



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

**NEUMANN JÁNOS
INFORMATIKAI KAR**



SZAKDOLGOZAT

**OE-NIK
2022**

Hallgató neve:
Hallgató törzskönyvi száma:

**Tomayer Dávid
T/006406/FI12904/N**

Óbudai Egyetem
Neumann János Informatikai Kar
Kiberfizikai Rendszerek Intézet

SZAKDOLGOZAT FELADATLAP

Hallgató neve: **Tomayer Dávid**
Törzskönyvi száma: **T/006406/FI12904/N**
Neptun kódja: **[REDACTED]**

A dolgozat címe:

**Targoncák statisztikai adatainak adattárházban történő feldolgozása és
ennek vizualizációja webalkalmazáson keresztül**
**Processing forklift statistics in a data warehouse and visualizing it via a
web application**

Intézményi konzulens: **Rusznák Attila**
Külső konzulens:

Beadási határidő: **2022. május 15.**

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

A feladat

A céloom egy targoncákkal és nehézgépeket bérbeadó és értékesítésével foglalkozó vállalat fontosabb döntéseit megkönnyíteni, analizálni, vagy átláthatóságot nyújtani számára. Ezeknek az adatoknak a vizualizálása Web alkalmazáson keresztül történik, valamint ebben az alkalmazásban kerül feltöltésre, módosításra a cég áldal rendelkezésre bocsátott adatokat is. A hallgató feladata továbbá, hogy megtervezze és megvalósítsa az ehhez szükséges rendszert, adattárházat és webalkalmazást. Mérje fel és mutassa be ezek megvalósításához szükséges lehetőségeket, majd ezek közül válassza meg a legmegfelelőbbet és indokolja a döntését!

A dolgozatnak tartalmaznia kell:

- a webalkalmazás felületének megtervezését és megvalósítását,
- az adattárház megtervezését és megvalósítását,
- az adatbázis technológiák összehasonlító elemzését,
- az ETL folyamatot,
- a választott fejlesztői eszközök ismertetését és azok választásának indoklását,
- végül a továbbfejlesztési lehetőségeket



Fleiner Rita

Dr. Fleiner Rita
intézetigazgató

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

A dolgozatot beadásra alkalmasnak tartom:

.....
külső konzulens

Russek AAik

.....
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, 2022.05.05



.....
hallgató aláírása



KONZULTÁCIÓS NAPLÓ

Hallgató neve: Neptun kód: Tagozat:
Tomayer Dávid [REDACTED] Nappali
Telefon: Levelezési cím (pl: lakcím):
[REDACTED]

Szakdolgozat / Diplomamunka¹ címe magyarul:

Targoncák statisztikai adatainak adattárházban történő feldolgozása és ennek vizualizációja webalkalmazáson keresztül

Szakdolgozat / Diplomamunka² címe angolul:

Processing forklift statistics in a data warehouse and visualizing it via a web application

Intézményi konzulens: Külső konzulens:
Rusznák Attila

Kérjük, hogy az adatokat nyomtatott nagybetűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	2021.09.20	Az adatbázis és az adattárház megbeszélése	Rusznák AAik
2.	2021.10.05	Pontosítás a felépítéssel kapcsolatban	Rusznák AAik
3.	2021.11.02	Formai követelmények átbeszélése	Rusznák AAik
4.	2021.11.21	Az irodalomkutatás véglegesítése	Rusznák AAik

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. december 7

Rusznák AAik
.....
intézményi konzulens

¹ Megfelelő aláhúzendő!

² Megfelelő aláhúzendő!

³ Megfelelő aláhúzendő!

⁴ Megfelelő aláhúzendő!



KONZULTÁCIÓS NAPLÓ

Hallgató neve: Neptun kód: Tagozat:
Tomayer Dávid [REDACTED] Nappali
Telefon: Levelezési cím (pl: lakcím):
[REDACTED]

Szakdolgozat / Diplomamunka¹ címe magyarul:

Targoncák statisztikai adatainak adattárházban történő feldolgozása és ennek vizualizációja webalkalmazáson keresztül

Szakdolgozat / Diplomamunka² címe angolul:

Processing forklift statistics in a data warehouse and visualizing it via a web application

Intézményi konzulens: Külső konzulens:
Rusznák Attila

Kérjük, hogy az adatokat nyomtatott nagybetűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	2022.03.11	Az adatbázis és az adattárház átbeszélése	Rusznák AAik
2.	2022.03.25	A webes környezet bemutatása	Rusznák AAik
3.	2022.04.15	A vizualizációs réteg átbeszélése	Rusznák AAik
4.	2022.05.03	Szakdolgozat bemutatása, prezentáció	Rusznák AAik

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, 20. 22. 05. 05.

Rusznák AAik
.....
intézményi konzulens

¹ Megfelelő aláhúzendő!

² Megfelelő aláhúzendő!

³ Megfelelő aláhúzendő!

⁴ Megfelelő aláhúzendő!

TARTALOMJEGYZÉK

1.	Bevezetés	1
2.	A Megoldás kiválasztása, indoklás.....	2
2.1	Adattárház tárolására alkalmas szoftver megválasztása.....	2
2.1.1	Adatbázis szerver kiválasztása.....	2
2.2	SQL Menedzser kiválasztása	4
2.2.1	További lehetőségek.....	4
2.2.2	Összefoglalás és döntés	5
2.3	Az ETL eszköz kiválasztása.	5
2.3.1	Összefoglalás és Döntés	6
2.4	Webes környezet megoldásának kiválasztása.....	7
2.4.1	A webkeretrendszer kiválasztása	7
2.4.2	Döntés.....	9
2.5	Adatbázis terv	10
2.5.1	Az adatok jelenlegi állapota.....	10
2.5.2	A vizualizációs séma	11
2.5.3	Adattárház tervezése	12
3.	Követelményspecifikáció	13
4.	A Webes környezet megtervezése	14
4.1	Szerepkörök	14
4.2	A Webes környezet felületének megtervezése	15
5.	Adatbázis kialakítása	18
5.1	Adatbázis	19
5.2	Táblák és a benne lévő adatok	20
5.2.1	Targoncák tábla.....	20
5.2.2	Szerelőnaplók tábla	21
5.2.3	Munkások tábla	22
5.2.4	Helyszínek tábla.....	22
6.	Adattárház kialakítása.....	23
7.	Az ETL folyamat	24

7.1	Extract réteg.....	24
7.2	Transform réteg.....	25
7.3	Load réteg	26
8.	A Webes környezet megvalósítása	31
8.1	Modulok és működésük.....	31
8.1.1	Targoncák oldal	31
8.1.2	Szerelőnaplok oldal	34
8.1.3	Alkalmazottak modul	35
8.2	Vizualizációs réteg	37
9.	Továbbfejlesztési lehetőségek.....	40
10.	Összefoglalás.....	41
11.	Summary	42
	Irodalomjegyzék.....	43
	Ábrajegyzék	45

1. BEVEZETÉS

A szakdolgozatom témája egy targoncák és nehézgép bérbeadó és értékesítő cég számára adattárház készítése, majd ezt egy webalkalmazásként megalkotni. A cég számára fontos, hogy a webes felületen elérhető legyen az összes olyan funkció, amellyel nyomon követhetik és analizálhatják az adataikat. Ezen adatok vizualizált formában való megjelenítése lényeges a vezetőség számára.

A célom, hogy az elkészített riportokból kiderüljön, hogy az egyes típusú targoncák a jövőben mennyi bevételt eredményeznek jelenleg a vállalkozás számára. Bemutatom, hogy a cég szerelői hol és hányszor végeztek szervízt megvénykként, a szerelők az öt napos munkahéten naponta átlagosan mennyi időt töltenek gép szerelésével. Ami még fontos, hogy könnyen módosítható, illetve bővíthető is legyen az adattárház a cég növekedéséhez mérve.

A cég szeretne egy webes felületet, ahol megtörténik egy újabb targonca hozzáadása, módosítása, törlése, ezzel az adminisztrátor könnyedén felviheti a futási adatokat, költségeket és a szervizelési munkalapokat. Egy külön oldalon jelennek majd meg a riportok is a megfelelő adatokkal. Lesz egy oldal, amely a szerelők számára egy szerelőnaplót tud felvinni. A munkások csak kizárólag hozzáadni tudnak egy új naplót. Fontos, hogy az adatok módosítását egy adminisztrátor végezze el, a vezetőség tagjai csak az ezekből készült riportokat tekinthessék meg.

Az adatokat több Excel fájlban tárolják, ebből kell az adatokat megszerezni. A jelenlegi adatfájl tartalmazza a cég targoncáinak az összes rendelkezésre álló adatát. Nekem fontos ebből a targonca üzemóra állása, a típusa, az elvégzett munka időtartama és az alkalmazottak javítási naplói és a fontosabb adatai. Tartalmazhat sok olyan adatot, ami számomra felesleges és a feladat megoldásánál nem releváns vagy nem szükséges.

A szakdolgozatomban be szeretném mutatni az adattárházam felépítését, a bizonyos rétegek megalkotását és annak módját, az adatbázis, az integrálás, a rétegek és a webalkalmazás létrehozásánál előkerülő problémákat és a megoldási lehetőségeket, valamint a továbbfejlesztési lehetőségeket.

2. A MEGOLDÁS KIVÁLASZTÁSA, INDOKLÁS

2.1 Adattárház tárolására alkalmas szoftver megválasztása

Adatok tárolására többféle szempontot veszek figyelembe megoldásként. A lehetőségek közül gondolok ezalatt relációs, illetve nem relációs adatbázis megoldásokra.

Nagyobb adat mennyiség és a dokumentumszerű állományok miatt akár gondolok NoSQL¹ megoldáson is. Ez azonban számomra nem releváns, mivel a cég által nyújtott adatok szerkezete a relációs adatbázisokra jellemző strukturált formában használják. A feladat során aztán tapasztalhatom, hogy nem is mindre teljesen igaz ez az állítás, de átalakíthatóak relációs adatbázis felhasználásra.

A kiválasztás után a mai technológiai fejlődések miatt lehetőség adódik arra, hogyha alkalmasnak látom a felhőalapú megoldást választani. Azonban ez nem mindenkinek jó megoldás.

Ugyan nagyon sok érv szól mellette - ezek közé tartozik a komolyabb hibátűrése és az adatok folyamatos elérése -, viszont a legtöbb ilyen rendszer nagyobb adathalmazokkal működik jól. Itt már a rendszer hardver és fenntartási költségei nagyon magasak lehetnek. Emellett az árazása egy kisebb cég számára kevésbé kedvező, hiszen a legtöbb felhőben működő rendszer legtöbbször havidíjas szolgáltatásokkal rendelkezik, melynek olykor egy-egy komponense óránként való fizetést igényel.

Mivel nekem az adatok nem túl komplexek és nem túl nagyok, ezért a bemutatott vállalkozás számára a földi (On-Premise) szerver lesz a legjobb.

2.1.1 Adatbázis szerver kiválasztása

Az adatbázis szerverek megválasztásánál fő célom, hogy minél jobban integrálva legyen a szerver a rendszerrel a könnyű használat érdekében. Itt főként arra gondolok, hogy a cég bizonyos alkalmazások használatához hozzá van szokva, ennek megfelelően a későbbi választásaimban is ezen szoftver családhoz tartozó megoldásokat kell előtérbe helyeznem.

A kiszolgálói rétegnek a legnépszerűbbek közül választottam a Microsoft SQL Server-t, illetve a PostgreSQL-t. Mivel az integráltság a legfontosabb tényező, ezért a nevesebb operációs rendszerek -a Linux és a Microsoft Windows - egy-egy népszerűbb SQL kiszolgálóját választottam.[1]

¹ Nem relációs strukturálatlan adatok nagy tömegének kezelésére képes

Microsoft SQL Server

A Microsoft SQL Server a Microsoft által kifejlesztett és üzemeltetett relációs adatbázis-kezelő rendszer. A strukturált lekérdezési nyelvek egy változatát, a Transact SQL-t vagy T SQL-t használja. Futtatható Linux operációs rendszereken Kubernetes támogatással vagy Windows-on.

A felhasználók könnyen használhatónak és nagyon megbízhatónak írják le, és erős .NET-kompatibilitással rendelkezik, azaz támogatja a Microsoft .NET keretrendszer által támogatott nyelvek (C#) tárolt eljárásait, amely a webes fejlesztés során számomra szükséges lehet.

PostgreSQL

A PostgreSQL egy fejlett relációs adatbázis kezelő rendszer, amely saját procedurális nyelve, a PL vagy pgSQL mellett strukturált lekérdezési nyelvet is használ. A PostgreSQL könnyen használható, és az adatok kezeléséhez az RDBMS² adatbázisok teljes körű funkcióival és képességeivel rendelkezik. Könnyen telepíthető Linux-os környezetekre.

Az SQL-t kiegészítő funkciói közé tartoznak a felhasználó által definiált típusok, egyéni modulok is, a JSON-támogatás, a trigger-ek-et kiegészítő funkciók, valamint a fejlettebb típusok.

Árazás

A PostgreSQL egy nyílt forráskódú adatbázis, bármilyen célú használata, beleértve a kereskedelmi célú felhasználást is, ingyenesen elérhető.

Az MSSQL³ kereskedelmi licencekben kapható, Az MSSQL két fő kiadásban, az Enterprise Edition és a Standard Edition változatban elérhető ugyan mindkettő kiadás fizetős opció de a Microsoft fejlesztők és diákok számára ingyenes verziót is kínál. [1]

Döntés

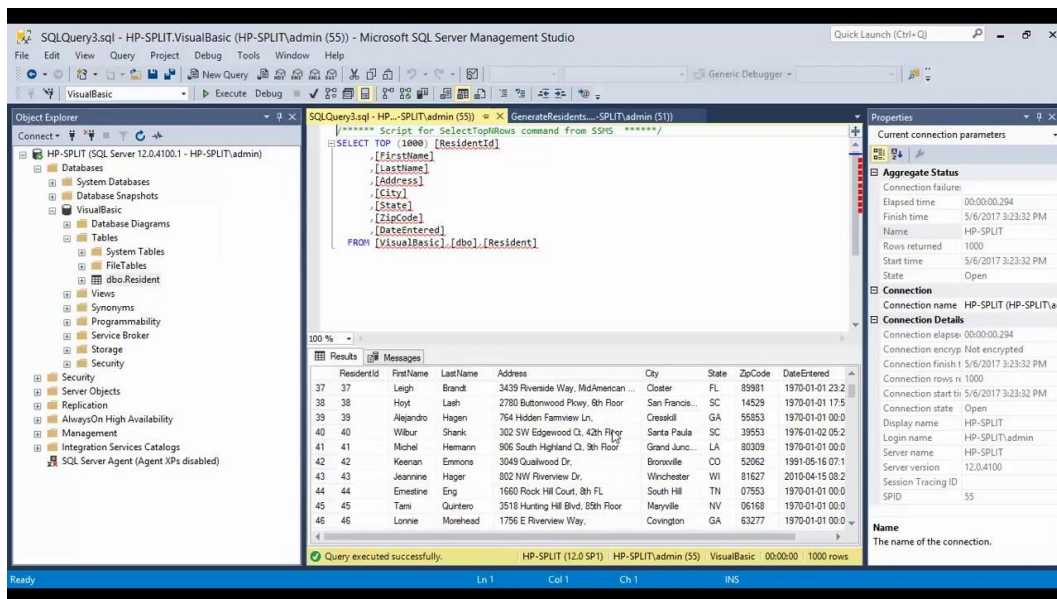
Számomra mindkettő megoldás megfelelő választás, hiszen alkalmazzák a mai technológiákat megfelelő szinten és a népszerűségük miatt rengeteg segítséget kaphatok felhasználóiktól.

Viszont mivel az MSSQL a.NET keretrendszerrel és más egyéb Microsoft szoftverekkel integráltan együtt tud dolgozni, ezért számomra ez lesz a jó megoldás.

² RDBMS: Relational Database Management System

³ MSSQL:Microsoft SQL Server

2.2 SQL Menedzser kiválasztása



2.2-1. ábra Adatbázis menedzser [13]

Az SQL adatbázis menedzselő szoftveremet - mivel szükséges lesz valahol tárolni az adatokat - ezért itt számomra az SSMS⁴ vagy a DBeaver került a lehetséges eszközök listájára. Az SSMS egy Microsoft-os termék, amely szoros kapcsolatban áll a cégen belüli másik ETL szoftverrel az SSIS⁵-el. A DBeaver egy ingyenes alkalmazás, amely nyílt forráskódú. Nagy előnye, hogy más SQL nyelvek, - mint a MySql vagy a PostgreSQL nyelvet is ismeri. [2] [3] [4]

2.2.1 További lehetőségek

Mivel az SSMS egy integrált szoftver a Microsoft Windows operációs rendszerben (2.2-1-es ábra), ezért több lehetőségem van. A cég saját termékeit - így például az Azure felhős szolgáltatásait - is használatba vehetjük. A DBeaver egy népszerű, ingyenes adatbázis menedzser. Ez a cég számára esetleg azért jobb, mert nem csak egy operációs rendszeren működik, hanem akár MAC-en vagy akár Linux-on is könnyedén elérhetőek. [3] [4]

⁴ SSMS: SQL Server Management Studio

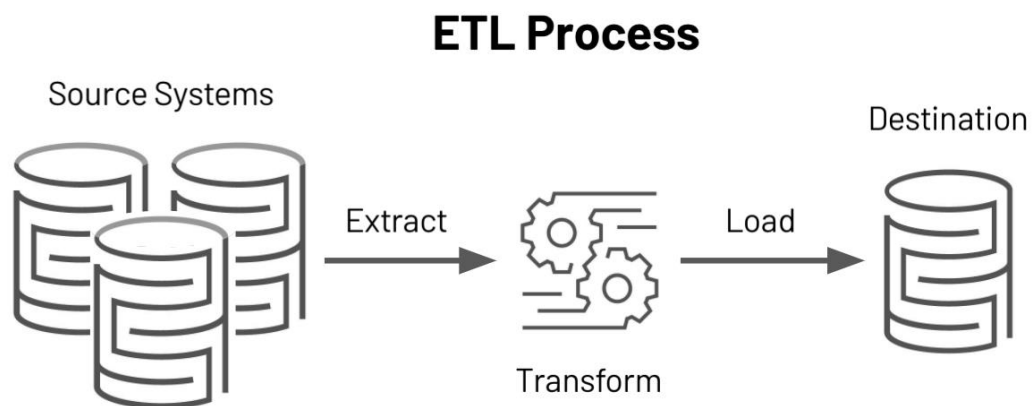
⁵ SSIS: SQL Server Integration Services

2.2.2 Összefoglalás és döntés

Mind az SSMS és a DBeaver is megfelelő megoldás az SQL táblák tárolására. A DBeaver nyílt forráskódú, több SQL nyelvet ismer és több operációs rendszeren is működik. Az SSMS kényelmesebb a beépített funkciói miatt és szorosan együttműködik a Microsoft termékeivel. Fontos még megemlíteni, hogy mind a kettő szoftver ingyenes.

A fentebb említettek miatt az SSMS mellett döntök. [3] [4]

2.3 Az ETL eszköz kiválasztása.



2.3-1. ábra ETL folyamat [14]

ETL (Extract, Transform, Load)

Az ETL, ami a kinyerés, betöltés és átalakítás rövidítése, az a folyamat (2.3-1-es ábra), amelyet az adatokkal foglalkozó szakemberek arra használnak, hogy különböző forrásokból adatokat nyerjenek ki. Az adatokat használható és megbízható adathalmazzá alakítsák át, és betöltsék ezeket az adatokat egy adattárházba, amelyhez a végén az elemzők és más felhasználók hozzáférhetnek.[5]

Az ETL mellett számos más adatátalakító módszer is létezik, ilyen például az ELT. Utóbbi azért kerülhet szóba, mert ez egy újabb technológia és számos előnye van. Az adatok átalakítása után viszont az ETL gyorsabb, hatékonyabb és stabilabb adatelemzést tesz lehetővé, ezért jobb választás az ETL folyamatot alkalmazni.[6]

ETL eszköz lehetőségek

Az ETL eszköz kiválasztásakor a korábbi döntéseimre alapozva, itt is egy Microsoft-hoz tartozó terméket helyeztem előtérbe lehetőségként, az SSIS -t. Ugyanis ma a piacon ez az egyik legnépszerűbb adatátalakító eszköz. Emellett még a nyílt forráskódú PDI-t választottam.

SSIS

Az SSIS szorosan kötődik az SQL szerverhez. Nagyon jól használható a Dashboard-ja és nagyon jól használható a többi Microsoft termékkel. Az alkalmazás a Microsoft Visual stúdiójában elérhető, amely csak Windows operációs rendszeren fut. Az SSIS-t sokan ajánlják, akik napi szinten használják is.[7]

PDI

A PDI egy univerzális szoftver - tehát bármilyen operációs rendszeren fut - a community verziója ingyenes. Azonban, ha már komolyabb eszközöket szeretnénk alkalmazni, akkor havi fix összegért elő kell fizetnünk az enterprise verziót. A szoftver nagyon népszerű és sok területen alkalmazzák hatékonysága és lehetőségei miatt.

A Pentaho támogatja a betöltés előtti és utáni transzformációk széles skáláját. Működésének alapelve, hogy több művelet típust húz be munkaterületekre és dobja el. Ezenkívül a felhasználók Python, Java, JavaScript és SQL nyelven írt egyéni szkripteket is használhatnak fel a munkafelületen.[8] [9]

2.3.1 Összefoglalás és Döntés

Az SSIS és a PDI is megfelelő lenne a folyamat elvégzéséhez. A PDI a népszerűségét előnyeinek köszönheti. Így például a transzformálási lehetőségek, valamint, hogy bármely operációs rendszeren elérhető.

Mivel már az előző döntésem is egy Microsoft termék mellett szólt, ezért az SSIS egy sokkal jobb választás. Az SSIS ugyanis sokkal jobban dolgozik az SQL szerverrel. A vállalkozás jövőbeni tervei szerint nem fog más operációs rendszeren dolgozni a Windowson kívül, ezért is jó döntés az SSIS használata. Valamint a Visual Studio folyamatos fejlesztési lehetőséget biztosít a program számára.

2.4 Webes környezet megoldásának kiválasztása

A cég szeretné az adatbázist egy webes felületen megjeleníttetni, melyen keresztül egy bizonyos módon szerkeszteni tudja az adatokat. Ezen információkat a későbbiekben fel lehet használni a különböző statisztikai kimutatásokban és a döntéshozásban.

A webalkalmazásnak alapvető követelménye, hogy a cégen belüli munkacsoportokat helyesen kezelje. Az ügyvezető ne lássa ugyanazt, amit egy adminisztrátor és több funkcionális érthessen el. Erre a webes felületen külön felhasználói bejelentkező felületet kell létrehozni.

Szükség lesz a megfelelő felhasználói utasításokat helyesen, a felületen keresztül módosítani és ezt a helyi adatbázissal szinkronizálni. Ezt egy korlátozott felhasználói kör használhatja csak és a szinkronizációs funkció után a biztonsági mentés nem szükséges. A kimutatások valós időben való feldolgozása során a mentés napi egyszeri ütemezésben indokolt.

Ezek alapján kellett kiválasztani a webfejlesztői szoftvereket, amiket a feladat során alkalmaznom kell.

2.4.1 A webkeretrendszer kiválasztása

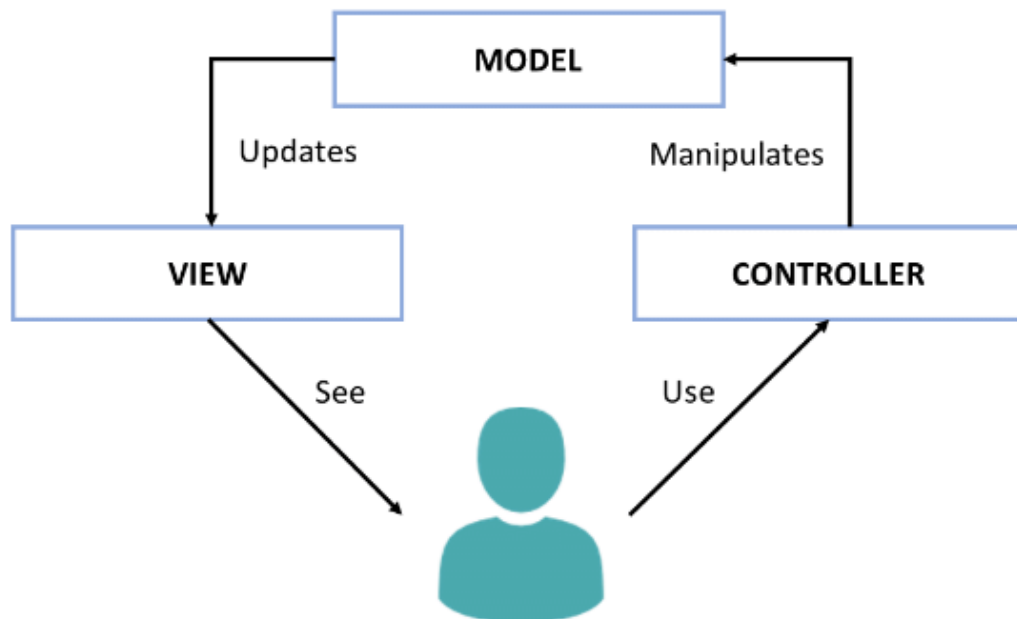
A keretrendszer kiválasztásakor az eddig felsorolt követelményeket figyelembe véve kellett választani. A főbb szempontok a könnyű fejleszthetőség, az eddig választott szoftverek közötti együttműködés.

A választásnál több technológiát vizsgáltam meg. Ilyen például az ASP.NET Core, Ruby on rails és Flask. A kutatásnál fontos szempont volt, hogy nagy támogatottságú szoftvereket keressek, valamint az, hogy manapság széles körben használt legyen.

A döntéshozatalhoz a keretrendszerek számát leszűkítettem háromra, ezek pedig az ASP.NET Core, a Laravel és a Spring MVC. Az ASP.NET Core a Microsoft által fejlesztett keretrendszer. A fejlesztést C#-ban végezzük, hiszen ebben lett megírva az ASP.NET-nek egy továbbfejlesztett verziója. A Laravel nyílt forráskódú PHP keretrendszer, Symfony-n alapul és alapértelmezetten MVC⁶ tervezési mintát támogat. A Spring MVC egy nyílt forráskódú, Java nyelven írt, valamint a Spring keretrendszer egy része. Ez az egyik jelenleg legnépszerűbb. [10] [11] [12] [13]

⁶ Model View Controller

MVC



2.4-1. ábra MVC működése [15]

Az MVC (Model-View-Controller) egy szoftvertervezési minta, amelyet általában felhasználói felületek, szoftver tervezéskor logikai vezérlő megvalósítására használnak. Hasonló szoftvertervezési minták az MVVM (Model-View-Viewmodel) és az MVP (Model-View-Presenter). Mindkettő tervezési minta az MVC alapjaira épülnek. Az MVC tervezési mintának három része van: A Modell, a Nézet és a Vezérlő (2.4-1-es ábra). A Modell kezeli az adatokat és az üzleti logikát. A Nézet kezeli az elrendezést és a megjelenítést. A Vezérlő pedig kezeli a parancsokat a Modell és a Nézet között. [10]

ASP

Ezt leginkább a Microsoft Visual Studio-n keresztül használjuk, tekintve, hogy ez a Microsoft egyik legnépszerűbb alkalmazása. Az ASP.NET Core főbb jellemzői a Host függetlenség, azaz bármilyen webszerveren futtatható, a keretrendszer maga platform független. Tehát lehet futtatni Windows-on Linux-on és akár MAC-en is.

Egy programozási nyelvben (C#-ban) lehet a megjelenítést és a szervert oldali részt is megírni, azaz egyszerűbb a fejlesztést kivitelezni. A keretrendszer nyílt forráskódú és közösség orientált, amely a kódolás folyamatát megkönnyíti. [11]

Laravel

A Laravel egy szerveroldali PHP-keretrendszer. Ennek segítségével, olyan alkalmazásokat lehet készíteni, amelyek jellemzően Backend-et igénylő funkciókkal rendelkeznek. Ilyen például a felhasználói fiókok készítése vagy rendeléskezelés.

A keretrendszer a maga szintjén népszerű a fejlesztők körében a beépített funkciói miatt. Ezen lehetőségei révén egyszerűbbé és gyorsabbá teszi a webes alkalmazások fejlesztését. Ilyen funkciók például az automata tesztelés, Objektum-relációs leképezés. Utóbbinak az a tulajdonsága, hogy osztályokként reprezentálja az adatbázis tábláit az adatok könnyebb eléréséhez és alakításához.

Hasonlóan ez a keretrendszer is ingyenes, nyílt forráskódú. [12]

Spring MVC

A Spring MVC a Spring keretrendszeren belüli könyvtár, amely leegyszerűsíti a HTTP-kérések és a válaszok kezelését. A Servlet API-ra épül, és a Spring keretrendszer alapvető összetevője. A Spring MVC kész komponenseit felhasználva teljesen funkcionális Java webalkalmazásokat építhetünk.

A JUnit keretrendszerrel könnyebb tesztelni. A Spring Container-t fel lehet használni az alkalmazások fejlesztéséhez, valamint a moduljaim könnyebb teszteléséhez.[13]

2.4.2 Döntés

A döntéshozatalban a már eddigi elért tudásomat vettem figyelembe, melyet egyetemi éveim és munkahelyi tapasztalataimra alapozok.

Az árazást nem kellett szem előtt tartani hiszen mindegyik szoftver vagy nyílt forráskódú, vagy ingyenesen elérhető.

Fontos volt számomra eddig az integráltság is, hiszen ez megkönnyíti a munkám nagy részét. Az egyik legjobb választás az ASP.NET MVC, mivel az MSSQL-lel és az SSIS-el is valamilyen formában kapcsolatot tud létesíteni, illetve a fejlesztési környezet - Visual Studio-ban kell dolgozni - is számomra ismerősebb.

2.5 Adatbázis terv

2.5.1 Az adatok jelenlegi állapota

A kézhez kapott adatok rengeteg olyan adatot tartalmaznak, amely nem lesz releváns. A cégtől több adathalmazt is megkaptam Microsoft Excel táblázat formájában. Az adatok a társaság majdnem kezdeti állapotáig visszamenőleg tartalmazza az eddig megvásárolt tárgoncákat és más egyéb speciális eszközöket.

Az előbb említett adatszerkezetben sok olyan információ is van, amelyeket a későbbi vizuális elemzéshez nem fogok alkalmazni. Például sok helyen tartalmaz a munkavédelem frissítése miatt olyan dátum oszlopokat, amelyek bizonyos időszakokra vannak beállítva. Ez a cég munkáját ugyan segíti, viszont a saját alkalmazásomban nem biztos, hogy az összes oszlop szükséges lesz számomra. Ez mindenképpen módosításokat igényel. Továbbá ahhoz, hogy egy konzisztens adatbázist létrehozhassak sokszor aggregálásra lesz szükségem, mivel a régebbi tárgonca készlet nyilvántartások többször hiányosak és különböző kisebb fájlokban találhatók meg.

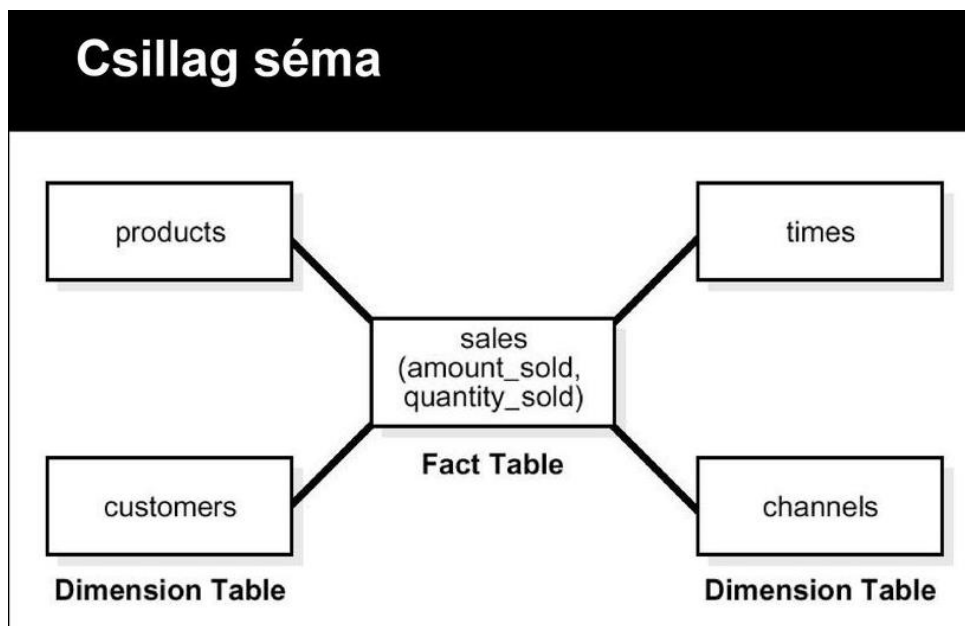
Kaptam még kettő másik adathalmazt is. Egy raktárkészlet nyilvántartást és egy, a szerelők és más egyéb alkalmazott adatait tartalmazó fájlt is. Nekem a szerelő adatbázis lesz fontosabb, mivel a raktár nyilvántartásból sok releváns információt nem nyerek ki.

A szerelő adatbázis terjedelmében a legnagyobb. Tulajdonképpen egy-egy munkához adott munkanapló tartozik. Ebben van a legtöbb információ egy adott azonosítószámhoz tartozó nehézgéphez. Ez az adattartalom a legfrissebb - 2016 óta vezetik - előtte a kevesebb munkaerő miatt még nem kontrollálták ilyen szinten, csak az adott tárgoncához tartozó problémákat jegyezték fel.

Az Excel a mostani időben hanyatlik, mivel a vállalat át akar állni egy SQL nyelvű adatbázisra. Speciális linkek és képek is megtalálhatóak, ezekkel tették könnyebbé a felsővezetők számára, hogy a tárgoncákat jobban nyomon tudják követni és a szerelési munkanaplókat is egy hivatkozás megnyomásával elérhessék Ezt a funkciót a webes kezelőfelületen is el szeretném készíteni, valamint egy-egy diagram segítségével a fontosabb kimutatásokat is elérhessék.

Az adatok titkosításra kerültek. Főleg a személyes információk bizonyultak átalakításra. Mindegyik munkatárs neve kicserélésre került. Ez könnyen megoldható volt a cég esetében, mivel nem rendelkezik sok alkalmazottal. Amely nagyon fontos volt, hogy a bérbe vevő társaság neve és pontos címe meg lett változtatva. Ez már egy jóval egyszerűbb séma szerint lett átalakítva, viszont annak érdekében, hogy a későbbi adatelemzéskor jobban is kivehető legyenek az adatok a nevek kategorizálva lettek. Ki lett alakítva például egy agrár szektorhoz tartozó cég neve esetében: „Mezogazdasagiceg1” nevet kapja.

2.5.2 A vizualizációs séma



2.5-1.Ábra: A Csillag-séma [16]

Az adatbázisistervemnek a végcélja egy vizualizációs megoldás, amelynek a legfontosabb része egy olyan séma, amellyel hatékonyan tudunk bizonyos diagramokat és kimutatásokat beállítani. Erre több megoldás is létezik. Ezek közül három vezető típus van: a Csillag, a Hópehely és a Galaxy-séma.

Úgy gondolom az utóbbi - Galaxy séma - sajnos nem ideális, mivel az adatbázisom szerkezetileg nem elég szerteágazó. Ennek a szerkezeti modellnek az a lényege, hogy több ténytablát tartalmaz és a dimenzió táblái közösek néhány kivételével. A dimenzió táblái között hierarchia van. A projektem számára ez nem ideális, mivel nincs elég adatom arra, hogy megfelelő hierarchikus felépítést készíthessek.

A Snowflake-séma vagy Hópehely a Csillag-séma kiterjesztése, mely további dimenziókat ad hozzá a meglévőkhöz. A dimenziós táblák normalizálva vannak, amelyek az adatokat további táblákra osztják. Habár nagyon csábítóak a tulajdonságai. Így például az, hogy ez a megoldás kevesebb tárhelyet foglal és a táblák implementálása jóval egyszerűbb. Viszont amennyi előnye van annyi hátránya is. Egy kevésbé jó tulajdonsága, hogy a lekérdezés sebessége lassabb, mint a Csillag-sémáé, illetve a lekérdezések is komplexekké válnak a táblák közötti kapcsolódás miatt.

A Csillag-séma - ahogy a 2.5-1-es ábrán látható - alakja legjobban egy csillaghoz hasonlít, középen egy ténytablával és körülötte a dimenzió táblákkal. Ez ideális megoldás, az egyszerűsége és nem mellesleg a lekérdezések gyorsasága miatt. Ugyan több helyet foglal, azonban a feladatom nem igényel akkora helyet, ezért egy praktikus megoldás.

2.5.3 Adattárház tervezése

Első lépésként a nyers adatokat ki kell exportálnom olyan formában, amellyel könnyen be tudom importálni az adatbázisomba. Így az adataimat egy .csv formátumba kell átalakítanom, amely könnyedén megtehető. Ez a legtöbb programban egy beépített funkció. Ez után készítek egy TARGO_ADATBÁZIS megnevezésű adatbázist. Ebben fogom tárolni kinyert adataimat, ahol már elvégzem az adatok megfelelő átalakítását.

A jelenlegi adatok átmozgatása egy SQL szerverre még nem elegendő. Ahhoz, hogy ebből a későbbiekben riportokat is tudjak alkotni, egy adattárházat kell, hogy készítsék. Ezáltal historizálva tudom tárolni a releváns információkat. Az adattárház létrehozásával azon adatokat, amelyek a cég számára fontosak lesznek a jövőben, a felhasználók nyomon tudják követni és analizálni.

A séma megvalósítása érdekében az adatokat megfelelően át kell alakítanom, hogy a végén egy Csillag-séma formájában tudják vizualizálni. Ehhez a tanulmányaim során használt módszereket fogok felhasználni.

Miután az adatokat kinyertem, a megfelelő típusokra kell átalakítanom az adataimat. Az átalakított információkat pedig be kell töltssem a szerveremre. Ezután létrehozok egy adattárház réteget. Itt az adatokat már historizált állapotban fogom tárolni, azaz nem csak felülírom az adatokat, hanem időben visszamenőleg is tárolom a változtatásokat. Illetve bizonyos változó információkat, amelyet nem szükséges historizálva tárolnom, mint például egy targoncának szervíz idejét, amely mindig lejár és újra kezdődik.

Végezetül az adatokat egy Data Mart rétegbe töltöm fel, így megvalósítom a Csillag-sémát. Az adattárház rétegből nyerek ki az adatokat és ezt napi frissítésben töltöm át. Ezen adatok csak egy bizonyos részét tartalmazzák az átalakított rétegnek, csak a riportok számára releváns adatok találhatók meg itt.

3. KÖVETELMÉNYSPECIFIKÁCIÓ

A cég szeretné, hogy a jelenlegi kevésbé integrált adathalmazait és más forrásait egy egységes és normalizált adatbázisba költöztessék át, amelyet a későbbiekben könnyen továbbfejleszthetnek. Mivel sok információ már elavult és nem felhasználható, - vagy nem kívánják felhasználni - ezért csak a számukra és a munkám során releváns adatokat fogom az elkészített adatbázisba betölteni.

A vállalat továbbá tervezi, hogy egy adattárházat valósítsak meg, ahol a fontosabb információkat historikusan tárolom. A változásokat pedig az ETL folyamat mindennapos lefuttatásával szeretné elérni.

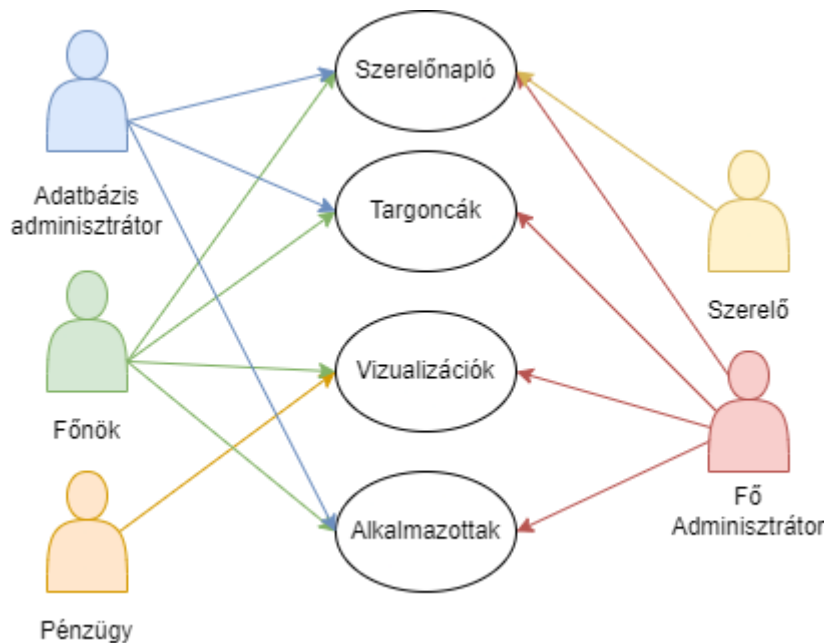
További követelmény, hogy egy webes felületet alkossak meg. Az oldalon ezeket az adatokat használhatják fel, valamint adhatnak hozzá újakat a társaság tagjai. Így gyorsabban és egyszerűbben tudják feltölteni adatokkal a későbbi analízis érdekében. A web alkalmazásban követelmény, hogy az egyes felhasználócsoporthoz számára más-más funkciócsoportok legyenek biztosítva, valamint bizonyos csoportok csak korlátozottan érhesse el egyes, mindannyiuk számára elérhető funkciókat.

A felhasználók számára egy linken keresztül - amely az általuk megadott e-mail címükre kerül elküldésre - regisztrálhatnak az oldalra, a felületen egy felhasználónév és egy jelszó párossal tudnak belépni, így azonosítva az őket és a megfelelő szerepkörükre elérhető funkciókat.

A felületen csak a jelenlegi adatokat használjuk fel. A cég szeretné, hogy a feldolgozott rekordjaikat analizálhassák és a webes felületen ezt a vezetőség számára elemezhetővé tegyem. Ennek érdekében a Csillag-séma felhasználásával egy vizualizációs oldalt is megjelenítek. Itt a tárgoncák és a szerelőnaplók statisztikai és pénzügyi vonzatait is szemléltetem.

4. A WEBES KÖRNYEZET MEGTERVEZÉSE

4.1 Szerepkörök



4.1-1. ábra A Szerepkörök (Saját készítésű ábra)

A szerepkörök fontos rész, hiszen a cég nem minden tagjának lehet hozzáférése az összes webfelülethez. Ahogy a 4.1-1-es ábrán látható a csapatot 5 részre osztottam a vállalat belüli munkákhoz viszonyítva.

A különböző felhasználók egy felhasználónév és jelszó párossal tudnak belépni. Új felhasználó hozzáadása az alkalmazott felvétele után történik. Ezután a Főnök felhasználó hozzáadja az adatbázishoz, illetve ilyenkor generálja le a linket, amellyel később adott alkalmazott beléphet. Bármilyen változást a Fő adminisztrátor és az Adatbázis adminisztrátor kezel.

Az Adatbázis adminisztrátor fogja a legtöbbet használni a webes felületet, ő mindegyik modulhoz hozzáférhet, kivéve a riportokhoz. Általában egy személy lesz csak Adatbázis adminisztrátor, mivel ezt a foglalkozást jelenleg is egy ember látja el. A Szerelőnapló és a Targoncák oldalhoz teljeskörű engedélye van, azaz hozzáadni, törölni és módosítani tud. A szerelőnaplók esetében utólagosan tud módosítani szerelőjegyzéket. Az alkalmazottak felületénél csak módosítani tud, viszont hozzáadni és törölni nem lesz képes. Így neki kevesebb funkció lesz elérhető.

A Főnök, ahogy a 4.1-1-es ábra mutatja, mindenhez hozzáfér. A Főnök a szerelőnaplók és a targoncák oldalon csak olvasni képes, hozzáadni, törölni és módosítani nem. Mivel ezt az Adatbázis adminisztrátor fogja elvégezni számára. A vizualizáció felületen minden riport elérhető lesz számára. Például a szerelőkkel kapcsolatos diagram, targoncákhoz tartozó riport vagy pénzügyi szempontból fontos adatok.

Az Alkalmazottak oldalon teljeskörű hozzáférése van, mivel amennyiben új alkalmazottat akar felvenni vagy el akar egy meglévő alkalmazottat bocsátani, akkor ez mindenképpen az ő döntése lesz.

A Pénzügy felhasználó a könyvelők és egyéb pénzügyi dolgozók számára ad segítséget. Ők kizárólag a vizualizációs oldalt érik el, viszont ezeknek csak egy részét fogják látni. Számukra csak a pénzügyi szempontból érdemleges kimutatások fontosak, így csak azokat a riportokat érik el.

A szerelőknek egy-egy targoncához gyakran a helyszínen ki kell menni kisebb vagy beütemezett szervízhez, ezért a dolguk megkönnyítésére létrehoztam egy szerelőnapló oldalt. Ez nem helyettesíti a papíron megírt munkanaplót, de a cég életét könnyíti meg, mert eddig mindig az adatbázis adminisztrátor manuálisan vitte fel ezeket az adatokat. Itt már nagyobb létszámban tudják felvinni az adatokat. Fontos, hogy nincsen teljes hozzáférésük, csak olvasni tudják az eddig felvitt naplóikat és hozzá tudnak adni újat. Amennyiben esetleges módosításra kerülne sor, akkor már az Adatbázis menedzsert kell kérniük.

A Fő adminisztrátor mindenhez hozzáfér és mindegyik funkciót használni tudja. Ő kevésbé gyakran fogja látogatni az oldalt, mivel az ő szerepköre csak technikai problémák megoldásához szükséges.

4.2 A Webes környezet felületének megtervezése

A felület már az előbb említett négy fő részre lesz lebontva, de minden már előbb említett felhasználó részére más és más funkció lesz elérhető. A jelszó és felhasználónév megadása után a megadott szerepkörnek megfelelő - a jogosultsághoz tartozó oldal – menü fogad. Ebben kijelölheti melyik modulra szeretne lépni és a jogosultság körének megfelelő felületre léphet csak be.

Targoncák modul

A targoncák modul egy kereső egységgel nyílik. Ide írhatunk be keresőszavakat. Itt, egy rádiógombos menü segítségével keressük ki a megadott gépet. Ha keresési értéket megadjuk és mellé a keresési feltételt akkor a keresés gomb megnyomásával kiadja egy listában az adott mód szerinti keresés eredményeit.

A keresést követően a listában megjelennek a gépek legfontosabb tulajdonságai. Ilyen például a gyári száma, pontos neve, üzemórája és emlékeztetők, hogy az adott gépnek mennyi üzemóra után kell a szervizelését elvégezni.

A szervizelésig visszaszámláló szám automatikusan változik. Ennek a számnak az értékét a szerelőnaplók változása befolyásolja. Ezen adat a szervizelési igény megérkezéséig becslést ad arról, hogy mikor kell a legközelebbi szervízt előreláthatólag megtenni.

Egy adott gép módosítására úgy van lehetőségünk, hogy a listában szereplő munkagépek közül ki kell választani a kérdéses tárgoncát. Ezután egy módosítás gombbal érjük el a módosító felületet, ahol minden paraméterét változtathatjuk.

Új gép hozzáadása esetén egy még ki nem töltött oldal jön fel. Ilyenkor egy a módosításhoz hasonló felületet érünk el. Fontos, hogy csak azon információkat kell a gépekről kitölteni, amelyeket csak manuálisan tudunk lekezelni. A maradék adatot a program automatikusan beilleszti. Ilyen például a jelenlegi lokációja a gépnek.

Az oldalon elérhető lesz egy Törlés funkció is esetleges hiba esetén. Eladás esetén egy eladás gombon keresztül kell megadni az eladás időpontját és árát. Ezzel a vizualizációban meg lehet jeleníteni az eladások gyakoriságát és annak pénzügyi vonzatát. Az adattárházunkból nem fog eltűnni, hiszen csak eladás/ selejtezésre kerül.

Szerelőnapló modul

Ebben a modulban a szerelő munkatársak számára egy munkát megkönnyítő oldalt szeretnénk elkészíteni, mivel eddig a vállalatnál minden beérkező naplót az Adatbázis adminisztrátornak kellett felvinnie. Ezáltal ezt a köztes lépést kikerülve a feladat elvégzése után azonnal feltölthetik a szerelők az adatokat. Így egyszerűsítve a cégen belüli adatáramlást. Fontos, hogy itt csak új szerelőnaplót tudnak hozzáadni a szerelő felhasználók, módosítani és törölni csak is az Adatbázis adminisztrátor és a Fő adminisztrátor tud.

A belépéskor minden User mást fog tudni elérni. Egy szerelő a lap tetején egy „új szerelőnapló hozzáadása” gombot és alatta a már meglévő, általa felrögzített naplókat, illetve egy keresőt. A Főnök felhasználó csak a keresőt érheti el, amely hasonló kivitelben fog látszani, mint amely a tárgonca modulban is megtalálható azzal a különbséggel, hogy ő ezeket az adatokat csak olvasni tudja, törölni és hozzáadni nem tud. A Fő adminisztrátor és az Adatbázis menedzser ezeken kívül lát egy módosítás gombot is.

Új munkalap hozzáadása esetén egy új fülben a munkalap minden információját manuálisan kell rögzíteni, amely egy kibővítése egy általános szerelési naplónak. A munkafolyamat megnevezése részben tudja az Alkalmazott egy legördülő menüben a folyamat nevét megadni. Ha azonban speciális vagy nem általánosítható volt az a feladat melyet fel kívánnak rögzíteni, akkor az egyéb opcióra kattintva manuálisan ütheti be.

A Módosítás hasonlóan, mint az előző modulban egy minden információt felsorakoztató oldalon jelenik meg. Ezután minden információ megváltoztatható.

Alkalmazottak modul

Ebben a modulban találjuk meg a cég összes alkalmazottját, valamint a hozzá tartozó információkat és belépési adataikat. Ezt a részt kevesebben érik el és legtöbbször a Főnök felhasználó fér majd hozzá.

Az oldal megtervezésében az alkalmazottak leglényegesebb adatait találjuk meg. Ez az oldal lesz a legritkább látogatottságú.

Új alkalmazott hozzáadásakor először meg kell adnunk a beosztását, vagyis, hogy milyen jogosultságai lesznek. Ilyenkor lehet az új munkatárs adatbázis adminisztrátor, pénzügyes vagy szerelő.

Ezután a felhasználó megadja a felhasználónevét, amellyel a későbbiekben be fog lépni. Egyelőre egy megbeszélte e-mail-címre küldjük azt a linket, amivel be tudja állítani a felhasználónevét, illetve a jelszavát.

Módosításkor kevesebb opciónk van, mint az előző modulokban. Nem minden információt módosíthatunk. Szerkeszthető valamilyen hiba miatt az adott felhasználó beosztása vagy a neve.

Törlés esetén a felhasználói név és jelszó teljesen törlődik, viszont a szerelőnaplók miatt az adott munkás lényegi információi, mint például a neve megmarad továbbra is. Erre ugyanúgy csak a Főnök képes. Ezt neki kell jóváhagynia, de ha egy volt alkalmazott visszatérne a céghez akkor újra hozzá kell adnia.

Vizualizációk

Ebben a modulban az eddigiekkel szemben kevesebb opció elérhető. A diagramokat csak értelmezni tudjuk, újat hozzáadni nem lehetséges.

A riportokban eltérés van. Példaként említve, mivel a pénzügyes számára nem relevánsak bizonyos tárgyonca adatok, - a szerelések száma megyékre bontva - ezért számára csak a pénzügyi statisztikák lesznek megjelenítve, azaz csak azok a vizualizációk és diagramok.

5. ADATBÁZIS KIALAKÍTÁSA

A cég által nyújtott adatok kevésbé strukturáltak, azok kezelését eddig egy ember végezte el. Ahogy említettem Excel fájlokban vannak tárolva az adatok. Nagy részük 2016-tól egységes. A vállalat számára ezek relevánsak, mivel 2016 előtt csak a targoncákról tároltak információkat, a szerelőnaplóról elenyészve vagy a targoncákhoz integrálva, amely átláthatatlanná tette a munkafolyamatot.

A cél, hogy ebből a nagy Excel fájlból kiszedjem a fontos adatokat, amelyekkel tovább szeretnének dolgozni és fejleszteni. A további terv, hogy a munkafolyamatot könnyebbé tegyem és a szerelőket is bevonjam a munkafolyamatba, ne csak egy alkalmazott manuálisan frissítse a szerelőnaplókat. Ugyan a papírforma, szükséges ezért a naplók kialakításakor odafigyeltem, hogy ugyan vannak olyan adatok, amiket a későbbiekben nem használok fel, de a továbbfejlesztési lehetőségek miatt az összes munkalapon megnevezett adatot felhasználtam.

A vállalat mivel dinamikusan nő, ezért a munkaereje is nagyon változik. Jelenleg nem túl sok emberrel rendelkezik, ezért nem kellett külön nyilvántartani őket. A későbbiekben azonban szükséges lesz kezelésük, ezért egy külön alkalmazottak táblát is implementáltam egyelőre egyszerűbb adatokkal, amelyeket a felhasználókezelésben alkalmazok.

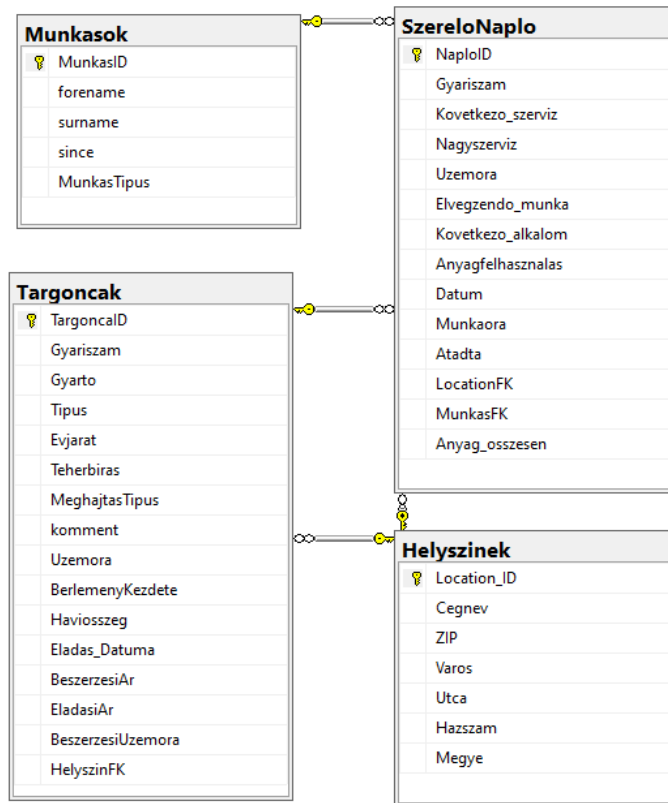
A nehézgépek nagy része anyagmozgató gép. A cég számára fontos a márkahűség, ezért egy-egy géptípusnál egy márkát használnak. Így is növelve a gyors munkát a javításoknál. Emiatt a legtöbb gyáriszám hierarchikus módszerben tűnik fel, illetve egy-egy gép típusjegyében is visszaköszönnek fontosabb információk az adott gépről. A legfőbb szempont a bérbeadás időtartama, a beszerzési és az eladási ár közötti mozgás, továbbá a cég folyamatosan vezeti jelenleg a targoncák üzemóra állását. Eddig manuálisan lett fentiek közül minden változtatva és megjelenítve. Ezeket az adatokat a szerelőnaplóból kellett kiszedni, a legutóbbi naplójegyzékből és emiatt duplikáció lépett fel.

A cég mivel jelenleg kis-közép vállalat, ezért sok olyan hanyagolt adat van, amit kevésbé vagy egyáltalán nem töltöttek fel. Ilyen például a bérbeadásnál a bérbe vevő cégeknél a targonca üzemeltetési helyének adatai. Ezeket csak a cégnév és város adatokkal kaptam meg, a nagyobb géppark és a folyamatosan növekvő bérlemények miatt ezt átalakítottam, és hozzáadtam további információkat, amellyel könnyedén tudják azonosítani a cégeket, illetve további statisztikákat tudnak készíteni és feldolgozni.

A források megszerzése után nekiláttam az adatbázis kialakításának és a meglévő adatok átalakításának. Az adatbázis kialakítása során odafigyeltem a jelenleg fellépő redundancia kiékelésére. A cég kéréseit is figyelembe vettem, ezért bizonyos adatokat az adatbázisban hagytam a továbbfejlesztési lehetőségeket is figyelve.

5.1 Adatbázis

Az adatbázis feltöltéséhez az általam kialakított Excel fájlt kellett exportálnom .csv fájl formátumba és az SQL szerveren pedig létrehoznom az adatbázist és benne a táblákat. A betöltéshez az SSIS-t használtam fel egyszeri betöltéssel.



5.1-1. ábra Az adatbázis terv (saját készítésű ábra)

Az átalakítás után az adatokat szétszedtem, az 5.1-1-es ábrán látható táblákra. A megfelelő adatokat, amelyek redundánsak voltak elhagytam, a kapcsolatokat pedig megfelelően használtam fel.

5.2 Táblák és a benne lévő adatok

5.2.1 Targoncák tábla

Oszlopnév	Típus	Magyarázat	Kulcs érték
TargoncaID	int	A gépekhez generált ID oszlop	Elsődleges kulcs
Gyariszam	varchar(50)	Egy adott gép gyáriszáma	Unique identifier Idegen kulcs
Gyarto	varchar(20)	A nehézgép márkája vagy gyártója	
Tipus	varchar(20)	A gép típusjele	
Evjarat	int	A gép gyártásának éve	
Teherbiras	int	A gép felhasználható teherbírása kg-ban megadva	
MeghajtasTipus	varchar(20)	A gépek meghajtásának típusa pl.:Elektromos,Diesel...	
komment	varchar(256)	Egy adott géphez tartozó megjegyzés ez általában hibakód vagy emlékeztető	
Uzemora	int	A gép jelenlegi üzemórája	
BerlemenYKezdet	date	A gép bérleményének kezdeti dátuma	
Haviosszeg	int	A géphez tartozó jelenlegi haviösszege euróban	
Eladas_Datuma	date	A gép eladásának dátuma	
BeszerzesiAr	int	A géphez tartozó Beszerzési ár euróban	
EladasiAr	int	A géphez tartozó eladási ár euróban	
BeszerzesiUzemora	int	A gép beszerzésekor megmért üzemóra	
HelyszínFK	int	A gép jelenlegi tártózkodási helyének kulcsa	Idegen kulcs

A Targoncák táblázat tartalmazza a 2016-tól beszerzett és meglévő targoncák összes aktuális adatát a jelenlegi állapotukban. A vállalat általában bérlemények kezdetét veszi figyelembe, hiszen innen tudják megmondani, hogy mennyi ideje dolgozik egy adott nehézgép a bérbeadás helyén. A havi összeg nagyjából egységes, azonban egy nehézgépre olykor speciális igények miatt más emelőadaptert vagy juttatást igényelnek. Ezért sokszor változik a havi bérleti díj, illetve a piac vagy pénzügyi helyzet miatt is változhat ez a szám.

5.2.2 Szerelőnaplók tábla

Oszlopnév	Típus	Magyarázat	Kulcs érték
NaploID	int	A naplókhoz generált ID oszlop	Elsődleges kulcs
Gyariszam	varchar(50)	Egy adott gép gyáriszáma	Idegen kulcs
Kovetkezo_szerviz	int	A gép következő szervizéhez tartozó üzemóra	
Nagyszerviz	int	A gép nagy szervizéhez tartozó üzemóra	
Uzemora	int	A gép aktuális üzemórája	
Elvegzendő_munka	varchar(256)	Az elvégzendő munkafolyamat	
Kovetkezo_alkalom	varchar(256)	A következő szervízhez maradt munkafolyamatok	
Anyagfelhasznalas	varchar(256)	A géphez felhasznált alkatrészek és anyagok	
Datum	date	A javításhoz tartozó dátum	
Anyag_osszesen	varchar(256)	A géphez eddig felhasznált alkatrészek és anyagok	
Munkaora	decimal(10,2)	A munkafolyamathoz tartozó munkaidő 2 tizedes pontossággal	
Atadta	varchar(20)	A kolléga, aki javítást végezte	
LocationFK	int	A gép tartózkodási helyének kulcsa	Idegen kulcs
MunkasFK	int	A munkás idegen kulcsa	Idegen kulcs

A Szerelő Napló tábla létrehozásánál fontos szempont volt a továbbfejlesztési lehetőség. Jelenleg ugyanis a cég csak azokat az adatokat töltötte fel, amely statisztikai szempontból fontos. A fizikai papíron több adatot írnak fel, amelyet az adminisztrátor manuálisan begépett. Ez több szempontból is rossz, hiszen egy adott munkafolyamatot kétszer kell elvégezni és sok információ nem kerül eltárolásra az adatbázisba. Az anyagfelhasználás és az anyag összesen oszlopok nagyon ritkák és sok esetben helytelenül voltak vezetve. Ezt próbáltam kijavítani. A későbbiekben így, hogy ez az adat is bekerül, több információ kerül eltárolásra.

5.2.3 Munkások tábla

Oszlopnév	Típus	Magyarázat	Kulcs érték
MunkasID	int	A munkásokhoz generált ID oszlop	Elsődleges kulcs
forename	varchar(256)	Az adott alkalmazott vezetéknéve	
surname	varchar(256)	Az adott alkalmazott keresztnéve	
since	date	Az alkalmazott mióta dolgozik a cégnél	
MunkasTípus	varchar(20)	Az alkalmazott beosztása	

Az alkalmazottak jelenleg nem túl sok adattal vannak feltöltve, azonban a jövőben továbbfejlesztési lehetőségként részletesebb adatokkal kívánom ellátni az egyes alkalmazottakat.

5.2.4 Helyszínek tábla

Oszlopnév	Típus	Magyarázat	Kulcs érték
Location_ID	int	A helyszínekhez generált ID oszlop	Elsődleges kulcs
Cegnev	varchar(150)	A cég neve	Unique identifier
ZIP	varchar(256)	A helyszínhez tartozó irányítószám	
Varos	varchar(100)	A helyszínhez tartozó városnak a neve	
Utca	varchar(256)	A helyszínhez tartozó utcanév	
Hazszam	int	A helyszínhez tartozó házszám	
Megye	varchar(256)	A helyszínhez tartozó megyének a neve	

6. ADATTÁRHÁZ KIALAKÍTÁSA

A cég egyik elvárása egy vizualizációs réteg létrehozása. Ennek érdekében egy adattárházat kellett megterveznem. A feladatomban létrehoztam a megfelelő rétegeket, amelyek során az első folyamatot csináltam meg, mely az Extract vagyis a kinyerés volt.

Első lépésként a megkapott Excel állományt alakítottam át a saját használathoz. Ez egy nagyobb állomány, amely tartalmazza az összes eddigi adatot. A fő része a targoncák munkalap, amelyet hivatkozásokkal oldottak meg a cég alkalmazottai, annak érdekében, hogy a megfelelő targoncához hozzárendelje az adott szerelési naplót.

A targoncák táblához a legtöbb adatot a fő munkalapból tudtam kinyerni. Azonban voltak olyan adatok, amelyeket - például a bérlemények kezdetét - a géphez tartozó szerelési naplóból értem el. Itt írták fel a targoncák cégek közötti mozgását.

A szerelőnapló táblánál az összes rekord rendelkezésre állt. A nagyszervíz oszlopot ki kellett számolnom, mivel annak az adatát nem mindig adták hozzá, így ez csak a fizikai munkalapra került rá. A probléma még a rekordok elrendezésével volt, mivel az adatok az általános elrendezéssel ellentétben transzponáltan volt. A munkájuk megkönnyítése, illetve az átláthatóság fényében ezt át kellett alakítanom a hagyományos elrendezéshez.

A helyszíneknél nagyon kevés információ állt a rendelkezésemre, így a legtöbbnél magamtól kellett hozzáadnom adatokat. A titkosítás érdekében a legtöbb utca és házszám nem valós vagy nagyon általánosított. Annak érdekében, hogy reprezentálja a valós adatokat a város, illetve a megyék a jelenlegi adatokkal egyezők. Az adatok struktúrájából következtetve a cégnévet egyedi azonosítóval láttam el.

A munkások táblát magamtól kellett elkészítenem az alkalmazottak kis létszáma miatt ez nem volt hosszú munkafolyamat. Az adatok jelenleg csak alapvető adatokkal vannak feltöltve, azonban ez könnyen továbbfejleszthető.

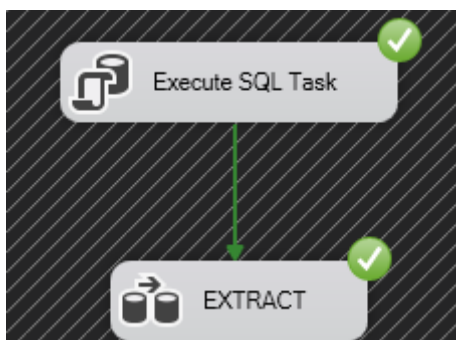
Az Extract után az adattárház réteg megvalósítása következik. Itt a cégnek fontos attribútumokat historizálok, illetve állítok be változó adatra. A fő adatbázis minden oszlopát áttöltöm. A réteg előtt a megfelelő adatokat átalakítom, mivel bizonyos oszlopok értékei megváltozhatnak. További lépésként létrehozok a Data Mart réteghez szükséges úgynevezett nézeteket, amelyek a Csillag-séma adattábláihoz szükségesek.

Ezután megvalósítom a Csillag-sémát, amely egy bizonyos részét képezi csak le az adattárházamnak. A réteget TARGO_DM néven hozom létre. A rétegben egy generált dimenzió tábla is megtalálható DimDate néven. Ez a kimutatások részletezésért szükséges.

7. AZ ETL FOLYAMAT

7.1 Extract réteg

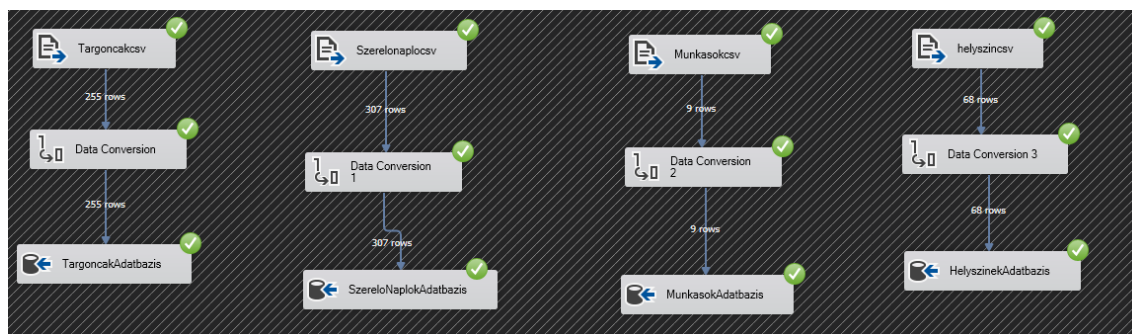
Az Extract folyamán az adatokat csak egyszer töltöm át. Ebben a lépésben a létrehozott adatbázis sémát töltöm fel a megfelelően kinyert adatokkal. A nyers adatokat először egy átalakított Excel fájlban tároltam, ezeket exportálnom kellett .csv fájl formátumban, amelyet az első rétegben forrásként betöltöttem.



7.1-1. ábra Extract csomag tartalma (Saját készítésű ábra)

Első lépésként - ahogyan a 7.1-1-es ábrán látható - egy SQL Task-ot futtatok le. Ez azért szükséges, hogy a táblának a tartalma az áttöltéskor biztosan üres legyen. Ez egy egyszerű Transact-SQL parancsot futtat le, amely az adatok és bármilyen más feltétel nélkül az összes adatbázisban létrehozott táblának a tartalmát törli.

A metódus lefutása után megkezdődik a betöltési folyamat, amely az aktuális naprakész adatokkal tölti fel az adatbázist.

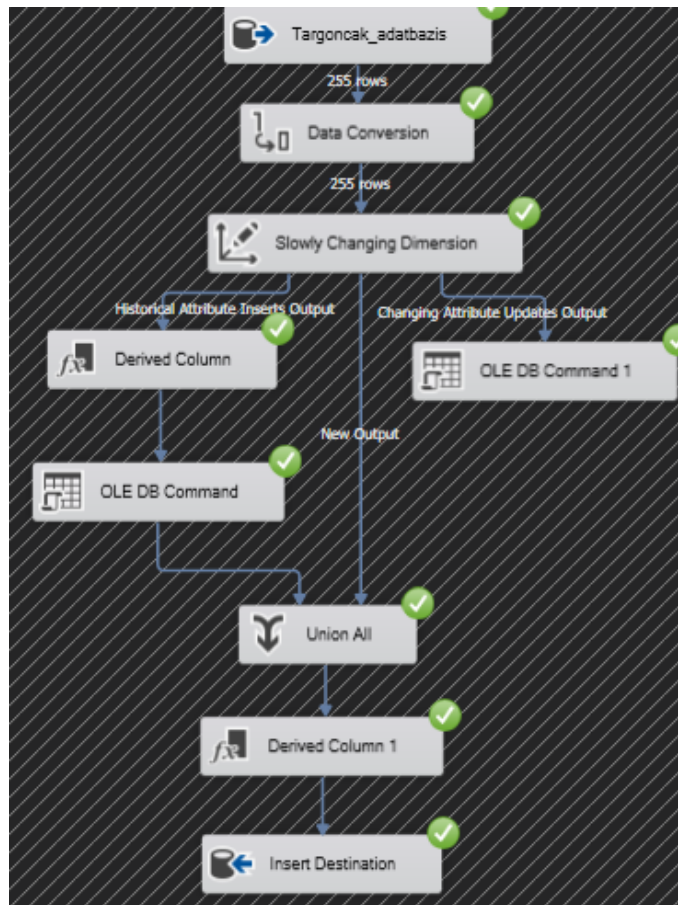


7.1-2. ábra Az Extract metódus (Saját készítésű ábra)

A táblák forrás állományai - mivel nyers állapotban vannak - alapértelmezetten szöveges típusúak, ezért ahogyan a 7.1-2-es ábrán látszik, egy adatkonverziós metódussal átalakítottam az adattáblák helyes típusaira. Odafigyeltem, hogy adatvesztés ne legyen és a dátum típusokra megfelelő régiókat állítottam be.

7.2 Transform réteg

A Transform rétegben végzem el az adatokhoz azokat a transzformációkat, melyek a feladatom végén a vizualizációhoz szükségesek. A rekordok változásait az SCD (Slowly Changing Dimension) metódusokkal állítom be a megfelelő típusra, amely lehet SCD type 0, amely fix attribútum. Ez azt jelenti, hogy az az oszlop nem fog és nem szabad, hogy változzon. SCD type 1 olyan metodika, amely az adatoknak megengedi a változásokat és ezeket a változtatásokat dinamikusan cseréli és frissíti. Továbbá lehet SCD type 2, amely historikusan tölti fel adatokkal az adattáblát. Tehát az adatok változásait végig követhetjük időrendi sorrendben ValidFrom és ValidTo oszlopok segítségével.



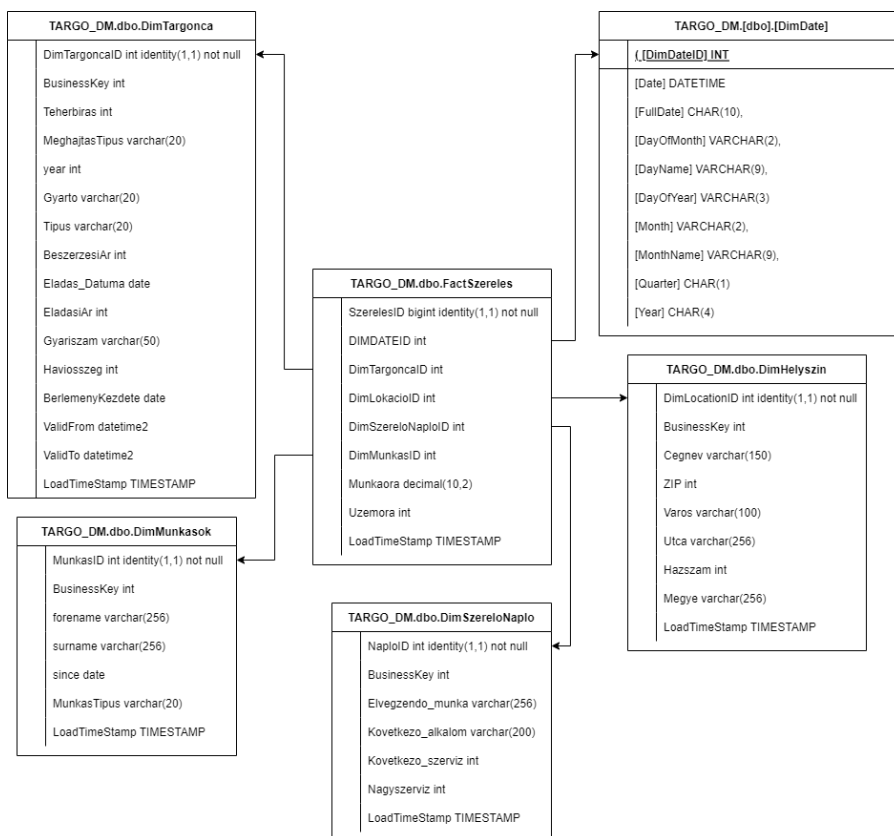
7.2-1. ábra Transform példa (Saját készítésű ábra)

Az adatváltozások követése csak bizonyos adatbázison belüli attribútumokra szükségesek. A cég számára legfontosabbakat historizáltam le, ilyen például egy targoncának bérleti időszakának kezdeti időpontja (7.2-1-es ábra). Ugyanúgy fontos az is, hogy az árak hogyan változtak egy bizonyos targoncánál.

A legtöbb attribútumra azonban nem szükséges a változásokat megfigyelni. Ilyen például a szerelőnaplóban a munkafolyamatok megnevezése, hiszen általában csak akkor változik meg egy már meglévő adaton, ha valamit kihagytak egy adott javításkor. Olyan is előfordulhatott, hogy olyan anyagot vezettek fel a szerelőnaplóra, amelyet ténylegesen nem használtak fel. Mivel a módosítás egyébként sem a szerelő feladata, ezért ezt az adminisztrátor fogja végrehajtani. Emiatt szükségtelen az adatok értékeinek az időbeli nyomon követése.

7.3 Load réteg

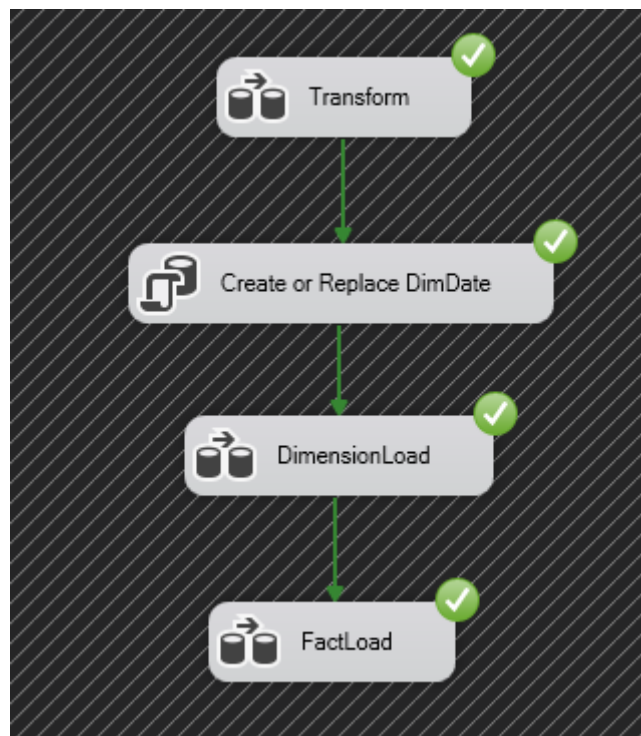
Ebben a rétegben valósítom meg a cég számára a vizualizációhoz szükséges adatok kiválasztását. Elsőként egy Data Mart adatbázist hozok létre, amelyben a tervhez specifikus adatokat válogatok ki és ebből megalkotok egy Csillag-séma alapú tábla rendszert (7.3-1-es ábra).



7.3-1. ábra A csillag séma terve (Saját készítésű ábra)

A betöltési réteg rögtön az adattárház áttöltés után folytatódik, mivel a Data Mart egy bizonyos leképezését teszi ki. Ehhez mindenekelőtt nézeteket kellett létrehoznom. A leképezéseket azért készítettem, hogy csak azokat az adatokat kapja meg a tény - illetve a dimenzió táblák, amelyeket kiválasztottam és a megfelelően összekapcsoltam. Így adom már át az SSIS munkafolyamatomnak.

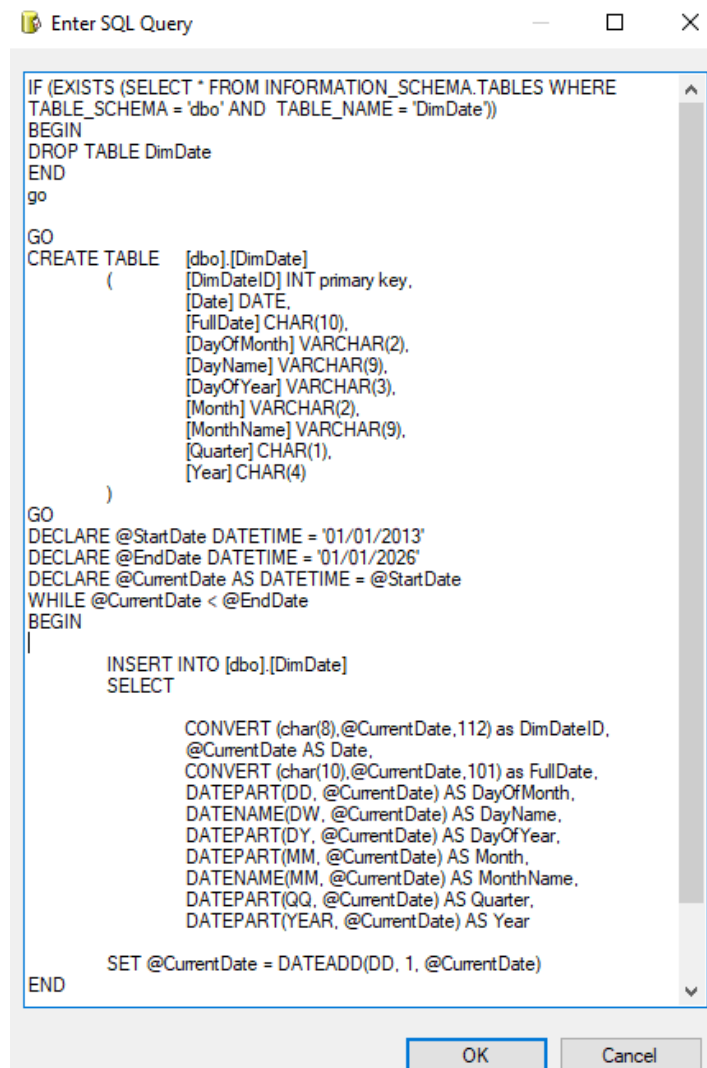
A létrejött nézetek után kialakítottam a Csillag-séma tábláit és megkezdtem a betöltési folyamatot.



7.3-2. ábra Data Mart feltöltés folyamatára (Saját készítésű ábra)

Mivel a ténytáblában többek között dátumok is szerepelnek, - ahogyan ezt lehet látni a 7.3-1-es ábrán - ezért egy Dátum dimenziót is hozzárendeltem a sémához, amellyel könnyebben és hatékonyabban tudjuk elemezni a ténytábla adatait.

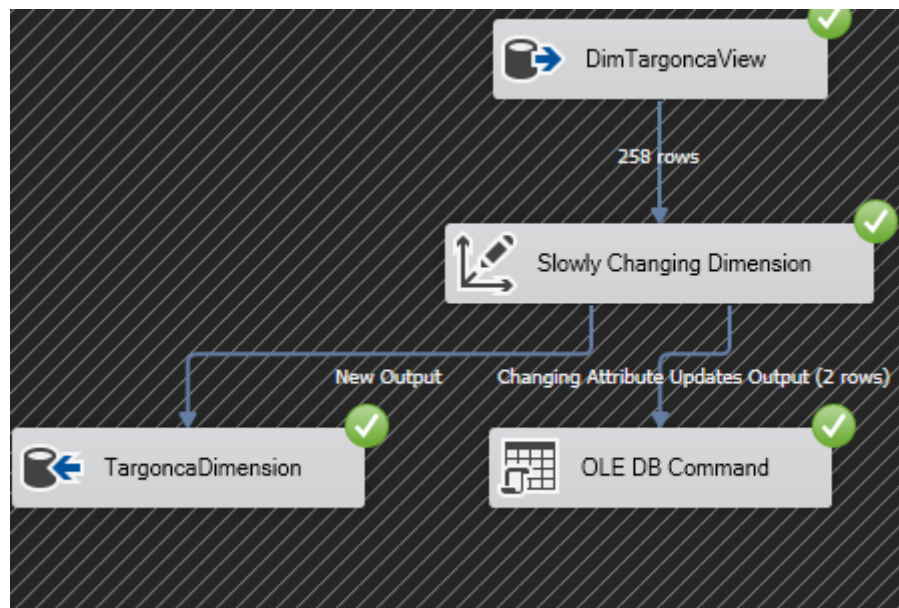
Ezt azonban egy külön eljárással fel kellett töltenem és a dátumot integer típusúvá kellett átalakítanom a gyorsabb kapcsolat érdekében. Ehhez egy Transact-SQL parancsot futtattam le, amely átkonvertálja a dátumot számmá és a dátum alapján kiválogatja azokat az információkat, amely fontos a feladat számára (7.3-2-es ábra).



7.3-3. ábra Dátum dimenzió feltöltése adatokkal (Saját készítésű ábra)

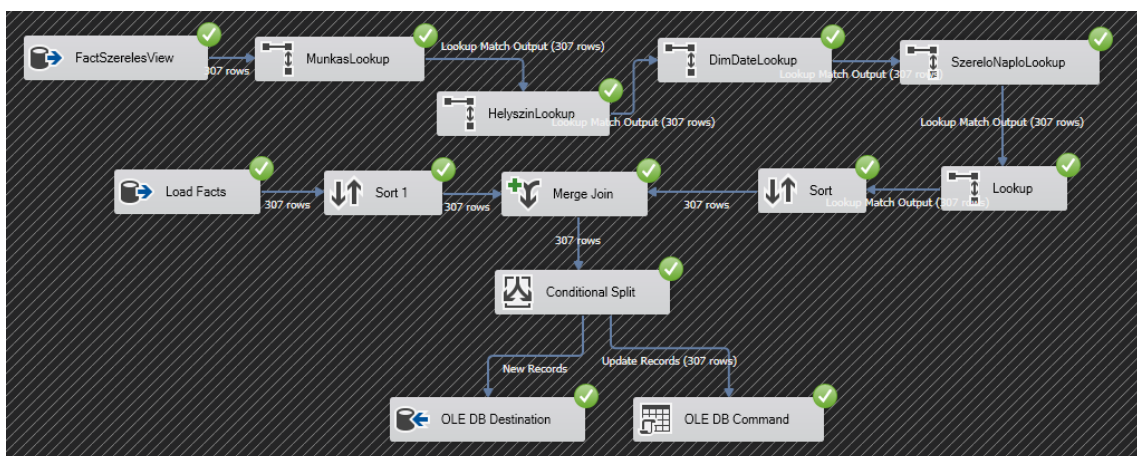
Az eljárás azzal kezdődik, hogy megkeresi a már létező dátum dimenzió táblát. Amennyiben talál ilyet, akkor azonnal törli ezeket az adatokat, majd elindítja az eljárást, amellyel létrehozza a táblát (7.3-3-as ábra). Az adattábla a cég számára jelenleg legfontosabb attribútumokkal rendelkezik. Ilyen a hét napjainak megnevezése, illetve a negyedévek jelzése. A metódus 2026-ig tölti fel a táblát, azonban ez bármikor módosítható, illetve tovább is fejleszthető.

Következő lépésként a dimenzió táblákat töltöttem fel. Ezen művelethez újra felhasználtam az SCD⁷-t. Az adatokat egy nézetből szedem ki, így már csak arra kell figyelnem, - ahogyan a 7.3-4-es ábrán is látszik - hogy az újonnan bekerülő adatokat frissítse.



7.3-4. ábra Dimenzió táblák feltöltése részlet (Saját készítésű ábra)

Miután elkészültem ezzel a részfeladattal már csak a ténytáblát kellett betöltenem. A ténytáblát a többi dimenzió táblával összekapcsoltam a LOOKUP transzformáció segítségével (7.3-5-ös ábra).



7.3-5. ábra A ténytábla feltöltése (Saját készítésű ábra)

⁷ Slowly Changing Dimension

Ez a folyamat nagyon hatékonyan szűri ki az egyező és nem egyező adatokat a céltáblában. A feladatot minden egyes dimenzió táblával elvégeztem, majd a végén rendeztem a ténytábla elsődleges kulcsa szerint.

A DataMart rétegben napi frissítéseket alkalmazunk (az adatok frissülnek), ezért célszerű egyesíteni a jelenlegi táblát és az újonnan beillesztendő adatokat. Erre azért van szükség, hogy elkerüljem azt, hogy folyamatosan új adatok kerüljenek beillesztésre. A folyamat végén pedig egy feltétellel megadva tölti fel a táblákat (7.3-5-ös ábra).

A cég az adattárház réteg és a DataMart réteg folyamatát ütemezni szeretné. Ehhez több fajta megoldás létezik. Leggyakrabban az SQL szerveren belül egy munkafolyamatot hozunk létre, az úgynevezett SQL Server Agent⁸-en keresztül. Mivel ez a szolgáltatás csak a standard verzióban elérhető, ezért más megoldáshoz folyamodtam.

Egy .bat fájlt hoztam létre, amellyel elindítom a megadott SSIS csomagot. Ez jelen esetben az adattárház transzformációs és a betöltési rétege.

Indítási feltétel szerkesztése

Feladat megkezdése: Ütemezésnél

Beállítások

☐ Egyszer

☒ Naponta

☐ Hetente

☐ Havonta

Indítás: 2022. 04. 04. 16:29:00 ☐ Időzónák közti szinkronizálás

Ismétlődés: 1 nap után

7.3-6. ábra Az ütemezés beállítása (Saját készítésű ábra)

Ez után beállítottam a Windows feladat ütemezőben a munkafolyamat nevét és megadtam a fájlt, amelyet el kell indítania, majd utána megadtam az indítási feltételt. A 7.3-6-os ábrán megadott napi ütemezést választottam és a pontos időpontot, illetve egy indulási dátumot.

⁸ Az SQL Server Agent egy SQL szerveren belüli szolgáltatás, ami adminisztrációs feladatokat hajt végre, amelyeket a szerveren belül feladatoknak vagy jobs-nak neveznek.

8. A WEBES KÖRNYEZET MEGVALÓSÍTÁSA

A webes környezet megvalósításánál figyelembe vettem az olyan speciális kéréseket, mint például a könnyebb kezelhetőség és az olyan jelző eszközök, amelyeket a saját adathalmazukban használtak - így például a szervízhez tartozó szám.

Az SQL szerverhez való kapcsolathoz megoldásként egy beépített csomagot használtam fel, amely leképezte számomra és hozzáadta könyvtáramhoz a megfelelő adattáblákat.

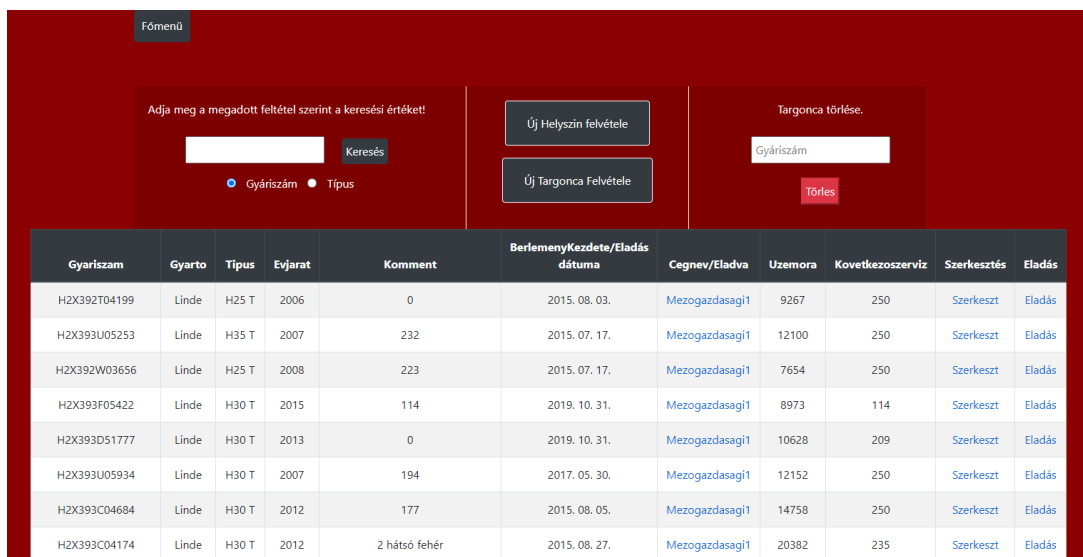
Az oldal fontosabb tulajdonsága még a bejelentkezési rendszer. Ehhez egy egyszerű felhasználónév és jelszó kombinációt adtam meg. Ezeket az adatokat titkosítva az SQL szerveren tároltam. Az adatbázis szerveren létrehoztam egy Felhasználók adattáblát, amelyben csak az alapvető adatok kerülnek tárolásra. Ez pedig a felhasználónév és a generált link, amellyel regisztrálva van.

A webes felületem a belépés után egy főmenüvel kezd, ahol a 4 fő részt tudjuk elérni, a nehézgépekhez tartozó targoncák menüpont, a szerelői jegyzékhez egy szerelőnapló, illetve az alkalmazottak menedzseléséhez és a vizualizációkhoz is egy-egy hivatkozás.

8.1 Modulok és működésük

8.1.1 Targoncák oldal

A fő modul, amelyen a lényegi információkat találjuk az egyes gépekről a Targoncák weboldal. Ez a legtöbb tag számára elérhető, azonban bizonyos funkciók nem. Ilyen például a szerkesztés és az eladás.



Gyáriszám	Gyarto	Típus	Évjarat	Komment	BerlemlenyKezdete/Eladás dátuma	Cegnev/Eladva	Uzemora	Kovetkezoserviz	Szerkesztés	Eladás
H2X392T04199	Linde	H25 T	2006	0	2015. 08. 03.	Mezogazdasagi1	9267	250	Szerkeszt	Eladás
H2X393U05253	Linde	H35 T	2007	232	2015. 07. 17.	Mezogazdasagi1	12100	250	Szerkeszt	Eladás
H2X392W03656	Linde	H25 T	2008	223	2015. 07. 17.	Mezogazdasagi1	7654	250	Szerkeszt	Eladás
H2X393F05422	Linde	H30 T	2015	114	2019. 10. 31.	Mezogazdasagi1	8973	114	Szerkeszt	Eladás
H2X393D51777	Linde	H30 T	2013	0	2019. 10. 31.	Mezogazdasagi1	10628	209	Szerkeszt	Eladás
H2X393U05934	Linde	H30 T	2007	194	2017. 05. 30.	Mezogazdasagi1	12152	250	Szerkeszt	Eladás
H2X393C04684	Linde	H30 T	2012	177	2015. 08. 05.	Mezogazdasagi1	14758	250	Szerkeszt	Eladás
H2X393C04174	Linde	H30 T	2012	2 hátsó fehér	2015. 08. 27.	Mezogazdasagi1	20382	235	Szerkeszt	Eladás

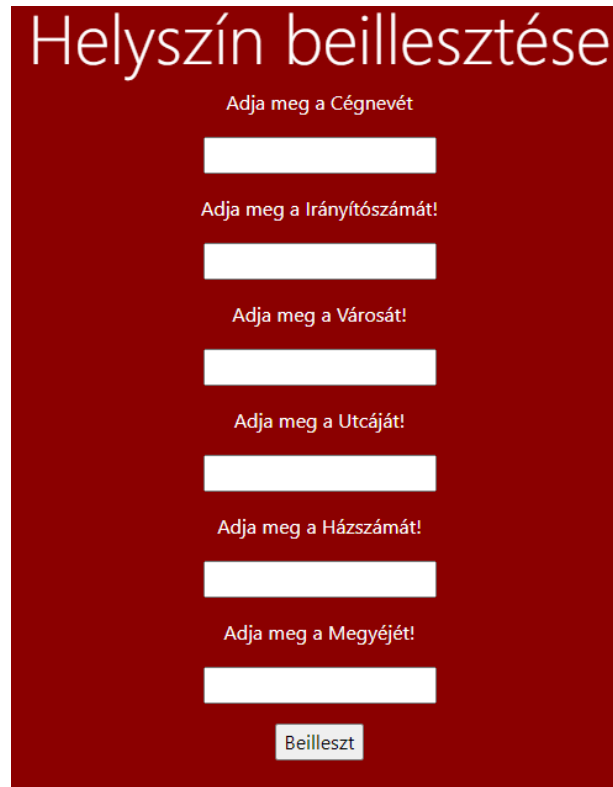
8.1-1. ábra Targoncák oldal részlet (Saját készítésű ábra)

Az oldal tetején található egy Főmenü gomb, amely visszavisz az alap oldalra. A folyamatot meg szerettem volna könnyíteni ezért, ha a felhasználó az egérrel a Főmenü gombra navigál, akkor mellette megjelenik az összes másik oldalnak a hivatkozásai is.

A táblázat felett megtalálható még egy kereső mező is. Itt megadtam kétfajta feltételt, amely alapján elvégezhető a keresés. Az egyik a Gyáriszám (ha egy specifikus munkagépet szeretnének megtalálni), a másik pedig egy Típus, amivel a nehézgépek típusjeleit szűkítheti le (8.1-1-es ábra).

Az oldal jobb oldalán megtalálható ezen kívül egy Törlés mező is. Itt csak pontosan egyező Gyáriszámú targoncát lehet törölni. Ezt főképp hiba esetén fogják használni.

Egy új targonca felvételekor a legtöbb esetben első lépésként egy új helyszínt kell megadni, melyet az „Új Helyszín felvétele” gombbal érhet el az Adatbázis adminisztrátor.



Helyszín beillesztése

Adja meg a Cégnevét

Adja meg a Irányítószámát!

Adja meg a Városát!

Adja meg a Utcáját!

Adja meg a Házszámát!

Adja meg a Megyáját!

Beilleszt

8.1-2. ábra Új helyszín beillesztése (Saját készítésű ábra)

A helyszín feltöltésekor az adatokat manuálisan kell bevinni (8.1-2-es ábra). Azonban legtöbb esetben automatikusan beilleszti a program a kívánt adatokat. Ilyen például a helyszíneknél az elsődleges kulcs, ugyanis ez automatikusan inkrementálódik.

Bizonyos esetekben, a targonca alap adatainak módosításánál a targonca meghajtásának típusát nem kell manuálisan beütni, hanem dinamikusan az eddig bevitt munkagépek meghajtásai közül egy lenyíló menüben választhatunk. A targoncák felvételekor a helyszínt nem változtathatjuk, mivel alapértelmezetten a cég telephelyére kerül és ilyenkor az összes dátum attribútumú oszlop is alaphelyzetbe kerül.

Módosítás

Adja meg a Gyáriszámot

H2X392T04199

Adja meg a Gyártó-t

Linde

Adja meg a gép típusát

H25 T

Adja meg a gép Évjáratát!

2006

Adja meg a gép Teherbírását!

2500

8.1-3. ábra Targonca módosítása részlet (Saját készítésű ábra)

Az oldalon megtalálható még a módosítás gomb is, amely az egyes targoncákkal egy sorban helyezkedik el (8.1-1-es ábra). A 8.1-3-as ábra szerinti adatfrissítés esetén minden adatot át lehet állítani, ezt főleg a dátumok változtatására fogják leggyakrabban használni. Ezenkívül bármilyen olyan adatot, amely esetleg nem megfelelően lett betöltve, azt is meg lehet változtatni.

Az oldalon még megtalálható egy Eladás hivatkozás is, amely egy speciális módosítást végez. Ezt főleg a munka megkönnyítésére alkalmazzák, mivel van olyan eset, amikor több targoncát egyszerre visznek el. Ilyenkor csak azokat az adatoszlopokat módosítja, amelyek eladásnál relevánsak.

A helyszínek szerkesztésénél egy hasonló munka felgyorsítást kíséreltem meg megoldani. A helyszínek megnevezésénél egy hivatkozást hoz létre, amely arra a lokációs adatra mutat (8.1-1-es ábra).

8.1.2 Szerelőnaplok oldal

A szerelőjegyzék felületét főleg a szerelő munkatársak fogják használni. A fő szempont, amely ennél a felületnél fontos volt, az a jegyzékek felvitelének gyorsítása és a munka megfelelő megosztása.

Az oldalon megtalálható a Főmenü menüsor. Ez ugyanúgy működik, mint az előző felületen. Egy kereső beviteli mező található meg, itt csak a targoncák gyáriszáma alapján tudunk lekérdezni. Ide azért nem került választási opció, mivel az egyes jegyzékeket a gyáriszámuk alapján legcélszerűbb csoportosítani.

A funkció mellett még megtalálható ezen kívül egy új jegyzék hozzáadása gomb is, amellyel megadhatók egyesével a szerelőjegyzék adatai. A lehető legtöbb egyszerűsítést beletettem. Így például a munka megnevezésekor egy lenyíló menüsort, amellyel megadhatjuk a legnépszerűbb szervizelési munkákat, valamint egy egyéb opciót, amelynél beütheti a pontos megnevezést. Új jegyzéket általában a szerelő visz fel, ezért a munkás adatait nem kell minden egyes alkalommal beütni, hanem azt a program lekezeli és automatikusan felviszi.

Ezen funkciók alatt természetesen megtalálható táblázatos megjelenítés a legfontosabb adatokkal. A munkagéphez tartozó következő szervíz és nagy szervíz oszlopok, valamint a munka elvégzésének pontos dátuma. A Szerelő beosztású felhasználóknak azonban csak azok az adatok jelenítődnek meg, amelyeket ők maguk töltöttek fel. Ezzel is ellenőrizhetik az egyes targoncákon elvégzett munkájukat, valamint a következő alkalommal elvégzendő feladataik megjegyzéseit is nyomon követhetik. Az adminisztrátorok és a főnök felhasználó csoport számára az összes rekord látszik, illetve az adminisztrátorok számára egy szerkesztési funkció is elérhető egy esetlegesen kihagyott vagy egyéb problémás adattag esetén.

8.1.3 Alkalmazottak modul

Az alkalmazottak menüpontot kevesen érik el, ezt főként a cégvezető fogja használni új munkatársak felvételére. A feladat gyorsítása érdekében ezen az oldalon történik meg a felhasználók regisztrálása, illetve bármilyen felhasználó törlése.

Az oldal csak az alapvető funkciókkal van ellátva, mint például a beillesztés, a módosítás, illetve a Törlés. Azzal a különbséggel, hogy mivel a szerelőjegyzék miatt a Munkások adattáblából nem kívánunk konkrét rekordot törölni, ezért csak a munka idejének a kezdetét kívánjuk alapértékre állítani. Ezzel jelezzük, hogy a munkás valaha itt dolgozott, azonban megszűnt már a céggel kötött munkaviszonya. Ugyan az adattáblából egyértelműen nem törölünk, azonban az oldal felhasználókat kezelő rendszeréből ténylegesen kitöröljük, ezzel meggátolva, hogy az adott személy az oldalon be tudjon jelentkezni.



Új Munkaerő Felvétele!

Adja meg a Vezetéknévét!

Teszt

Adja meg a Keresztnevét!

János

Adja meg a Kollega típusát!

Szerelő

Hozzáad

8.1-4. ábra Új alkalmazott felvétele (Saját készítésű ábra)

Az oldal fő funkciója a cégvezető számára az új munkaerő felvétele, amelynek a hivatkozása az oldal tetején helyezkedik el. Beillesztéskor a cég számára releváns információkat kér el, ahogyan a 8.1-4-es ábrán is látszik. Ezzel az adatbázisban elérhető Munkások táblát tölti fel. A munka egyszerűsítése és a felhasználói réteg beazonosításáért a munkások típusát egy legördülő menüben választhatja ki a cégvezető. A kiválasztott szerepkörrel fog rendelkezni később az adott kolléga.

A bejelentkezési link:

<http://localhost:63397/User/Register/c47debe9-ce46-4913-8edb-6ad34a06e456>

[Vissza](#)

8.1-5. ábra A bejelentkezéshez alkalmazott link (Saját készítésű ábra)

Miután megadtuk a munkások adattáblának az adatait ez után már következik az adott munkásnak a regisztrációja, amelyhez legenerál egy linket. Ez szöveges formában jelenik, meg ahogy 8.1-5-ös ábrán látható. Ezt a linket küldi el a megadott munkásnak az e-mail címére, azonban ez a link nem örökös, ugyanis egy adott link csak a nap végéig érvényes. Tehát, ha nem sikerül regisztrálni még azon a napon, akkor egy új linket kell generálnunk az adott alkalmazottnak.



The screenshot shows a web browser window with the address bar displaying the URL: localhost:63397/User/Register/c47debe9-ce46-4913-8edb-6ad34a06e456. The page has a dark red background and is titled "Regisztrációs Felület" in white text. Below the title, there are three input fields: "Felhasználónév:" (Username), "Jelszó:" (Password), and "Jelszó újra:" (Password again). Each field has a white input box. Below these fields is a white button labeled "Regisztrálás" (Registration).

8.1-6. ábra Regisztrációs oldal (Saját készítésű ábra)

A felhasználón már csak az adott linkre kattintva be is írhatja választott felhasználónevét, illetve a jelszavát, amelynél fontos volt a megfelelő jelszóerősség, azaz elég hosszúnak kell lennie, illetve tartalmaznia kell számot, kis és nagy betűket, illetve egy speciális karaktert (8.1-6-os ábra). A jelszavát még egyszer megerősítésként elkérjük. A regisztrációs adatokat a program titkosítva tárolja Hash kód alapján az SQL serveren.

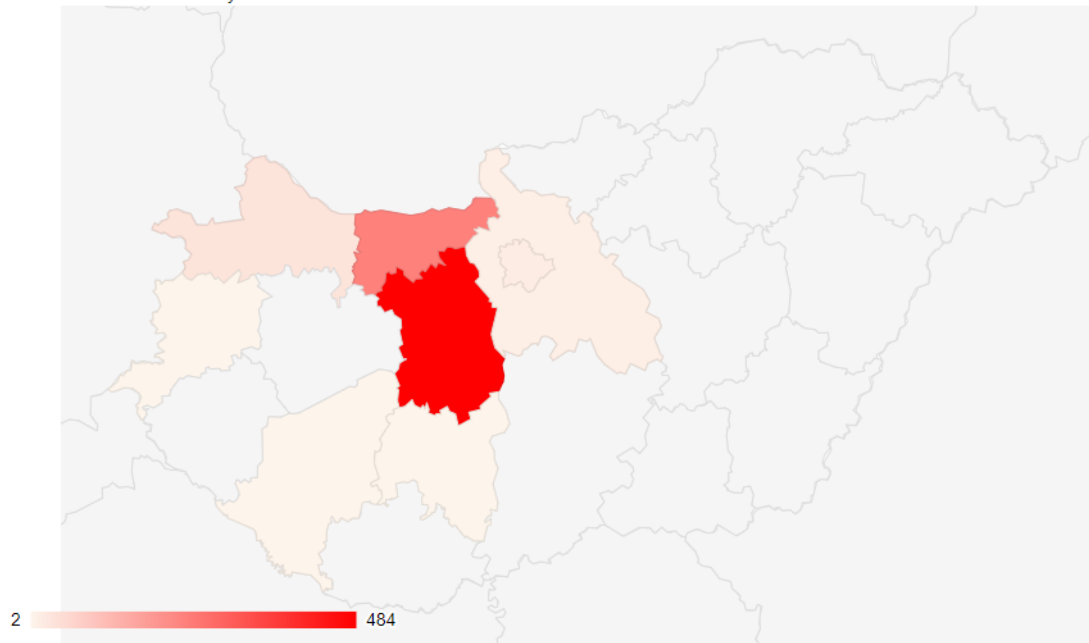
Az oldalon még megtalálható egy szerkesztés gomb is. A fő funkciója a hibajavítás, illetve egy esetleges promóció esetén a munkás adott beosztásának a megváltoztatása. Ilyenkor a felhasználó kezelő rendszer is frissíti az adott munkás szerepkörét.

8.2 Vizualizációs réteg

A vizualizációs réteg nem mindenkinél elérhető, mivel ez a cégvezető, illetve jelenleg és később majd a döntéseket elemző pénzügyi szektor számára fontos. A vezető számára azonban fontos, hogy mindegyik diagramot lássa. Ezért az oldalon a belépett felhasználótól függően kevesebb vagy több riport jelenik meg, két részre osztva pénzügyi és nem pénzügyi riportokra.

A vizualizációs oldal fő szempontja az adatok részletesebb analizálása volt. Itt csak a naponta változó adatokból vagyis a Data Mart adatbázisból kértem le az adatokat. A vizualizációk megjelenítésére GoogleCharts került felhasználásra. [19]

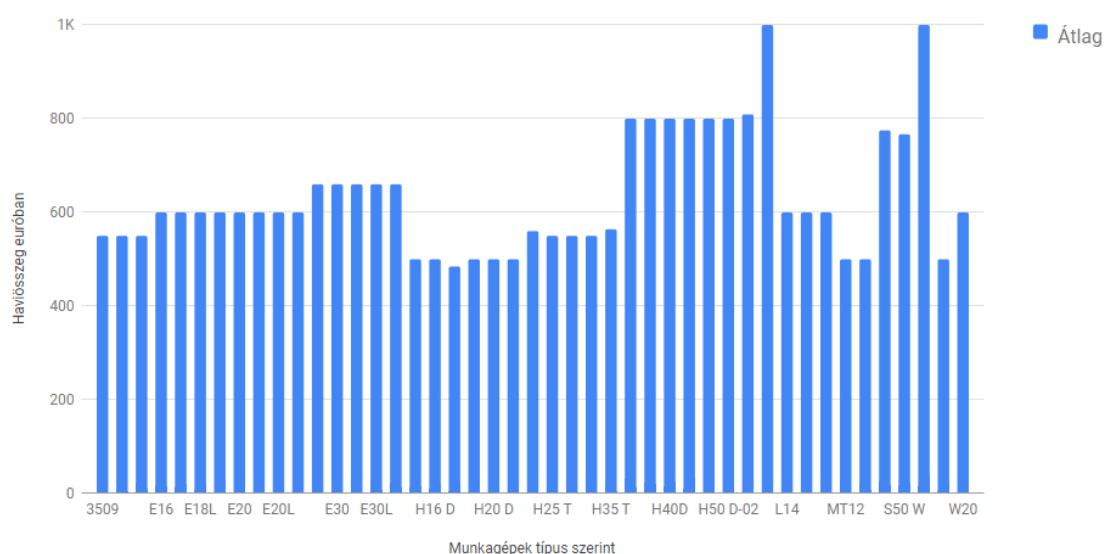
Alkalmazottak hol és hányszor szereltek



8.2-1. ábra Térkép a szerelések számáról megyékre bontva (Saját készítésű ábra)

A 8.2-1-es ábrán elemezni lehet azt, hogy a jelenleg kiadott gépeken a szerelő munkatársak milyen intenzitással végeznek egy-egy megyében munkát. Ez a későbbiekben, illetve most is nagy segítséget nyújt, mivel ezáltal láthatja a vezető, hogy összesen hányszor voltak adott időszak alatt az egyes megyékben. Így a későbbiekben akár a következő telephelyének a lokációját az adott megyére választja vagy akár bizonyos kollégát fog kihelyezni egy megyének a kiszolgálására. Ezzel is növelve a munka hatékonyságát.

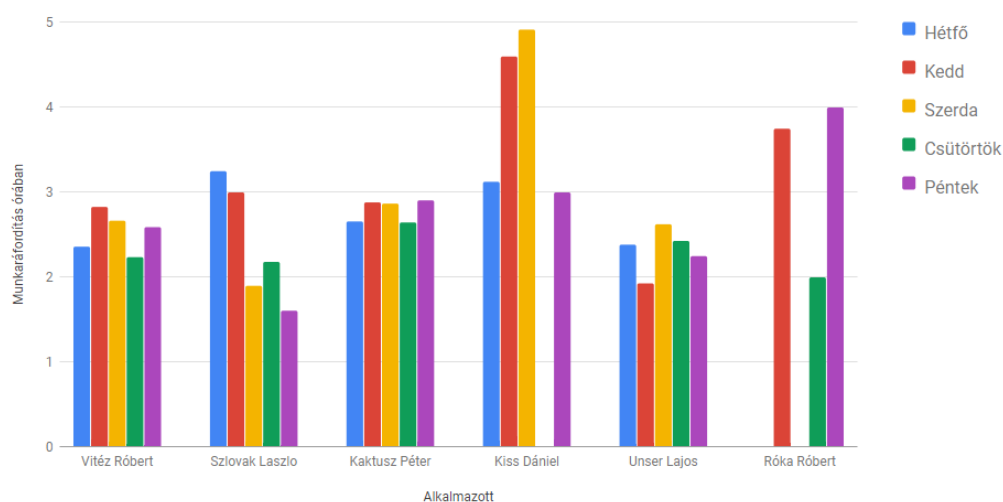
Targoncák típusaihoz tartozó átlagos haviösszeg



8.2-2. ábra A targoncák típusaira bontott haviösszegek (Saját készítésű ábra)

A pénzügyi részleg számára egy olyan diagramot hoztam létre, amelyen az éppen aktuális targoncák típusaira lebontva láthatjuk a havi bevételi összegek átlagát. A 8.2-2-es ábrán bemutatom, hogy jelenleg a társaság targoncatípusonként havonta milyen bevételi összegeket ér el. Az adatok pontosítására az adott oszlop fölé húzva az egeret látjuk, a konkrét bevételi értéket.

Alkalmazottakhoz tartozó átlagos munkaóra



8.2-3. ábra A cég szerelőinek az átlagos munkaráfordítása munkaórában a hét napjaira lebontva (Saját készítésű ábra)

A következő riporton a pénzügy, illetve a vezető számára is egy fontos szempontot vettem figyelembe, mégpedig azt, hogy a legtöbb embernek a munkamorálja, illetve a munkára fordított ideje hogyan változik.

Ahogy a 8.2-3-as ábrán látszik az egyes emberekhez köthető munkára fordított idő a legtöbb esetben csökkenő tendenciát mutat a hét vége felé haladva. Azaz igaz, hogy a munkahét vége felé már kevesebbet dolgoznak egy-egy szerelésnél. Ez azonban nem minden esetben igaz. Itt lehet elemezni a munka idejét, hogy átlagban mennyit dolgoztak aznap, illetve azt is, hogy ki mennyire érdemelhet meg egy esetleges promóciót vagy akár hosszabb szabadságot.

Továbbá elkészítettem egy, a bérlemények idejét vizualizáló diagramot, amelyen targoncának a gyáriszámára lebontva látható a jelenleg bérbeadott gépek kezdeti ideje és a jelenlegi idő közötti különbség, amelyen nyomon követhető a targonca mennyi ideje van már kiadva egy bizonyos cégnél.

Bemutatom még a targoncák átlagos életkorát elemző oszlop diagramot, amelyen - hasonlóan a havi összegeket felsorakoztató 8.2-2-es ábrához - a munkagépek típusokra felosztva tekinthetők meg, ezzel is segítve a jelenlegi állomány fiatalításának szükségességét.

9. TOVÁBBFEJLESZTÉSI LEHETŐSÉGEK

Továbbfejlesztési lehetőségként főképp a munka még gyorsabbá tételét szeretném megcélózni. Javaslatom a szerelési jegyzék feltöltése. Jelenleg egy lappal mennek ki a szerelők a munkagépeket üzemeltető társaság telephelyére általános szervízt végezni, vagy bejelentett hibát orvosolni. Az első dolguk az a targonca fontosabb adatainak kiolvasása vagy a hibakódok elérése.

Ilyenkor a webes felületen meg szeretném valósítani a lekért adatok átvitelét és más egyébb, a jövőben fontosabb részletek beolvasását. Így csak a munkafolyamat megnevezését és a munka időbeli ráfordítását kell csak manuálisan beütniük.

További fejlesztésnek véltem még az útmutatók készítését egy-egy általános hiba megoldására. Ezáltal a cég újabb, még kevésbé felkészült vagy még jelenleg nem túl sok tapasztalattal rendelkező munkatársainak egy kicsivel egyszerűbb feladata legyen.

A hibakód beolvasásakor az oldalon megjelenne a megfelelő hibakódhoz tartozó szerelési útmutató, azonban nem minden hibakódnak megfelelő útmutatás lenne készen, hiszen ezt majd az adminisztrátor tudná feltölteni. A gyakoribb meghibásodások megoldáshoz vezető út itt nemcsak szöveges, hanem kép vagy akár videó formájában elérhető lenne.

A későbbiekben megjegyzéseket is lehetne tenni az útmutató mellé, amellyel kijavítható a meghibásodott rész. A továbbiakban az újabb targonca alkatrészek megfelelő változásokat is praktikus és célszerű lenne beleilleszteni, hiszen az alkatrészek is változnak az évek során.

Fejlesztési lehetőség még, a szerelők számára a papírforma megírásának automatikus kitöltése, ezzel is csökkentve a telephelyen töltött időt.

Jelenleg minden egyes cég számára egy adott papírt ki kell tölteni minden javítás után a szakembereknek. Ezeken ugyanazok az adatok érhetők el, amely a jelenlegi adatbázisban is megtalálhatóak.

Jelenleg tollal kell kitölteni a szerelőknek egy javítási mellékletet, mely igazolja mindkét fél számára a munka elvégzését. Ezen fizikai jegyzék kevésbé kényelmes, ezért szeretnék egy elektronikus változatot is létrehozni, amelyen minden webalkalmazáson beütött adatra generálna a program egy .pdf formátumú dokumentumot. Ezt már csak digitálisan alá kell írniuk, aztán ezt e-mailen továbbítani kell annak a cégnek, ahol a munkát elvégezték.

10. ÖSSZEFOGLALÁS

Szakdolgozatomban egy cég számára kellett elkészítenem egy olyan adattárolási megoldást, amellyel a mostani tárolási módjukat reformáltam meg. Elsősorban a tárolási módszereket és tárolási rendszereket tekintettem át. Főleg az árazás, illetve a Microsoft termékeivel való integráció volt fontos, illetve az adatbázis kezelés.

Miután ezt kiválasztottam következő lépésként az ETL folyamatról írtam és forrás irodalmat kutattam. Erre azért került sor, mert nem minden esetben egyértelmű az ETL metodikát használni adatok áttöltésére és az adattárház feltöltésére. A folyamat megválasztása és indoklása után pedig kiválasztottam a konkrét ütemezést, valamint az adatok átalakításának megfelelő szoftvert.

A feladatomban megfogalmazott kérések alapján belekezdtem a Webes környezet megalkotásába. Megkerestem a megfelelő keretrendszert, illetve a megfelelő szoftvertervezési mintát. Előzőekben megemlítetteket a korábbi döntéseknek megfelelően és a későbbi továbbfejlesztésekre tekintettel választottam.

Az adattárházam megtervezésekor figyelembe vettem a kiválasztott ETL folyamatot kezelő programot és az adatok jelenlegi állapotát is, amellyel megterveztem végül az adattárházam betöltési lépéseit. A következő lépésben összehasonlítottam a vizualizációs sémákat a Csillag-sémát, a Snowflake-et illetve a Galaxy-t. Ezek közül a Csillag-sémát választottam végül. Ezt követően megadtam egy részletes Követelményspecifikációt.

Következő lépésként a Webes felületem kialakítását terveztem meg. Itt főleg az oldal elrendezését készítettem el, illetve a felhasználói szerepkörökről egy komplexebb magyarázatot adtam, amelyet el fogok készíteni.

Ezek után megkezdtem a konkrét feladat elkészítését, amelyben először az adatbázisomnak a megvalósítását készítettem el. Itt beszéltem a mostani állapotról és a megvalósított adatbázis rendszerről és a benne lévő adatokról.

Ezután az SSIS-en belüli ETL folyamat megvalósításáról írtam, amelyben minden lépését bemutattam. Végül a feladatok ütemezésének az implementációját is elmagyaráztam.

Végül a Webalkalmazás egyes részeit, illetve azoknak a belső funkcióinak a működését mutattam be, valamint a megalkotott vizualizációk részleteiről beszéltem.

A feladatom végén pedig beszélek az elkészített munkám továbbfejlesztési lehetőségeiről is, amelyben részleteztem az új vagy a már meglévő funkciók jövőbeni értékeit.

11. SUMMARY

In my thesis, I had to create a data storage solution for a company to reform their current way of storing data. I mainly looked at storage methods and storage systems. The main focus was on pricing and integration with Microsoft products and database management.

Once I had selected this I next wrote about the ETL process and researched source literature. This was done because it is not always straightforward to use the ETL methodology to migrate data and populate the data warehouse. And after choosing and justifying the process, I selected the specific process and the appropriate software to transform the data.

Based on the requirements of my task, I started to create the Web environment. I looked for the appropriate framework and the appropriate software design pattern. I have chosen the ones mentioned above according to the previous decisions and with a view for future developments.

When designing my data warehouse, I also took into account the selected ETL process manager and the current state of the data, with which I finally designed the loading steps of my data warehouse. In the next step, I compared the visualization schemas Star, Snowflake and Galaxy respectively. Of these, I finally chose the Star schema. I then gave a detailed Requirement Specification.

Next, I designed the design of my Web interface. Here I mainly created the layout of the site and gave a more complex explanation of the user roles, which I will provide.

After that, I started to prepare the concrete task, in which I first prepared the implementation of my database. Here I talked about the current state and the database system I implemented and the data it contains.

Then I wrote about the implementation of the ETL process within SSIS, in which I described all the steps. Finally, I explained the implementation of the task scheduling.

Finally, I presented the different parts of the Web application and their internal functions, and discussed the details of the visualizations created.

And at the end of my assignment, I also talk about the possibilities for further development of my work, detailing the future value of new or existing features.

IRODALOMJEGYZÉK

- [1] Microsoft-sql-server-mssql-vs-postgresql-comparison-details-what-differences <https://www.enterprisedb.com/blog/microsoft-sql-server-mssql-vs-postgresql-comparison-details-what-differences> utoljára megtekintve 2021.10.9
- [2] What are database management tools and what features are must-have? <https://blog.toadworld.com/what-are-database-management-tools-and-what-features-are-must-have> utoljára megtekintve 2021.10.10
- [3] What is Microsoft SQL Server Management Studio? <https://docs.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver15> utoljára megtekintve 2021.10.10
- [4] DBeaver vs SQL server management studio <https://www.stackshare.io/stackups/dbeaver-vs-microsoft-sql-server-management-studio> utoljára megtekintve 2021.10.10
- [5] Kinyerés, átalakítás és betöltés (ETL) <https://docs.microsoft.com/hu-hu/azure/architecture/data-guide/relational-data/etl> utoljára megtekintve 2021.10.10
- [6] ETL vs ELT <https://www.xplenty.com/blog/etl-vs-elt/#elt> utoljára megtekintve 2021.10.11
- [7] SQL Server Integration Services (SSIS) <https://docs.microsoft.com/en-us/sql/integration-services/sql-server-integration-services?view=sql-server-ver15> utoljára megtekintve 2021.10.11
- [8] SSIS and Pentaho <https://sqlkover.com/ssis-and-pentaho-a-quick-comparison/> utoljára megtekintve 2021.10.12
- [9] SQL Server Integration Services (SSIS) vs. Pentaho Data Integration (Kettle) vs. Stitch <https://www.stitchdata.com/vs/ssis/pentaho/> utoljára megtekintve 2021.10.12
- [10] Interaction within MVC pattern https://www.researchgate.net/figure/Interaction-within-MVC-pattern-The-Model-component-correlates-with-all-the-data-related_fig2_328716094 utoljára megtekintve 2021.10.20
- [11] MDN Web Docs Glossary: Definitions of Web-related terms MVC <https://developer.mozilla.org/en-US/docs/Glossary/MVC> utoljára megtekintve 2021.10.20
- [12] APS.NET Core <https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-6.0> utoljára megtekintve 2021.10.25
- [13] What is Laravel? https://dev.to/creativetim_official/what-is-laravel-explain-it-like-i-m-five-19eb utoljára megtekintve 2021.10.25
- [14] Spring MVC <https://docs.spring.io/spring-framework/docs/3.2.x/spring-framework-reference/html/mvc.html> utoljára megtekintve 2021.10.25

- [15] SQL-Server-Management-Studio <https://dbmstools.com/tools/sql-server-management-studio> utoljára megtekintve 2021.10.10
- [16] What is ETL? <https://databricks.com/glossary/extract-transform-load> utoljára megtekintve 2021.10.11
- [17] MVC Pattern https://www.researchgate.net/figure/Interaction-within-MVC-pattern-The-Model-component-correlates-with-all-the-data-related_fig2_328716094 utoljára megtekintve 2021.10.20
- [18] Adattárházak oracle <https://docplayer.hu/5435799-Adattarhazak-fekete-zoltan-bi-w-termekmenedzser-oracle-hungary.html> utoljára megtekintve 2021.10.29
- [19] Google Charts: <https://developers.google.com/chart/interactive/docs> utoljára megtekintve 2022.04.28

ÁBRAJEGYZÉK

2.2-1. ábra Adatbázis menedzser [13]	4
2.3-1. ábra ETL folyamat [14]	5
2.4-1. ábra MVC működése [15]	8
4.1-1. ábra A Szerepkörök (Saját készítésű ábra)	14
5.1-1. ábra Az adatbázis terv (saját készítésű ábra)	19
7.1-1. ábra Extract csomag tartalma (Saját készítésű ábra)	24
7.1-2. ábra Az Extract metódus (Saját készítésű ábra)	24
7.2-1. ábra Transform példa (Saját készítésű ábra)	25
7.3-1. ábra A csillag séma terve (Saját készítésű ábra)	26
7.3-2. ábra Data Mart feltöltés folyamatára (Saját készítésű ábra)	27
7.3-3. ábra Dátum dimenzió feltöltése adatokkal (Saját készítésű ábra)	28
7.3-4. ábra Dimenzió táblák feltöltése részlet (Saját készítésű ábra)	29
7.3-5. ábra A ténytábla feltöltése (Saját készítésű ábra)	29
7.3-6. ábra Az ütemezés beállítása (Saját készítésű ábra)	30
8.1-1. ábra Targoncák oldal részlet (Saját készítésű ábra)	31
8.1-2. ábra Új helyszín beillesztése (Saját készítésű ábra)	32
8.1-3. ábra Targonca módosítása részlet (Saját készítésű ábra)	33
8.1-4. ábra Új alkalmazott felvétele (Saját készítésű ábra)	35
8.1-5. ábra A bejelentkezéshez alkalmazott link (Saját készítésű ábra)	36
8.1-6. ábra Regisztrációs oldal (Saját készítésű ábra)	36
8.2-1. ábra Térkép a szerelések számáról megyékre bontva (Saját készítésű ábra)	37
8.2-2. ábra A targoncák típusaira bontott haviösszegek (Saját készítésű ábra)	38
8.2-3. ábra A cég szerelőinek az átlagos munkaráfordítása munkaórában a hét napjaira lebontva (Saját készítésű ábra)	38