

Connecting & Shaping Data

- Κάνουμε import το csv αρχείο MavenMarket_Customers.
- Έλεγχω αν η πρώτη σειρά του csv αρχείου είναι header στα δεδομένα και παρατηρώ πως έγινε header αυτόματα.
- Μετονομάζω τον πίνακα σε Customers.

Name

- Ελέγχω να δω αν οι στήλες έχουν τα σωστά Data types. Αλλάζω την στήλη “customer_acct_num” και την στήλη “customer_postal_code” σε “text”.

customer_id	customer_acct_num	first_name	last_name	customer_address	customer_city	customer_state_province	customer_postal_code
-------------	-------------------	------------	-----------	------------------	---------------	-------------------------	----------------------

- Προσθέτω νέα στήλη “full_name” κάνοντας merge τις στήλες “first_name” και “last_name”, διαχωρίζοντας τες με ένα κενό “ ”.

full_name
Sheri Nowmer
Derrick Whelply
Jeanne Derry
Michael Spence
Maya Gutierrez

- Προσθέτω νέα στήλη “birth_year” κάνοντας extract το έτος από τη στήλη “birth_date” και το μετατρέπω σε “text”.

birth_year
1961
1915
1910
1969
1951

- Προσθέτω νέα στήλη “has_children”, δημιουργώντας conditional column με βάση το αν η στήλη “total_children” έχει το 0 ή όχι. Το “has_children” έχει “Y”, αν έχει η στήλη “total_children” έχει μεγαλύτερο αριθμό του 0, αλλιώς δείχνει “N”.

ABC 123	has_children
	Y
	Y
	Y
	Y

- Κάνουμε Close & Apply για τον πίνακα Customers.
- Κάνουμε import το csv αρχείο MavenMarket_Products.
- Έλεγχω αν η πρώτη σειρά του csv αρχείου είναι header στα δεδομένα και παρατηρώ πως έγινε header αυτόματα.
- Μετονομάζω τον πίνακα σε Products.
- Ελέγχω να δω αν οι στήλες έχουν τα σωστά Data types. Αλλάζω την στήλη “product_sku” σε “text” και την στήλη “product_retail_price” και “product_cost” σε “Decimal Number”.
- Χρησιμοποιώ το εργαλείο “statistics” για να βρω τις ξεχωριστές (distinct) τιμές της στήλης “product_brand” και “product_name”.

product_brand distinct values

111

product_name distinct values

1560

- Προσθέτω τη στήλη “discount_price” πολλαπλασιάζοντας τη στήλη “product_retail_price” * 0,9.

\$ discount_price
2,57
0,67
0,75
3,28

- Υπολογίζω το μέσο retail price ανά brand, χρησιμοποιώντας το εργαλείο Group By και δημιουργώ μια νέα στήλη “AVG Retail Price”.

product_brand	\$ AVG Retail Price
Washington	2,18
Jeffers	2,10
Blue Label	2,22
Green Ribbon	2,21

- Παρατηρώ ότι οι στήλες “recyclable” και “low_fat” έχουν πολλές null τιμές, τις οποίες θα αντικαταστήσω με 0.

Πριν την αντικατάσταση

recyclable	low_fat
null	null
null	1
1	1

Μετά την αντικατάσταση

recyclable	low_fat
0	0
0	1
1	1

- Κάνουμε Close & Apply για τον πίνακα Products.
- Κάνουμε import το csv αρχείο MavenMarket_Stores.
- Έλεγχω αν η πρώτη σειρά του csv αρχείου είναι header στα δεδομένα και παρατηρώ πως έγινε header αυτόματα.

- Μετονομάζω τον πίνακα σε Stores.
- Ελέγχω να δω αν οι στήλες έχουν τα σωστά Data types και δεν φαίνεται να υπάρχει κάποιο λάθος.
- Δημιουργώ νέα στήλη “full_address”, κάνοντας merge τις στήλες “store_city”, “store_state”, “store_country”, έχοντας σαν separator το comma και το κενό “, ”.

A ^B C full_address
2853 Bailey Rd, Acapulco, Guerrero
5203 Catanzaro Way, Bellingham, WA
1501 Ramsey Circle, Bremerton, WA
433 St George Dr, Camacho, Zacatecas

- Δημιουργώ νέα στήλη “area_code”, κάνοντας extract τους αριθμούς πριν την πρώτη παύλα “-” από τη στήλη “store_phone”.

A ^B C area_code
262
605
509

- Κάνουμε Close & Apply για τον πίνακα Stores.
- Κάνουμε import το csv αρχείο MavenMarket_Regions.
- Έλεγχω αν η πρώτη σειρά του csv αρχείου είναι header στα δεδομένα και παρατηρώ πως έγινε header αυτόματα.
- Μετονομάζω τον πίνακα σε Regions.
- Ελέγχω να δω αν οι στήλες έχουν τα σωστά Data types και δεν φαίνεται να υπάρχει κάποιο λάθος.
- Κάνουμε Close & Apply για τον πίνακα Regions.
- Κάνουμε import το csv αρχείο MavenMarket_Calendar.

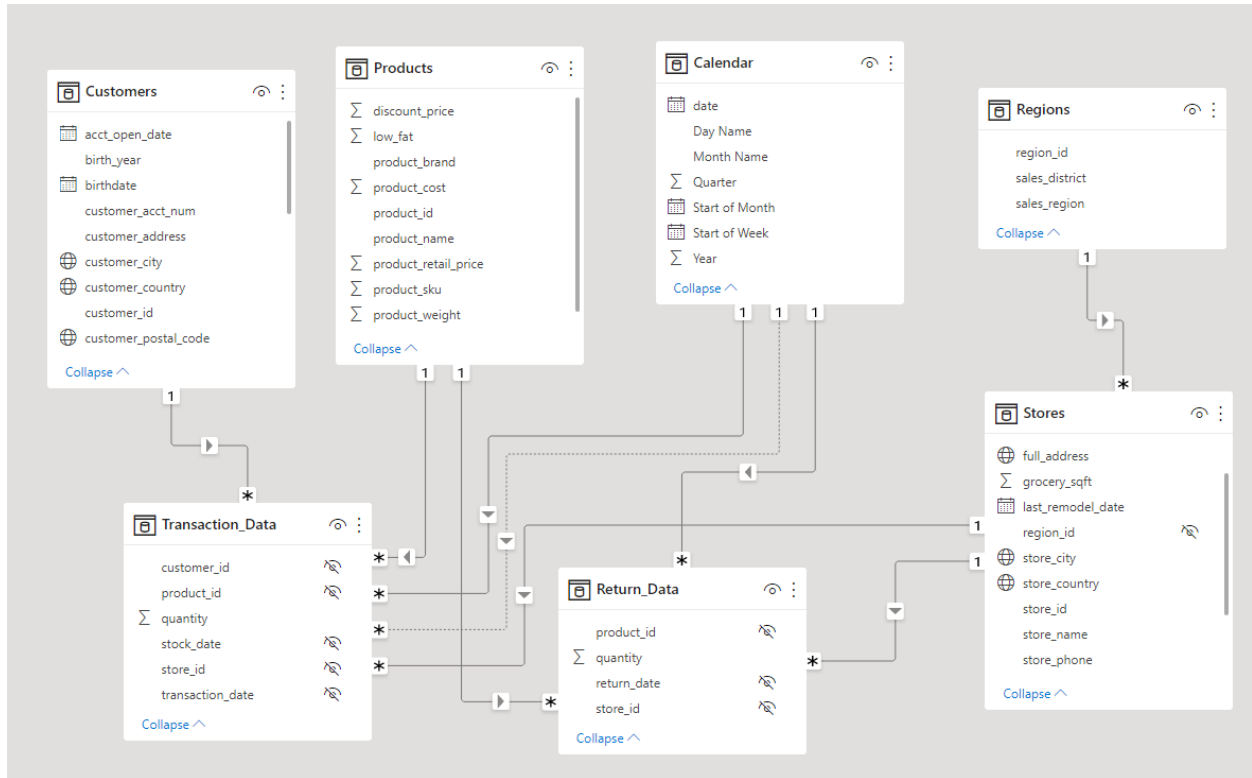
- Έλεγχω αν η πρώτη σειρά του csv αρχείου είναι header στα δεδομένα και παρατηρώ πως έγινε header αυτόματα.
- Μετονομάζω τον πίνακα σε Calendar.
- Ελέγχω να δω αν οι στήλες έχουν τα σωστά Data types και δεν φαίνεται να υπάρχει κάποιο λάθος.
- Χρησιμοποιώντας το εργαλείο date, δημιουργώ τις στήλες “Start of Week”, “Name of Day”, “Start of Month”, “Name of Month”, “Quarter of Year”, “Year”.

date	Start of Week	A ^B _C Day Name	Start of Month	A ^B _C Month Name	t ² ₃ Quarter	t ² ₃ Year
1/1/1997	29/12/1996	Wednesday	1/1/1997	January	1	1997
2/1/1997	29/12/1996	Thursday	1/1/1997	January	1	1997
3/1/1997	29/12/1996	Friday	1/1/1997	January	1	1997

- Κάνουμε Close & Apply για τον πίνακα Calendar.
- Κάνουμε import το csv αρχείο MavenMarket_Returns_1997-1998.
- Έλεγχω αν η πρώτη σειρά του csv αρχείου είναι header στα δεδομένα και παρατηρώ πως έγινε header αυτόματα.
- Μετονομάζω τον πίνακα σε Return_Data.
- Ελέγχω να δω αν οι στήλες έχουν τα σωστά Data types και δεν φαίνεται να υπάρχει κάποιο λάθος.
- Κάνουμε Close & Apply για τον πίνακα Return_Data.
- Τοποθετώ σε ένα φάκελο τα αρχεία που αφορούν τα transactions και τα κάνω import μέσω της επιλογής “Get Data by Folder”.
- Αφαιρώ τη στήλη “Source.Name” και μετονομάζω τον πίνακα σε “Transaction_Data”.
- Κάνουμε Close & Apply για τον πίνακα Transaction_Data.

Modelling Data

- Στη συνέχεια, δημιούργησα αυτές τις σχέσεις στο modeling tab και επιλέγω το “Hide from Model” για όλα τα πεδία που λειτουργούν σαν foreign keys.



DAX Measures

Calculated Columns

- Ενώ βρισκόμαστε στο Data View, προσθέτουμε νέα στήλη στον πίνακα calendar που θα δείχνει ποιες μέρες είναι σαββατοκύριακο.

```
Weekend = IF(('Calendar'[Day Name] = "Saturday" || 'Calendar'[Day Name] = "Sunday"), "Weekend", "Weekday")
```

- Προσθέτουμε νέα στήλη στον πίνακα calendar που θα δείχνει την τελευταία ημέρα του μήνα.

```
End of Month = ENDOFMONTH('Calendar'[date])
```

- Προσθέτουμε νέα στήλη στον πίνακα customers που θα δείχνει την ηλικία των πελατών.

```
Current Age = DATEDIFF(Customers[birthdate], TODAY(), YEAR)
```

- Προσθέτουμε νέα στήλη στον πίνακα customers που θα δείχνει αν οι πελάτες είναι σε high priority (αν κατέχουν σπίτι και έχουν χρυσή κάρτα μέλους) .

```
Priority = IF(Customers[homeowner] = "Y" && Customers[member_card] = "Golden", "High", "Standard")
```

- Προσθέτουμε νέα στήλη στον πίνακα customers που θα επιστρέφει τα 3 πρώτα γράμματα της χώρας του πελάτη με κεφαλαία.

```
Short_Country = UPPER(LEFT(Customers[customer_country],3))
```

- Προσθέτουμε νέα στήλη στον πίνακα customers που θα δείχνει τον αριθμό σπιτιού του πελάτη (χαρακτήρες πριν το πρώτο κενό στη στήλη "customer_address").

```
House Number = LEFT(Customers[customer_address], SEARCH(" ", Customers[customer_address]))
```

- Προσθέτουμε νέα στήλη στον πίνακα products που θα δείχνει το price range των προϊόντων.
(High>3, Mid>1, Low)

```
Price_Tier = IF(Products[product_retail_price] > 3, "High", (IF(Products[product_retail_price] > 1, "Mid" , "Low")))
```

- Προσθέτουμε νέα στήλη στον πίνακα stores που θα υπολογίζει πόσα χρόνια πέρασαν από την τελευταία φορά που έγινε remodel.

```
Years_Since_Remodel = DATEDIFF(Stores[last_remodel_date], TODAY(), YEAR)
```

Measures

- Measure στον πίνακα Transaction_Data για υπολογισμό της ποσότητας που πουλήθηκε.

```
Quantity_Sold = SUM(Transaction_Data[quantity])
```

- Measure στον πίνακα Return_Data για υπολογισμό της ποσότητας που επιστράφηκε.

```
Quantity_Returned = SUM(Return_Data[quantity])
```

- Measure στον πίνακα Transaction_Data για υπολογισμό των transactions.

```
Total Transactions = COUNT(Transaction_Data[quantity])
```


- Measure στον πίνακα Return_Data για υπολογισμό των returns.

`Total Returns = COUNT(Return_Data[quantity])`

- Measure στον πίνακα Return_Data για υπολογισμό του ποσοστού των επιστροφών προς τις πωλήσεις.

`Return Rate = [Quantity_Returned] / [Quantity_Sold]`

- Measure στον πίνακα Transaction_Data για υπολογισμό των transaction που έγιναν τα σαββατοκύριακα.

`Weekend Transactions = CALCULATE([Total Transactions], 'Calendar'[Weekend] = "Weekend")`

- Measure στον πίνακα Transaction_Data για υπολογισμό του ποσοστού των transaction που έγιναν τα σαββατοκύριακα προς τα συνολικά transaction.

`% Weekend Transactions = [Weekend Transactions] / [Total Transactions]`

- Measure στον πίνακα Transaction_Data για υπολογισμό του συνολικού revenue(quantity * product_retail_price).

`Total Revenue = SUMX(Transaction_Data, Transaction_Data[quantity] * RELATED(Products[product_retail_price]))`

- Measure στον πίνακα Transaction_Data για υπολογισμό του συνολικού cost(quantity * product_cost).

`Total Cost = SUMX(Transaction_Data, Transaction_Data[quantity] * RELATED(Products[product_cost]))`

- Measure στον πίνακα Transaction_Data για υπολογισμό του συνολικού profit.

$$\text{Total Profit} = [\text{Total Revenue}] - [\text{Total Cost}]$$

- Measure στον πίνακα Transaction_Data για υπολογισμό του profit margin.

$$\text{Profit Margin} = [\text{Total Profit}] / [\text{Total Revenue}]$$

- Measure στον πίνακα Products για υπολογισμό των μοναδικών (unique) προϊόντων.

$$\text{Unique Products} = \text{DISTINCTCOUNT}(\text{Products}[\text{product_name}])$$

- Measure στον πίνακα Transaction_Data για υπολογισμό Year to Date Revenue.

$$\text{YTD Revenue} = \text{CALCULATE}([\text{Total Revenue}], \text{DATESYTD}('Calendar'[\text{date}]))$$

- Measure στον πίνακα Transaction_Data για υπολογισμό του revenue κάθε 60ήμερου (rolling).

$$60\text{-day Rolling Revenue} = \text{CALCULATE}([\text{Total Revenue}], \text{DATESINPERIOD}('Calendar'[\text{date}], \text{MAX}('Calendar'[\text{date}]), -60, \text{DAY}))$$

- Measure στον πίνακα Transaction_Data για υπολογισμό Year to Date Revenue.

$$\text{YTD Revenue} = \text{CALCULATE}([\text{Total Revenue}], \text{DATESYTD}('Calendar'[\text{date}]))$$

- Measure στον πίνακα Transaction_Data για τα transaction του προηγούμενου μήνα.

$$\text{Last Month Transactions} = \text{CALCULATE}([\text{Total Transactions}], \text{DATEADD}('Calendar'[\text{date}], -1, \text{MONTH}))$$

- Measure στον πίνακα Transaction_Data για υπολογισμό του revenue του προηγούμενου μήνα.

```
Last Month Revenue = CALCULATE([Total Revenue], DATEADD('Calendar'[date],-1,MONTH))
```

- Measure στον πίνακα Transaction_Data για υπολογισμό του profit του προηγούμενου μήνα.

```
Last Month Profit = CALCULATE([Total Profit], DATEADD('Calendar'[date],-1,MONTH))
```

- Measure στον πίνακα Return_Data για υπολογισμό των returns του προηγούμενου μήνα.

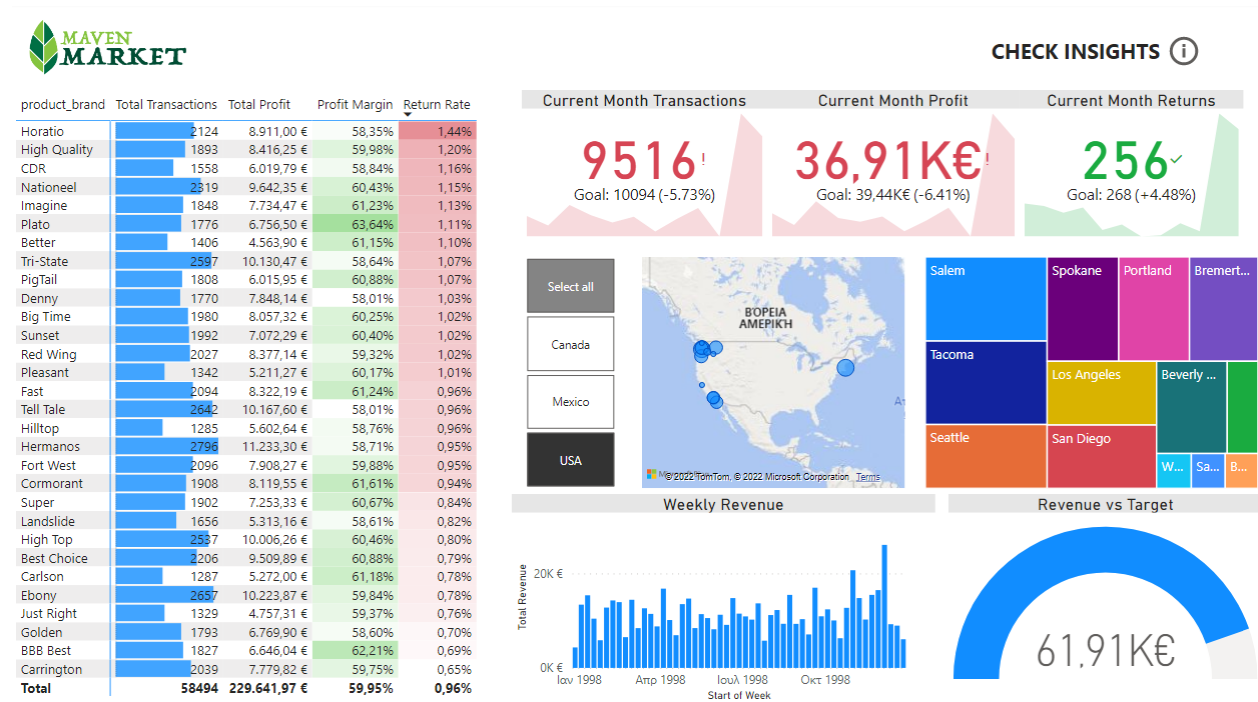
```
Last Month Returns = CALCULATE([Total Returns], DATEADD('Calendar'[date],-1,MONTH))
```

- Measure στον πίνακα Transaction_Data για υπολογισμό του Target revenue (5% αύξηση επί του προηγούμενου μήνα).

```
Revenue Target = [Last Month Revenue] * 1.05
```

Visual Report

Δημιούργησα αυτό το διαδραστικό dashboard.



Αν πατήσουμε στο button “CHECK INSIGHTS”, θα δούμε και αυτά τα συμπεράσματα.

1. Portland hits 1000 sales in December!
2. San Francisco is growing the fastest!
3. Los Angeles has the most decreasing popularity!
4. We need to sell more products from Plato, as it has the biggest profit margin!
5. Horatio products have the most returns!