UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN
INFORMACIJSKE TEHNOLOGIJE

Zaključna naloga

Urejanje in prikaz podatkov v interaktivni obliki

(Manipulating and displaying data in an interactive way)

Ime in priimek: Patrik Pepelko
Študijski program: Računalništvo in informatika
Mentor: doc. dr. Uroš Godnov

Koper, marec 2019

Ključna dokumentacijska informacija

Ime in PRIIMEK: Patrik PEPELKO

Naslov zaključne naloge: Urejanje in prikaz podatkov v interaktivni obliki

Kraj: Koper

Leto: 2019

Število listov: 39 Število slik: 30 Število referenc: 11

Mentor: doc. dr. Uroš Godnov

Ključne besede: podatki, obdelava, SQL strežniki, Power BI, interakcija, poročila, Excel, Tabele

Izvleček:
V zaključni nalogi bo opisana obdelava večje količine podatkov in njihova predstavitev v obliki interaktivnega poročila. Opisana bo uporaba integracijskih storitev SQL strežnika – SSIS, njenih orodij za združevanje podatkov iz različnih virov, kot tudi orodij za preurejanje podatkov s pomočjo kode SQL. Po preureditvi podatkov po želenih kriterijih je opisan tudi Power BI in njegova uporaba. Opisane bodo njegove lastnosti, zmožnosti, vizualizacije in njihova konfiguracija, kot tudi uvoz slik v tabelo in njena povezava na podatkovno tabelo. Na koncu je opisano tudi, kako se poročilo shrani in deli, da lahko uporabniki poročilo vidijo in uporabijo.

Key words documentation

Name and SURNAME: Patrik PEPELKO

Title of final project paper: Manipulating and displaying data in an interactive way

Place: Koper

Year: 2019

Number of pages: 39 Number of figures: 30 Number of references: 11

Mentor: Assist. Prof. Uroš Godnov, PhD

Keywords: Data, Manipulation, SQL Server, Power BI, Interaction, Report, Excel, Table

Abstract: This paper will describe the manipulation of larger amounts of data and their presentation in an interactive way. The paper will describe the usage of SSIS – SQL Server Integration Services and its tools to combine data from multiple sources. The tools to manipulate data with SQL code in SSIS will also be described, after the data is transformed according to the desired criteria, Power BI will be described and its usage. Its capabilities, properties, visualizations and their configurations, as well as importing images as a table and linking that table to the data table will all be described. Finally it will show how to publish the report and share it with users, who can view and use the report.

Zahvala

Iskreno se zahvaljujem mentorju docentu dr. Urošu Godnovu, ki mi je omogočil delo na tem projektu in me povezal s podjetjem Brandstock. Projekt je nastal zato, da zaposlenim na podjetju omogoči hitrejši in lažji pregled nad podatki. Zahvala gre tudi podjetju Brandstock za ponujeno priložnost in zaupanje, kot tudi za ponujeno nadaljno sodelovanje po diplomi. Posebno zahvalo si zaslužijo tudi družina in prijatelji, tako za pregled slovnične in strukturne oblike diplomske naloge, kot tudi za podporo, ki sem jo dobival skozi študijska leta.

Kazalo vsebine

Kazalo slik

Seznam kratic

npr.	na primer
oz.	oziroma
t.i.	tako imenovano
itd.	in tako dalje
tj.	to je
SQL	Structured Query Language [4]
SSIS	Integracijske Storitve SQL Strežnika - SQL Server Integration Services
SSMS	SQL Server Management Studio
CTE	Common Table Expression

1 Uvod

Po zadnjem opravljenem izpitu sem misli usmeril k diplomski nalogi. Od nekdaj sem bolj praktične kot akademske narave, zato je bila moja želja izdelati nekaj realnega in s tem pridobiti znanje, ki mi bo koristilo tudi v prihodnosti. V tretjem letniku sem obiskoval predmet, pri katerem sem podrobneje spoznal, kako manipulirati oz. urejati podatke s kodo. Programski jezik SQL omogoča veliko manipulacije z malo kode, je pa potrebno biti pri pisanju pazljiv in natančen. Potrebno je tudi natančno poznati problem in ga pravilo izraziti s kodo [4]. Poučeval nas je docent dr. Uroš Godnov. Kljub temu, da je bilo študijsko leto 2017/2018 zanj prvo na naši fakulteti, je predavanja izpeljal odlično. Ta predmet je bil obvezni izbirni predmet, ki sem ga moral izbrati, kljub temu, da ga takrat nisem želel. Vesel sem, da sem ga, saj mi je predavatelj vzbudil zanimanje za podatkovno smer računalništva, ki ga do takrat nisem imel.
Zaradi novega zanimanja do podatkovne smeri računalništva sem želel ustvariti konkreten projekt iz tega področja. Ker so mi bila predavanja in vaje doc. dr. Godnova zanimiva, sem se odločil, da ga kontaktiram in povprašam za nasvet. Zelo hitro sem dobil pozitiven odgovor, še več, v razmaku nekaj dni je že imel v mislih projekt, ki bi ga ustvaril pod njegovim mentorstvom za podjetje Brandstock.
Podpisati sem moral pogodbo o nerazkritju informacij (NDA - *a nondisclosure agreement)*. Zaradi NDA so poročila in ostali prikazani podatki spremenjeni. Splošnih podatkov, ki se pojavijo v poročilu, nisem spreminjal (npr. države), občutljivejše podatke pa sem moral (imena blagovnih znamk, cene, stanja). Zaradi NDA prav tako nisem opisal, kaj pomenijo oznake razredov, ampak so te navedene le kot število.
Po podpisu je sledil prvi sestanek z doc. dr. Godnovom. Na tem sestanku mi je predstavil cilj projekta. Cilj je bil, da iz več podatkovnih tabel podatke združim, uredim po kriterijih in pretvorim v interaktivno poročilo. V poročilu bi moralo biti vidno stanje licence prodaje blagovne znamke (*Trademarka*) po državah. Cilj je, da ima uporabnik možnost izbire med *blagovnimi znamkami* in *državami*. Stanja so tri: aktivna licenca, v procesu pridobivanja licence in neaktivna licenca. Če je licenca neaktiva, se mora v poročilu izpisati tudi cena aktivacije v tisti državi.
Ker česa takega nisem še nikoli počel, mi je mentor svetoval, naj si pogledam integracijske storitve SQL strežnika (SSIS - SQL Server Integration Services), Power BI in Tableau. Zadnja dva sta oba programski orodji za pripravo interaktivnih poročil.

Kljub temu, da sem poročila začel delati v obeh programskih orodjih, je v končnem produktu uporabljen samo Power BI, ker Tableau ne omogoča praktične uporabe slik. To pa sem nujno potreboval, saj sem v veliko primerih moral uvoziti logotipe blagovnih znamk.

Vsa uporabljena orodja so Microsoftovega porekla. Klient za uporabo SSIS se namesti znotraj *Microsoft Visual Studia*, SQL kode sem pisal in preverjal v SSMS - *SQL Server Management Studiu*. Ta v končni verziji nima vloge, ker je bil uporabljen samo za pisanje kode, ne pa tudi za izedbo. Kljub temu se mi ga je zdelo primerno omeniti, saj mi je zelo olajšal delo in pregled nad kodo, kot pa če bi pisal v navadnem urejevalniku besedil (*Notepadu*). SQL koda se je izvedla v SSIS. Zadnje orodje pa je že omenjeni *Microsoft Power BI* za ustvarjanje interaktivnih poročil.

Po prvem sestanku sem začel s spoznavanjem omenjenih delovnih orodij. Slaba dva tedna po sestanku sem dobil podatke. V tem času sem že dovolj spoznal SSIS, da sem lahko začel z delom. Osnovne operacije za delo s SSIS sem spoznal preko uradnih spletnih delavnic SSIS [5].

Delo je potekalo večinoma samostojno, redno sem se o delu posvetoval z mentorjem po elektronski pošti. Nekajkrat sva se dobila tudi na sestankih, kjer sva opravila pregled nad do takrat narejenim delom. Večkrat sem moral popravljati stvari ”za nazaj”, saj je bil to zame prvi resen projekt nad podatki in sem zato velikokrat napačno razumel probleme.

1.1 Podatki

Podatki so mi bili posredovani s strani mentorja. Vsi podatki so bili shranjeni v *Microsoft Excel* datotekah. Dobil sem skupno 13 Excelovih datotek, od tega osem podatkovnih, ki sem jih moral v SSIS združiti v eno. Več o tem v naslednjem poglavju.
Ostale Excel datoteke so bile pomožne. Najpomembnejša je bila Excel datoteka "Trademarks", ki vsebuje kriterije. V njej so napisane vse blagovne znake (*Brand* in *Trademark*) in tako imenovani razredi prodaje (*Class*), ki morajo biti v poročilu. Za vsako blagovno znamko je napisana tudi skupina držav (*Countries of interest*), za katere mora imeti blagovna znamka v poročilu prikazana stanja po razredih.
V Excelu "Countries" so navedene vse tri skupine omenjene v Excelu "Trademarks". Vsaka skupina ima naštete države, ki spadajo vanjo. Skupine so: *prve in druge države (Primary & Secondary Countries), države latinske amerike (Latam)* in *Evropska unija (EU).*
Naslednja Excel datoteka je "Cost estimation", ki vsebuje cene. Cene so pomembe v primeru, ko blagovna znamka v neki državi nima aktivne licence, bi jo pa želela imeti. Za vsako državo so naštete tri cene, odvisno od tega, koliko razredov nekege blagovne znamke ima neaktivno stanje. Maksimalno število razredov katere koli blagovne znamke v Excelu "Trademarks" je tri, zato so naštete tri cene za vsako državo.
Naknadno sem dobil še dve Excel datoteki. Novi Excel s cenami (*Cost*), ki je po strukturi enak staremu, le cene so spremenjene. Druga dodatna Excel datoteka je "Regions", ki ima navedene regije ter naštete ustrezne države za vseh pet regij. Regija služi kot dodatni filter pri izbiri podatkov v poročilu.

1.2 Namestitev

Pred samim začetkom sem moral namestiti vsa potrebna orodja. Prvi je na vrsto prišel *Microsoft Visual Studio - Community* različica [7], nato podatkovna orodja SQL strežnika za *Microsoft Visual Studio* - dodatek k programu [8]. Sledila je lokalna namestitev SQL strežnika. Izbral sem razvijalsko (*Developer*) različico [9]. Četrti korak je bil namestitev *SQL Server Management Studio*, ki sem ga uporabil kot programsko okolje za pisanje in testiranje SQL kode [10]. Na koncu sem poleg *Excela* moral namestiti še *Microsoft Power BI* [11].

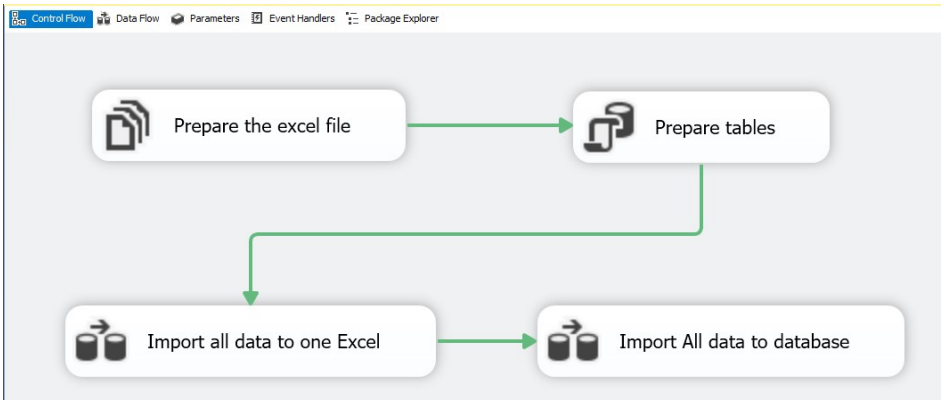
2 Integracijske storitve SQL strežnika

2.1 Predpriprava

V predpripravo spadajo naloge, ki sem jih moral opraviti pred začetkom oz. pred zagonom delovnega orodja *Microsoft Visual Studio*. Pripraviti sem moral podatke. To pomeni, da sem moral podatke, ki sem jih kasneje uvozil in so shranjeni v osmih Excelovih datotekah, shraniti v skupno mapo. To sem storil zato, da si olajšam delo in vzdrževanje, če pride v prihodnosti do sprememb v podatkih ali do nepravilnosti, npr. različno število stolpcev, različna imena stolpcev, dodatne podatkovne datoteke itd. Podatke sem moral pregledati zato, da sem ugotovil, ali so vsi v ustrezni obliki, kot tudi to, katere od navedenih podatkov potrebujem. Ugotovil sem, da ne potrebujem vseh stolpcev, pač pa le določene. Podrobneje je ta del opisan v nadaljevanju.
Pripravil sem tudi novo Excelovo datoteko, ki služi kot vir za podatkovno bazo. V to Excelovo datoteko sem združil vse podatke iz omenjenih osmih Excelovih datotek.
V *Microsoft Visual Studiu* sem zagnal integracijke storitve SQL strežnika (SSIS).
SSIS ima dva dela uporabe. Prvi, imenovan kontrolni tok (*Control Flow*), prikazuje proces spreminjanja oz. urejanja podatkov po korakih. Drugi del, imenovan podatkovni tok (*Data Flow*), pa prikazuje, kako se podatki spreminjajo znotraj vsakega koraka procesa. Ta del je bil v končni rešitvi zelo malo uporabljen, saj sem spretnejši z urejanjem podatkov s pomočjo programskega jezika SQL. Tega se lahko uporablja v kontrolnem toku, medtem ko podatkovni tok uporablja grafični vmesnik, s katerim pa nisem tako spreten. Grafični vmesnik potrebuje tudi veliko več vmesnih tabel, kar ni optimalo za vzdrževanje. V predpripravo spada tudi priprava podatkovne baze, v katero sem kasneje uvozil podatke.
V SSIS se naloge lahko razdeli na dele - pakete. Pripravil sem tri pakete: 1. Priprava, 2. Program in 3. Zagon. Pri tretjem paketu s pomočjo orodja za zagon paketov (*Execute package*) samo zaženem prva dva, ki sta opisana v nadaljevanju, zato tretji paket v nadaljevanju ni posebej opisan.

2.2 Paket: Priprava

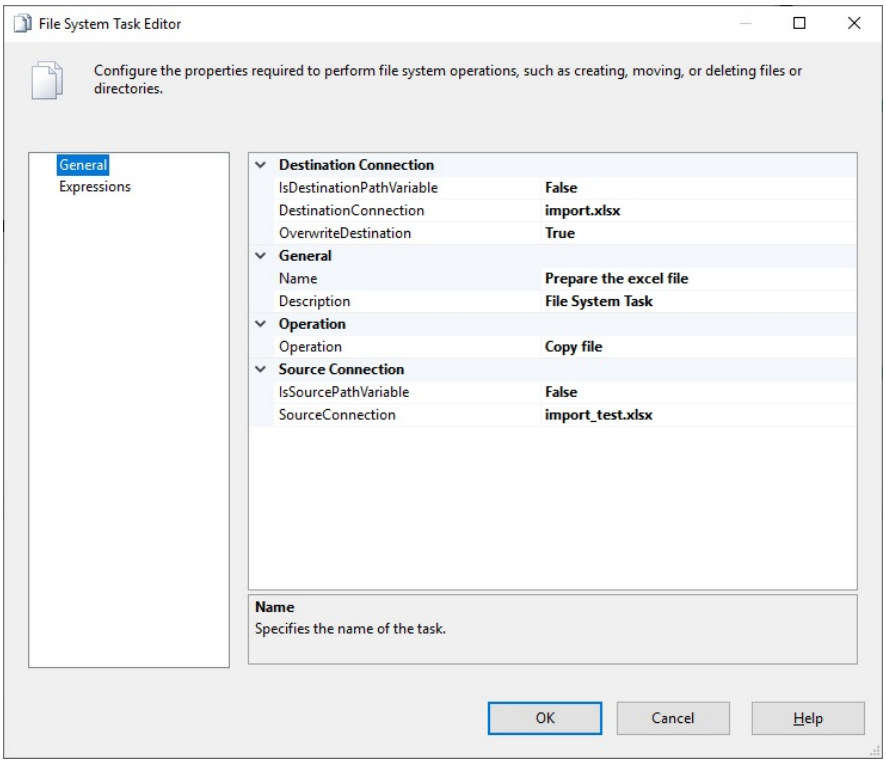
V paketu *Priprava* pripravim podatke. Delo poteka večinoma v kontrolnem toku programa, nekaj pa tudi v podatkovnem toku (podpoglavje 2.2.3). Vsak del paketa je podrobno opisan v nadaljevanju, izgled celotnega paketa pa je viden na **sliki 1**.



Slika 1: Izgled paketa *Priprava*

2.2.1 Priprava Excelove datoteke

Za pripravo Excelove datoteke sem uporabil orodje imenovano "orodje datotečnega sistema" (*File System Task*), ki omogoča osnovne operacije nad datoteko ali mapo. Te operacije so kopiraj, premakni, izbriši, preimenuj in ustvari. Uporabil sem le kopiranje s prepisovanjem, saj sem že v predpripravi pripravil Excelovo datoteko, v kateri bodo shranjeni združeni podatki. Zato, da se podatki ob vsakem zagonu programa ne dodajajo, ampak ponovno v celoti naložijo, sem pripravil dve datoteki. Prva, v katero se združujejo podatki in druga, ki je po stolpcih identična prvi, le da nima nobenega vnosa. Drugo s tem orodjem ob vsakem zagonu programa kopiram pod imenom prve in jo prepišem. Na ta način sem dosegel, da se podatki ne dodajajo, ampak se stari podatki izbrišejo in se naložijo novi. Na **Sliki 2** je vidna uporaba tega orodja. Datoteka *import.xlsx* je Excel datoteka s podatki, datoteka *import_ test.xlsx* pa je omenjena prazna Excelova datoteka, ki prepiše podatkovno.



Slika 2: *File System Task* za pripravo Excelove datoteke.

2.2.2 Priprava tabel

Za pripravo tabel sem uporabil orodje za izvedbo SQL kode (*Execute SQL task*). Tudi to je orodje kontrolnega toka. Kot že samo ime pove, to orodje izvede SQL kodo. Za pisanje in testiranje vseh SQL kod sem uporabil delovno orodje *Microsoft SQL Server Management Studio*, ki pa v končni rešitvi nima nobene vloge.

Koda za pripravo tabel je enostavna. Najprej sem za vsako tabelo preveril, če že obstaja in če ja, jo izbrišem. Nato sem jo z enakimi lastnostmi ponovno ustvaril. To sem storil z ukazema DROP TABLE IF EXISTS "ime tabele" in CREATE TABLE "ime tabele" (stolpec1 tip stolpca1, stolpec2 tip stolpca2,...). **Slika 3**

Enako bi dosegel tudi, če bi uporabil ukaz TRUNCATE TABLE, ki izbriše vse vnose v tabeli, ne pa tabele same.

Izbral sem izbris in ponovno ustvarjanje, da imam zbrane vse tabele na enem mestu, v primeru, da bom moral v prihodnosti tabele spreminjati.

Zato, da je orodje za izvedbo SQL kode vedelo, katero kodo mora izvesti, sem moral ustvariti dve povezavi. Prva je povezava do podatkovne baze in druga je povezava do kode SQL (datoteka s končnico ".sql").

Prva povezava je ob vsaki uporabi orodja vedno enaka, druga pa se spreminja glede na to, kateri del programa za izvedbo je na vrsti (katera koda se mora izvesti). Vse

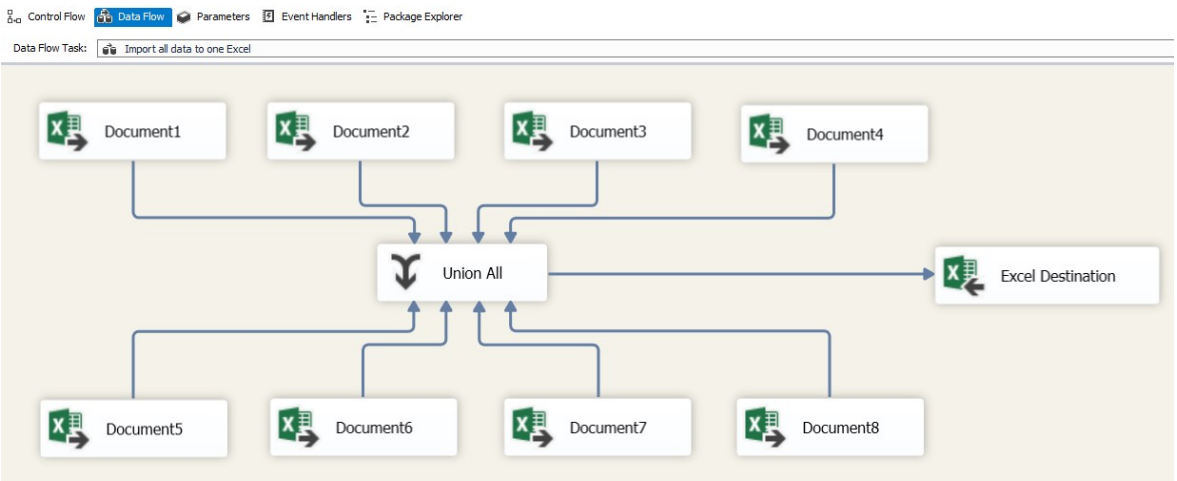
povezave sem ustvaril v tako imenovanem "upravitelju povezav" (*Connection manager*).

```
1  DROP TABLE IF EXISTS [dbo].[Excel_Destination];
2  GO
3  CREATE TABLE [dbo].[Excel_Destination](
4      CC NVARCHAR(255),
5      Country NVARCHAR(255),
6      "BRAND family" NVARCHAR(255),
7      Trademark NVARCHAR(255),
8      Status NVARCHAR(255),
9      "1st Class" NVARCHAR(255),
10     "2nd Class" NVARCHAR(255),
11     "3rd Class" NVARCHAR(255),
12     "4th Class" NVARCHAR(255),
13     "5th Class" NVARCHAR(255),
14     "6th Class" NVARCHAR(255),
15     "7th Class" NVARCHAR(255),
16     "8th Class" NVARCHAR(255),
17     "9th Class" NVARCHAR(255),
18     Source NVARCHAR(255)
19 )
20 GO
```

Slika 3: Vzorec za pripravo tabel

2.2.3 Združevanje podatkov

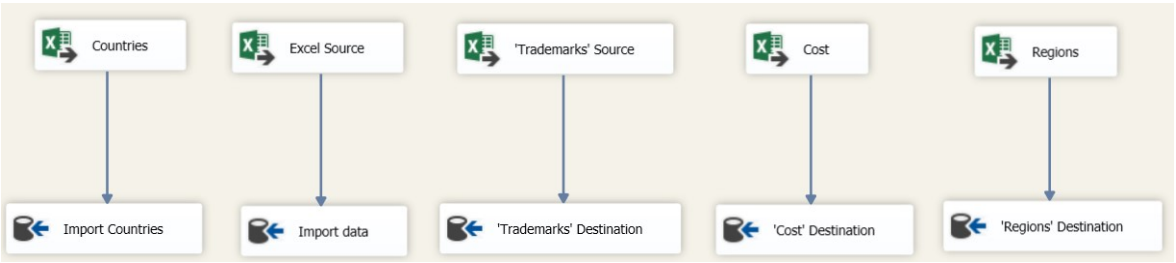
Naslednje je na vrsto prišlo združevanje podatkov. V "upravitelju povezav" sem moral narediti povezave do vseh osmih Excelovih datotek. Nato sem uporabil orodje kontrolnega toka "orodje podatkovnega toka" (*Data Flow Task*), ki omogoči nadaljno izvedbo v drugem delu programa, v podatkovnem toku. V drugem delu programa SSIS sem združil vseh osem Excelovih dokumentov. Za vsakega sem moral dodati orodje "Excel vir" (*Excel Source*), ki omogoča uvoz Excelove datoteke v SSIS. Znotraj tega orodja sem moral določiti, po kateri povezavi dostopa do dokumenta in dodatno še določiti katere stolpce naj uvozi, saj vseh nisem želel. Uvozil sem sledečih 15 stolpcev: *CC, Country, Brand family, Trademark, Status, 1st Class, 2nd Class, 3rd Class, 4th Class, 5th Class, 6th Class, 7th Class, 8th Class, 9th Class in Source*. Nato sem vse Excel vire združil s pomočjo orodja *Union All* ter jih uvozil v prej pripravljen Excel s pomočjo orodja "Excel destinacija" (*Excel Destination*). **Slika 4**



Slika 4: Združenje podatkov

2.2.4 Uvoz podatkov

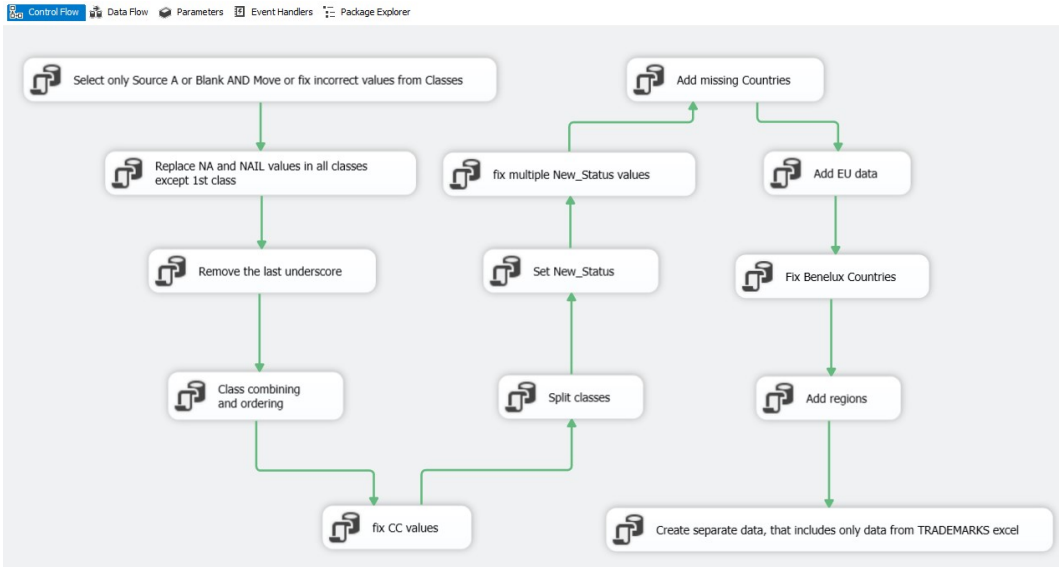
Vse Excel datoteke, ki sem jih v tem koraku uvozil, sem dobil od podjetja Brandstock. Razlika je le ta, da sem tistih osem datotek pred uvozom združil v eno. To sem storil zato, da imam vse podatke na enem mestu (viru) brez neželenih stolpcev. V tej stopnji je bilo vse pripravljeno na uvoz podatkov. Uvozil sem združene podatke, podatke o državah, regijah, cenah in kriterije. To sem storil v podatkovnem toku s pomočjo parov orodij "Excel vir" - "destinacija podatkovna baza" (*Excel Source - OLE DB Destination*). **Slika 5**



Slika 5: Uvoz podatkov

2.3 Paket: Program

V času študija sem podrobneje spoznal programski jezik SQL. Zato so vsa orodja v tem paketu orodja za izvedbo SQL kode. Določene dele bi lahko izvedel tudi s pomočjo grafične rešitve podatkovnega toka, vendar sem se odločil za SQL kodo, saj mi v primeru sprememb omogoče lažje nadaljno vzdrževanje.
Paket izvedbe sem razdelil na 13 delov, v vsakem delu pa je le delček celotne kode. Ta način mi ponovno omogoča lažje vzdrževanje in preverjanje vmesnih rezultatov. Izgled celotnega paketa je prikazan na **sliki 6**.
Kriteriji, po katerih sem moral urediti podatke, so mi bili ponovno posredovani s strani podjetja Brandstock. Zanimali so me samo podatki, ki imajo v stolpcu *Source* vrednost A ali pa vrednosti nimajo. Ostali kriteriji so opisani v uvoženi Excelovi datoteki imenovani "Trademarks". V njej so navedena *Podjetja (Brand), Blagovne znamke (Trademark), Razredi (Class)* in *Skupina držav (Countries of interest)*, ki morajo biti v končnem poročilu.



Slika 6: Izgled paketa *Program*

2.3.1 *Source* A ali prazno in napačne vrednosti

Prvi pogoj je bil, da vzamem samo tiste podatke, ki imajo v stolpcu *Source* vrednost A ali pa vrednosti nimajo. Ker tisti podatki, ki imajo vrednost B niso napačni, ampak me enostavno niso zanimali, sem jih iz podatkovne baze izbrisal. Izbrisal sem jih z ukazom DELETE "Ime tabele" WHERE *Source* NOT LIKE A OR *Source* IS NOT NULL. Nato sem pregledal vseh devet razrednih (*Class*) stolpcev in ugotovil, da je

veliko nepravilnih vrednosti. Nepravilna vrednost je vsaka vrednost, ki ni število, saj so razredni stolpci prikazani s številom. Vsako število pove, za katero kategorijo/smer prodaje gre. Podobno, kot če bi pri predmetih na fakulteti z "1" označili predavanja, z "2" pa vaje. Katere so te kategorije ni pomembno, pomembno je le to, da so v teh stolpcih samo števila. Ker pa so bile vrednosti vnešene na različne načine, ti stolpci niso nastavljeni kot število (int) ampak kot besedilo (nvarchar).
Npr. v prvem primeru so podatki vnešeni kot zaporedje števil v *1st Class* stolpec, števila pa so ločena s podpičjem ";". V drugem primeru pa so razredi vnešeni pravilno, v vsakem razrednem stolpcu največ eno število, po vrsti od prvega do devetega razreda. To sem odpravil v enem od naslednjih korakov, omenjeno je bilo le kot pojasnilo, zakaj ti stolpci niso shranjeni kot število (int).
Vse napačne vrednosti so bile premaknjene v tabelo za napačne vrednosti. Napačna vrednost je tudi prazno polje, ampak le v primeru, ko so vsi razredni stolpci prazni. Če vrednost v prvem stolpcu obstaja in ta ni napačna, potem prazna polja v ostalih razrednih stolpcih niso napaka. **Slika 7**

```
insert into dbo.wrongValues
select * from dbo.Excel_Destination
where [1st Class] like '%CO%' or [1st Class] like '%DO%' or [1st Class] like '%VAR%' OR [1st Class] like '%N%/%A%' OR [1st Class] like '0' or [1st Class] is null or [1st Class] like 'na'
go
delete dbo.Excel_Destination
where [1st Class] like '%CO%' or [1st Class] like '%DO%' or [1st Class] like '%VAR%' OR [1st Class] like '%N%/%A%' OR [1st Class] like '0' or [1st Class] is null or [1st Class] like 'na'
GO

insert into dbo.wrongValues
select * from dbo.Excel_Destination
where [1st Class] like '%(%' OR [2nd Class] like '%(%' OR [3rd Class] like '%(%' OR [4th Class] like '%(%' OR
[5th Class] like '%(%'OR [6th Class] like '%(%'OR [7th Class] like '%(%' OR [8th Class] like '%(%' OR [9th Class] like '%(%'
GO

delete dbo.Excel_Destination
where [1st Class] like '%(%' OR [2nd Class] like '%(%' OR [3rd Class] like '%(%' OR [4th Class] like '%(%' OR
[5th Class] like '%(%'OR [6th Class] like '%(%'OR [7th Class] like '%(%' OR [8th Class] like '%(%' OR [9th Class] like '%(%'
GO

insert into dbo.wrongValues
select * from dbo.Excel_Destination
where [1st Class] like '%int%' OR [1st Class] like '%M%' OR [2nd Class] like '%int%' OR [2nd Class] like '%M%' OR [3rd Class] like '%int%' OR [3rd Class] like '%M%' OR
[4th Class] like '%int%' OR [4th Class] like '%M%' OR [5th Class] like '%int%' OR [5th Class] like '%M%' OR [6th Class] like '%int%' OR [6th Class] like '%M%' OR
[7th Class] like '%int%' OR [7th Class] like '%M%' OR [8th Class] like '%int%' OR [8th Class] like '%M%' OR [9th Class] like '%int%' OR [9th Class] like '%M%'
GO
delete dbo.Excel_Destination
where [1st Class] like '%int%' OR [1st Class] like '%M%' OR [2nd Class] like '%int%' OR [2nd Class] like '%M%' OR [3rd Class] like '%int%' OR [3rd Class] like '%M%' OR
[4th Class] like '%int%' OR [4th Class] like '%M%' OR [5th Class] like '%int%' OR [5th Class] like '%M%' OR [6th Class] like '%int%' OR [6th Class] like '%M%' OR
[7th Class] like '%int%' OR [7th Class] like '%M%' OR [8th Class] like '%int%' OR [8th Class] like '%M%' OR [9th Class] like '%int%' OR [9th Class] like '%M%'
GO
```

Slika 7: Obdelava napačne vrednosti stolpcev *Class*

Nato sem preveril še vrednosti v stolpcu s podjetjem (*Brand family*). Ponovno sem vse napačne vrednosti prenesel v tabelo za napačne vrednosti. Pred tem sem vrednosti tega stolpca popravil, če je to bilo mogoče. To sem storil tako, da sem si pomagal s kriterijskimi podatki "Trademarks". Enačil sem stolpca *Trademark* iz obeh tabel in dopolnil oz. popravil vrednost *Brand family*, če sta se stolpca blagovnih znamk *Trademark* ujemala.
Dodatno sem dopolnil še manjkajoče vrednosti *Brand family* tako, da sem preveril stolpce *Trademark* (samo iz tabele podatkov). Zgodilo se je, da je včasih enak *Trade-*

mark imel vrednost *Brand family*, včasih pa je ni imel. V teh primerih sem vrednost stolpca *Brand family* dopolnil tam, kjer te ni bilo. **Slika 8**

```
update dbo.Excel_Destination
set [BRAND family] = b.[BRAND]
from dbo.Excel_Destination as a
inner join dbo.Trademarks as b on a.Trademark = b.Trademark
where b.[BRAND] IS NOT NULL AND a.[BRAND family] != b.BRAND
GO
update a
set a.[BRAND family] = b.[BRAND family]
from dbo.Excel_Destination as a
left join dbo.Excel_Destination as b on a.Trademark = b.Trademark
where a.[BRAND family] is null and b.[BRAND family] is not null
GO
```

Slika 8: Popravljanje in dopolnjevanje stolpca *Brand family*

Na enak način, kot sem na koncu dopolnjeval stolpec *Brand family*, sem dopolnjeval tudi stolpec *CC* - stolpec z oznako države (*Country*). V nekaj primerih se je zgodilo, da je ena država (*Country*) včasih imela oznako (*CC*), včasih pa ne. Ko oznake ni bilo, sem jo dopolnil. V primeru, da pravilne oznake nisem našel, sem pustil prazno. Če je samo oznaka prazna, ne pa tudi država, to ni napaka.

2.3.2 Oblikovanje podatkov

Kot že povedano, so bili podatki vnešeni na različne načine, zato sem moral podatke preoblikovati tako, da so bile vrednosti, ki naj bi bile enake, tudi v enaki obliki.
Prve razlike sem opazil že pri urejanju razrednih stolpcev. Pri vnosih, kjer so razredni stolpci (razen *1st Class*) prazni, torej nimajo vrednosti, je bilo to označeno na različne načine (z različnimi vrednostmi). Vse različne oznake sem zato spremenil tako, da tisti stolpec ni vseboval ničesar, torej sem določil NULL vrednost. **Slika 9**

```
update [dbo].[Excel_Destination]
set [2nd Class] = Null where [2nd Class] like 'N/A' OR [2nd Class] like 'Nail' OR [2nd Class] like 'na';
update [dbo].[Excel_Destination]
set [3rd Class] = Null where [3rd Class] like 'N/A' OR [3rd Class] like 'Nail' OR [3rd Class] like 'na';
update [dbo].[Excel_Destination]
set [4th Class] = Null where [4th Class] like 'N/A' OR [4th Class] like 'Nail' OR [4th Class] like 'na';
update [dbo].[Excel_Destination]
set [5th Class] = Null where [5th Class] like 'N/A' OR [5th Class] like 'Nail' OR [5th Class] like 'na';
update [dbo].[Excel_Destination]
set [6th Class] = Null where [6th Class] like 'N/A' OR [6th Class] like 'Nail' OR [6th Class] like 'na';
update [dbo].[Excel_Destination]
set [7th Class] = Null where [7th Class] like 'N/A' OR [7th Class] like 'Nail' OR [7th Class] like 'na';
update [dbo].[Excel_Destination]
set [8th Class] = Null where [8th Class] like 'N/A' OR [8th Class] like 'Nail' OR [8th Class] like 'na';
update [dbo].[Excel_Destination]
set [9th Class] = Null where [9th Class] like 'N/A' OR [9th Class] like 'Nail' OR [9th Class] like 'na';
GO
```

Slika 9: Oblikovanje praznih vrednosti stolpcev *Class*

Naslednji korak je bil oblikovati števila v razrednih stolpcih. Števila sem moral preoblikovati tako, da so ta bila v enaki obliki. Npr. vrednost ”9” je bila napisana kot ”09” in kot ”9”. Take primere sem pretvoril v drugo obliko, torej brez ničle. Dodatno sem spremenil tudi vsa ločila v podpičja ter odstranil presledke. Zato, da sem to lahko čim enostavneje dosegel, sem vseh devet razrednih stolpcev predhodno združil v enega in se nato lotil preoblikovanja. To sem dosegel s funkcijo CONCAT_WS (”ločilo”, *stolpec1, stolpec2*,...). Nato sem vrednosti preoblikoval s pomočjo funkcije REPLACE (celoten besedilni niz (*stolpec*), iskani besedilni niz, novi besedilni niz). **Slika 10**

```
insert into[dbo].[tmp]
select [CC],[Country],[Brand family],[Trademark],[Status],
CONCAT_WS(';',[1st Class],[2nd Class],[3rd Class],[4th Class],[5th Class],[6th Class],[7th Class],[8th Class],[9th Class]) as 'Classes',
[Source]
from dbo.Excel_Destination
GO
update dbo.tmp
set [Classes] = replace ([Classes],'01','1');
GO
update dbo.tmp
set [Classes] = replace ([Classes],'02','2');
GO
update dbo.tmp
set [Classes] = replace ([Classes],'03','3');
GO
update dbo.tmp
set [Classes] = replace ([Classes],'04','4');
GO
update dbo.tmp
set [Classes] = replace ([Classes],'05','5');
GO
update dbo.tmp
set [Classes] = replace ([Classes],'06','6');
GO
update dbo.tmp
set [Classes] = replace ([Classes],'07','7');
GO
update dbo.tmp
set [Classes] = replace ([Classes],'08','8');
GO
update dbo.tmp
set [Classes] = replace ([Classes],'09','9');
GO
update dbo.tmp
set [Classes] = replace ([Classes],'.',';');
GO
update dbo.tmp
set [Classes] = replace ([Classes],',',';');
GO
update dbo.tmp
set [Classes] = replace ([Classes],' ','');
GO
```

Slika 10: Združitev *Class* stolpcev in oblikovanje vrednosti

2.3.3 Razdelitev in ”barvanje”

Za oblikovanje sem razrede združil, v tem koraku pa sem jih ponovno razdelil. Tokrat sem jih razdelil po vrsticah in ne po stolpcih. S tem sem dosegel, da sem v vsaki vrstici dobil izključno en razred. To mi je omogočilo, da sem v nadaljevanju lažje primerjal vnose med seboj in tako popravljal oz. dodajal vnose. Razrede sem razdelil s kombinacijo dveh ukazov/funkcij. CROSS APPLY je ukaz, ki sproži izbrano funkcijo nad vsemi vnosi, izbrana funkcija pa je STRING_SPLIT (vrednost (*stolpec*), separator). STRING_SPLIT razdeli izbran besedilni niz (po vrsticah) glede na določeno ločilo. Izbrani besedilni niz je vrednost združenih razredov, separator pa je podpičje, zato sem v enem od prejšnjih korakov tudi nadomestil vsa ločila s podpičjem. **Slika 11**

```
insert into[dbo].[final]
select [CC],[Country],[Brand family],[Trademark],[Status],value as 'Class',[Source]
from dbo.tmp
cross apply string_split([Classes],';')
GO
```

Slika 11: Razdelitev *Class* po vrsticah

”Barvanje” pravim procesu določanja novega stanja vsakemu vnosu. Prvotnih stanj

je šest, vendar je nekaj stanj med sabo enakovrednih. Zato sem določil nova stanja, tista, ki so zahtevana v poročilu. Končna stanja so tri: "Protection granted", ki pove, da ima blagovna znamka za izbrani razred aktivno licenco prodaje v izbrani državi in je označeno z zeleno barvo. "Application pending" pove, da je proces za pridobitev licence v teku in je označeno z modro barvo ter "To be filed", ki pove, da licence ni in jo je treba pridobiti. Zadnje stanje je najpomembnejše, saj tukaj kasneje pridejo v poštev tudi cene. Označeno je z oranžno barvo.
Paziti sem moral tudi na podvojene vrednosti. V primeru, ko sem dobil enak vnos (enaki stolpci *Country, Brand family, Trademark* in *Class*), vendar z več različnimi stanji, sem moral izbrati pravilnega glede na prioriteto stanja. Zeleno stanje ima najvišjo, sledi modro, najnižjo prioriteto pa ima oranžno stanje. To sem rešil tako, da sem že pri določanju novih stanj določil tudi števila pred imenom stanja. Zeleno stanje sem poimenoval "1. Protection granted", modro "2. Application pending" in oranžno "3. To be filed". Nato sem z uporabo začasne tabele (*CTE - Common table expression*) in funkcije ROW_NUMBER pridobil vse primere, ko je stanj več. **Slika 12**

CC	Country	Brand family	Trademark	Status	Class	Source	New_Status	RowNumber
SI	Slovenia	Brand A	A_1	Registered	9	A	1. Protection granted	1
SI	Slovenia	Brand A	A_1	Filed	9	A	2. Application pending	2
SI	Slovenia	Brand A	A_1	Inactive	9	A	3. To be filed	3

Slika 12: Primer večih stanj za isti primer

Ker sem podatke urejedil po "barvanem" stanju (*New_Status* na **Sliki 12**), sem dobil zaradi števil pred stanjem na prvem mestu vedno tisti vnos, ki vsebuje stanje z najvišjo prioriteto. Zato sem izbrisal vse ostale vnose, kjer je *RowNumber* večji od števila ena.
Za lepši izgled v poročilu sem odstranil tudi števila pred imenom stanja. **Slika 13**

```
alter table dbo.final
add New_Status nvarchar(255)
GO
update dbo.final
set New_Status = case
when UPPER(Status) like 'REGISTERED' then '1. Protection granted'
when UPPER(Status) like 'FILED' then '2. Application pending'
else '3. To be filed'
end
GO
with CTE as (
select  CC, Country, [Brand family], Trademark, Status, Class, Source, New_Status,
row_number() over(partition by Country, Trademark, Class order by Country,Trademark, Class, New_Status) as RowNumber
from dbo.final
)
delete CTE where RowNumber > 1
GO

update dbo.final
set New_Status = case
when New_Status like '1. Protection granted' then 'Protection granted'
when New_Status like '2. Application pending' then 'Application pending'
else 'To be filed'
end
GO
```

Slika 13: "Barvanje" in odstranjanje duplikatov

2.3.4 Dodajanje manjkajočih držav

V kriterijskem Excelu "Trademarks" so navedene skupine držav (*Countries of interest*) za vsak *Brand, Trademark* in *Class* stolpec. Zato sem moral preveriti, če moji podatki vsebujejo za vsak *Brand, Trademark* in *Class* vsako državo, ki spada v želeno skupino držav. Skupine držav so tri: Prve in druge države (*Primary & Secondary countries*), države latinske amerike (*LATAM*) in države evropske unije (*European union*). Postopek za dodajanje manjkajočih držav iz vsake skupine držav je enak, zato bo opisan le za skupino prvih in drugih držav (*Primary & Secondary countries*).
V prvo začasno tabelo, imenovano *#PSC*, sem shranil vse države in njihove oznake iz prve skupine. V drugo začasno tabelo, imenovano *#TAC*, sem vnesel vse *Brand, Trademark* in *Class* stolpce za vsako državo iz prve skupine. Torej sem imel vse te stolpce ponovljene za vsako državo iz prve skupine. To sem dosegel s pomočjo CROSS JOINa. CROSS JOIN je kartezični produkt [1] vseh vrednost dveh tabel.
Kot prvo množico (tabelo) sem izbral stolpce *Brand family, Trademark* in *Class* iz tabele *Trademarks*, za drugo množico (tabelo) pa sem izbral le stolpec *primary & secondary countries* iz tabele *countries*. Na ta način sem dosegel, da sem vsakemu vnosu, kjer je skupina držav *primary & secondary countries* dodal le stolpec z državo.
Na **Sliki 14** je prikazan primer za skupino *Benelux*. V to skupino spadajo tri države:

[1] Kartezični produkt je množica vseh urejenih parov (a,b), kjer je a element prve množice, b pa element druge množice. [3]

Belgija, Luksemburg in *Nizozemska*. V prvi vrstici je prikazana vrednost, kot je napisana v kriterijski Excel datoteki "Trademarks", v ostalih pa kako izgledajo vrednosti v začasni tabeli *#TAC*.

	Brand	Trademark	Classes	Countries of interest
1	Brand C	C_1	9,25,35	benelux
2	/	/	/	/
3	Brand C	C_1	9	Belgium
4	Brand C	C_1	25	Belgium
5	Brand C	C_1	35	Belgium
6	Brand C	C_1	9	Luxemburg
7	Brand C	C_1	25	Luxemburg
8	Brand C	C_1	35	Luxemburg
9	Brand C	C_1	9	Netherlands
10	Brand C	C_1	25	Netherlands
11	Brand C	C_1	35	Netherlands

Slika 14: Primer začasne tabele *#TAC*

V tretjo začasno tabelo *#TAC2* sem združil prvi dve tako, da sem imel stolpce *CC, Country, Brand, Trademark, Class*. To sem dosegel tako, da sem z levim zunanjim stikom (*left outer join)* tabeli *#TAC* dodal vrednosti iz tabele *#PSC*. **Slika 15**

```
DROP TABLE if exists #PSC;
GO
Create table #PSC (
CC nvarchar(255),
Country nvarchar(255),
[Count] int
)
GO
insert into #PSC
select b.CC, a.PRIMARY_SECONDARY_COUNTRIES, COUNT(*) from dbo.Countries as a
join dbo.final as b on a.PRIMARY_SECONDARY_COUNTRIES = b.Country
where PRIMARY_SECONDARY_COUNTRIES is not null
group by b.CC, a.PRIMARY_SECONDARY_COUNTRIES
order by PRIMARY_SECONDARY_COUNTRIES
GO
DROP TABLE if exists #TAC;
GO
select a.BRAND, a.TRADEMARK, a.Class, b.PRIMARY_SECONDARY_COUNTRIES
into #TAC
from dbo.Trademarks2 as a
cross join (select PRIMARY_SECONDARY_COUNTRIES from dbo.Countries where PRIMARY_SECONDARY_COUNTRIES is not null) as b
where a.[Countries of interest for future filling] like 'primary & secondary countries'
GO
DROP TABLE if exists #TAC2;
GO
select b.CC, a.PRIMARY_SECONDARY_COUNTRIES, a.BRAND, a.TRADEMARK, a.Class
into #TAC2
from #TAC as a
left outer join #PSC as b on a.PRIMARY_SECONDARY_COUNTRIES = b.Country
GO
```

Slika 15: Koda začasnih tabel *#PSC, #TAC* in *#TAC2*

Zato, da v podatkih ne bi podvajal tistih vnosov, ki jih že imam, sem združil zadnjo začasno tabelo *#TAC2* s podatkovno tabelo z ukazom MERGE. MERGE omogoči enačenje dveh tabel na podlagi želenih stolpcev in nato dodatne operacije takrat, ko se vrednosti stolpcev ujemajo in/ali takrat, ko se vrednosti stolpcev ne ujemajo. V mojem primeru so me zanimali samo tisti primeri, pri katerih se vrednosti niso ujemale po *Country, Brand, Trademark* in *Class*. Takrat sem dodal manjkajoč vnos z oranžnim "To be filed" stanjem. **Slika 16**

```
Merge dbo.final as T
using #TAC2 as S
on (T.Country = S.PRIMARY_SECONDARY_COUNTRIES) AND (T.[Brand family] = S.Brand)
AND (T.Trademark = S.Trademark) AND (T.Class = S.Class)
when not matched by Target then
insert (CC, Country, [Brand family], Trademark, [Status], Class, [Source], New_Status)
values (S.CC, S.PRIMARY_SECONDARY_COUNTRIES, S.Brand, S.Trademark, 'Inactive',S.Class,'A','To be filed');
GO
```

Slika 16: Dodajanje manjkajočih držav v podatkovno tabelo

2.3.5 Popravljanje statusov

Med podatki sta se pojavili dve državi, ki to nista. Ti dve sta *European Union* in *Benelux*. Obe vsebujeta tudi svoje vnose in stanja. Veljata za univerzalni vrednosti, v primerih ko države, ki spadajo v njuno skupino, vrednosti nimajo. Ker pa sem v prejšnjem koraku dodal vrednosti glede na skupine držav (z oranžnim stanjem), sem moral določenim primerom to stanje popraviti.
Najprej sem popravil primere, ko je država članica evropske unije. Primerjati sem moral vse vnose, kjer je država članica Evropske unije z vsemi vnosi, kjer je država *European Union*. V primerih, ko se stanja razlikujejo, sem moral izbrati tisto, ki je imelo stanje z višjo prioriteto. Ponovno, zeleno "Protection granted" stanje je stanje z najvišjo prioriteto, sledi modro "Application pending", najnižjo prioriteto pa ima oranžno stanje "To be filed". To sem storil tako, da sem najprej v začasno tabelo *UrejenSource* vnesel vse podatke, "razdelil" podatke tam, kjer je država *European Union*, le da sem namesto države *European Union* vnesel vsako državo članico Evropske unije. To sem ponovno dosegel s CROSS JOIN, torej ustvaril sem kartezični produkt med podatki, kjer je država *European Union* in državami članicami Evropske unije.
Nato sem uporabil še UNION, ki združi te podatke z ostalimi, brez duplikatov. To pomeni, če nek primer v podatkih že obstaja (CROSS JOIN del), se ta ne bo ponovil, če se pojavi tudi v prvotnih podatkih (UNION del). **Slika 17**

```
--Prepare source table
SELECT distinct tt.CC,tt.Country,st.[Brand family],st.Trademark, st.Status, st.CLASS, st.Source, st.New_Status
INTO  dbo.UrejenSource
from dbo.final as st
cross join #EU as tt
where st.CC = 'EU'
union
SELECT st.CC,st.Country,st.[Brand family],st.Trademark, st.Status, st.CLASS, st.Source, st.New_Status
FROM dbo.final AS st;
GO
```

Slika 17: Vnašanje podatkov v začasno tabelo *UrejenSource*

Nadaljeval sem s ponovno uporabo funkcije za združevanje (MERGE). Tokrat sem koristil oba pogoja, takrat, ko se na podlagi stolpcev *Country, Trademark* in *Class* enači in takrat, ko se ne.
Ko se je enačilo, sem moral dodati še pogoj za pregled stanj (zaradi prioritet stanj). Stanje sem moral prepisati le, če je stanje v tabeli *UrejenSource* imelo višjo prioriteto kot stanje v podatkih.
V primerih, ko pa se ni enačilo po stolpcih *Country, Trademark* in *Class*, sem moral vnos dodati državi, saj ima tisti *Trademark* za tisti *Class* licenco Evropske unije. Stanje je tisto, ki je navedeno pri državi *European Union*. **Slika 18**
Celoten postopek sem ponovil še za države *Benelux*-a.

```
Merge dbo.final as T
using dbo.UrejenSource as S
on (T.Country = S.Country)and(T.Trademark = S.Trademark) and (T.Class = S.Class)
when matched AND T.New_Status not like S.New_Status AND
((T.New_Status not like 'Protection granted' AND S.New_Status like 'Protection granted') OR
(T.New_Status like 'To be filed' AND S.New_Status not like 'To be filed')) then
update set New_Status = S.New_Status
when not matched by Target then
insert (CC, Country, [Brand family], Trademark, [Status], Class, [Source], New_Status)
values (s.CC, s.Country, s.[Brand family], s.Trademark, s.[Status], s.Class, s.[Source], s.New_Status);
GO
```

Slika 18: Združevanje in pogoj prepisovanja stanj

2.3.6 Dodajanje regij

V naknadno dobljenem Excelu "Regions", so omenjene tudi regije, po katerih se mora razdeliti države. Regij je šest: EMEA - Evropa, Srednji vzhod in Afrika (*Europe, the Middle East and Africa*), APAC - azijsko-pacifiška regija (*Asia-Pacific*), LATAM - države latinske amerike (Latin America), EU - Evropska unija (European Union), NA - severna Amerika (*North America*) in CINA - Kitajska (*China*) . Če država ne spada v eno od teh skupin, se določi skupuna "Other". Pred tem sem moral tabeli s podatki

dodati stolpec *Region*. To sem storil z ukazom ALTER TABLE "ime tabele" ADD "ime stolpca" *tip stolpca*. Dodatno sem moral nekaterim državam ročno dodati regijo, saj so države shranjene na različne načine. Slovaška je na primer v podatkih shranjena kot "Slovakia (Slovak Republic)", v regijah pa kot "Slovakia". **Slika 19**

```
alter table dbo.final
    add Region nvarchar(255)
GO
update dbo.final
set Region =case
when Country in (select [EMEA] from dbo.Regions) then 'EMEA'
when Country in (select [APAC] from dbo.Regions) then 'APAC'
when Country in (select [LATAM] from dbo.Regions) then 'LATAM'
when Country in (select [EU] from dbo.Regions) then 'EU'
when Country in (select [NA] from dbo.Regions) then 'NA'
when Country in (select [CINA] from dbo.Regions) then 'CINA'
when Country like 'Korea, Republic of' then 'APAC'
when Country like 'United States' then 'NA'
when Country like 'Slovakia (Slovak Republic)' then 'EU'
when Country like 'European Union' then 'EU'
when Country like 'Benelux' then 'EU'
else 'Other'
end
GO
```

Slika 19: Koda dodajanja regij

2.3.7 Končna tabela

Zadnji korak je bil upoštevanje vseh kriterijev iz kriterijske Excel datoteke "Trademarks" in dodajanje cen iz Excela "Cost". S pomočjo začasnih tabel sem prvo izbral samo tiste podatke, katerih vrednosti stolpcev *Brand family, Trademark* in *Class* sovpadajo z istoimenskimi stolpci iz kriterijske tabele "Trademarks". To sem naredil v dveh korakih. V prvo, začasno tabelo *costData*, sem iz podatkovne tabele vnesel samo tiste podatke, katerih *trademark* se nahaja v kriterijskih podatkih "Trademarks".
Ker sem kot kriterij gledal samo *Trademark*, sem vnesel preveč podatkov, zato sem v naslednjem koraku izbrisal nezaželene vnose. To sem naredil tako, da sem k tabeli *costData* z LEFT OUTER JOIN dodal stolpce iz "Trademarks" tabele tako, da sem enačil vrednosti *Brand family, Trademark* in *Class*. Ker sem z LEFT OUTER JOIN dodal stolpce na desno stran tabele, so se pojavile prazne vrednosti (NULL) povsod, kjer se ne enači. Zato sem izbrisal tiste vrednosti iz tabele *costData*, kjer je bil *Class* stolpec brez vrednosti. Lahko bi določil katerega koli od treh stolpcev, po katerih sem

enačil, vendar ni bilo potrebe, ker če je bil prazen eden, so bili prazni vsi trije. **Slika 20**

```
insert into costData
select CC, Country, [Brand family], Trademark, Class, [New_Status], Region
from dbo.final
where Trademark in (select distinct Trademark from dbo.Trademarks2)
order by Country,[Brand family],Trademark
GO

delete a
from dbo.costData as a
left outer join dbo.Trademarks2 as b on (a.[Brand family] = b.Brand AND a.Trademark = b.TRADEMARK and a.Class = b.Class)
where b.Class is null
GO
```

Slika 20: Koda urejanja podatkov *costData* tabele

V tej stopnji sem imel podatke, ki ustrezajo trem od štirih kriterijev (*Brand family, Trademark* in *Class*). Moral sem urediti tudi skupino držav. To sem storil tako, da sem ponovno uporabil LEFT OUTER JOIN po *Trademark* in *Class*, dodatno je moral biti izpolnjen eden od treh pogojev:

1. Skupina držav je "primary & secondary countries" in izbrana država spada v to skupino držav. Ta podatek sem dobil iz stolpca *PRIMARY_SECONDARY_COUNTRIES* iz tabele "Countries" ALI
2. Skupina držav je "LATAM" in država spada v to skupino ALI
3. Skupina držav je "EU & UK" in država spada v to skupino. **Slika 21**

```
select a.CC, a.Country, a.[Brand family], a.Trademark, a.Class, a.Status, a.Region
into #tmpData
from dbo.costData as a
left outer join dbo.trademarks2 as b on (a.Trademark = b.TRADEMARK AND a.Class = b.Class)
where
(b.[Countries of interest for future filling] like 'primary & secondary countries' and a.Country in
(Select [PRIMARY_SECONDARY_COUNTRIES] from dbo.Countries))
OR
(b.[Countries of interest for future filling] like 'LATAM' and a.Country in
(Select LATAM from dbo.Countries))
 OR
(b.[Countries of interest for future filling] like 'EU & UK' and a.Country in
(Select [EUROPEAN UNION] from dbo.Countries))
GO
```

Slika 21: Koda preverjanja držav

Nato sem pripravil tabelo, imenovano *data_final*, ki kot samo ime pove, je zadnja tabela. Njena vsebina je tista, ki gre v interaktivno poročilo. Vanjo sem vnesel do sedaj pripravljene podatke, dodatno sem dodal še stolpec *Number of Classes*, ki za vsako državo pove, koliko *Class* vrednosti je za vsak *trademark*, pri katerih je stanje

oranžno "To be filed". Ta podatek mi je omogočil izbiro prave cene. V Excelu "Cost" so za vsako državo navedene tri cene. Izbere se tisto, ki ustreza stolpcu *Number of Classes*. Cene sem dodajal v dveh korakih. Prvi korak je bil dodajanje cen tistim državam, ki imajo vrednosti navedene v Excelu "Cost", drugi korak pa dodajanje državam, ki niso posebej navedene, ampak vem, da spadajo v Evropsko unijo. V prvem koraku sem uporabil LEFT JOIN, v drugem pa CROSS JOIN. Cene sem zaokrožil na celo število. **Slika 22**

```
update a
set [Cost] = case
when a.[Number Of Classes] = 1 AND a.[Status] like 'To be filed' then cast (ROUND(b.[1 CLASS in USD],0) as int)
when a.[Number Of Classes] = 2 AND a.[Status] like 'To be filed' then cast (ROUND(b.[2 CLASSES in USD],0) as int)
when a.[Number Of Classes] = 3 AND a.[Status] like 'To be filed' then cast(ROUND(b.[3 CLASSES in USD],0)as int)
end
from dbo.data_final as a
left join dbo.Cost as b on a.Country = b.Country
GO

update a
set a.Cost = case
when a.[Number Of Classes] = 1 then cast (ROUND(b.[1 CLASS in USD],0) as int)
when a.[Number Of Classes] = 2 then cast (ROUND(b.[2 CLASSES in USD],0)as int)
when a.[Number Of Classes] = 3 then cast (ROUND(b.[3 CLASSES in USD],0)as int)
end
from dbo.data_final as a
cross join dbo.Cost as b
where
b.Country like 'European Union' AND a.Country in (select [EUROPEAN UNION] from dbo.Countries where [EUROPEAN UNION] is not null)
AND a.Status like 'To be filed' AND a.Cost is null
GO
```

Slika 22: Koda dodajanja cen v dveh korakih

2.3.8 Dodatno

V tabeli *data_final* sem v tej stopnji imel vse potrebne podatke. Dodatno sem jih preurejal samo zato, da sem lahko naredil učinkovitejše interaktivno poročilo. Najprej sem spremenil imena vseh razredov iz "25" v "Class 25". Nato sem dodal vnose samo zato, da imam v enem stolpcu prikazane vse statuse ter cene. To sem moral storiti, ker v interaktivnem poročilu sem lahko kot vsebino (*Value*) vizualizacije *Matrika* nastavil največ en stolpec (več o vizualizacijah v nadaljevanju). Prav tako sem dopolnil podatke na tak način, da tam kjer cene ni, bi pa morala biti (manjkajoča država v Excelu "Cost"), izpiše vrednost "No data!". **Slika 23**

```
update a
set a.Class = case
when Class like '9' then REPLACE ([Class],'9','Class 09')
when Class like '25' then REPLACE ([Class],'25','Class 25')
when Class like '35' then REPLACE ([Class],'35','Class 35')
else Class
end
from dbo.data_final as a
GO
insert into dbo.data_final
select a.CC, a.Country, a.[Brand family], a.Trademark, 'Cost ($)' as 'Class',
a.Status, a.Region, a.[Number Of Classes], a.Cost
from dbo.data_final as a
where a.Status like 'To be filed'
GO
update dbo.data_final
set Status = Cost
where class like 'Cost ($)'
GO
update dbo.data_final
set Status = 'No data!' where Cost is null AND Class like 'Cost%' AND Status is null
GO
```

Slika 23: Koda preurejanja za učinkovitejše poročilo

Kot zadnjo stvar sem dodal stolpec *NewStatus*, v katerem sem določil pravilno vrednost cene tam, kjer mora biti, tam kjer cene ni, pa sem določil vrednost 0. Ta stolpec sem shranil kot število (int). Dodal sem ga po predstavitvi poročila na podjetju Brandstock, saj je bila želja, da se v poročilu lahko vidi skupne oz. delne cene po državah in po blagovnih znamkah.
Dodatno sem moral paziti, da delim ceno s številom pojavitev statusa "To be filed", saj bi drugače imel zabeleženo dvakrat oz. trikrat previsoko ceno. **Slika 24**

```
alter table dbo.data_final
add NewStatus int
go
update dbo.data_final
set NewStatus = case
when Cost is not null AND Status not like 'To be filed' and [Number Of Classes] = 1 then [Cost]
when Cost is not null AND Status not like 'To be filed' and [Number Of Classes] = 2 then ([Cost]/2)
when Cost is not null AND Status not like 'To be filed' and [Number Of Classes] = 3 then ([Cost]/3)
else 0
end
go
```

Slika 24: Koda dodatnega stolpca *NewStatus*

3 Power BI

Power BI je Microsoftovo orodje za izdelovanje interaktivnih poročil. Power BI Desktop omogoča tri poglede: *Okno odnosov*, ki omogoča grafični prikaz tabel in njihovih povezav, *okno podatkov*, ki prikazuje podatke izbrane tabele in *okno poročil*, kjer se izdeluje interaktivna poročila. Ker sem podake že uredil, sem veliko večino dela opravljal v zadnjem oknu.
Naredil sem dve poročili. V prvem sem prikazal vse napačne vrednosti. Za to poročilo sem potreboval le tisto tabelo, v katero sem v prejšnjem poglavju vnašal napačne vrednosti.
Drugo (pomembnejše) poročilo pa prikazuje stanje blagovnih znamk (*trademarkov*) za izbran *Brand family* glede na državo oz. države in primerne cene.
Za seznanjanje z delom v Power BI in njegovimi funkcijami sem pred začetkom pregledal spletni vodič za Power BI [6].

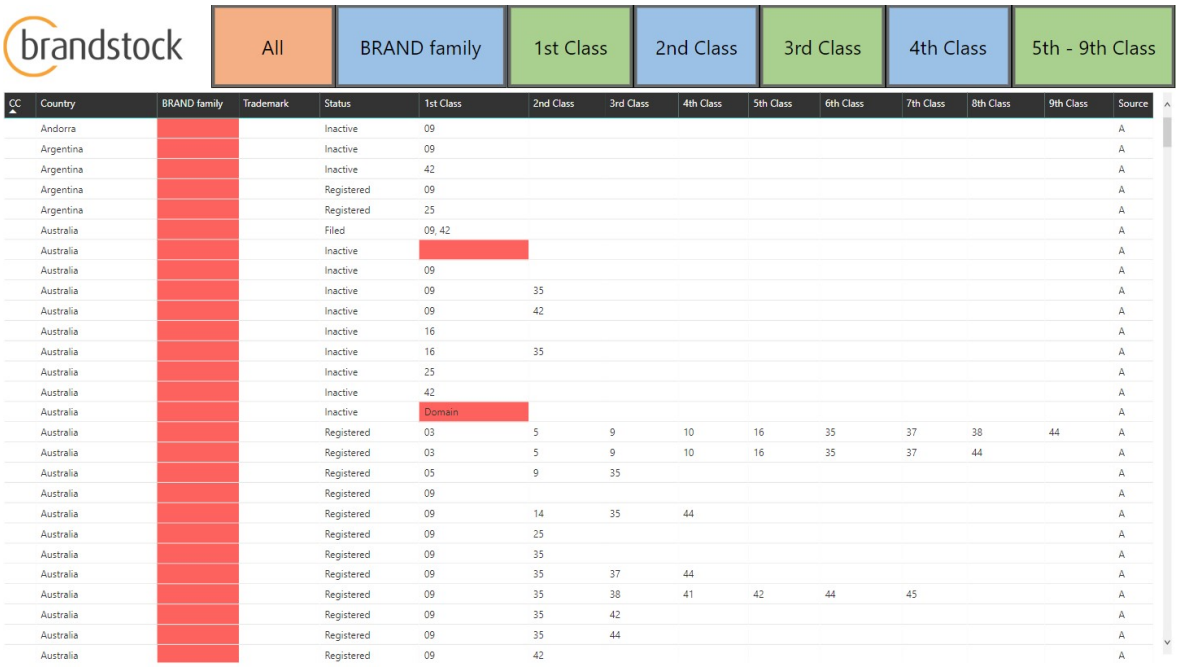
3.1 Poročilo napačnih vrednosti

Tabela z napačnimi vrednostmi je že bila v SSIS delu predelana v ustrezno obliko, tako da je v Power BI nisem rabil dodatno predelovati, le uvoziti sem jo moral. Podatke sem prikazal z vizualizacijo *Table* - Tabela. V njej sem prikazal vse stolpce.
Vse nepravilne vrednosti sem pobarval z rdečo barvo, zato da izstopajo, saj niso vsi stolpci v vrstici napačni. To sem naredil tako, da sem za vsak stolpec dodal novo "meritev" (*measure*). Ta meritev doda vsaki napačni vrednosti število "1", ostalim pa "2". To sem dosegel z DAX kodo. **Slika 25**
S funkcijo SELECTEDVALUE (*Tabela[stolpec]*) sem dobil vrednost stolpca in jo shranil v spremenljivko. Nato sem s funkcijo SWITCH preveril različne vrednosti. SWITCH funkcija s TRUE() pogojem je enakovredna CASE stavku SQL kode. Napačnih vrednosti ni bilo veliko, zato sem jih ročno izpisal (pravilnih vrednosti je bilo veliko več, zato sem raje izpisal nepravilne). Če je program ugotovil, da vrednost ustreza eni od napisanih napačnih vrednosti, je določil število "1", v nasprotnem primeru pa "2".

```
ErrorNumberClass2 =
VAR
    ErrorNumber1 = SELECTEDVALUE(diplomaWrong[2nd Class])
RETURN
SWITCH(TRUE()
    ,ErrorNumber1 = "N/A",1
    ,ErrorNumber1 = "21 (de)",1
    ,ErrorNumber1 = "42 (de)",1
    ,ErrorNumber1 = "44 (English specification is unavailable)",1
    ,ErrorNumber1 = "35 (English specification is unavailable)",1
    ,ErrorNumber1 = "50(10)",1
    ,ErrorNumber1 = "09 (de)",1
    ,ErrorNumber1 = "25 (de)",1
    ,ErrorNumber1 = "05 (English specification is unavailable)",1
    ,ErrorNumber1 = "25 (English specification is unavailable)",1
    ,ErrorNumber1 = "Nail",1
    ,2)
```

Slika 25: DAX koda določanja števila "1" napačnim vrednostim stolpca *2nd Class*

Nato sem vrednost stolpca obarval le v primeru, ko je imela meritev te vrednosti število "1". Na poročilo sem postavil tudi pet tipk, s katerimi izbiram med prikazi napačnih vrednosti. Prva tipka *All* prikaže vse napačne vrednosti, tipka *Brand family* prikaže napačne vrednosti stolpca *Brand family*, tipka *1st Class* napačne vrednosti stolpca *1st Class*,... V zadnjem primeru so napačne vrednosti popolnoma enake za stolpce *5th - 9th Class*, zato sem jih lahko nastavil na eno tipko, ne pa posebej, kot sem moral v ostalih primerih. Izgled poročila je viden na **Sliki 26**.



CC	Country	BRAND family	Trademark	Status	1st Class	2nd Class	3rd Class	4th Class	5th Class	6th Class	7th Class	8th Class	9th Class	Source
	Andorra			Inactive	09									A
	Argentina			Inactive	09									A
	Argentina			Inactive	42									A
	Argentina			Registered	09									A
	Argentina			Registered	25									A
	Australia			Filed	09, 42									A
	Australia			Inactive										A
	Australia			Inactive	09									A
	Australia			Inactive	09	35								A
	Australia			Inactive	09	42								A
	Australia			Inactive	16									A
	Australia			Inactive	16	35								A
	Australia			Inactive	25									A
	Australia			Inactive	42									A
	Australia			Inactive	Domain									A
	Australia			Registered	03	5	9	10	16	35	37	38	44	A
	Australia			Registered	03	5	9	10	16	35	37	44		A
	Australia			Registered	05	9	35							A
	Australia			Registered	09									A
	Australia			Registered	09	14	35	44						A
	Australia			Registered	09	25								A
	Australia			Registered	09	35								A
	Australia			Registered	09	35	37	44						A
	Australia			Registered	09	35	38	41	42	44	45			A
	Australia			Registered	09	35	42							A
	Australia			Registered	09	35	44							A
	Australia			Registered	09	42								A

Slika 26: Izgled poročila napačnih vrednosti

3.2 Poročilo podatkov

Pri tem poročilu je bilo več dela. Poleg tabele s podatki, sem moral uvoziti tudi slike (logotipe). Nato sem moral ti dve tabeli pravilno združiti in prikazati v pravilni obliki.

3.2.1 Izvoz logotipov iz Excela in njihov uvoz v Power BI

V kriterijskem Excelu "Trademarks" imam poleg stolpcev *Brand family, Trademark, Class* in *Countries of interest* shranjene tudi logotipe blagovnih znamk. Nima vsaka blagovna znamka shranjenega logotipa, vendar za tiste, ki ga imajo, sem moral logotip prenesti v poročilo.
V nastavitvah Excela sem moral aktivirati (prikazati) skrit zavihek "Razvijalec" (*Developer*), s katerim sem odprl dostop do manipulacije podatkov odprtega Excela s pomočjo kode. Programski jezik je *Visual Basic for Applications - VBA*. Slike so shranjene kot oblike (*shapes*), zato sem uporabil zanko *for each shape*. Znotraj zanke sem preveril, če ima stolpec *Trademark* vrednost v vrstici, v kateri najde obliko. Če ja, shrani vrednost *Trademark* stolpca kot spremenljivko in nato izvozi (*Export*) obliko kot sliko ("Pot do mape" & *Spremenljivka z imenom* & ".jpg"). **Slika 27**

```
Sub ExportPictures()
For Each oShape In ActiveSheet.Shapes
    On Error GoTo NextShp:
    If Not IsEmpty(ActiveSheet.Cells(oShape.TopLeftCell.Row, 2).Value) Then
        strImageName = ActiveSheet.Cells(oShape.TopLeftCell.Row, 2).Value
        oShape.Select
        Application.Selection.CopyPicture
        Set oDia = ActiveSheet.ChartObjects.Add(0, 0, oShape.Width, oShape.Height)
        Set oChartArea = oDia.Chart
        oDia.Activate
        With oChartArea
            .ChartArea.Select
            .Paste
            .Export ("C:\Users\Pepsi\Desktop\Logos\" & strImageName & ".jpg")

        End With
        oDia.Delete
    Else
    End If

NextShp:
    i = i + 1
    Resume NextShp2
NextShp2:
    Next oShape
End Sub
```

Slika 27: Koda izvoza slik iz Excel datoteke

Programiranje v VBA programskem jeziku mi je tuje, kodo sem našel na forumu

Stackoverflow in jo predelal po mojih potrebah. [1]

Power BI omogoča uvoz vsebine izbrane mape kot tabelo. To lastnost sem koristil pri uvozu logotipov. Ko sem uvozil mapo z logotipi, sem dobil dva pomembna stolpca: *Content* in *Name*. Stolpec *Name* sem kopiral in odstranil končnico ".jpg". Po tem stolpcu sem lahko kasneje enačil tabelo z logotipi in tabelo s podatki glede na stolpec *Trademark*. Stolpec *Content* je pomemben, saj vsebuje sliko (logotip), vendar v taki obliki, da je še nisem mogel prikazati. S pomočjo funkcije *BinaryToBase64*, narejene s strani *G. Brueckl* [2], sem dodal stolpec *Image*, v katerem se nahaja slika v primerni obliki za prikaz. **Slika 28**

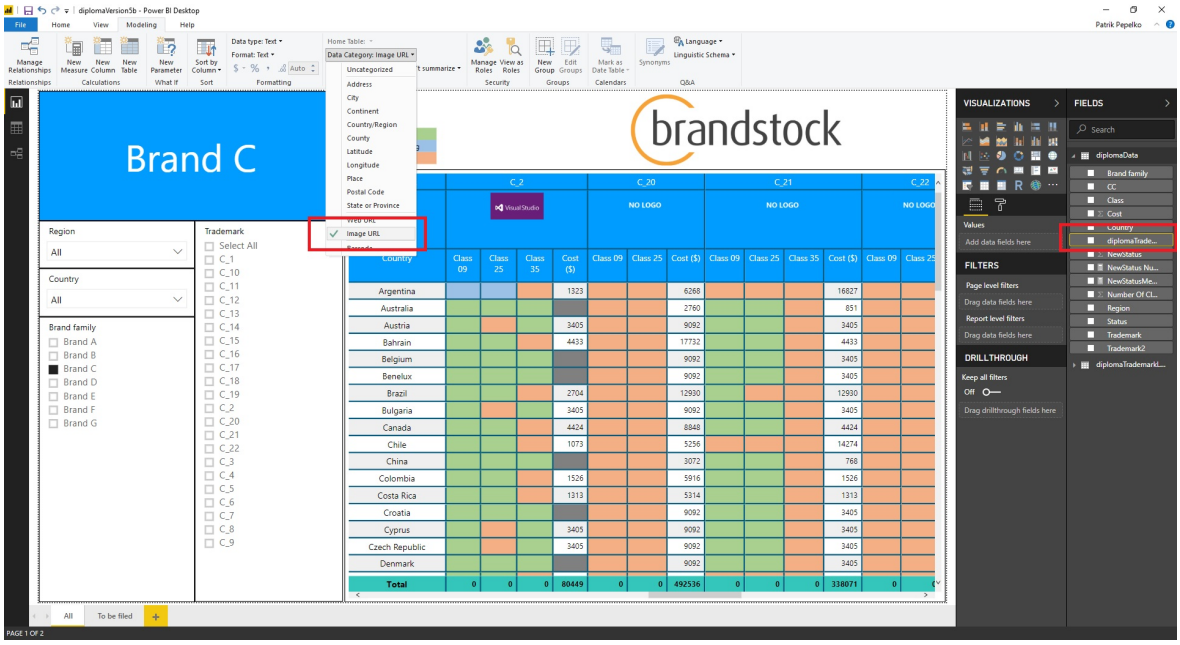
	Content	Name	Extension	Attributes	Folder Path	Name - Copy	Image
1	Binary	A_1.jpg	.jpg	Record	C:\Users\Pepsi-PC\Documents\Faks\...	A_1	/9j/4AAQSkZJRgABAQEAYABgAAD/4QBYRXh...
2	Binary	B_1.jpg	.jpg	Record	C:\Users\Pepsi-PC\Documents\Faks\...	B_1	/9j/4AAQSkZJRgABAQAAAQABAAD/2wCEAA...
3	Binary	C_1.jpg	.jpg	Record	C:\Users\Pepsi-PC\Documents\Faks\...	C_1	/9j/7QAsUGhvdG9zaG9wIDMuMAA4QklNA+...
4	Binary	C_2.jpg	.jpg	Record	C:\Users\Pepsi-PC\Documents\Faks\...	C_2	/9j/4AAQSkZJRgABAQEASABIAAD/4QBcRXhp...
5	Binary	C_3.jpg	.jpg	Record	C:\Users\Pepsi-PC\Documents\Faks\...	C_3	/9j/4AAQSkZJRgABAQEAYABgAAD/4QA6RXh...
6	Binary	C_4.jpg	.jpg	Record	C:\Users\Pepsi-PC\Documents\Faks\...	C_4	/9j/4AAQSkZJRgABAQEAYABgAAD/4QA6RXh...

Slika 28: Tabela z logotipi

3.2.2 Priprava podatkov v Power BI

Uvozu tabele z logotipi je sledil uvoz tabele s podatki. Po tem sem moral tabeli združiti. Združiti sem ju moral zato, da lahko prikažem tako podatke, ki jih izberem v poročilu, kot tudi logotipe. Če bi imel ločeni tabeli, bi izbral prikaz samo določenih *Trademarkov* vendar, ker so logotipi izbranih *Trademarkov* v drugi, ločeni tabeli, se le ti ne bi prikazali.
Združil sem ju z MERGE funkcijo Power BI. Tabeli s podatki sem z LEFT OUTER JOIN dodal stolpec iz tabele z logotipi, in sicer tisti, ki je vseboval sliko - *Image*. Tabeli sem enačil po stolpcih *Trademark* iz tabele s podatki in z imenskim stolpcem *Name - copy*, kateremu sem odstranil končnico *.jpg*.
Prva verzija poročila je bila taka, da je bil poseben (privzet) znak nastavljen tam, kjer logotipa ni bilo. Po predstavitvi podjetju je bila izražena želja, da bi namesto tega znaka pisalo "NO LOGO". Ker pa Power BI, v času pisanja diplome, tega še ne omogoča, sem moral to doseči na drugačen način. Pripravil sem sliko z napisom "NO LOGO" in jo shranil v isto mapo, kjer so shranjeni logotipi. Nato sem moral tabeli s podatki dodati nov stolpec, poimenoval sem ga *Trademark2*, kateremu sem dodal vrednost iz stolpca *Trademark*, vendar le tistim vrednostim, ki imajo logotip. Ostalim sem določil vrednost "NO LOGO". Novi stolpec sem uporabil samo takrat, ko sem združeval (MERGE) tabelo s podatki s tabelo z logotipi. Tako sem dosegel, da povsod

kjer logotipa ni na voljo, prikaže sliko z napisom "NO LOGO".
Nato sem v oknu poročila začel izdelovati interaktivno poročilo. Najprej sem stolpcu, ki vsebuje sliko, spremenil lastnost, da je to stolpec s sliko. **Slika 29**



Slika 29: Spreminjanje lastnosti stolpcu s sliko (logotipom)

Sledilo je ustvarjanje poročila, tj. postavljanje izbranih vizualizacij na želeno mesto in njihova konfiguracija. Uporabil sem štiri različne vizualizacije.

1. Vizitka (*Card*): Uporabil sem jo za prikaz izbrane vrednosti stolpca *Brand family* v zgornjem levem kotu.

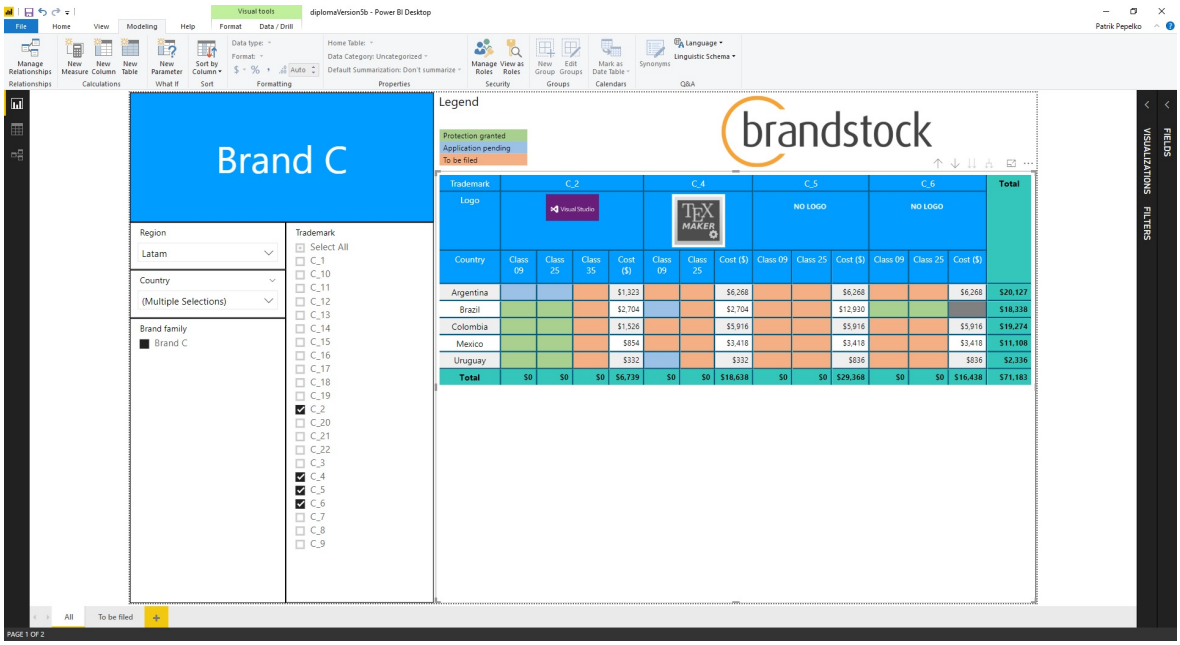
2. Rezalnik (*Slicer*): Uporabil sem ga kar štirikrat. Rezalnik ponudi uporabniku možnost izbire različnih vrednosti enega stolpca, ki jih želi uporabnik videti. Uporabil sem ga za izbor *Regije, Države, Brand family* in *Trademarka*. Vsem razen *Brand family* sem dopustil možnost izbire več vrednosti hkrati, za *Brand family* pa se tako nisem odločil, saj pride pri prikazu v naslednji vizualizaciji (matriki) do veliko praznih vrstic in posledično do grdega in nepreglednega izgleda poročila.

3. Matrika (*Matrix*): Glavna vizualizacija tega poročila. V njej sem prikazal vse pomembne podatke. Na levi strani so po vrsticah prikazane izbrane *države*, zgoraj pa vse izbrane blagovne znamke (*Trademark*) skupaj z ustreznimi logotipi in *Razredi (Class)*. V sredini, kot vrednosti, pa so prikazana stanja z barvami za vsak razred ter ceno. Zelena barva pomeni "Protection granted" stanje, modra pomeni "Application pending", oranžna "To be filed", rdeča "No data!" ter siva, ki pove, da tam cene ni in da ta ni potrebna (v primeru, ko so za *Trademark* ni stanje nobenega *Classa* oranžno).

Dodatno sem pri tej vizualizaciji uporabil tudi vgrajeno funkcijo SUBTOTAL, ki ponudi delne rezultate za neko vrednost. Določil sem, da prikaže skupno ceno po *državi* (na desni strani matrike) in skupno ceno po *Trademarku* (na dnu matrike).
4. Tabela (*Table*): Zadnjo vizualizacijo sem uporabil samo kot legendo stanj, na kateri so vidne barve ter njihov pomen.
5. Dodatno: Dodaten element, ki sem ga dodal v poročilo, je logotip podjetja Brandstock. Uvozil sem ga kot samostojno sliko, ki se nikoli ne spreminja, zato lahko ta slika, za razliko od logotipov, stoji kot samostojen element.
Po predstavitvi se je pojavila tudi želja, da ima poročilo možnost prikaza samo tistih primerov, kjer je stanje oranžno. Zato sem narejeno poročilo dupliciral na novo stran, jo preimenoval v "To be filed" in določil filter celotne strani, ki prisili vse vizualizacije, da pokažejo le primere s stanjem oranžnim "To be filed" stanjem.

Pri določanju zgornjih vrednosti matrike (*Trademark, logo, Class*) Power BI ne omogoča postavitve dveh vrednosti na isto raven. Ker sem želel imeti stolpca *Class* in *Cost* na isti ravni, sem moral preurediti vrednosti *Class* tako, da sem pri vseh *Trademakrih*, kjer se pojavi oranžno stanje, dodal vrednost "Cost ($)". Ker v vsebinskem delu matrike prikazujem stanje, sem moral v primerih, ko je *Class* vrednost "Cost ($)", posodobiti stolpec s stanjem na vrednost stolpca *Cost*. S tem sem dosegel, da sem v vsebinsko (*Value*) polje postavil vrednosti stolpca *Stanje (Status)*, pod vrednosti *Class* sem dobil izpisano stanje, pod *Cost ($)* pa ceno. **Slika 23**
Nato je prišla želja, da bi bilo v poročilu možno videti skupne cene po državi (*Country*) in po *Trademarku*. Spremeniti sem moral vsebinski del matrike, saj za izračun skupnih cen Power BI kot vrednosti potrebuje števila in ne besedila, kot je to bilo do sedaj.
Ponovno sem moral preurediti svojo vhodno tabelo. Ker ključnih podatkov ni dobro preveč spreminjati, sem dodal nov stolpec *NewStatus*, ki služi kot novo vsebinsko polje matrike. Vanj sem vstavil ceno le v tistih primerih, ko vrednost stolpca *Cost* ni prazna, vrednost stolpca *Status* ni "To be filed" IN stolpec *Number of Classes* vsebuje število ena, dve ali tri (1, 2 ali 3). Če vsi trije pogoji niso izpolnjeni, dobi *NewStatus* vrednost 0. Na ta način sem dosegel, da ta stolpec vsebuje samo števila. Če je vrednost stolpca *Number of Classes* dve ali tri (2 ali 3), moram ceno deliti z dva oz. tri, če ne dobim dvakrat oz. trikrat previsoko ceno. Prepričan sem, da tako pravilno določim cene, saj pri ostalih dveh *stanjih* je vrednost stolpca *Number of Classes* prazna. **Slika 24**
V poročilu sem stolpcu *NewStatus* spremenil lastnost, da število prikazuje valuto - ameriški dolar.

Izgled končnega poročila je viden na **Sliki 30**



Slika 30: Izgled končnega poročila

3.3 Objava poročil

Ko sta bili obe poročili pripravljeni za uporabo, sem ju moral objaviti na svoj Power BI portal. To sem storil z enostavnim pritiskom na tipko *Publish*. Ko sta bili poročili objavljeni, sem lahko do njiju dostopal kar preko spletnega brskalnika. Kot lastnik in avtor, sem lahko preko spletnega brskalnika tudi urejal. Te lastnosti sicer nisem koristil, saj je program Power BI Desktop, v primerjavi s spletno verzijo, bolj uporabniško prijazen oz. preglednejši in lažji za uporabo. Bistvo objave na svoj Power BI portal je, da lahko spletno različico poročila delim z uporabniki, ki za uporabo poročila ne potrebujejo programa Power BI Desktop. Za uporabo morajo ustvariti le Power BI račun.

4 Zaključek

K projektu sem sem pristopil z željo, da ustvarim nekaj resničnega in uporabnega. Docent dr. Godnov je mojo željo razumel in hitro prišel na idejo, da pod njegovim mentorstvom ustvarim interaktivno poročilo za podjetje Brandstock. Najprej sem moral pripraviti vse podatke in nato s pomočjo SSIS v SQLu pripraviti ustrezne pakete in jih med seboj povezati. Da sem to dosegel, sem moral izostriti svoje znanje o manipulaciji podatkov s SQL kodo, kot tudi spremeniti oz. nadgraditi zorni kot gledanja na problem. Bolj kot sem se bližal koncu, bolj je prišlo do izraza t.i. razmišljanje oz. iskanje rešitve "zunaj škatle", saj sem moral po tem, ko sem že imel ustrezne podatke za poročilo, te preurediti. Preurediti sem jih moral tako, da se njihova vrednost in veljavnost ni spremenila, spremeniti sem moral le obliko, saj sem le tako lahko ustvaril boljše oz. lepše in učinkovitejše poročilo, saj ima Power BI še veliko omejitev pri prikazovanju podatkov. Sledilo je delo v PowerBI, kjer sem moral poleg podatkov uvoziti tudi logotipe in pripraviti ustrezne vizualizacije. Zadnji korak je bila objava na portal, zato, da lahko uporabniki na podjetju Brandstock poročilo uporabijo.

5 Literatura

[1] J. Robert, *Export pictures from excel file into jpg using VBA*, https://stackoverflow.com/questions/18232987/export-pictures-from-excel-file-in (Datum ogleda: 03. 01. 2019.) *(Citirano na strani 26.)*

[2] G. Brueckl, *Export pictures from excel file into jpg using VBA*, https://blog.gbrueckl.at/base64/. (Datum ogleda: 22. 12. 2018.) *(Citirano na strani 26.)*

[3] E-um, *Kartezični produkt*, http://www.nauk.si/materials/932/out/#state=1. (Datum ogleda: 26. 01. 2019.) *(Citirano na strani 15.)*

[4] *SQL*, https://en.wikipedia.org/wiki/SQL. (Datum ogleda: 31. 1. 2019.) *(Citirano na straneh VII in 1.)*

[5] *Integration Services Tutorials*, Microsoft. https://docs.microsoft.com/en-us/sql/integration-services/integration-services-tutorials?view=sql-server-2017. (Datum ogleda: 28. 09. 2018.) *(Citirano na strani 2.)*

[6] *Getting started with Power BI Desktop*, Microsoft. https://docs.microsoft.com/en-us/power-bi/desktop-getting-started. (Datum ogleda: 20. 12. 2018.) *(Citirano na strani 23.)*

[7] *Visual Studio 2017 - Downloads*, Microsoft. https://visualstudio.microsoft.com/downloads/. (Datum ogleda: 29. 09. 2018.) *(Citirano na strani 3.)*

[8] *Download and install SQL Server Data Tools (SSDT) for Visual Studio*, Microsoft. https://docs.microsoft.com/en-us/sql/ssdt/download-sql-server-data-tools-ssdt?v (Datum ogleda: 29. 09. 2018.) *(Citirano na strani 3.)*

[9] *SQL Server - Downloads*, Microsoft. https://www.microsoft.com/en-us/sql-server/sql-se (Datum ogleda: 29. 09. 2018.) *(Citirano na strani 3.)*

[10] *Download SQL Server Management Studio (SSMS)*, Microsoft. `https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio` (Datum ogleda: 29. 09. 2018.) *(Citirano na strani 3.)*

[11] *Power BI*, Microsoft. `https://powerbi.microsoft.com/en-us/desktop/`. (Datum ogleda: 29. 09. 2018.) *(Citirano na strani 3.)*