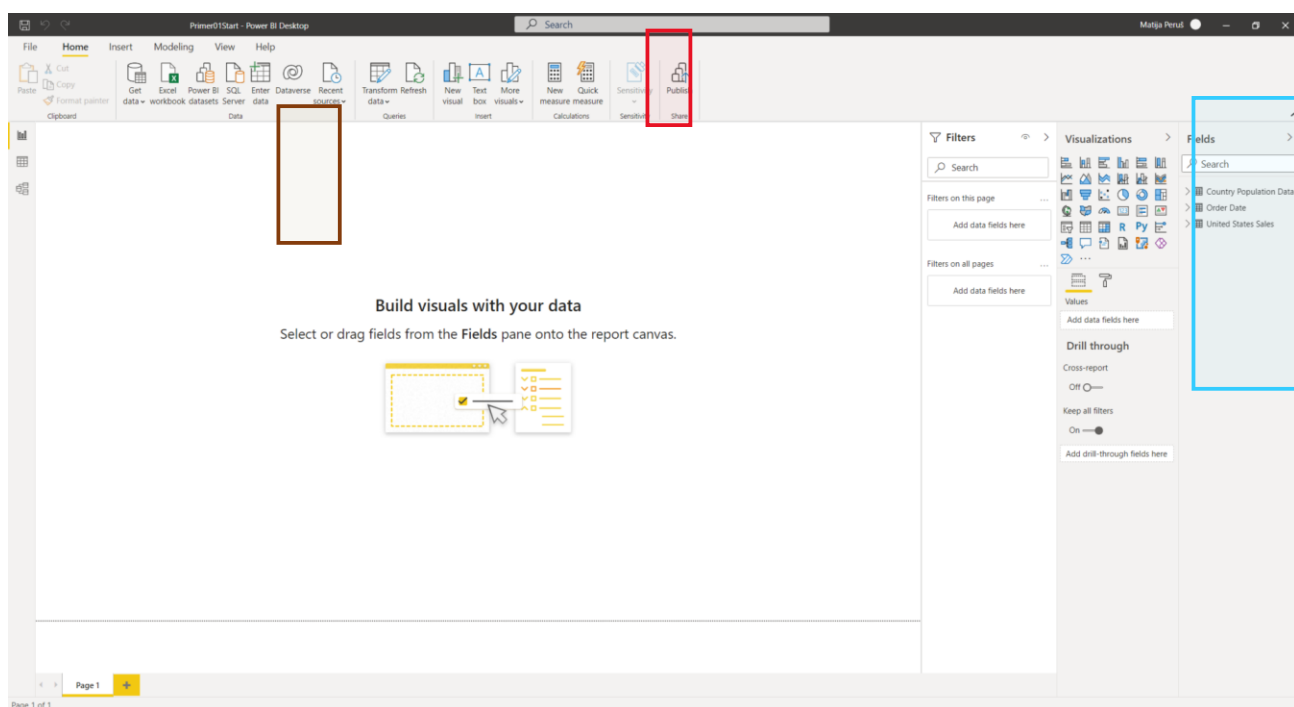
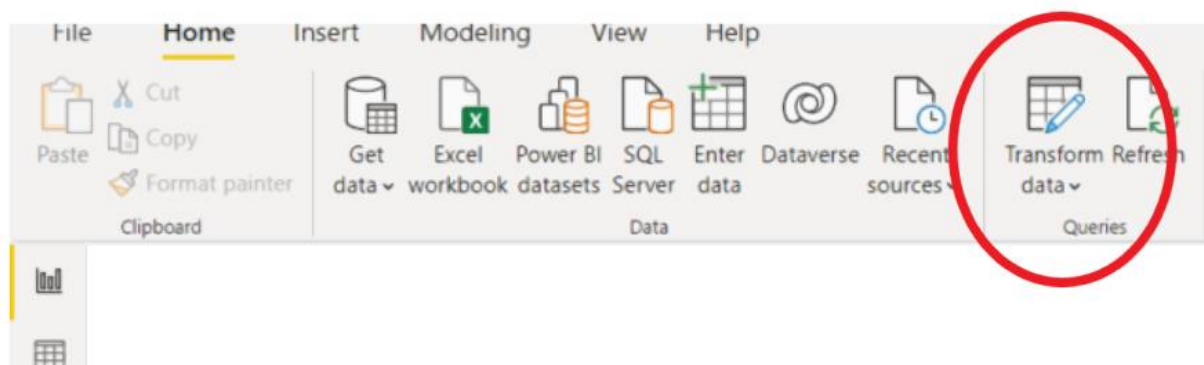


Kratka ponovitev 1. srečanja



Slika 1 Delovno okolje

Na sliki so izpostavljeni najpomembnejši elementi. Z rjavo barvo je obkrožena lokacija pogledov (od zgoraj navzdol) pogled platna – kjer ustvarjamo vizualizacije, pogled podatkov – kjer lahko pregledamo naše podatke in pogled modela – kjer ustvarjamo naše relacije. Z modro so označeni filtri, z oranžno možnosti vizualizacij, z zeleno pa podokno polja, kjer vidimo katere poizvedbe (queries) imamo naložene v našem modelu. Za nas je zaenkrat najpomembnejša rdeča lokacija, kjer se nahaja naš Power Query Editor (v nadaljevanju PQE).



Slika 2 Lokacija Power Query Editorja

Primer01Start - Power Query Editor

Order Date	Day Number Of Week	English Day Name of Week	Day Number of Month	Day Number of Year	Week Number of Year	English Month Name
1. 01. 2010	6	Friday	1	1	1	January
2. 01. 2010	7	Saturday	2	2	1	January
3. 01. 2010	1	Sunday	3	3	2	January
4. 01. 2010	2	Monday	4	4	2	January
5. 01. 2010	3	Tuesday	5	5	2	January
6. 01. 2010	4	Wednesday	6	6	2	January
7. 01. 2010	5	Thursday	7	7	2	January
8. 01. 2010	6	Friday	8	8	2	January
9. 01. 2010	7	Saturday	9	9	2	January
10. 01. 2010	1	Sunday	10	10	3	January
11. 01. 2010	2	Monday	11	11	3	January
12. 01. 2010	3	Tuesday	12	12	3	January
13. 01. 2010	4	Wednesday	13	13	3	January
14. 01. 2010	5	Thursday	14	14	3	January
15. 01. 2010	6	Friday	15	15	3	January
16. 01. 2010	7	Saturday	16	16	3	January
17. 01. 2010	1	Sunday	17	17	4	January
18. 01. 2010	2	Monday	18	18	4	January
19. 01. 2010	3	Tuesday	19	19	4	January
20. 01. 2010	4	Wednesday	20	20	4	January

Slika 3 Power Query Editor (PQE)

Close & Apply, New Source, Recent Sources, Enter Data, Data source settings, Manage Parameters, Refresh Preview, Properties, Advanced Editor, Manage, Choose Columns

Queries [3]

- Order Date
- United States Sales
- Country Population Data

Order Date

Day Number Of Week

1. 01. 2010

2. 01. 2010

3. 01. 2010

4. 01. 2010

5. 01. 2010

6. 01. 2010

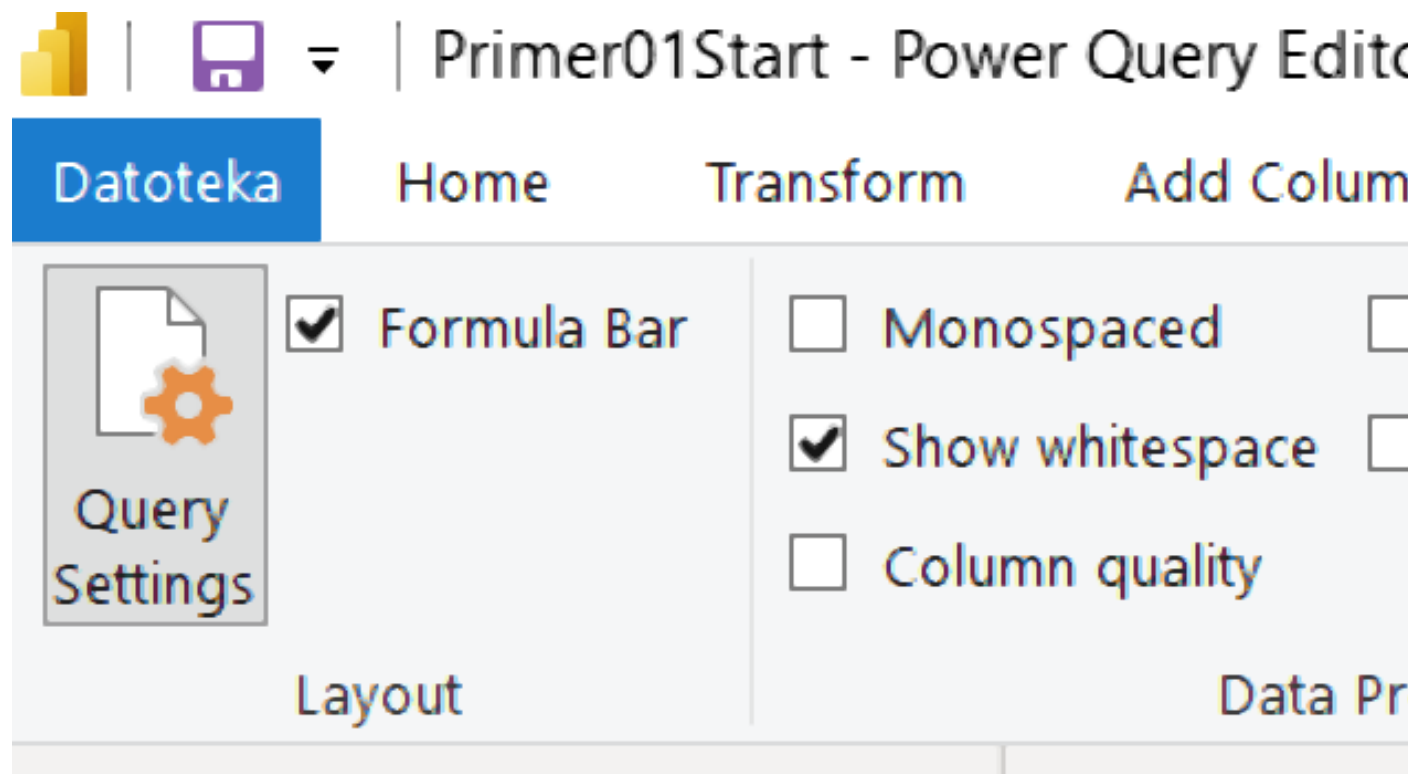
7. 01. 2010

8. 01. 2010

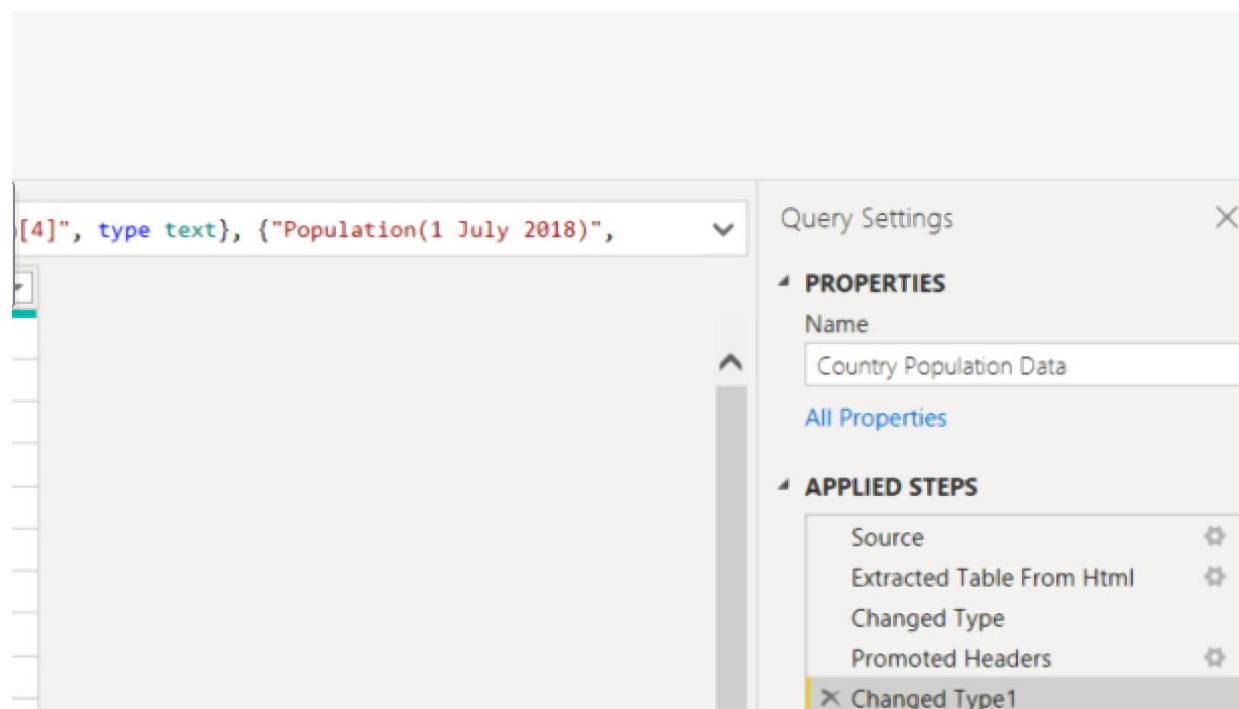
Slika 4 Leva stran PQE

Na rdeče označenem mestu dodamo nov vir. Vsak dodan vir se bo shranil pod pod Queries (poizvedbe), ki je obarvan z zeleno. S svetlo modro je označena lokacija, kjer smo uporabili funkcijo, s katero smo prvo vrstico spravili v glavo stolpca (use first row as header), z rumenim krogom pa lokacija, kjer smo spremenili podatkovni tip. Na zgornjem traku je s temno modro barvo označena opcija »Remove Rows«, tam smo odstranili nepotrebne vrstice. V sredini pa imamo še vnosno vrstico

formule, kjer lahko sami vpisujemo M jezik, ali pa samo spremljamo, kako v M jeziku izgledajo naši »kliki«. Če se želimo vrstice za vnos formule znebiti, gremo na zavihek »View« in levo zgoraj obkljukamo »Formula bar«, če pa po pomoti (ali namerno) zapremo »Query Settings« (nastavitve poizvedb) pa se opcija za ponoven vklop nahaja ob »Formula baru« (glej sliko 5).



Slika 5 Vnos formule in nastavitve poizvedb

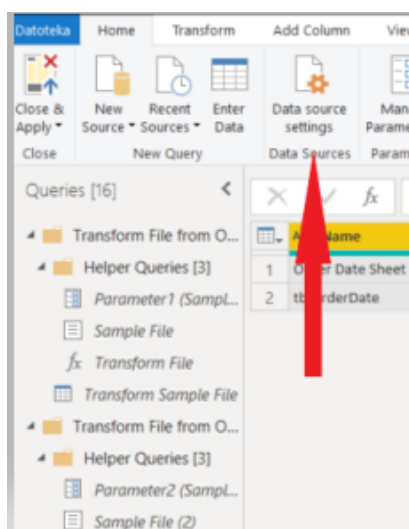


Slika 6 Desna stran PQE

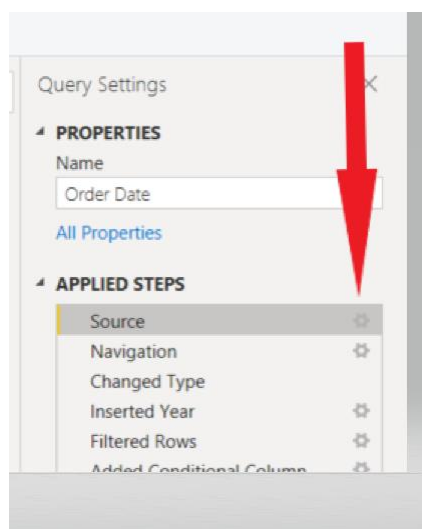
Na desni strani pa se nahajajo še nastavitve poizvedb. Pod properties lahko spremenimo ime, pod tem pa se nahajajo »Applied Steps« oz. uporabljeni koraki pri naših operacijah s podatki. Power BI zaenkrat še ne podpira možnosti »UNDO / REDO« oz. »razveljavi / ponovi vneseno«, tako da vse napake oz. popravke opravljamo tukaj.

Kratka ponovitev 2. in 3. srečanja.

Na 2. srečanju smo si pogledali kako v Power Query (v nadaljevanju PQ) uvozimo še par drugih tipov datotek. Tisti tabeli iz spleta smo dodali še excelove datoteke, tekstovne datoteko in mapo. To smo storili na tak način, da smo si iz spleta prenesli **henkel01.zip** datoteko in njeno vsebino spravili v mapo **c:\powerbi\henkel**. Pot je pomembna, oz. je bila na začetku, saj smo pokazali tudi možnosti kako nastaviti pot (glej sliko).



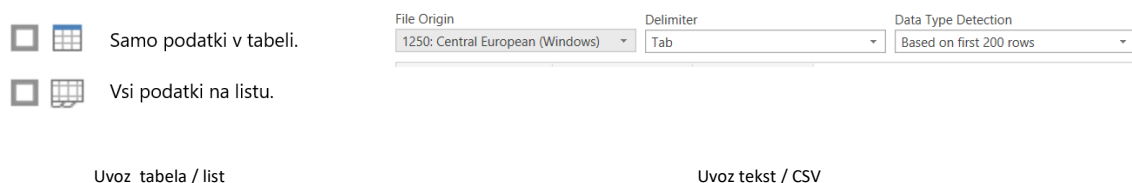
V zavihku home



Kliknemo na zobnik v korakih

Ko smo pgnali to našo datoteko **primer01start.pbix** smo v podoknu polja videli, da so v tem modelu že naložene 3 datoteke. Za tisto iz spleta ni problem, ker se naslov ni spremenil, ti dve pa smo uvozili iz diska in zato je pomembno, da je pot nastavljena pravilo.

Naš model smo pričeli dopolnjevati najprej z excel datoteko, kjer smo si pogledali razliko med uvozom tabele ali celotnega lista (če je urejena vedno uvozimo tabelo).



Pri tekst datoteki PQ prepozna tip po prvih 200 vrsticah ali če spremenimo, gre pogledat vse (načeloma ni potrebno) in je bil potreben še kak korak več, da smo sami spravili podatke v tip tabele (prvo vrstico smo povzdignili v glavo tabele).

Majhna razlika se pojavila pri uvozu mape. Če imamo tabele z isto vsebino, kot v našem primeru prodaja po državah, v isti mapi, jo lahko uvozimo kot mapo. Ne, da lahko, še bolje je, da uvozimo mapo, ker če bi se v mapo dodala ena datoteka npr. Sloveniasales, bi jo pri PowerBI pri posodobitvi avtomatsko dodal naslednjič, ko stisnemo refresh. Enako je mapa boljša opcija, če pride Corona in se npr. Avstralija odloči, da več naših uslug ne potrebuje in če mi uvažamo mapo, bo PowerBI gledal samo kaj je v mapi in bo uvozil kar pač ima. Če pa bi ročno dodajali datoteko po datoteko, pa bi se model sesul, ker ne bi našel Avstralije.

Opcija za uvoz mape pride prav, če npr. nimate dostopa do vseh podatkov, pa vam periodično nalagajo posodobitve na določeno podatki so po navadi v točno določeni obliki.

Pri uvozu mape še levo spodaj opazimo možnost »*skip errors*« (preskoči napake).

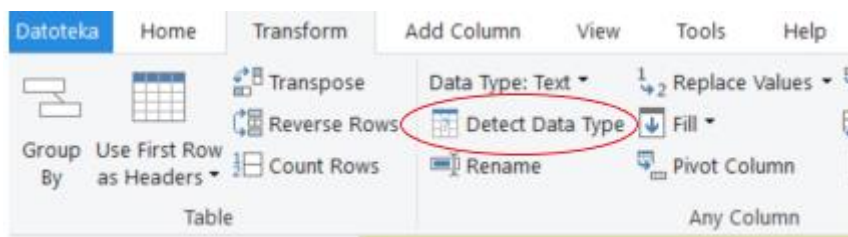


Pri uvažanju mape pa se lahko zgodijo tri stvari.

- Ali vse štima
- Ali preskoči napake (če označimo to možnost)
- Ali zablokira

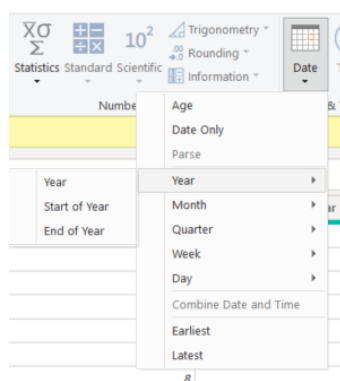
S tem dodajanjem smo prišli na iz naših 3 querijev (poizvedb) na 7. Teh 7 smo začeli pripravljati na nadaljnjo obdelavo – začeli smo jih čistiti. To pomeni, da smo stolpce preimenovali v večini primerov nastavili podatkovni tip, par pa smo jih pustili nalašč napačnih, da bomo danes videli, kako izgleda, ko pride do napak pri dodajanju in združevanju.

Pri nastavljanju pravilnega podatkovnega tipa smo si pomagali tudi s funkcijo »*auto detect*«, ki načeloma dela super, se pa zna zgoditi, da hoče včasih malo preveč pomagati – kar smo opazili pri *Order Date* – stolpec Month Year.



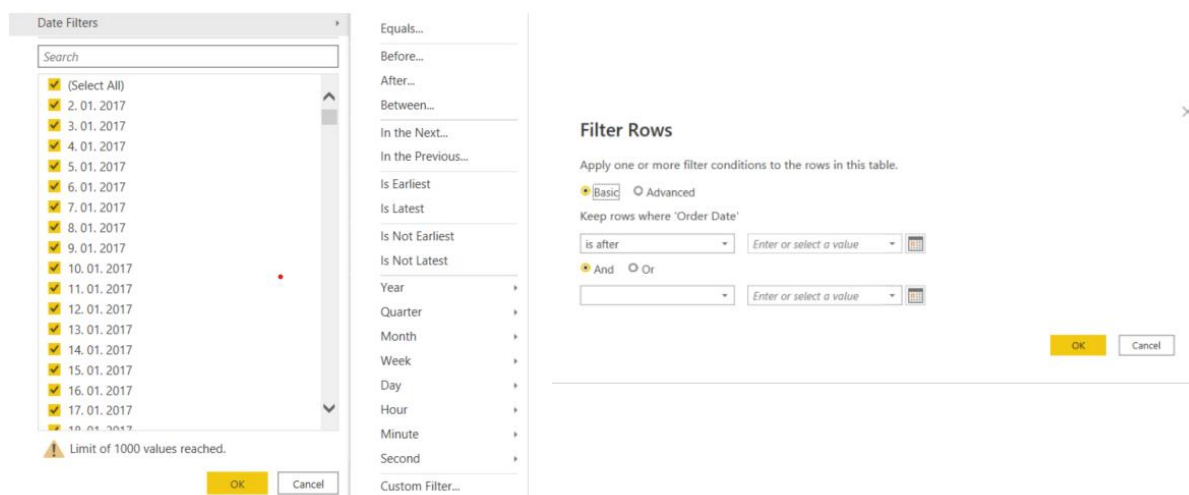
Napaka je bila vstavljena, da nas opozori, da moramo poznati svoje podatke in jih po uporabi avtomatskih funkcij še enkrat preveriti. Ker to niso vaši podatki, seveda tega niste mogli vedeti, razen, če ste si doma prej odprli to tabelo.

V tej tabeli smo še uporabili funkcijo izvažanja, da smo pokazali, kako lahko iz stolpcev dobimo podatke, odvisno od nastavljenega podatkovnega tipa (na sliki primer podatkovnega tipa »Date«).

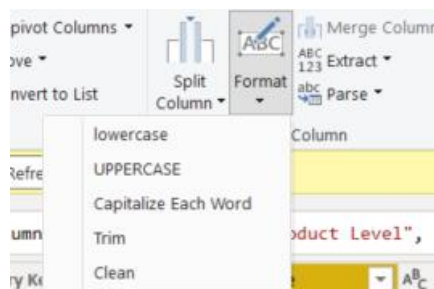


Verjetno ste že opazili, da se zgoraj na traku pri različnih zavihkih ukazi ponavljajo. Razlika je, da če ukaz izvedem pod zavihkom *Transform*, se bo transformacija zgodila tukaj, v tem stolpcu, če pa to opravim pod zavihkom *Add Column*, pa mi bo PQ na koncu vrnil nov stolpec, tako kot smo to storili mi.

Od nastavljenega podatkovnega tipa bodo odvisni tudi filtri, pri našem datumskem so tako na voljo (na sliki tipi filtrov in primer okna, ko izberemo možnost *After*)



V tabeli *Product* smo delali še največ čiščenja, v prvem stolpcu smo se znebili duplikatov (desni klik na stolpec in *Remove Duplicates*). Pri imenih pa smo prikazali delovanje formatiranja, kjer smo uporabili *Trim*, *Clean* in pa *Capitalize Each Word*, ker pač včasih pridejo podatki iz sistemov, ki se med sabo v celoti ne podpirajo in se ti »skrivni znaki« odstranijo.

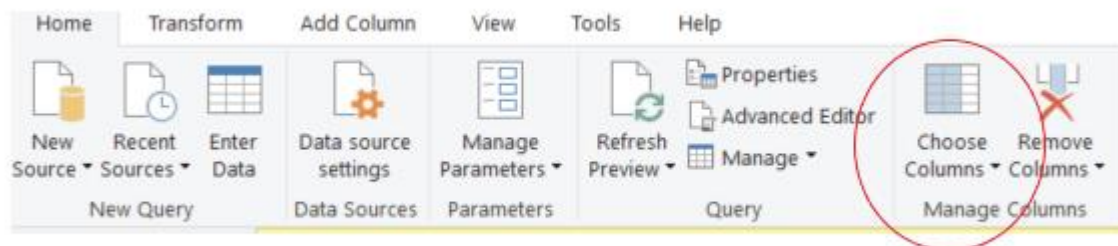


Da smo pokazali, koliko je teh znakov, smo dodali še dva stolpca in sicer dolžina pred čiščenjem in pa po, do tega smo prišli, da smo v zavihku *Add Column* extractali (izvozili) dolžino (lokacija na zgornji sliki desno od možnosti *Format*).

Prikazali smo tudi menjavo vrednosti pri opisu produktov, kjer smo menjali *null* z »brez opisa«. To smo storili z desnim klikom na stolpec in opcijo *Replace Values*.

V datoteki s teritoriji smo stolpec tudi razbili na več delov (*Split Column* – glej zgoraj). Delimeter (ločilo), po katerem delimo PQ prepozna avtomatsko, vedno pa še obstaja možnost, da si ga pod *Custom* nastavimo sami.

Pri čiščenju podatkov pa je potrebno odstraniti tudi, za analizo nekoristne, stolpce. Lahko jih odstranimo z desnim klikom in izberemo *Remove Columns*, boljša opcija pa je, da gremo na zavihek home in pod skupino ukazov *Manage Columns*, izberemo možnost *Choose Columns*.

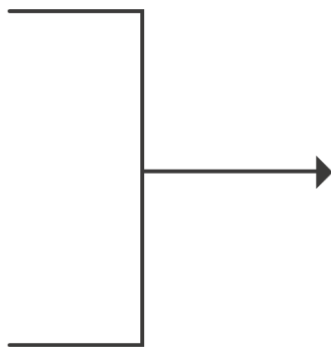


To je boljši in varnejši način brisanja, sploh, če gre za tabelo z ogromno stolpci. Tu imamo opcije da izbiramo stolpce ali gremo na stolpce. Označim katere želim obdržati, kliknem *Ok* in pogledam k našim korakom. Vidimo, da se ime koraka spremeni iz *Removed Column* na *Removed Other Columns*. In spet podobno, kot je bila situacija prej pri mapah, če uporabimo to možnost, ko nekdo doda kaj v tabelo bo model še vedno deloval, ker odvzame vse ostale. V nasprotnem primeru pa ne.

Potem smo si še pogledali kako deluje kombiniranje querijev. Obstajata dva načina *Append* (dodajanje) in *Merge* (združevanje). Da bo lažje razumljivo, si bomo prvo pogledali na slikah kako delujeta v principih.

A	B	C
1	1	1
2	2	2
3	3	3

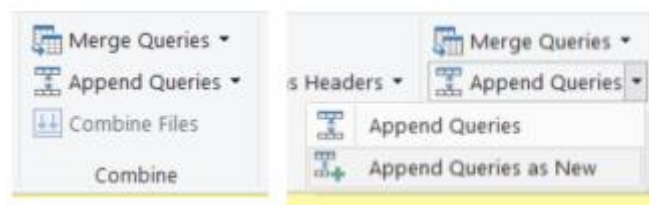
A	B	D
4	4	4
5	5	5



A	B	C	D
1	1	1	null
2	2	2	null
3	3	3	null
4	4	null	4
5	5	null	5

Append oz. dodajanje (<https://docs.microsoft.com/en-us/power-query/append-queries>)

Če imam dve tabeli in želim eno dodati drugi (eno pod drugo), stisnem *Append*. To možnost (kot tudi *Merge*) najdem na traku v skupini *Combine Columns*.



Pri obeh možnostih združevanja se ob kliku na puščico dol, pojavi možnost, da ustvari se poizvedbi združita ali pa se ustvari nova.

Če imata tabeli enako število stolpcev, ki so enakega podatkovnega tipa in imena, potem seveda ni nobenega problema, ker se zgodi, da če je leva tabela 1 in desna 2, bo desno samo prilepil spodaj k levi. To so idealne razmere, ki pa, kot si lahko predstavljate, so bolj kot ne redkost. V večini primerov se nam zgodi, da stolpci ne štimajo ali po tipu ali pa po imenu. In v takem primeru se zgodi to, da bo PBI dodal stolpec na tak način in dobili bomo en kup teh sivih vrednost, ki bojo v tabeli imeli vrednost *null*. Tega nočemo in če se le da to sami prejš očitimo.

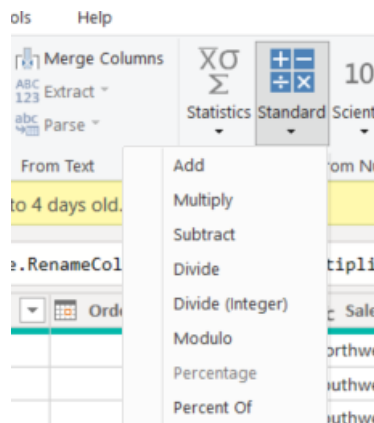
Končni uporabnik, ni nujno tisti, ki bo podatke pripravljaj, bo imel na voljo to okno. In v tem oknu, če se bo analizirala prodaja, bi bilo verjetno bolje, da bi imeli celotno prodajo, ne pa Ameriko ločeno od sveta, ker če želimo ločit, lahko to preprosto naredimo s filtri.

Prisotni na srečanju so imeli nalogo, da uredijo tabelo z vsemi državami, na način, da jo bomo lahko združili s prodajo po ZDA. Torej, da smo z opcijo *Choose Columns*, odstranili stolpce in obe tabeli preoblikovali, da sta imeli enako število stolpcev.

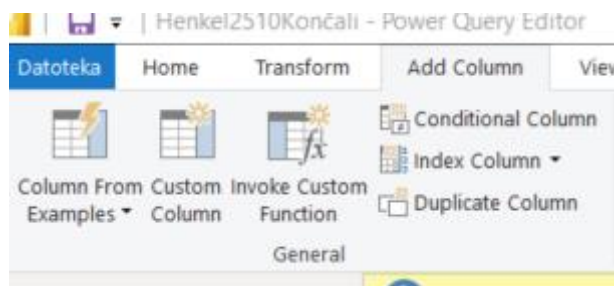
Pri dodajanju smo ustvari nov query, ki dobi privzeto ime *Append1*. Preimenovali smo ga v *Sales* in uredili napake, ki so nastale zaradi napačnega poimenovanja in napačnega tipa (smo jih pred tem pustili nalašč). Na kratko, če združimo dve tabeli s po pet stolpci in ime združen query več stolpcev, to pomeni, da nismo bili najbolj točni z našimi poimenovanji stolpcev. Če pa se nam ustvari podatkovni tip Any, pa niso bili pravilno nastavljeni podatkovni tipi. Napak ne popravljamo v združenem queriju, temveč gremo v izvirne (lahko bi jih tu, a ne vemo, če bomo tiste še kdaj potrebovali, ali jih bo kdo drug... tako nam narekuje dobra praksa). Ko so napake v izvornih querijih popravimo, se avtomatsko popravijo tudi v združenem.

Po brisanju stolpcev je sledilo še dodajanje le teh. Pogledali smo si načine dodajanja stolpcev v PQ. Pogledali smo si že opcijo *extract*, ko smo gledali dolžino. Ne pozabit enake komande so pod *Transform*, kot *Add Column*, tako da če nismo dobili novega, smo verjetno pozabili prestavit.

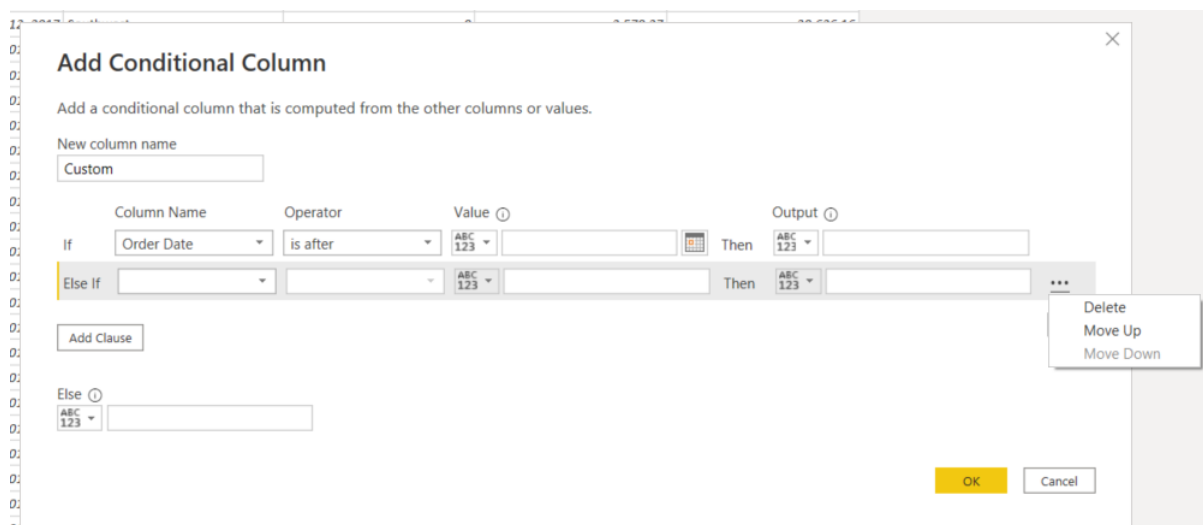
Za izvajanje osnovnih računskih operacij med stolpci je najenostavneje, da označimo željene stolpce (ctrl + levi klik), ter kliknemo na željeno operacijo.



V PQ obstaja tudi možnost, dodajanja stolpca iz primera, torej *Column From Example* (kot v Excelu). To pomeni, da vpišem vzorec iz enega stolpca. Kliknem zgoraj na opcijo in mu lahko naročim, da naj gleda samo označen stolpec ali pa vse. Ustvari se nov stolpec in začnem vpisovat, kar želim, ko sem zadovoljen stisnem Ctrl + Enter in PBI bo celoten stolpec zapolnil z istim vzorcem.



Pri ustvarjanju novih stolpcev, pa je potrebno še omenit *Conditional Columns*, torej stolpce, pri katerih pišemo pogoje – IF stavke.



Pogoje poljubno dodajam z opcijo *Add Clause*. Če želim kaj izbrisat ali spremeniti vrstni red, kliknem na tropižje ob strani in to uredim.

Če pozabimo vpisati vrednost – *Value*, nas PQ preprosto ne bo spustil naprej, če pozabimo *Output*, pa se bodo celice zapolnile z vrednostjo *null*.

Pogoje lahko pišemo tudi v *Custom Column*, ali pa izven PQ, v PowerBi, s čimer pa se bomo se srečali.

Custom column opcija nam pride prav tudi, če želimo dodati prazen stolpec v PQ, s tem, da pač njegovo vrednost nastavimo na *null*.

Custom Column

Add a column that is computed from the other columns.

New column name
Custom

Custom column formula ⓘ
= null

Available columns
Product Key
Order Date
Sales Territory Region
Order Quantity
Unit Price
Total Sales

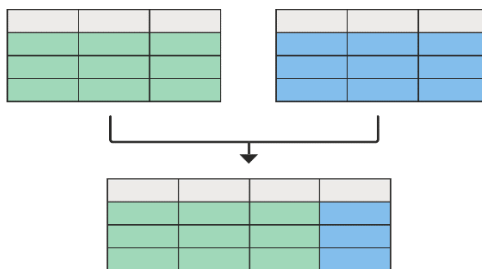
<< Insert

Learn about Power Query formulas

✓ No syntax errors have been detected.

OK Cancel

Torej za *Append* oz. dodajanje druge tabele pod prvo, je dobro, če štimajo stolpci. *Merge* oz. združevanje, pa ima malo več opcij, ker tukaj združujemo po vrsticah. Koncept združevanja na slikah:



Merge (<https://docs.microsoft.com/en-us/power-query/merge-queries-overview>)







Imamo dve tabeli, kjer se prvi dve vrstici ujemata, zadnja pa ne. Najpogostejši tip združevanja je *Left Join*, ki pomeni, da drugo tabelo, ki jo dodajamo pripojim k levi. Pripojile pa se bodo samo vrstice druge, ki se ujemajo z vrstico v prvi, ostale se izgubijo.

Right Join, deluje ravno obratno, tu se nam izgubijo vrstice iz prve, ki jo združimo z drugo.

Potem imamo tu še *Inner Join*, ki bo združil samo tiste vrstice, ki se ujemajo, izgubila se bosta tako tiger, kot slon.

In pa še *Full Join*, ki bo združil vse, ne glede na ujemanje. Kjer bi bil slon v desni tabeli in pa tiger v levi, bo vrednost *null*.

To niso vse opcije, sta še dve anti-opciji, leva in desna. *Left anti join* pripelje samo vrstice iz leve tabele v skupno, *Right anti join* deluje enako v nasprotni smeri.

Join kind	Icon	Description
Left outer		All rows from the left table, matching rows from the right table
Right outer		All rows from the right table, matching rows from the left table
Full outer		All rows from both tables
Inner		Only matching rows from both tables
Left anti		Only rows from the left table
Right anti		Only rows from the right table

<https://docs.microsoft.com/en-us/power-query/merge-queries-overview>

Da nam bo analiza lažja bomo združili querija *Product* in *Product Level*. Kliknem zgoraj na *Merge* (glej sliko pri *Append*), ki se nahaja poleg opcije *Append*. Dropdown meni (puščica navzdol) je spet enak, lahko ustvarimo nov merge ali pa ta dva querija združimo v enega. In če smo prej ustvarili nov query, dajmo tukaj samo mergeat.

Ker že imam označen *Product*, moram izbrati samo še drugo tabelo. Najdem naš *Product Level*.

Merge

Select a table and matching columns to create a merged table.

Product

Product Key	Subcategory Key	Product Name	Product Description	Standard Cost	Color	Text Before
AR-5381	null	Adjustable Race	Brez opisa	null	NA	AR
BA-8327	null	Bearing Ball	Brez opisa	null	NA	BA
BE-2349	null	Bb Ball Bearing	Brez opisa	null	NA	BE
BE-2908	null	Headset Ball Bearings	Brez opisa	null	NA	BE

Product Level

ProductSubcategoryKey	Product Subcategory	Product Category
1	Mountain Bikes	Bikes
2	Road Bikes	Bikes
3	Touring Bikes	Bikes
4	Handlebars	Components
5	Bottom Brackets	Components

Join Kind

Left Outer (all from first, matching from second)

☐ Use fuzzy matching to perform the merge

> Fuzzy matching options

✓ The selection matches 295 of 504 rows from the first table.

OK

Cancel

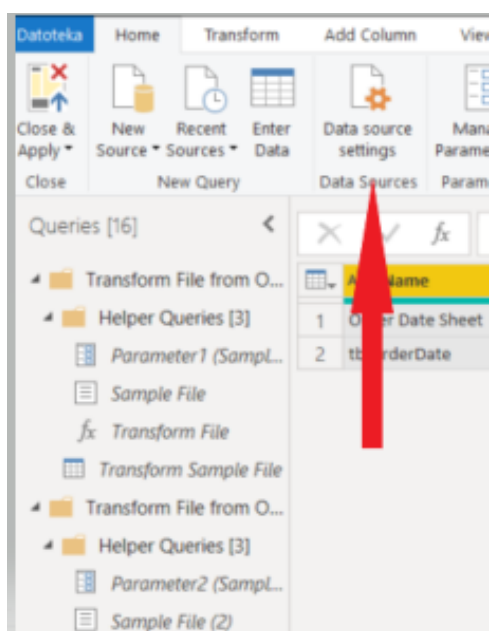
Prvi problem, ki se pojavi je, da če nimam določenih enaki podatkovnih tipov, nas PQ ne bo pustil združevati in moramo v izvornih poizvedbah te stvari popraviti.

Po združevanju se nam pokaže v queriju, ki je bil izbran kot prvi, dodaten stolpec. To v resnici ni stolpec, ampak gre za tabelo, kar vidimo po vrednostih v celicah in pa »razširjevalniku« ob strani (obkrožen na sliki).

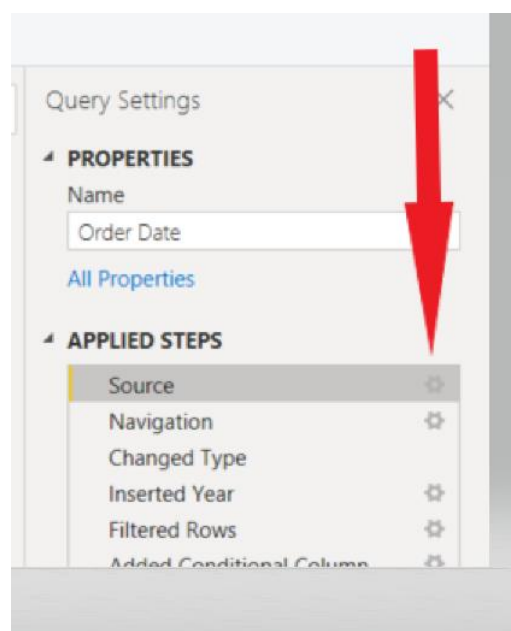
Product Level
Table
Table
Table
Table
Table
Table
Table
Table

Ko kliknemo nanj, se nam prikažejo vsi stolpci, katere smo prinesli iz druge tabele. Ko obkljukamo stolpce, ki jih hočemo, se nam dodajo vsi. Na tak način smo združili naši prvi dve tabeli. Za razumevanje smo še preverili, če znamo združiti tabeli s prodajnimi teritoriji in svetovnim prebivalstvom, na način, da se država pripoji samo vrstica s številom prebivalcev.

Na 2. srečanju smo si pogledali kako v Power Query (v nadaljevanju PQ) uvozimo še par drugih tipov datotek. Tisti tabeli iz spleta smo dodali še excelove datoteke, tekstovne datoteko in mapo. To smo storili na tak način, da smo si iz spleta prenesli **henkel01.zip** datoteko in njeno vsebino spravili v mapo **c:\powerbihenkel**. Pot je pomembna, oz. je bila na začetku, saj smo pokazali tudi možnosti kako nastaviti pot (glej slike).



V zavihku home



Kliknemo na zobnik v korakih

Ko smo pgnali to našo datoteko **primer01start.pbix** smo v podoknu polja videli, da so v tem modelu že naložene 3 datoteke. Za tisto iz spleta ni problem, ker se naslov ni spremenil, ti dve pa smo uvozili iz diska in zato je pomembno, da je pot nastavljena pravilo.

Naš model smo pričeli dopolnjevati najprej z excel datoteko, kjer smo si pogledali razliko med uvozom tabele ali celotnega lista (če je urejena vedno uvozimo tabelo).

- ☐ Samo podatki v tabeli.
- ☐ Vsi podatki na listu.

Uvoz tabela / list

File Origin	Delimiter	Data Type Detection
1250: Central European (Windows)	Tab	Based on first 200 rows

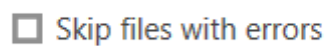
Uvoz tekst / CSV

Pri tekst datoteki PQ prepozna tip po prvih 200 vrsticah ali če spremenimo, gre pogledat vse (načeloma ni potrebno) in je bil potreben še kak korak več, da smo sami spravili podatke v tip tabele (prvo vrstico smo povzdignili v glavo tabele).

Majhna razlika se pojavila pri uvozu mape. Če imamo tabele z isto vsebino, kot v našem primeru prodaja po državah, v isti mapi, jo lahko uvozimo kot mapo. Ne, da lahko, še bolje je, da uvozimo mapo, ker če bi se v mapo dodala ena datoteka npr. Sloveniasales, bi jo pri PowerBI pri posodobitvi avtomatsko dodal naslednjič, ko stisnemo refresh. Enako je mapa boljša opcija, če pride Corona in se npr. Avstralija odloči, da več naših uslug ne potrebuje in če mi uvažamo mapo, bo PowerBI gledal samo kaj je v mapi in bo uvozil kar pač ima. Če pa bi ročno dodajali datoteko po datoteko, pa bi se model sesul, ker ne bi našel Avstralije.

Opcija za uvoz mape pride prav, če npr. nimate dostopa do vseh podatkov, pa vam periodično nalagajo posodobitve na določeno podatki so po navadi v točno določeni obliki.

Pri uvozu mape še levo spodaj opazimo možnost »skip errors« (preskoči napake).

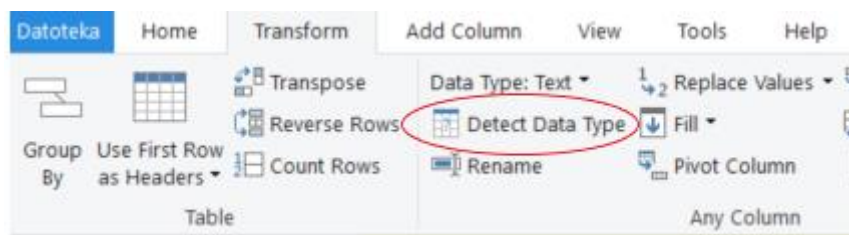


Pri uvažanju mape pa se lahko zgodijo tri stvari.

- Ali vse »štima«
- Ali preskoči napake (če označimo to možnost)
- Ali zablokira

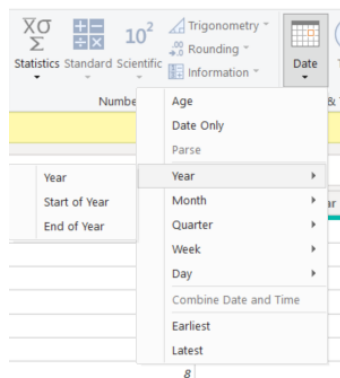
S tem dodajanjem smo prišli na iz naših 3 querijev (poizvedb) na 7. Teh 7 smo začeli pripravljat na nadaljnjo obdelavo – začeli smo jih čistiti. To pomeni, da smo stolpce preimenovali v večini primerov nastavili podatkovni tip, par pa smo jih pustili nalašč napačnih, da bomo danes videli, kako izgleda, ko pride do napak pri dodajanju in združevanju.

Pri nastavljanju pravilnega podatkovnega tipa smo si pomagali tudi s funkcijo »auto detect«, ki načeloma dela super, se pa zna zgodit, da hoče včasih malo preveč pomagat – kar smo opazili pri *Order Date* – stolpec Month Year.



Napaka je bila vstavljena, da nas opozori, da moramo poznat svoje podatke in jih po uporabi avtomatskih funkcij še enkrat preverit. Ker to niso vaši podatki, seveda tega niste mogli vedet, razen, če ste si doma prej odprli to tabelo.

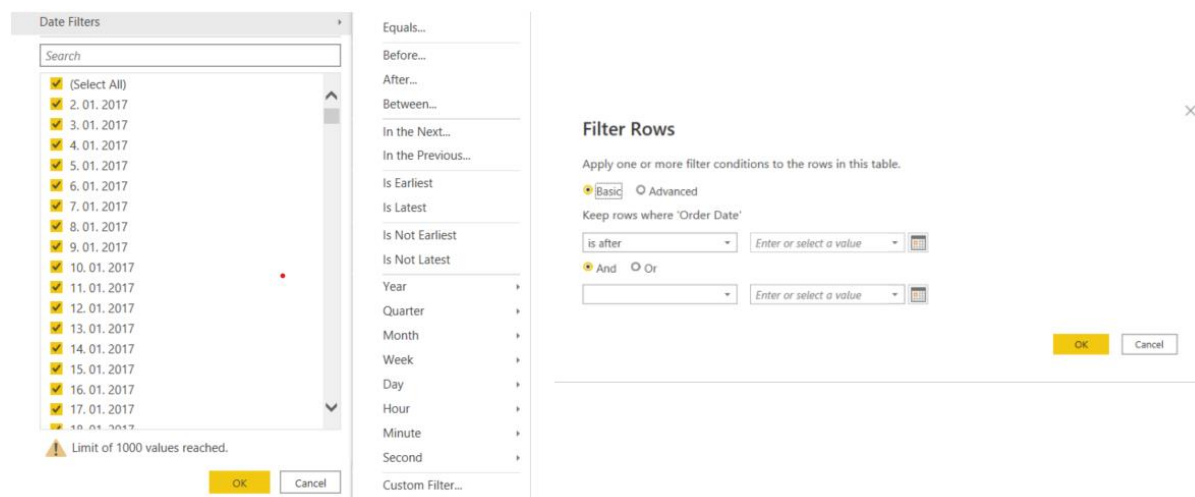
V tej tabeli smo še uporabili funkcijo izvažanja, da smo pokazali, kako lahko iz stolpcev dobimo podatke, odvisno od nastavljenega podatkovnega tipa (na sliki primer podatkovnega tipa »Date«).



Extract Date možnosti

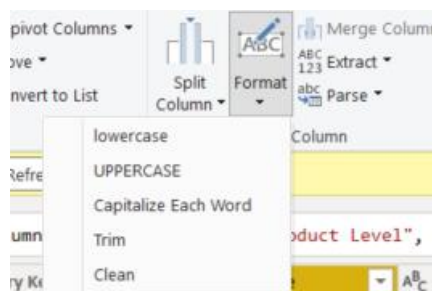
Verjetno ste že opazili, da se zgoraj na traku pri različnih zavihkih ukazi ponavljajo. Razlika je, da če ukaz izvedem pod zavihkom *Transform*, se bo transformacija zgodila tukaj, v tem stolpcu, če pa to opravim pod zavihkom *Add Column*, pa mi bo PQ na koncu vrgel nov stolpec, tako kot smo to storili mi.

Od nastavljenega podatkovnega tipa bodo odvisni tudi filtri, pri našem datumskem so tako na voljo (na sliki tipi filtrov in primer okna, ko izberemo možnost *After*)



Datumsko filtriranje

V tabeli *Product* smo delali še največ čiščenja, v prvem stolpcu smo se znebili duplikatov (desni klik na stolpec in *Remove Duplicates*). Pri imenih pa smo prikazali delovanje formatiranja, kjer smo uporabili *Trim*, *Clean* in pa *Capitalize Each Word*, ker pač včasih pridejo podatki iz sistemov, ki se med sabo v celoti ne podpirajo in se ti »skrivni znaki« odstranijo.



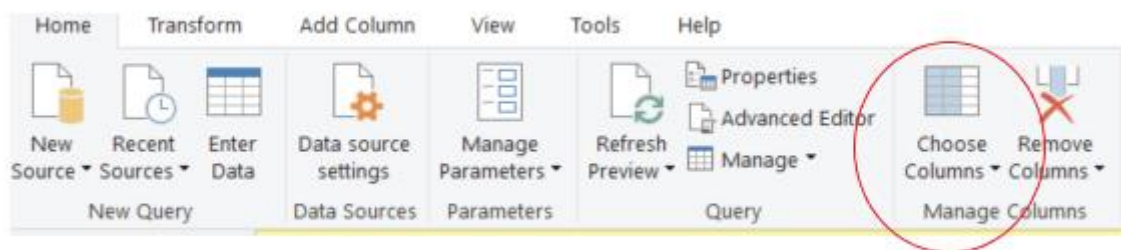
Format

Da smo pokazali, koliko je teh znakov, smo dodali še dva stolpca in sicer dolžina pred čiščenjem in pa po, do tega smo prišli, da smo v zavihku *Add Column* extractali (izvozili) dolžino (lokacija na zgornji sliki desno od možnosti *Format*).

Prikazali smo tudi menjavo vrednosti pri opisu produktov, kjer smo menjali *null* z »brez opisa«. To smo storili z desnim klikom na stolpec in opcijo *Replace Values*.

V datoteki s teritoriji smo stolpec tudi razbili na več delov (*Split Column* – glej zgoraj). Delimeter (ločilo), po katerem delimo PQ prepozna avtomatsko, vedno pa še obstaja možnost, da si ga pod *Custom* nastavimo sami.

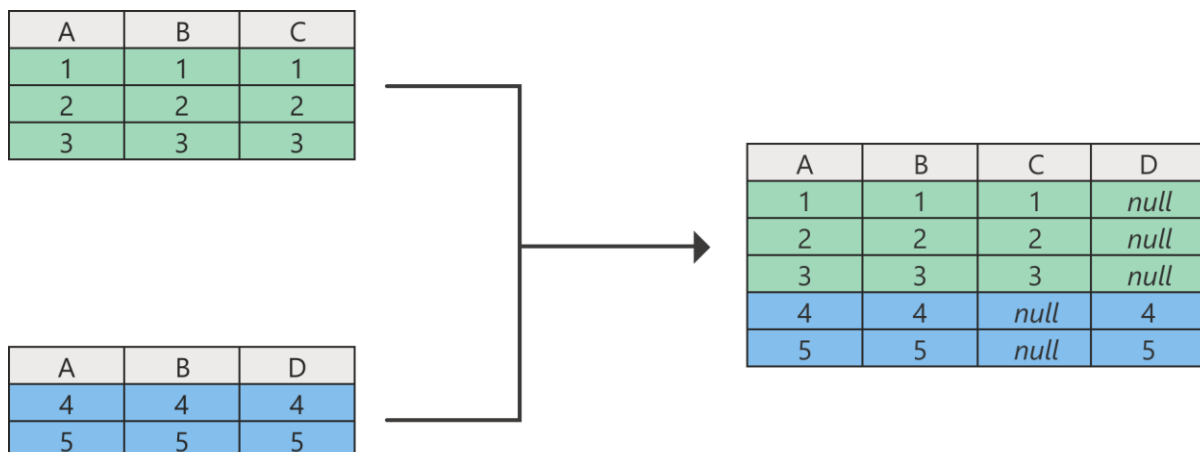
Pri čiščenju podatkov pa je potrebno odstraniti tudi, za analizo nekoristne, stolpce. Lahko jih odstranimo z desnim klikom in izberemo *Remove Columns*, boljša opcija pa je, da gremo na zavihek home in pod skupino ukazov *Manage Columns*, izberemo možnost *Choose Columns*.



Choose Columns

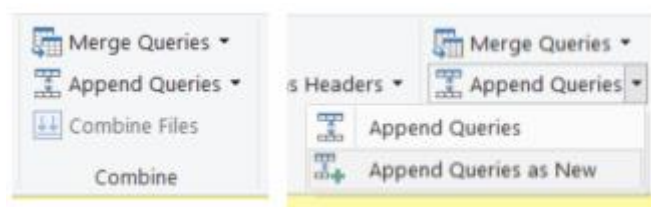
To je boljši in varnejši način brisanja, sploh, če gre za tabelo z ogromno stolpci. Tu imamo opcije da izbiramo stolpce ali gremo na stolpce. Označim katere želim obdržati, kliknem *Ok* in pogledam k našim korakom. Vidimo, da se ime koraka spremeni iz *Removed Column* na *Removed Other Columns*. In spet podobno, kot je bila situacija prej pri mapah, če uporabimo to možnost, ko nekdo doda kaj v tabelo bo model še vedno deloval, ker odvzame vse ostale. V nasprotnem primeru pa ne.

Potem smo si še pogledali kako deluje kombiniranje querijev. Obstajata dva načina *Append* (dodajanje) in *Merge* (združevanje). Da bo lažje razumljivo, si bomo prvo pogledali na slikah kako delujeta v principih.



Append oz. dodajanje (<https://docs.microsoft.com/en-us/power-query/append-queries>)

Če imam dve tabeli in želim eno dodati drugi (eno pod drugo), stisnem **Append**. To možnost (kot tudi **Merge**) najdem na traku v skupini **Combine Columns**.



Pri obeh možnostih združevanja se ob kliku na puščico dol, pojavi možnost, da ustvari se poizvedbi združita ali pa se ustvari nova.

Če imata tabeli enako število stolpcev, ki so enakega podatkovnega tipa in imena, potem seveda ni nobenega problema, ker se zgodi, da če je leva tabela 1 in desna 2, bo desno samo prilepil spodaj k levi. To so idealne razmere, ki pa, kot si lahko predstavljate, so bolj kot ne redkost. V večini primerov se nam zgodi, da stolpci ne štimajo ali po tipu ali pa po imenu. In v takem primeru se zgodi to, da bo PBI dodal stolpec na tak način in dobili bomo en kup teh sivih vrednosti, ki bojo v tabeli imeli vrednost *null*. Tega nečemo in če se le da to sami prej očistimo.

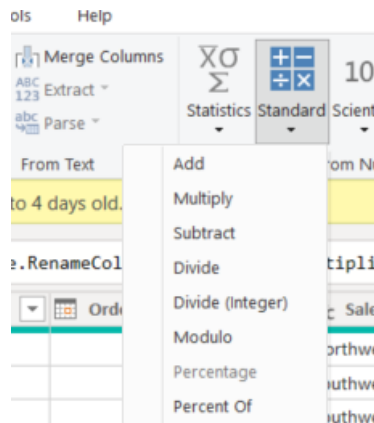
Končni uporabnik, ni nujno tisti, ki bo podatke pripravljala, bo imel na voljo to okno. In v tem oknu, če se bo analizirala prodaja, bi bilo verjetno bolje, da bi imeli celotno prodajo, ne pa Ameriko ločeno od sveta, ker če želimo ločit, lahko to preprosto naredimo s filtri.

Prisotni na srečanju so imeli nalogo, da uredijo tabelo z vsemi državami, na način, da jo bomo lahko združili s prodajo po ZDA. Torej, da smo z opcijo *Choose Columns*, odstranili stolpce in obe tabeli preoblikovali, da sta imeli enako število stolpcev.

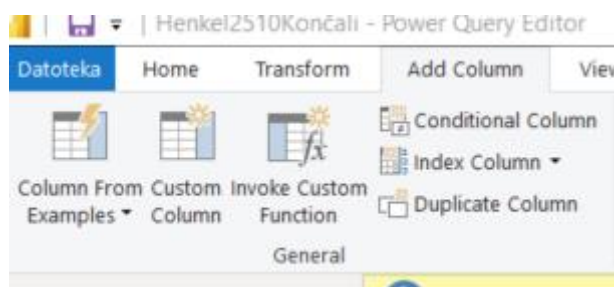
Pri dodajanju smo ustvarili nov query, ki dobi privzeto ime *Append1*. Preimenovali smo ga v *Sales* in uredili napake, ki so nastale zaradi napačnega poimenovanja in napačnega tipa (smo jih pred tem pustili nalašč). Na kratko, če združimo dve tabeli s po pet stolpci in ime združen query več stolpcev, to pomeni, da nismo bili najbolj točni z našimi poimenovanji stolpcev. Če pa se nam ustvari podatkovni tip Any, pa niso bili pravilno nastavljeni podatkovni tipi. Napak ne popravljamo v združenem queryju, temveč gremo v izvirne (lahko bi jih tu, a ne vemo, če bomo tiste še kdaj potrebovali, ali jih bo kdo drug... tako nam narekuje dobra praksa). Ko so napake v izvornih queryjih popravimo, se avtomatsko popravijo tudi v združenem.

Po brisanju stolpcev je sledilo še dodajanje le teh. Pogledali smo si načine dodajanja stolpcev v PQ. Pogledali smo si že opcijo *extract*, ko smo gledali dolžino. Ne pozabit enake komande so pod *Transform*, kot *Add Column*, tako da če nismo dobili novega, smo verjetno pozabili prestavit.

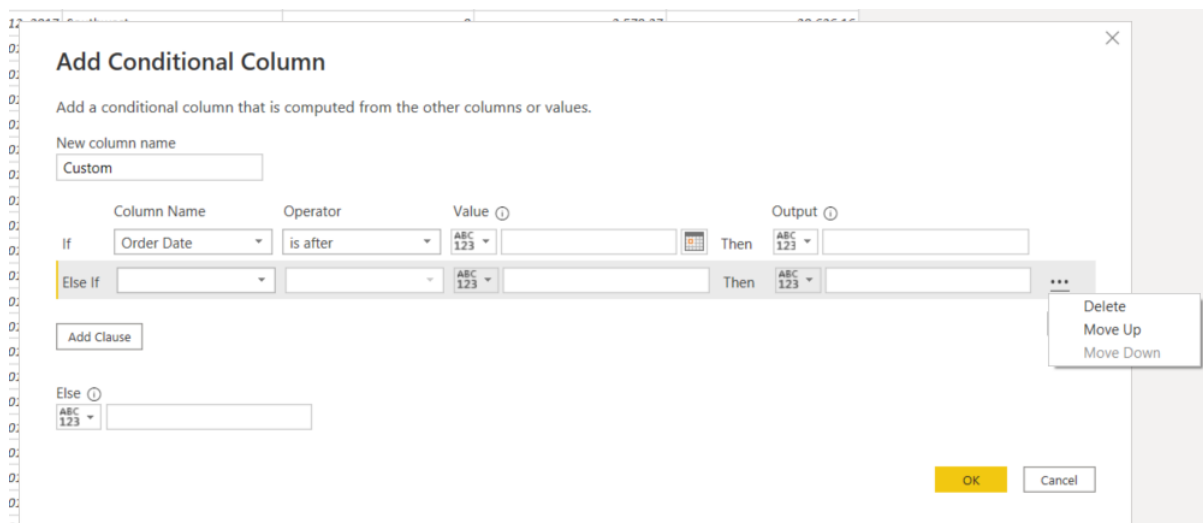
Za izvajanje osnovnih računskih operacij med stolpci je najenostavneje, da označimo željene stolpce (ctrl + levi klik), ter kliknemo na željeno operacijo.



V PQ obstaja tudi možnost, dodajanja stolpca iz primera, torej *Column From Example* (kot v Excelu). To pomeni, da vpišem vzorec iz enega stolpca. Kliknem zgoraj na opcijo in mu lahko naročim, da naj gleda samo označen stolpec ali pa vse. Ustvari se nov stolpec in začnem vpisovat, kar želim, ko sem zadovoljen stisnem Ctrl + Enter in PBI bo celoten stolpec zapolnil z istim vzorcem.



Pri ustvarjanju novih stolpcev, pa je potrebno še omenit *Conditional Columns*, torej stolpce, pri katerih pišemo pogoje – IF stavke.

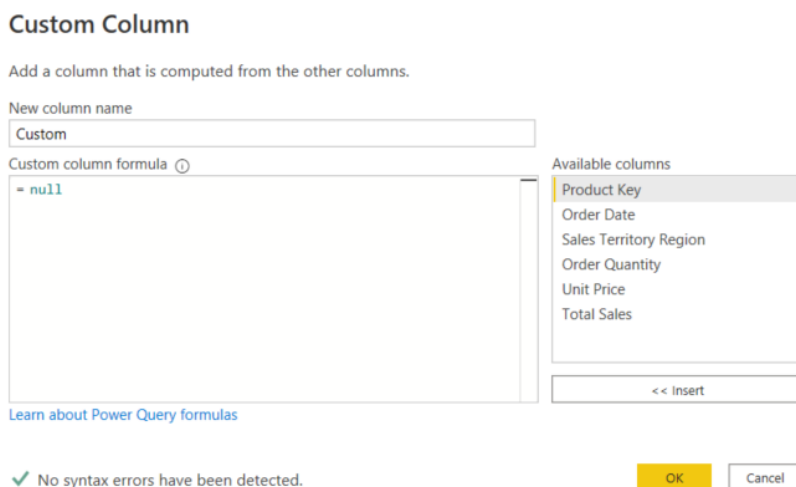


Pogoje poljubno dodajam z opcijo *Add Clause*. Če želim kaj izbrisat ali spremeniti vrstni red, kliknem na tropičje ob strani in to uredim.

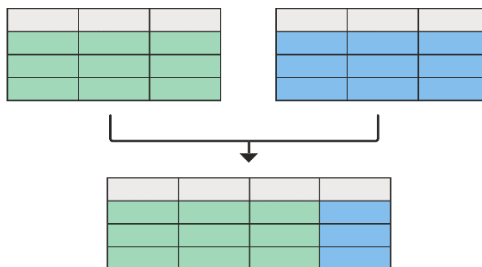
Če pozabimo vpisati vrednost – *Value*, nas PQ preprosto ne bo spustil naprej, če pozabimo *Output*, pa se bodo celice zapolnile z vrednostjo *null*.

Pogoje lahko pišemo tudi v *Custom Column*, ali pa izven PQ, v PowerBi, s čimer pa se bomo se srečali.

Custom column opcija nam pride prav tudi, če želimo dodati prazen stolpec v PQ, s tem, da pač njegovo vrednost nastavimo na *null*.



Torej za *Append* oz. dodajanje druge tabele pod prvo, je dobro, če štimajo stolpci. *Merge* oz. združevanje, pa ima malo več opcij, ker tukaj združujemo po vrsticah. Koncept združevanja na slikah:



Merge (<https://docs.microsoft.com/en-us/power-query/merge-queries-overview>)

Imamo dve tabeli, kjer se prvi dve vrstici ujemata, zadnja pa ne. Najpogostejši tip združevanja je *Left Join*, ki pomeni, da drugo tabelo, ki jo dodajamo pripojim k levi. Pripojile pa se bodo samo vrstice druge, ki se ujemajo z vrstico v prvi, ostale se izgubijo.

Right Join, deluje ravno obratno, tu se nam izgubijo vrstice iz prve, ki jo združimo z drugo.

Potem imamo tu še *Inner Join*, ki bo združil samo tiste vrstice, ki se ujemajo, izgubila se bosta tako tiger, kot slon.

In pa še *Full Join*, ki bo združil vse, ne glede na ujemanje. Kjer bi bil slon v desni tabeli in pa tiger v levi, bo vrednost *null*.

To niso vse opcije, sta še dve anti-opciji, leva in desna. *Left anti join* pripelje samo vrstice iz leve tabele v skupno, *Right anti join* deluje enako v nasprotni smeri.

Join kind	Icon	Description
Left outer		All rows from the left table, matching rows from the right table
Right outer		All rows from the right table, matching rows from the left table
Full outer		All rows from both tables
Inner		Only matching rows from both tables
Left anti		Only rows from the left table
Right anti		Only rows from the right table

<https://docs.microsoft.com/en-us/power-query/merge-queries-overview>

Da nam bo analiza lažja bomo združili querija *Product* in *Product Level*. Kliknem zgoraj na *Merge* (glej sliko pri *Append*), ki se nahaja poleg opcije *Append*. Dropdown meni (puščica navzdol) je spet enak, lahko ustvarimo nov merge ali pa ta dva querija združimo v enega. In če smo prej ustvarili nov query, dajmo tukaj samo mergeat.

Ker že imam označen *Product*, moram izbrati samo še drugo tabelo. Najdem naš *Product Level*.

Merge

Select a table and matching columns to create a merged table.

Product

Product Key	Subcategory Key	Product Name	Product Description	Standard Cost	Color	Text Before
AR-5381	null	Adjustable Race	Brez opisa	null	NA	AR
BA-8327	null	Bearing Ball	Brez opisa	null	NA	BA
BE-2349	null	Bb Ball Bearing	Brez opisa	null	NA	BE
BE-2908	null	Headset Ball Bearings	Brez opisa	null	NA	BE

Product Level

ProductSubcategoryKey	Product Subcategory	Product Category
1	Mountain Bikes	Bikes
2	Road Bikes	Bikes
3	Touring Bikes	Bikes
4	Handlebars	Components
5	Bottom Brackets	Components

Join Kind

Left Outer (all from first, matching from second)

☐ Use fuzzy matching to perform the merge

> Fuzzy matching options

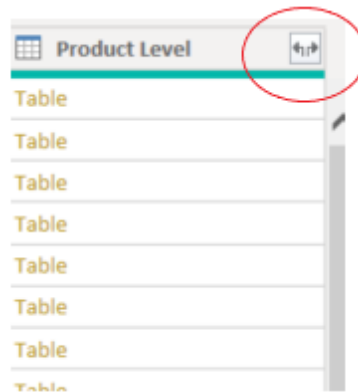
✓ The selection matches 295 of 504 rows from the first table.

OK

Cancel

Prvi problem, ki se pojavi je, da če nimam določenih enaki podatkovnih tipov, nas PQ ne bo pustil združevati in moramo v izvornih poizvedbah te stvari popraviti.

Po združevanju se nam pokaže v queriju, ki je bil izbran kot prvi, dodaten stolpec. To v resnici ni stolpec, ampak gre za tabelo, kar vidimo po vrednostih v celicah in pa »razširjevalniku« ob strani (obkrožen na sliki).



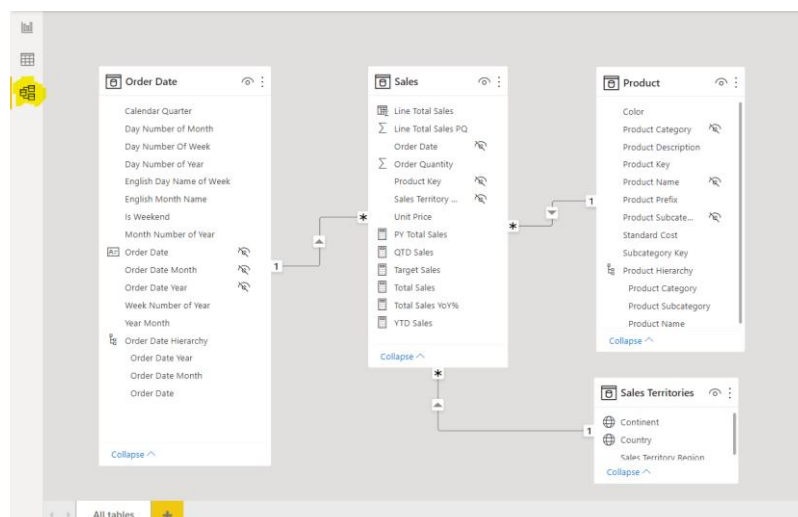
Ko kliknemo nanj, se nam prikažejo vsi stolpci, katere smo prinesli iz druge tabele. Ko obkljukamo stolpce, ki jih hočemo, se nam dodajo vsi. Na tak način smo združili naši prvi dve tabeli. Za razumevanje smo še preverili, če znamo združiti tabeli s prodajnimi teritoriji in svetovnim prebivalstvom, na način, da se država pripoji samo vrstica s številom prebivalcev.

POZOR!

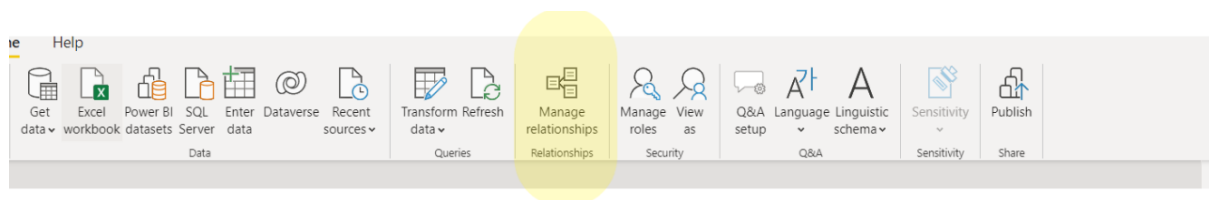
Takšen način dela ni optimalen. Je bolj enostaven za uporabnike Excela, ki prihajajo iz »stare šole«, kjer je navada, da se vse da na en kup in to je celo predstavljalo dobro prakso. Ključna beseda tu pa je **predstavljalo**, saj to ne velja več. Preden so bili PQE, Power Pivot in DAX na voljo v Excelu, pač ni blo orodij s katerimi se je dalo ustvariti kaj drugače in se je tako delalo, če smo želeli analizirati. Tradicionalna Pivot tabela -> črpa iz enega podatkovnega vira in potem imamo milijon Lookup in Match funkcij.

Na kratko, lahko naredite tako ni pa učinkovito. Preprosto se ustvari preveč nepotrebnih, odvečnih podatkov, ki porabijo neprimerno več spomina in procesorske moči.

Modele ustvarjamo v *Model View* (označeno z rumeno) .



Modele lahko ustvarimo na dva načina – Click & Drag ali kliknemo zgoraj na *Manage Relationship*. Pravzaprav obstaja še način avtomatske detekcije, a smo tega na začetku izklopili, saj nam tako narekujejo tudi pravila dobre prakse. Če pa nas zamika, da bi vseeno to zaupali PBI, pa lahko pod *Manage Relationship*, spet stisnemo autodetect.



Hitrejši način je Klikni in Potegni, sam pa vam priporočam, da se kar držimo gumba, saj imamo tako pregled nad še dodatnimi nastavitvami.

Če smo bili pridni pri naši pripravi podatkov v PQ, bo PBI že brez, da kliknemo na katerikoli stolpec, avtomatsko identificiral ključ, po katerem želimo ustvariti povezavo. Spodaj vidimo še možnosti *Cardinality* in *Cross Filter*. Se k temu še vrnemo, zaenkrat OK in Close in imamo našo prvo povezavo.

Ko grem z miško čez črto, se mi »highlighta« povezava med ključi vidim pa tudi smer filtra in pa kardinalnost. 1 predstavlja one, * pa many.

Edit relationship

Select tables and columns that are related.

Sales

Product Key	Order Date	Sales Territory Region	Order Quantity	Unit Price	Line Total Sales PQ
WB-H098	torek, 04. februar 2020	Southwest	3	\$4,99	\$14,97
WB-H098	torek, 04. februar 2020	Southwest	3	\$4,99	\$14,97
WB-H098	nedelja, 09. februar 2020	Southwest	3	\$4,99	\$14,97

Product

Product Key	Subcategory Key	Product Name	Product Description	Standard Cost	Color	Product Prefix
AR-5381	null	Adjustable Race	No description	null	NA	AR
BA-8327	null	Bearing Ball	No description	null	NA	BA
BE-2349	null	Bb Ball Bearing	No description	null	NA	BE

Cardinality

Many to one (*:1)

Cross filter direction

Single

☒ Make this relationship active

☐ Apply security filter in both directions

☐ Assume referential integrity

OK

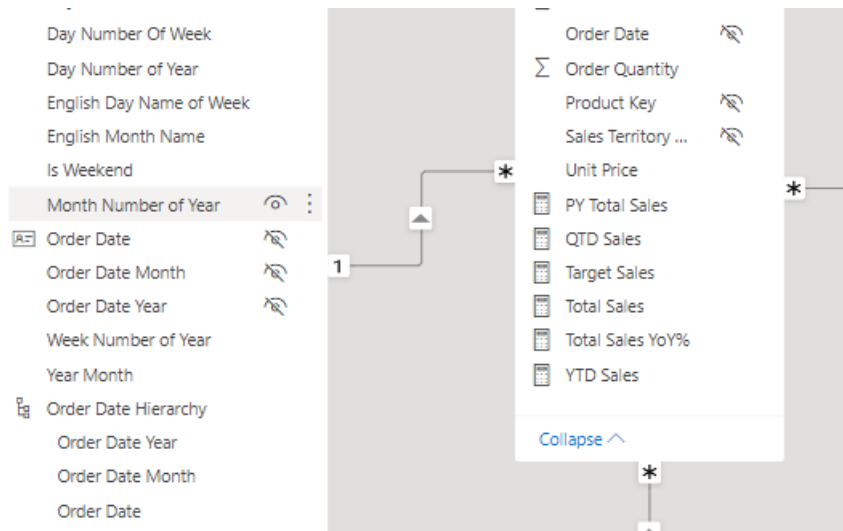
Cancel

Pri našem modeliranju smo izbrali za zvezdni model, ki je za PBI priporočljiv. Model (viden na zgornji sliki) je sestavljen iz Fact tabele (Data tabele) in dimenzijske tabele (Lookup tabele). Fakt tabela je na sredini, okoli nje pa se nahajajo dimenzije.

Sam izgled niti nima kakšnega pomena, je pa lažje berljiv. Pomembno pa je, da preverimo naše filtre in kardinalnost, kjer želimo, da puščice vedno kažejo v smeri iz dimenzij proti fakt tabelam in da je pri dimenziji število 1, pri Fakt tabeli pa * (zvezdica).

Če spet povežemo z Excelovimi vrtilnimi tabelami, v Fakt gre vse, kar gre tam pod values, v dimenzije pa vse kar gre tam v vrstice in stolpce.

Če se nam kakšne vrednosti podvajajo, jih lahko tudi skrijemo. Skrijemo jih, če ne želimo, da končni uporabnik (ali pa tudi mi sami), uporabi ključ, ki ima isto ime in se nahaja v napačni tabeli in to storimo tako, da gremo z miško čez željeno količino v modelu, kjer se nam pokaže in kliknemo nanj (slika spodaj)



V našem primeru smo skrili možnosti *Order Date*, tako v fakt, kot v dimenzijski tabeli, pustili pa smo »odkrito« hierarhijo, za katero smo predvideli, da se bo uporabljala.

Ko imamo vzpostavljen model, lahko začnemo to tudi izkoriščati. Pred povezavami nobena od tabel ni vedela za obstoj druge. Pri obdelavi modela smo si pomagali z DAX. DAX pomeni *Data Analysis Expressions*. Gre za funkcijski jezik, ki je povzet po jeziku, v katerem v Excelu pišete formule.

Z Daxom ustvarjamo stolpce (*Calculated Columns*), tabele (*Calculated tables*) in mere (*Measures*). Najredkeje se uporablja za tabele, večinoma, če potrebujemo kaj »na hitro«, ker smo to pozabili ustvariti prej, v PQE.

Calculated Columns se uporablja za kontekst – kjer rabimo vrednosti s katerimi bomo opisovali oz. filtrirali – ne pa ustvarjali nove vrednosti. Je pa treba na tem mestu omeniti, da zaradi načina delovanja PBI, je to boljše (hitreje in učinkoviteje), če se to že opravi v PQE.

Mere pa so, po domače povedano, DAX formule s katerimi ustvarimo nove izračunane vrednosti.

Pri pa se moramo zavedati naslednje stvari:

- Jih ne vidimo dokler jih ne vizualiziramo (podobno kot calculated field v Excel pivot)
- Mere imajo filter context - vrednosti se konstantno spreminjajo glede na filtre, ki obstajajo v tem trenutku.

Torej, če potrebujemo pri analizi dodatne vrstice, stolpce, slicerje ali filtre: *Calculate Column*. Če pa potrebujemo vrednosti, pa *Measure* (mero).

Primeri uporabljenih formul se nahajajo v datoteki *Uporabljene formule*, na githubu. Formul je preveč, da bi se jih naučili na pamet, bi pa posebej izpostavil dve. Prva je, po mojem mnenju, bolj enostaven zapis IF stavka. Namesto IF, uporabimo sintakso SWITCH(TRUE)):

Uspešnost prodaje =

```
SWITCH(
    TRUE(),
    'Sales Territories'[skupno št. prodaj] >= 8000, "Zlo dobr..",
    'Sales Territories'[skupno št. prodaj] >= 4000, "dobr..",
    'Sales Territories'[skupno št. prodaj] >= 1, "Zlom..",
    "N/A"
)
```

(Formule se lahko pišejo v eni vrstici, zaradi lepšega izgleda pa lahko pišemo v večih. V novo vrstico pridemo s standardno kombinacijo tipk Shift in Enter.)

Druga pa CALCULATE(). Calculate je absolutni prvak po številu uporab v PBI. Že sama po sebi je dokaj »močna« (od Excelovih formul še najbolj spominja na SUMIF() in COUNTIF() , s to razliko, da lahko opravlja vse, ne pa samo sešteva in šteje. Lahko si jo predstavljate kot CALCULATEIF.... Funkcijo se lahko naredi še bolj uporabno, največkrat v kombinaciji s funkcijami ALL() in FILTER().

ALL() zanemari filtre, s funkcijo Filter() pa lahko vpisujemo dodatne, komplicirane pogoje (pri sami Calculate() lahko pišemo bolj enostavne pogoje, brez mer in brez tipa True/False).

Oblika: **CALCULATE(izraz, filter1,filter2,...)**

Enostaven primer:

Prodaja šamponov = Calculate([Skupna prodaja], Produkt[Kategorija] = »Šamponi«)

V tem primeru že imamo hipotetično mero, Skupna prodaja, v kateri so zajete vse kategorije artiklov in filter, ki filtrira po kategorijah (iz stolpca).

Če pa bi me zanimala prodaja šamponov, katerih cena je višja od povprečne cene, pa bi moral uporabiti še filter

Oblika: Filter(Tabela, FilterIzraz)

Prodaja 2 = Calculate(
[Skupna Prodaja],
Filter(Prodaja, Prodaja[Prodajna Cena] > [Povprečna cena]))

V tem primeru imam v filtru (namišljeno) tabelo prodaja. V njej se nahaja stolpec s ceno posameznega šampona, stolpec ima ime prodajna cena. To ceno primerjam z mero Povprečna cena. Šele, ko Filter(), gredo vrednosti, ki ustrezajo pogoju, v Calculate, ki oceni mero [Skupna Prodaja], pod novimi pogoji.

Calculate() »povozi« vse ostale filtre, ker pa smo vrednosti primerjali z mero, smo mogli funkciji pomagati s filtrom.

DAX potrebujemo tudi za uporabo *Time Intelligence*. Pri tem mislimo na uporabo raznih funkcij, ki izračunajo tekoče vsote, prodajo tekočega leta, lanskega leta ipd. Lahko se uporabi funkcija TOTALYTD() ali pa več funkcij, v povezavi s CALCULATE(), kot so: DATESYTD, DATEADD in DATESINPERIOD. Pri uporabi *Time intelligence*, se priporoča uporaba hitrih mer – Quick Measure, kjer se nahaja par najpogostejše uporabljenih DAX funkcij.

2 primera izračuna Year-to-Date

Total YTD =
CALCULATE(
[TotalSales],
DATESYTD(OrderDate[Date]))

Total YTD = TOTALYTD([TotalSales],OrderDate[Date])

Quick measures

Calculation

Select a calculation

- Percentage difference from filtered value
- Sales from new customers
- Time intelligence
 - Year-to-date total
 - Quarter-to-date total
 - Month-to-date total
 - Year-over-year change
 - Quarter-over-quarter change
 - Month-over-month change
 - Rolling average

Fields

Search

- Order Date
- Product
- Sales
- Sales Territories

[Post an idea](#)

OK

Cancel

Za uporabo pa je nujno potrebno določiti *Date Table*, kar storimo v *Data View*.

The screenshot shows the Power BI interface with the 'Mark as date table' button highlighted. The 'Name' field is set to 'Order Date'. The 'Structure' tab is active, showing a table with columns: 'Order Date', 'Day Number Of Week', 'English Day Name of Week', and 'Day Number of'. The table contains data for January and February 2017.

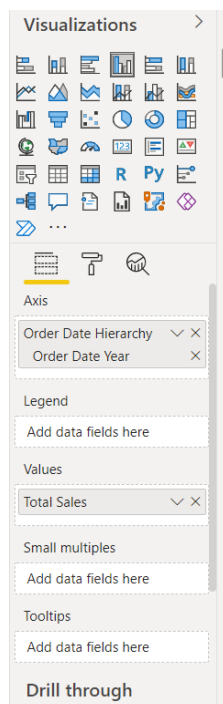
Order Date	Day Number Of Week	English Day Name of Week	Day Number of
ponedeljek, 02. januar 2017	2	Monday	
ponedeljek, 09. januar 2017	2	Monday	
ponedeljek, 16. januar 2017	2	Monday	
ponedeljek, 23. januar 2017	2	Monday	
ponedeljek, 30. januar 2017	2	Monday	
ponedeljek, 06. februar 2017	2	Monday	
ponedeljek, 13. februar 2017	2	Monday	
ponedeljek, 20. februar 2017	2	Monday	

Z uporabo hitrih mer, lahko naše količine preprosto potegnemo v polja, brez da bi napisali eno vrstico DAX-a.

Ko imamo naš model urejen (očiščen in imamo izračunane vse željene mere). Ga gremo vizualizirat. Vizualizacija se opravlja v *Report Viewu* (glej sliko 1 zgoraj). Mi smo si pri našem delu pomagali s prednaloženo predlogo slike (template). V kolikor pa sem uspel videti vaša poročila, boste vi svoja nalagali v že pripravljeno datoteko (vse ostalo pa je enako).

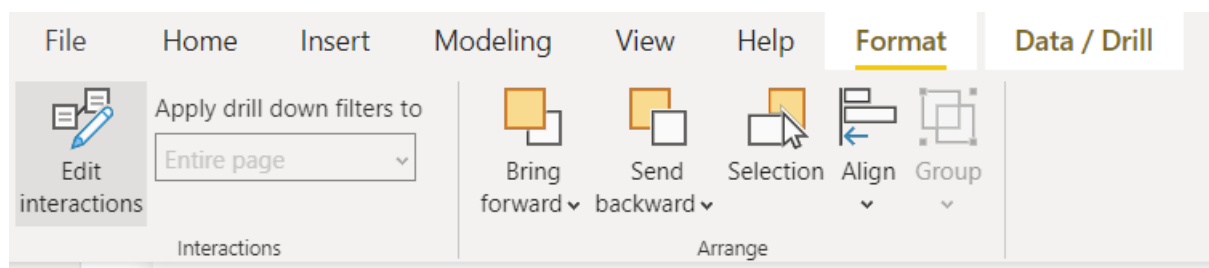
Takšen način uporabe se tudi priporoča, saj je PBI dokaj »lesen« kadar pride do ročnega vstavljanja teksta, slik ipd. To pa tudi pomeni, da boste že imeli določena mesta, tip in barve grafov.

Ponazoritve (Vizualizacije) lahko delamo na več načinov. Osebnost mi je najbolj všeč ta, da sam kliknem na izbrano grafiko (tabelo, KPI, line chart....) in potem nanjo potegnem vrednosti.

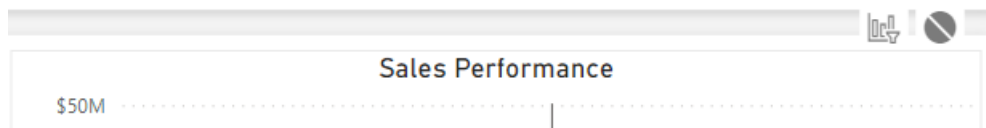


Pod ikonami ponazoritev, se nahajajo tri ikone, s katerimi določamo izgled in vsebino naše. Od leve proti desni si sledijo, polja (field), format in analitika. V polja vlečemo naše stolpce, vrstice in vrednosti, v format poskrbimo za izgled. Pri nekaterih grafih, pa se nam odprejo še dodatne možnosti analitike.

Vsi grafi v PBI, so kot privzeto, med seboj povezani in interaktivni. To pomeni, da ko kliknem na eno enega, se vsi spremenijo. Včasih si to želimo, ne pa vedno. To lahko uredimo tako, da kliknemo na izbrano vizualizacijo in izberemo možnost Format -> Edit Interactions.

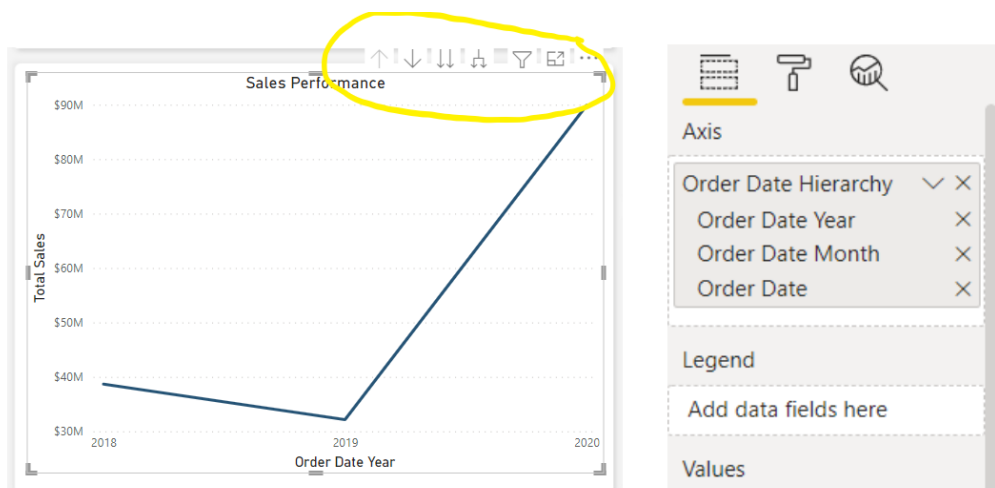


Po tem, se nam pri vsakem drugem grafu pokaže možnost interakcije (znak prepovedano parkiranje).



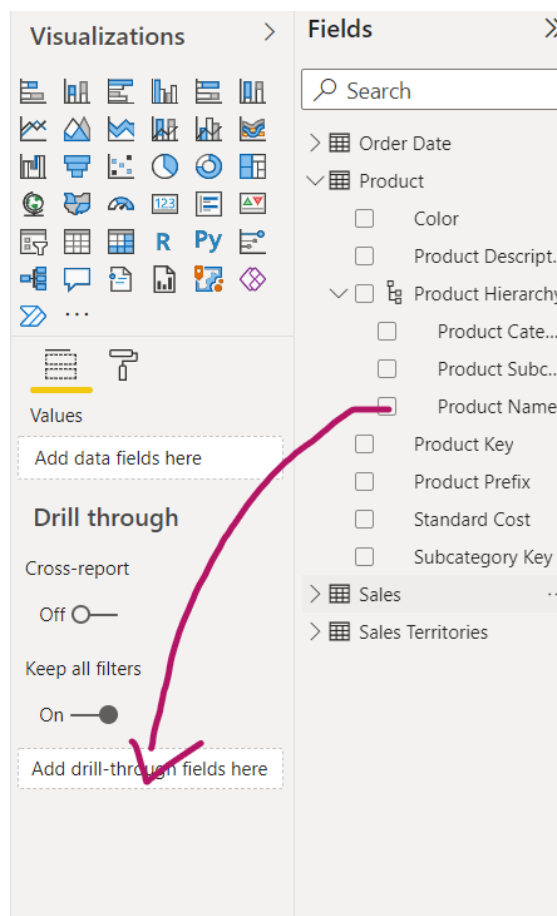
Če ga izberemo (se potemni), potem to pomeni, da graf ne bo več reagiral s prvotno označenim grafom.

Če smo v graf vstavili hierarhijo (v primeru na sliki časovno).

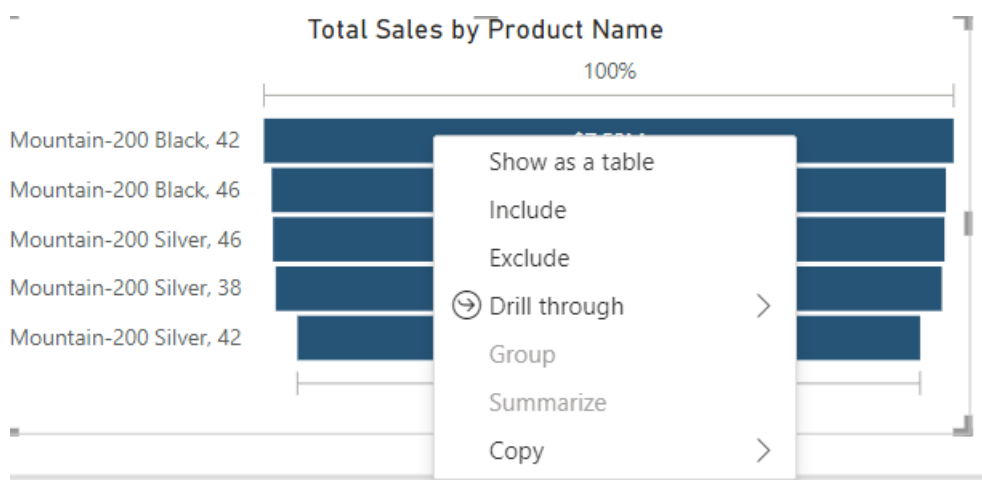


Se nam na grafu pokažejo dodatne puščice, ki nam omogočajo možnost vrtanja (drill-down). Vrtanje se vedno začne na najvišjem nadstropju hierarhije.

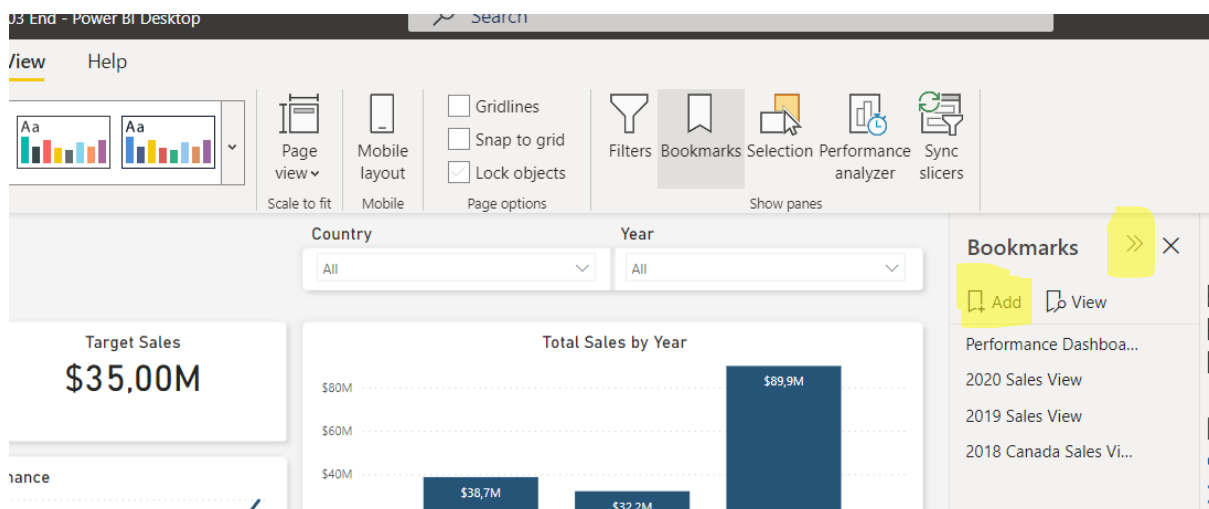
Če ne želimo preveč na eni strani poročila, lahko povezave povežemo tudi preko možnosti »Drill-Through«. Na grafu, ki ni na prvotni strani, polje drill-through potegnemo vrednost, po kateri želimo vrtati. Za vse ostalo poskrbi PBI avtomatsko.



Ko smo to naredili, se nam na grafu, na prvi strani, pokaže možnost, pri grafih, ki vsebujejo polje, po katerem vrtamo.



Za predstavitev pa je še posebej uporabna možnost zaznamkov (Bookmarks), saj si lahko z njimi ustvarimo tip PowerPoint predstavitve, ki ima dodano vrednost, da je interaktivna.



Vse kar moramo storiti je, da v zavihku *View* označimo ukaz *Bookmarks*. Ko si nastavimo željeno sliko, jo dodamo s tipko *Add*. Dodajamo lahko poljubno mnogo zaznamkov. V aplikaciji lahko poženeemo Report s pritiskom na sliki označeni puščici, je pa to še bolje storiti v on-line verziji, Power BI Service, kjer lahko gremo v celo zaslonski način. Za to pa je treba naše poročilo prvo naložiti na splet (lahko v My Workspace – tam gre vsem).

Dodatno: Pivot, UnPivot, Transpose

Kot se je izkazalo proti koncu naših srečanj, bo večina vašega dela (zaenkrat) potekala v PQE. Do datotek boste dostopali preko *Dataflows*. Postopek do uvažanja katerega koli vira v PBI je popolnoma enak. Gremo v PQE (stisnemo *Transform Data*) in izberemo tip vira. Pod *New Source* -> *More* najdemo *Dataflows* in se povežemo z željenim (sam se žal ne morem z vašim, zato tukaj ni slike).

Ko naložite tabelo, jo je treba zaradi velikosti in ker vas zanimajo samo podatki za vaš oddelek, prvo filtrirati. Ko jo sfiltriramo, pa se lotimo preoblikovanja tabele, kjer uporabljamo vse prijeme iz naših prvih srečanj, po potrebi pa dodamo katero od zgornjih funkcij.

Pivot / Unpivot -> RAZLIČNE VREDNOSTI

	1 ² ₃ 1994	1 ² ₃ 1995	1 ² ₃ 1996	1 ² ₃ 1997	1 ² ₃ 1998
1	286322	253787	155483	246491	130602

PIVOT

	1 ² ₃ Leto	1 ² ₃ Prodanih Enot
1	1994	286322
2	1995	253787
3	1996	155483
4	1997	246491
5	1998	130602

UNPIVOT

Pivot prestavi vrstice v stolpce, unpivot obratno. Pivot, bo če bo obstajalo več različnih vrednosti, npr. če bi imeli 2 vrstici vrednost 1994, obe združil v en stolpec.

Večkrat se uporablja tudi operacija *Transpose*, ki matriko transponira (obrne vrstice in stolpce). Sliši se podobno kot pivot/unpivot, s to razliko, da bi tukaj obe vrstici 1994, iz zgornjega primera, dobili svoj stolpec!