so that in the last layer, accordingly to the star scheme, there are only five tables included for the preparations of statements. The way this process works is that purchases are continuously loaded into the data warehouse as we do not perform any change on them. On the other hand, in the case of dimension tables, it monitors changes and overwrites them where necessary, or historically stores them in the data warehouse. After the completed source database, data integration process, and data warehouse, I created the web application which allows the interpretation of extracted information and the observation of conformation of revenue and performance of stores using visualizations. After all parts of the system were completed, I did the testing of each component and any errors that had been occurring while testing were corrected.

During the implementation of my dissertation, I got a better insight into the world of databases and had the chance to get practice in the use of the data integration tool. I got acquainted with the operation of data warehouses and the special needs of storing data from different types. Last but not least, I gained experience in developing a web application and creating data visualization.

15. Irodalomjegyzék

- [1] „Definition of DBMS,” TechTerms, [Online]. Available: <https://techterms.com/definition/dbms>. [Hozzáférés dátuma: 1 5 2021].
- [2] „What is a Relational Database Management System?,” codecademy, [Online]. Available: <https://www.codecademy.com/articles/what-is-rdbms-sql>. [Hozzáférés dátuma: 1 5 2021].
- [3] „RDBMS (Relational Database Management System) Definition,” Gartner, [Online]. Available: <https://www.gartner.com/en/information-technology/glossary/rdbms-relational-database-management-system>. [Hozzáférés dátuma: 1 5 2021].
- [4] „Comparing Database Management Systems: MySQL, PostgreSQL, MSSQL Server, MongoDB, Elasticsearch and others,” altexsoft, 2019. [Online]. Available: <https://www.altexsoft.com/blog/business/comparing-database-management-systems-mysql-postgresql-mssql-server-mongodb-elasticsearch-and-others/>. [Hozzáférés dátuma: 1 5 2021].
- [5] „Difference Between SQL Vs MySQL Vs SQL Server (With Examples),” 2021. [Online]. Available: <https://www.softwaretestinghelp.com/sql-vs-mysql-vs-sql-server/>. [Hozzáférés dátuma: 1 5 2021].
- [6] „System Properties Comparison Microsoft SQL Server vs. MySQL vs. PostgreSQL,” DB-Engines, [Online]. Available: <https://db-engines.com/en/system/Microsoft+SQL+Server%3BMySQL%3BPostgreSQL>. [Hozzáférés dátuma: 1 5 2021].
- [7] „Comparative Performance Analysis of MySQL and SQL Server Relational Database Management Systems in Windows Environment,” 2015. [Online]. Available: <https://www.ijarcce.com/upload/2015/march-15/IJARCCE%2039.pdf>. [Hozzáférés dátuma: 1 5 2021].
- [8] P. Pedamkar, „Difference Between MySQL and MSSQL,” Educba, [Online]. Available: <https://www.educba.com/mysql-vs-mssql/>. [Hozzáférés dátuma: 1 5 2021].
- [9] „Stack Overflow 2020 Survey,” [Online]. Available: <https://insights.stackoverflow.com/survey/2020#technology-databases>. [Hozzáférés dátuma: 1 5 2021].

- [10] „Relational Database Comparison – Which One To Choose For Your Next Project?,” BoostHigh, [Online]. Available: <https://boosthigh.com/relational-database-comparison/>. [Hozzáférés dátuma: 1 5 2021].
- [11] R. Sherman, Business Intelligence Guidebook - From Data Integration to Analytics, Waltham, Massachusetts: Elsevier Science, 2014.
- [12] „WHAT IS ETL?,” Matillion, [Online]. Available: <https://www.matillion.com/what-is-etl-the-ultimate-guide/>. [Hozzáférés dátuma: 1 5 2021].
- [13] „Data Integration & ETL softwares,” bitool.net, [Online]. Available: <https://www.bitool.net/data-integration.html>. [Hozzáférés dátuma: 1 5 2021].
- [14] R. Dhir, „Choosing the Best ETL Tool,” Grazitti Interactive, 2019. [Online]. Available: <https://www.grazitti.com/blog/choosing-the-best-etl-tool/>. [Hozzáférés dátuma: 1 5 2021].
- [15] „15 Best ETL Tools In 2021 (A Complete Updated List),” 2021. [Online]. Available: <https://www.softwaretestinghelp.com/best-etl-tools/>. [Hozzáférés dátuma: 1 5 2021].
- [16] „Talend Open Studio Features and Reviews,” wyzoo2, [Online]. Available: <https://wyzoo.com/technology/talend-open-studio.html>. [Hozzáférés dátuma: 1 5 2021].
- [17] P. Dr. Hegedűs és R. Dr. Ferenc, „Nagyméretű adatbázisok,” 2020. [Online]. Available: http://eta.bibl.u-szeged.hu/5031/2/EFOP343_A6_2e_BigData-data-transform-SPOC.pdf. [Hozzáférés dátuma: 1 5 2021].
- [18] W. H. Inmon, Building the Data Warehouse, Fourth Edition, Indianapolis, Indiana: Wiley Publishing, Inc., 2005.
- [19] „Definition of Data Warehouse,” [Online]. Available: <https://www.gartner.com/en/information-technology/glossary/data-warehouse>. [Hozzáférés dátuma: 1 5 2021].
- [20] „Fact Table - Definition, Examples and Four Steps Design by Kimball,” Zentut, [Online]. Available: <https://www.zentut.com/data-warehouse/fact-table/>. [Hozzáférés dátuma: 1 5 2021].
- [21] „Star Schema,” ScienceDirect, [Online]. Available: <https://www.sciencedirect.com/topics/computer-science/star-schema>. [Hozzáférés dátuma: 1 5 2021].

- [22] „Understanding Star Schemas,” [Online]. Available: http://gkmc.utah.edu/ebis_class/2003s/Oracle/DMB26/A73318/schemas.htm. [Hozzáférés dátuma: 1 5 2021].
- [23] „Making Sense of Data Warehouse Architecture,” [Online]. Available: <https://datawarehouseinfo.com/data-warehouse-architecture/>. [Hozzáférés dátuma: 1 5 2021].
- [24] „Star and Snowflake Schema in Data Warehouse with Examples,” guru99, [Online]. Available: <https://www.guru99.com/star-snowflake-data-warehousing.html>. [Hozzáférés dátuma: 1 5 2021].
- [25] „What is Data Visualization? Definitions, Graph Types and How to Use Them,” Kilpfolio, [Online]. Available: <https://www.klipfolio.com/resources/articles/what-is-data-visualization>. [Hozzáférés dátuma: 29 4 2021].
- [26] J. Schwabish, Better Data Visualizations: A Guide for Scholars, Researchers, and Wonks, New York, United States: Columbia University Press, 2021.
- [27] C. Pittenturf, „What is Data Visualization and Why is it Important?,” CIO Review, [Online]. Available: <https://data-visualization.cioreview.com/cxoinsight/what-is-data-visualization-and-why-is-it-important-nid-11806-cid-163.html>. [Hozzáférés dátuma: 29 4 2021].
- [28] „The Top 10 Types of Data Visualization Made Simple (Graphs & Charts),” Boost Labs, [Online]. Available: <https://boostlabs.com/blog/10-types-of-data-visualization-tools/>. [Hozzáférés dátuma: 29 4 2021].
- [29] S. Berinato, „Visualizations That Really Work,” Harvard Business Review, [Online]. Available: <https://hbr.org/2016/06/visualizations-that-really-work>. [Hozzáférés dátuma: 29 4 2021].
- [30] A. Lynch, „Top 10 Types of Charts and Graphs for Data Visualization,” Wondershare, [Online]. Available: <https://www.edrawsoft.com/chart-types-uses.html>. [Hozzáférés dátuma: 29 4 2021].
- [31] „The Data Visualisation Catalogue,” [Online]. Available: <https://datavizcatalogue.com/index.html>. [Hozzáférés dátuma: 29 4 2021].
- [32] „Top Charts & Graphs for Your Data,” [Online]. Available: <https://www.idashboards.com/guides/top-charts-and-graphs/>. [Hozzáférés dátuma: 29 4 2021].

- [33] C. Miller, „FinancesOnline: Pros & Cons of Microsoft Power BI: Analysis of a Leading Business Intelligence Software,” [Online]. Available: <https://financesonline.com/pros-cons-of-microsoft-power-bi-analysis-of-a-leading-business-intelligence-software/>. [Hozzáférés dátuma: 27 4 2021].
- [34] „Tableau vs QlikView vs Power BI,” 2019. [Online]. Available: <https://www.smetricinsights.com/wp-content/uploads/2019/05/Tableau-VS-QlikView-VS-Power-BI-2019-Update.pdf>. [Hozzáférés dátuma: 1 5 2021].
- [35] „Qlikview vs Power BI: Everything you need to know,” BetterBuys, [Online]. Available: <https://www.betterbuys.com/bi/qlikview-vs-power-bi/>. [Hozzáférés dátuma: 1 5 2021].
- [36] „Tableau Pros and Cons | Tableau Benefits for 2019,” DataFlair, [Online]. Available: <https://data-flair.training/blogs/tableau-pros-and-cons/>. [Hozzáférés dátuma: 1 5 2021].
- [37] R. Gour, „Top 5 BI Tools Widely used for Data Visualization,” towards data science, 22 5 2019. [Online]. Available: <https://towardsdatascience.com/top-5-bi-tools-that-you-must-use-for-data-visualization-7ccc2a852bd3>. [Hozzáférés dátuma: 1 5 2021].
- [38] „QlikView Advantages and Disadvantages | QlikView Benefits,” DataFlair, [Online]. Available: <https://data-flair.training/blogs/qlikview-advantages-and-disadvantages/>. [Hozzáférés dátuma: 1 5 2021].
- [39] „Tableau vs QlikView: What Is The Difference?,” MindMajix, 19 4 2021. [Online]. Available: <https://mindmajix.com/tableau-vs-qlikview>. [Hozzáférés dátuma: 1 5 2021].
- [40] „HTML: HyperText Markup Language,” MDN Web Docs, 14 4 2021. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/HTML>. [Hozzáférés dátuma: 28 4 2021].
- [41] „What is HTML - javatpoint,” javatpoint, [Online]. Available: <https://www.javatpoint.com/what-is-html>. [Hozzáférés dátuma: 28 4 2021].
- [42] D. G., „What is HTML? The Basics of Hypertext Markup Language Explained,” Hostinger, 25 11 2019. [Online]. Available: <https://www.hostinger.com/tutorials/what-is-html>. [Hozzáférés dátuma: 28 4 2021].

- [43] „CSS (Cascading Style Sheet) Definition,” TechTerms, [Online]. Available: <https://techterms.com/definition/css>. [Hozzáférés dátuma: 28 4 2021].
- [44] „Types of CSS (Cascading Style Sheet),” GeeksforGeeks, 19 6 2020. [Online]. Available: <https://www.geeksforgeeks.org/types-of-css-cascading-style-sheet/>. [Hozzáférés dátuma: 28 4 2021].
- [45] „JavaScript Definition,” TechTerms, [Online]. Available: <https://techterms.com/definition/javascript>. [Hozzáférés dátuma: 28 4 2021].
- [46] L. K. Cox, „Web Design 101: How HTML, CSS, and JavaScript Work,” HubSpot, [Online]. Available: <https://blog.hubspot.com/marketing/web-design-html-css-javascript>. [Hozzáférés dátuma: 28 4 2021].
- [47] A. HEDDINGS, „What Is Microsoft’s Blazor Web Framework, and Should You Use It?,” 5 8 2021. [Online]. Available: <https://www.cloudsavvyit.com/12493/what-is-microsofts-blazor-web-framework-and-should-you-use-it/>. [Hozzáférés dátuma: 1 12 2021].
- [48] C. Sainty, „What’s behind the hype about Blazor?,” 26 2 2020. [Online]. Available: <https://stackoverflow.blog/2020/02/26/whats-behind-the-hype-about-blazor/>. [Hozzáférés dátuma: 01 12 2021].
- [49] S. Subramanian, „What Is Blazor And How It Works,” c-sharpcorner, 25 03 2021. [Online]. Available: <https://www.c-sharpcorner.com/article/what-is-blazor-and-how-does-it-works/>. [Hozzáférés dátuma: 1 12 2021].

16. Ábrajegyzék

2.1. TÁBLÁZAT: ADDRESSES TÁBLA ELEMEI	10
2.2. TÁBLÁZAT: AUTHORS TÁBLA ELEMEI	11
2.3. TÁBLÁZAT: BOOKS TÁBLA ELEMEI	11
2.4. TÁBLÁZAT: CATEGORIES TÁBLA ELEMEI	11
2.5. TÁBLÁZAT: CUSTOMERS TÁBLA ELEMEI	11
2.6. TÁBLÁZAT: MANAGERS TÁBLA ELEMEI	12
2.7. TÁBLÁZAT: PUBLISHERS TÁBLA ELEMEI	12
2.8. TÁBLÁZAT: PURCHASES TÁBLA ELEMEI	12
2.9. TÁBLÁZAT: STOCKS TÁBLA ELEMEI	13
2.10. TÁBLÁZAT: STORES TÁBLA ELEMEI	13
2.1. ÁBRA: AZ ADATBÁZIS TÁBLA DIAGRAMJA	13
2.2. ÁBRA: AZ ADATBÁZIS EGYED-KAPCSOLAT DIAGRAMJA	14
3.1. TÁBLÁZAT: ADATBÁZIS-KEZELŐ RENDSZEREK ÖSSZEHASONLÍTÁSA	19
4.1. ÁBRA: AZ ETL FOLYAMAT	20
4.2. ÁBRA: ENTERPRISE DATA WAREHOUSE MODELL	21
4.1. TÁBLÁZAT: ADATINTEGRÁCIÓS ESZKÖZÖK ÖSSZEHASONLÍTÁSA	24
6.1. ÁBRA: KIVÁLASZTOTT TÁBLÁK	27
6.2. ÁBRA: KIALAKÍTOTT CSILLAGSÉMA	27
7.1. ÁBRA: TÁBLÁZAT ÉS MUTATÓSZÁM PÉLDA	33
7.2. ÁBRA: PÉLDA AZ OSZLOPDIAGRAM TÍPUSOKRA	33
7.3. ÁBRA: PÉLDA HISZTOGRAMRA	34
7.4. ÁBRA: PÉLDA SÁVDIAGRAM TÍPUSOKRA	34
7.5. ÁBRA: PÉLDA SZÓRÁSDIAGRAMRA	35
7.6. ÁBRA: PÉLDA VONALDIAGRAMRA	35
7.7. ÁBRA: PÉLDA A TERÜLETDIAGRAM TÍPUSOKRA	36
7.8. ÁBRA: PÉLDA KÖR- ÉS FÁNKDIAGRAMRA	36
7.9. ÁBRA: PÉLDA FATÉRKÉP DIAGRAMRA	37
7.10. ÁBRA: PÉLDA TÉRKÉPDIAGRAMRA	37
7.11. ÁBRA: PÉLDA TELJESÍTMÉNY DIAGRAMOKRA	38
8.1. ÁBRA: WEBALKALMAZÁS GRAFIKUS TERVE SZÁMÍTÓGÉPES NÉZETBEN	44
9.1. ÁBRA: TELJES ETL FOLYAMAT	45
9.2. ÁBRA: STAGE RÉTEG FOLYAMATAI	45
9.3. ÁBRA: STAGE RÉTEGBEN TÖRTÉNŐ FOLYAMATOK EGY TÁBLÁN SZEMLÉLTETVE	46
9.4. ÁBRA: TRANSFORM FÁZIS A „BOOKS” TÁBLÁN BEMUTATVA	47
9.5. ÁBRA: DM RÉTEG FOLYAMATAI	48
9.6. ÁBRA: TÉNYTÁBLÁN ELVÉGZETT MŰVELETEK	49
10.1. ÁBRA: PÉLDA AZ ADATOK MEGJELENÉSÉRE A BÖNGÉSZŐBEN	52
10.2. ÁBRA: A WEBALKALMAZÁS FŐOLDALA	52
10.3. ÁBRA: PÉLDA A TOVÁBBI FELÜLETEK KINÉZETÉRE	53
11.1. ÁBRA: FORRÁSADATBÁZIS EXPORTÁLT TÁBLADIAGRAMJA	55
11.1. TÁBLÁZAT: TÁBLÁK KÖZÖTTI SZIMPLA KAPCSOLATOK TESZTELÉSE	56
11.2. TÁBLÁZAT: ADATINTEGRÁCIÓS FOLYAMATOK- ÉS AZ ADATTÁRHÁZ KAPCSOLATAINAK TESZTELÉSE	57