Γεώργιος Παπουτσάκης 8200137

9/1/2025

# Εισαγωγή

Το συγκεκριμένο αρχείο word περιέχει την αναλυτική περιγραφή της διαδικασίας που ακολουθήθηκε για την υλοποίηση της εργασίας στο μάθημα της Επιχειρηματικής Ευφυΐας και Ανάλυσης Μεγάλων Δεδομένων. Εάν χρειάζεται επιπλέον οποιοδήποτε από τα υπόλοιπα αρχεία που αναφέρονται εντός του word, επικοινωνήστε μαζί μου και θα σας τα στείλω άμεσα.

# Περιεχόμενα

[Εισαγωγή 1](#_Toc187300168)

[Περιεχόμενα 1](#_Toc187300169)

[Επιλογή Dataset 2](#_Toc187300170)

[Dataset Cleaning 2](#_Toc187300171)

[Δημιουργία Βάσης 5](#_Toc187300172)

[ETL 7](#_Toc187300173)

[Δημιουργία Κύβου 13](#_Toc187300174)

[Visualizations 18](#_Toc187300175)

[Data mining 21](#_Toc187300176)

# Επιλογή Dataset

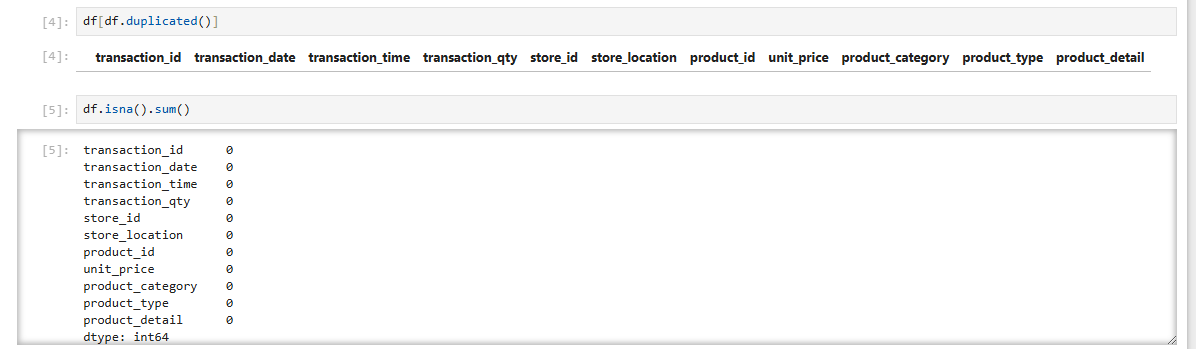
Για την εύρεση του dataset χρησιμοποιήθηκε η ιστοσελίδα kaggle.com και επιλέχθηκε το dataset από το παρακάτω link:

<https://www.kaggle.com/datasets/ahmedabbas757/coffee-sales>

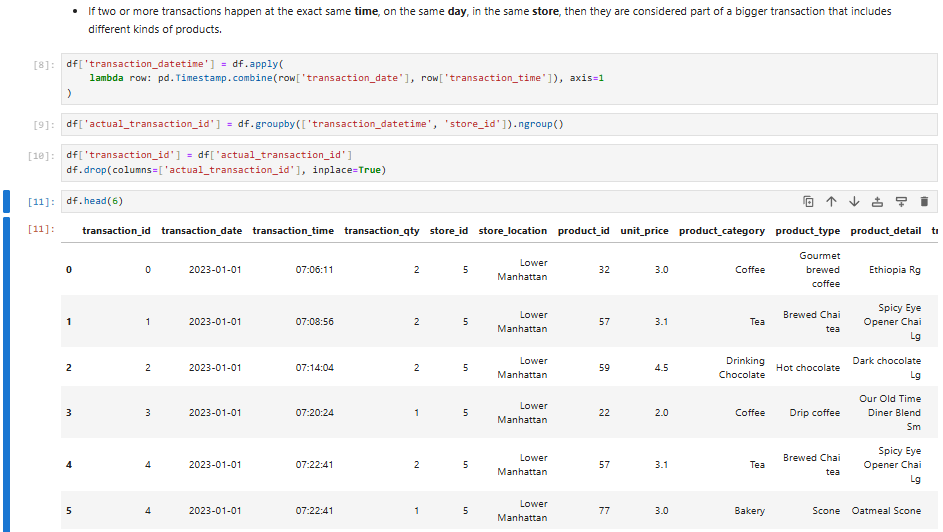
Αναφέρεται σε συναλλαγές που πραγματοποιήθηκαν σε 3 καταστήματα καφέ την περίοδο 1/1/2023 – 30/6/2023. Αποτελείται από 149.116 γραμμές και 11 στήλες ('transaction\_id', 'transaction\_date', 'transaction\_time', 'transaction\_qty', 'store\_id', 'store\_location', 'product\_id', 'unit\_price', 'product\_category', 'product\_type', 'product\_detail').

# Dataset Cleaning

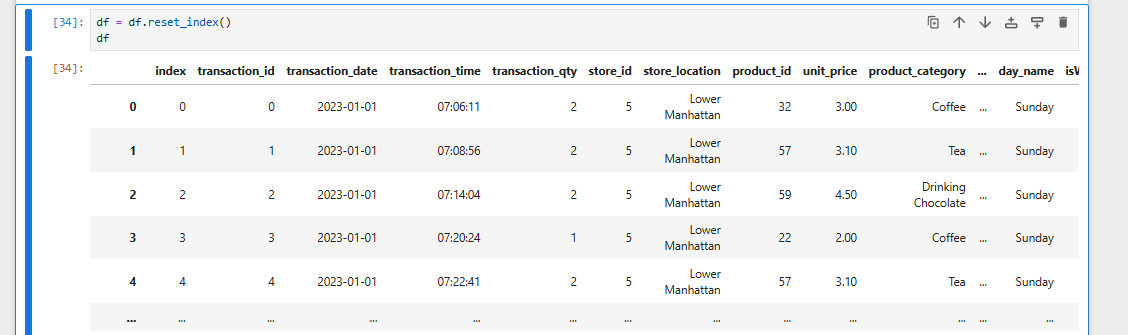
Για τον καθαρισμό του dataset χρησιμοποιήθηκε python σε jupyter notebook. Πρώτα από όλα ελέγχθηκε αν υπήρχαν διπλές εγγραφές και Null τιμές. Δεν υπήρξαν τέτοιες περιπτώσεις στο dataset.



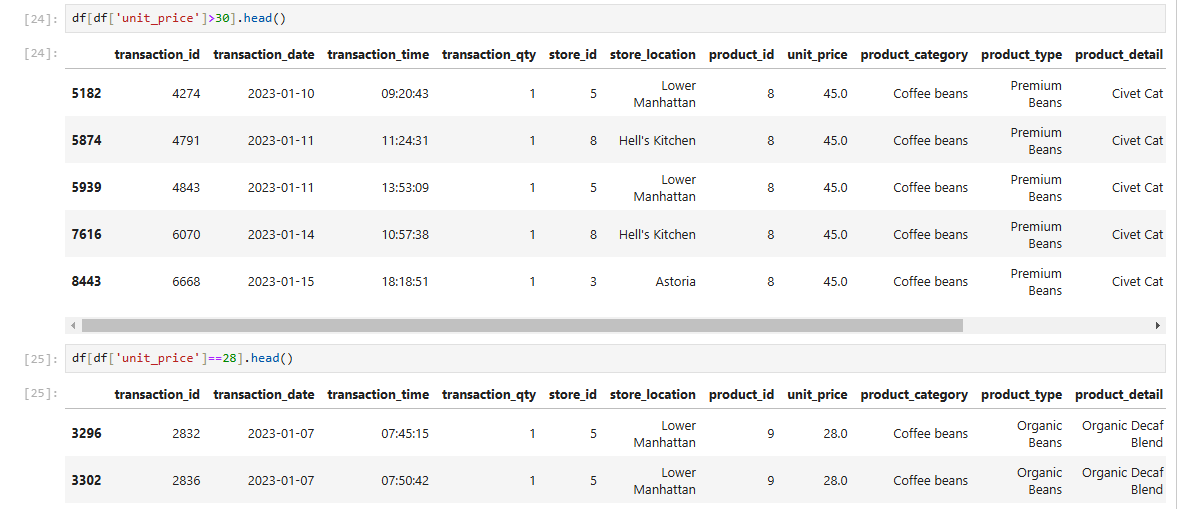
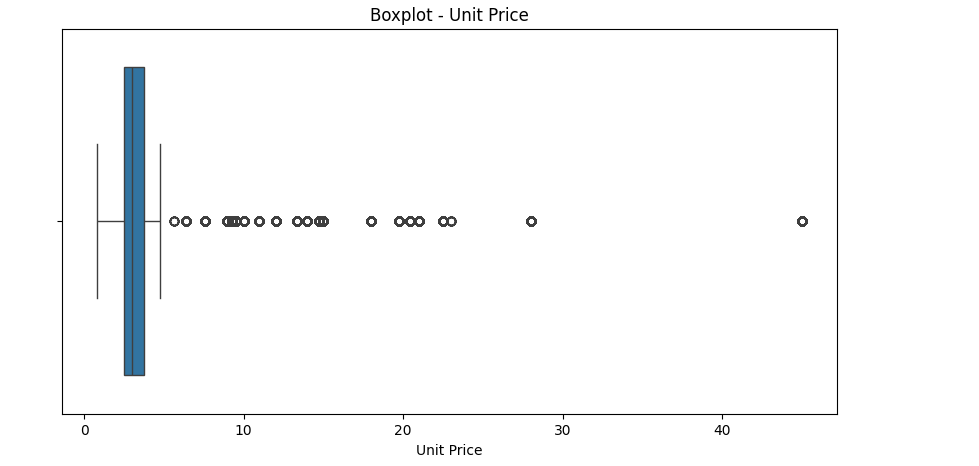
Το επόμενο βήμα του καθαρισμού αφορούσε την διόρθωση των transaction\_id. Παρατηρήθηκε πως κάθε γραμμή του πίνακα είχε διαφορετικό transaction\_id ακόμη και αν αφορούσε την ίδια συναλλαγή. Επομένως έγινε αντικατάσταση των transaction\_id για τις συναλλαγές που έγιναν ακριβώς την ίδια μέρα και ώρα στο ίδιο μαγαζί με ένα νέο κοινό transaction\_id.



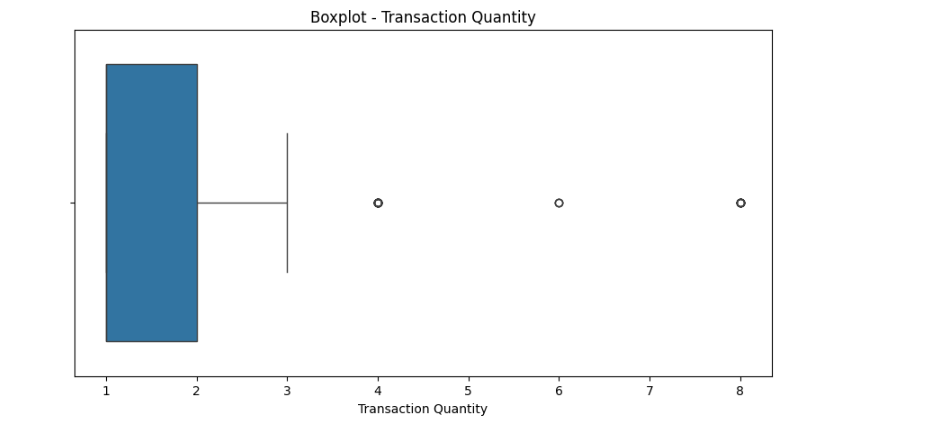
Επίσης, θα δημιουργηθεί μία νέα στήλη index ώστε να υπάρχει κάποιο μοναδικό στοιχείο για κάθε εγγραφή του αρχείου. Δεν είναι απαραίτητο για την python αλλά θα χρειαστεί ως πρωτεύον κλειδί όταν φτιαχτούν οι πίνακες της βάσης.



Ο επόμενος έλεγχος αφορούσε τα outliers. Αρχικά φαίνεται πως υπάρχουν κάποιες outliers στις τιμές των συναλλαγών. Ωστόσο κοιτάζοντας πιο συγκεκριμένα τις συγκεκριμένες συναλλαγές παρατηρούμε ότι δεν αφορούν συνηθισμένα προϊόντα καφέ, αλλά κάποια premium. Επομένως αποφασίζω να τις κρατήσω χωρίς να τις τροποποιήσω καθώς οι τιμές φαίνονται λογικές για τα συγκεκριμένα προϊόντα.



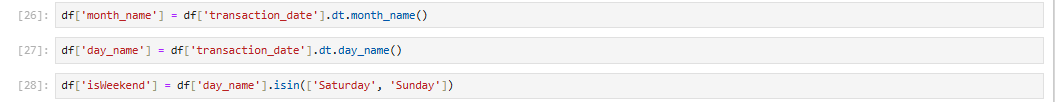
Σχετικά με την ποσότητα, και πάλι φαίνεται να υπάρχουν κάποιες τιμές που ξεχωρίζουν αλλά δεν απέχουν από την ανθρώπινη πραγματικότητα, επομένως αποφασίζω να τις κρατήσω και αυτές.



Δεν υπήρξαν συναλλαγές με μηδενικές ή αρνητικές τιμές στην τιμή ή την ποσότητα. Γενικότερα το dataset ήταν πολύ καθαρό εάν εξαίρεσουμε τις τιμές των transaction\_id.

Στην συνέχεια υπολογίστηκαν κάποιες νέες τιμές και προστέθηκαν ως νέες στήλες, κυρίως με βάση την ώρα και μέρα που πραγματοποιήθηκε η συναλλαγή καθώς επίσης και οι γεωγραφικές συντεταγμένες για την τοποθεσία των καταστημάτων καθώς γνωρίζαμε μόνο την περιοχή. Οι νέες στήλες είναι: 'index', 'time\_of\_day', 'month\_name', 'day\_name', 'isWeekend', 'day', 'month', 'year', 'hour', 'minutes', 'seconds', 'latitude', 'longitude'.

Ενδεικτικά ο υπολογισμός κάποιων από αυτών.



Τελευταίο βήμα στην python ήταν να αποθηκεύσω το ανανεωμένο dataframe ως νέο CSV αρχείο.



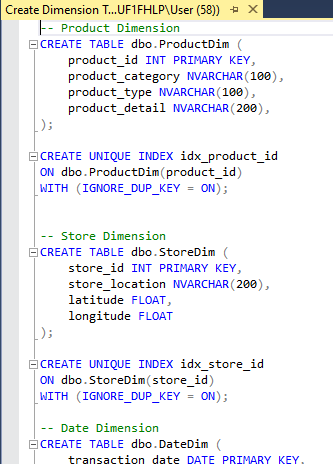
Για τα επόμενα βήματα της εργασίας χρειάστηκε να εγκατασταθούν και χρησιμοποιηθούν τα εργαλεία:

* SQL Server
* SQL Server Management Studio
* SQL Server Integration Services
* SQL Server Analysis Services
* Visual Studio
* Power Bi Desktop

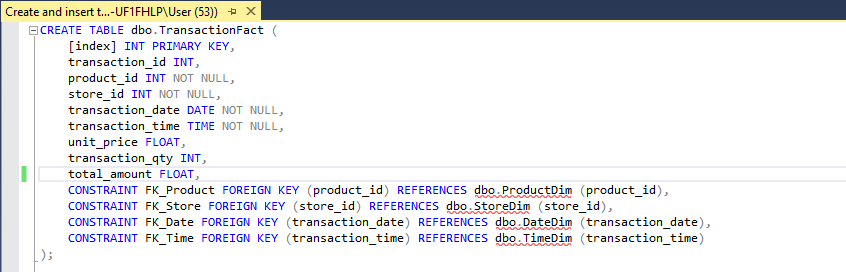
# Δημιουργία Βάσης

Για την δημιουργία της βάσης δεδομένων χρησιμοποιήθηκε τοπικά το SQL Server Management Studio. Σκοπός είναι να δημιουργηθεί ένα Star Schema για τα δεδομένα. Ο πίνακας Γεγονότων θα αναφέρεται προφανώς στις συναλλαγές ενώ οι πίνακες Διάστασης θα αναφέρονται στο Κατάστημα, στο Προϊόν, στην Ημέρα και στην Ώρα.

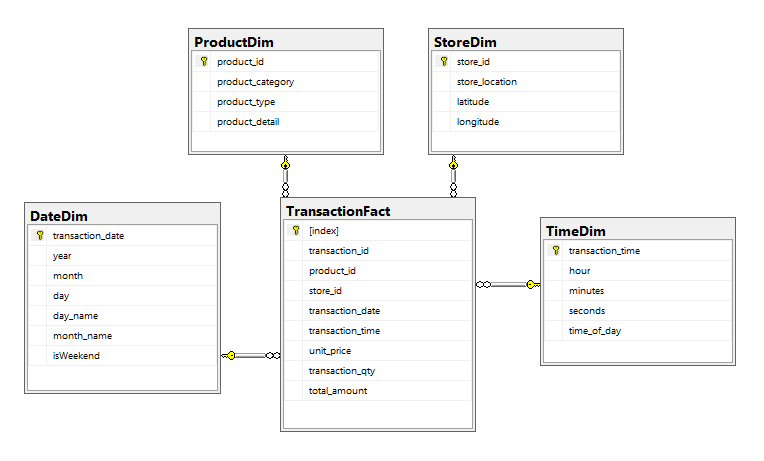
Εφόσον έχουν αποφασιστεί ποιοι πίνακες πρέπει να υπάρχουν στην βάση για το Star Schema, πρέπει πλέον και να δημιουργηθούν. Αυτό θα γίνει με sql queries εντός του Management Studio. Σε κάθε πίνακα θα προστεθεί επιπλέον ένας index που θα εμποδίζει το πρόγραμμα να σταματάει βγάζοντας σφάλμα, όταν επιχειρήσουμε να προσθέσουμε μια εγγραφή με πρωτεύον κλειδί που υπάρχει ήδη. Φυσικά όμως δεν θα προστίθεται η εγγραφή, απλά θα αποφεύγουμε το crash (θα φανεί χρήσιμο εισαγωγή των δεδομένων-ETL).



Στην συνέχεια δημιουργούμε και τον Fact Table.

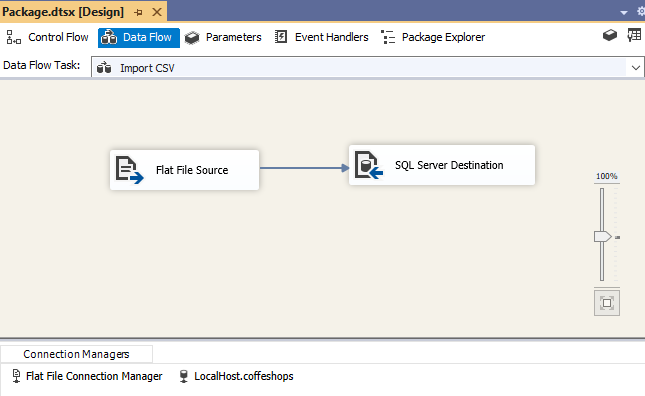


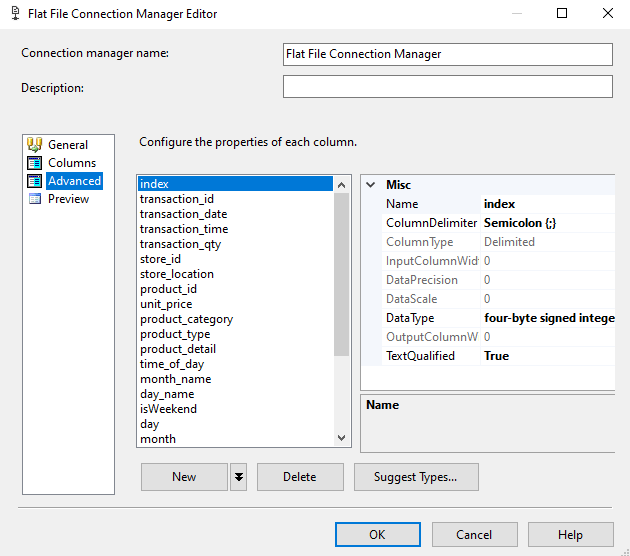
Η τιμή του προϊόντος βρίσκεται εντός του Fact Table γιατί όπως παρατηρήθηκε ένα προϊόν μπορεί να έχει διαφορετική τιμή ανάλογα την ημερομηνία που πωλήθηκε.

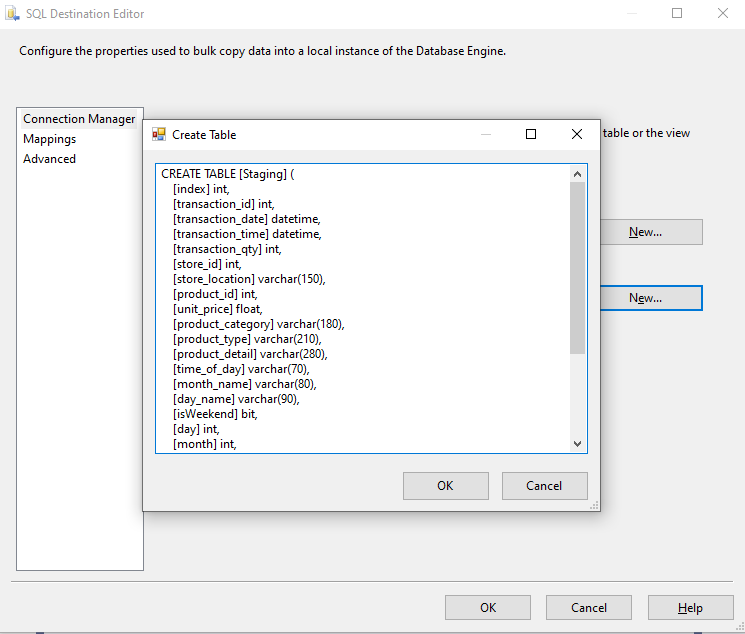


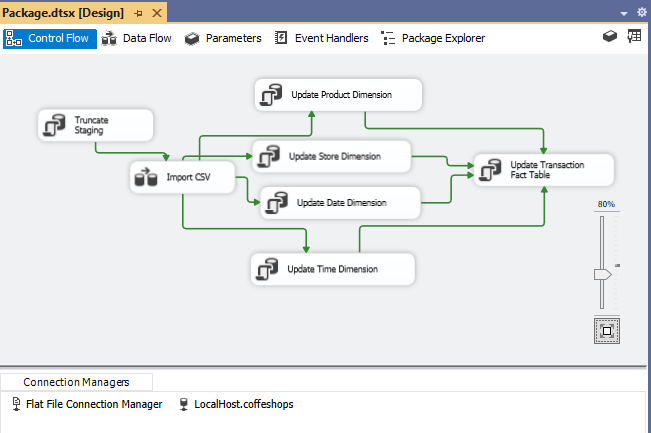
# ETL

Για το ETL χρησιμοποιήθηκε το SQL Server Integration Services (SSIS) μέσω του Visual Studio.

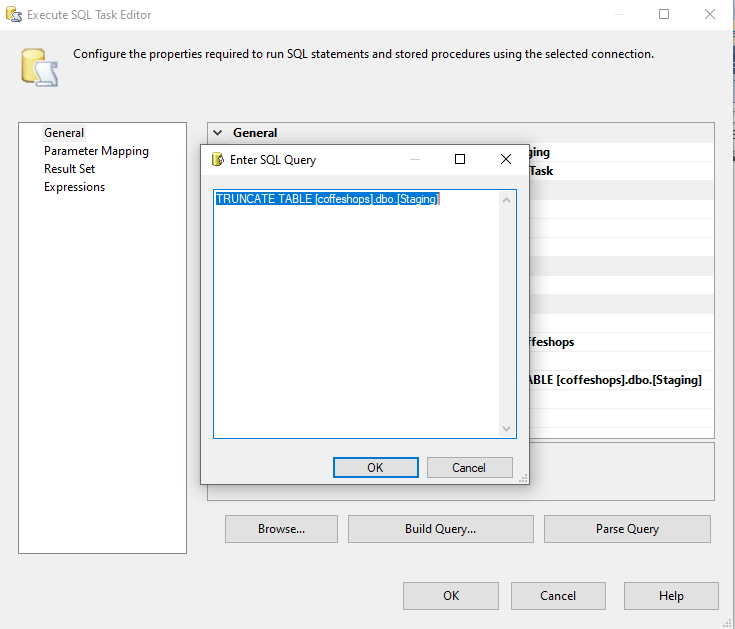
Ως πρώτο βήμα απαιτείται η φόρτωση των δεδομένων από το CSV αρχείο εντός της βάσης. 

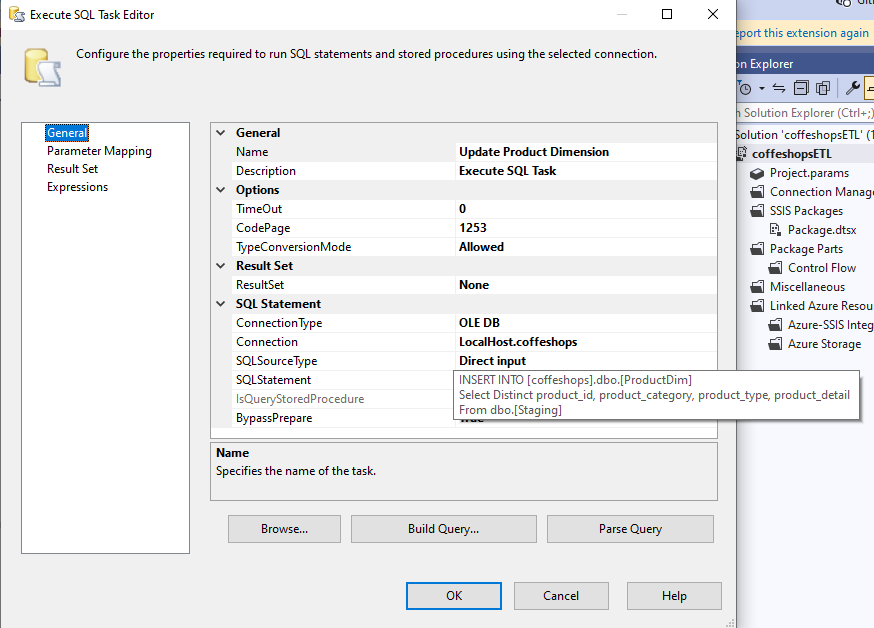
Στο κουτάκι του Flat File Source ρυθμίζουμε τον Connection Manager(τοποθεσία αρχείου και τύποι δεδομένων).

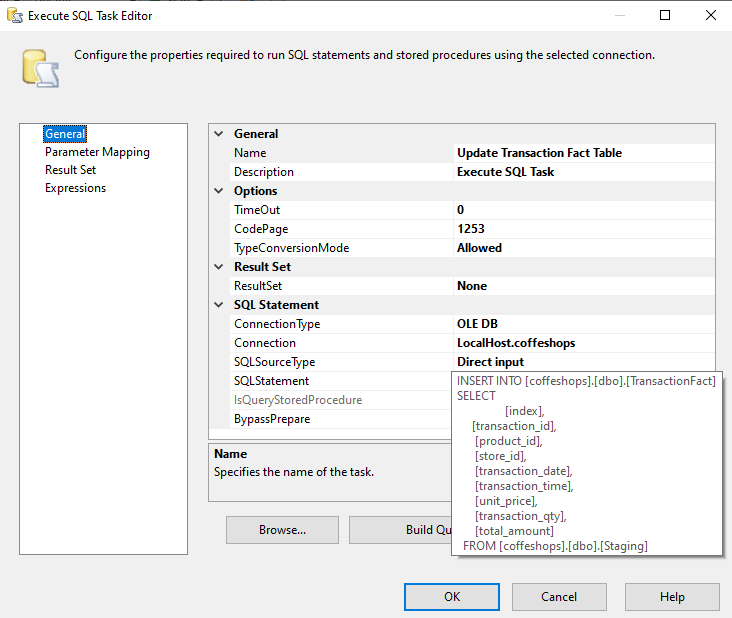
Στο κουτάκι του SQL Server Destination ρυθμίζουμε τον Connection Manager (στοιχεία βάσης, δημιουργία βοηθητικού πίνακα, map των στηλών του CSV με τις στήλες του βοηθητικού πίνακα). Ο βοηθητικός πίνακας θα ονομάζεται Staging και έχει ως σκοπό την προσωρινή φιλοξενία των εγγραφών στην βάση μέχρι να μεταφερθούν στου κατάλληλους πίνακες.

Το τελικό Control Flow θα είναι το παρακάτω

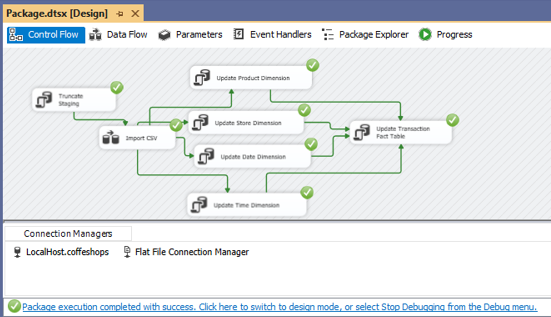
Αρχικά αδειάζουμε τον πίνακα Staging. Αυτό γίνεται ώστε εάν θελήσουμε να επαναλάβουμε την διαδικασία (στην πράξη αυτό έγινε πολλές φορές), οι ίδιες γραμμές να μην γίνονται εισαγωγή στους κύριους πίνακες πολλαπλές φορές.



Στη συνέχεια, θέλουμε να ανανεώσουμε τους πίνακες διάστασης με τις κατάλληλες γραμμές που περιέχονται εντός του Staging. Αυτό θα γίνει με το ακόλουθο query για τον πίνακα διάστασης του Προϊόντος (με αντίστοιχο τρόπο και για τους υπόλοιπους πίνακες διάστασης). Εδώ χρησιμεύει ο index που προσθέσαμε κατά την δημιουργία των πινάκων ώστε να μας προστατεύει από τα crashes σε περίπτωση διπλού πρωτεύοντος κλειδιού.

Τελευταίο βήμα είναι η ενημέρωση του Fact Table το οποίο θα γίνει με το ακόλουθο query.

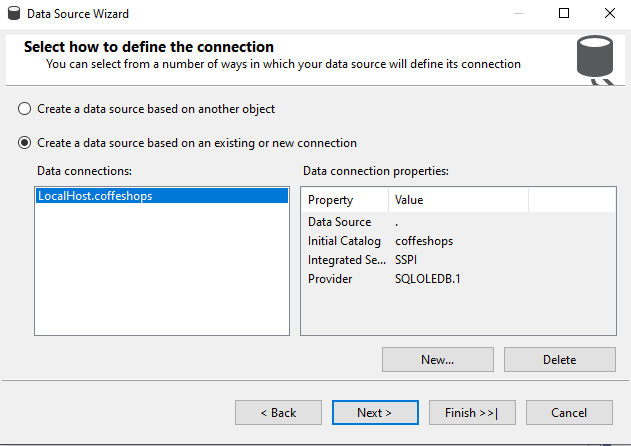
Πλέον, εάν τρέξουμε το Flow, εκτελείται με επιτυχία.



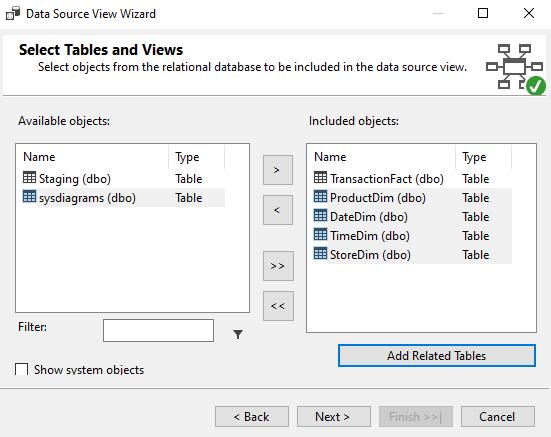
Με αυτόν τον τρόπο, μπορούμε αυτόματα να φορτώνουμε τα δεδομένα από οποιοδήποτε CSV αρχείο της ίδιας μορφής προς την τοπική βάση του SQL Server.

# Δημιουργία Κύβου

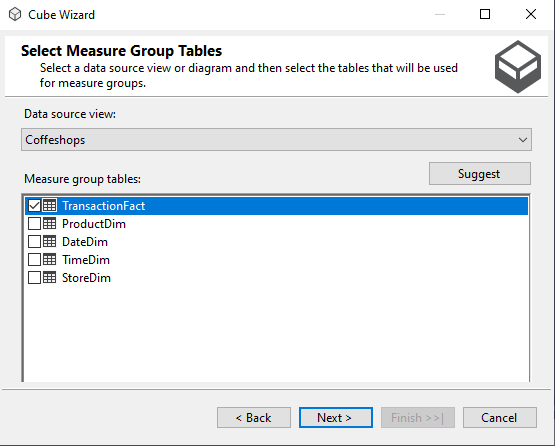
Για την δημιουργία του κύβου χρησιμοποιήθηκε το SQL Server Analysis Services (SSAS) μέσω του Visual Studio.

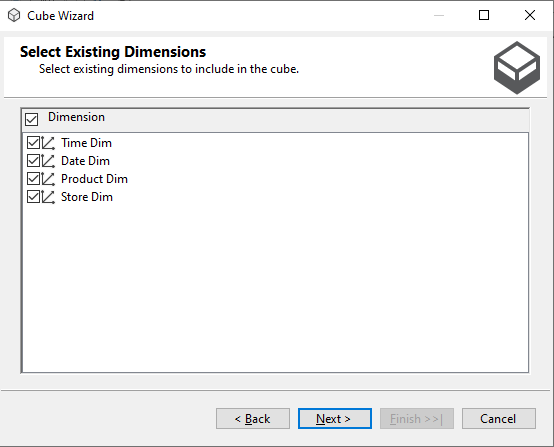
Αρχικά, δημιουργούμε το Data Source

Έπειτα δημιουργούμε το Source View επιλέγοντας μόνο πίνακες της βάσης που ανήκουν στο Star Schema.

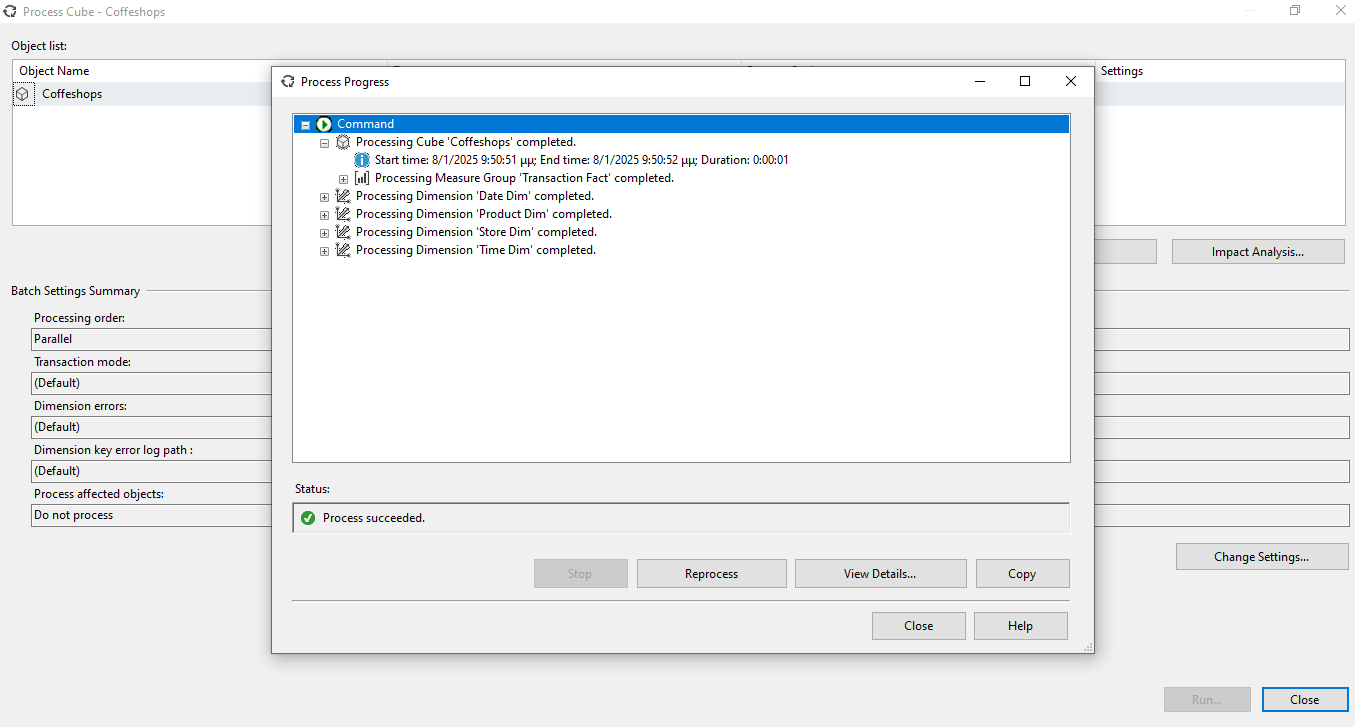


Δημιουργούμε τον κύβο επιλέγοντας μόνο τον Fact Table ως measure group table και οι διαστάσεις του θα έχουν προεπιλεγεί από τους πίνακες Διάστασης.

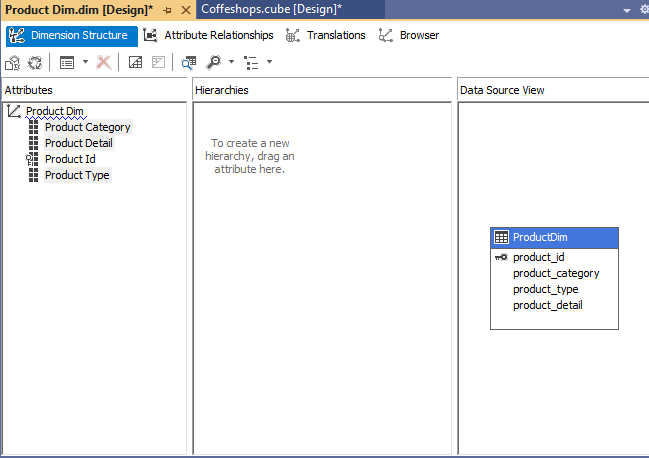




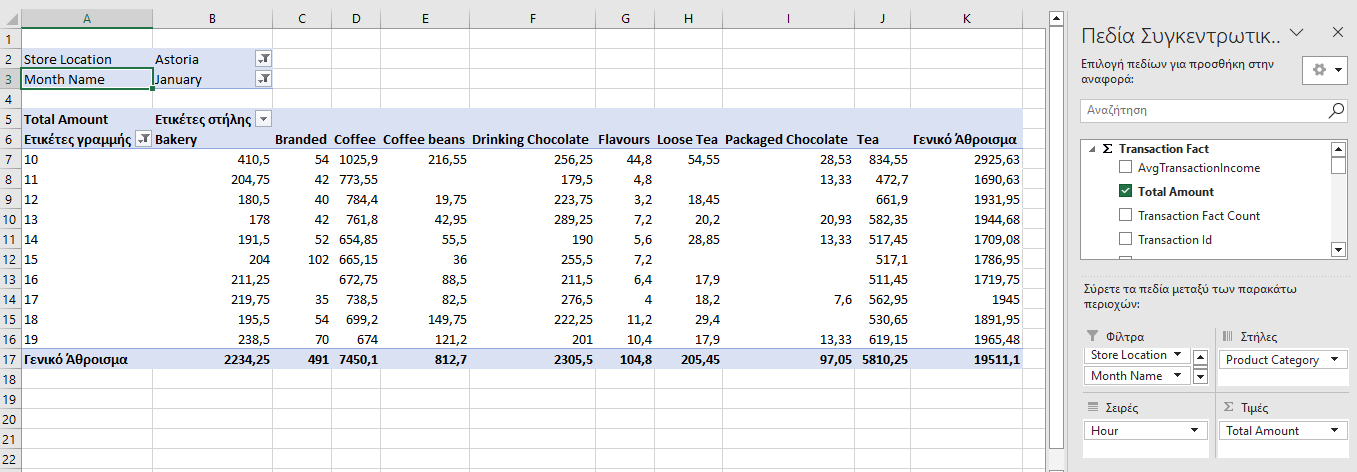
Κάνω process τον κύβο.



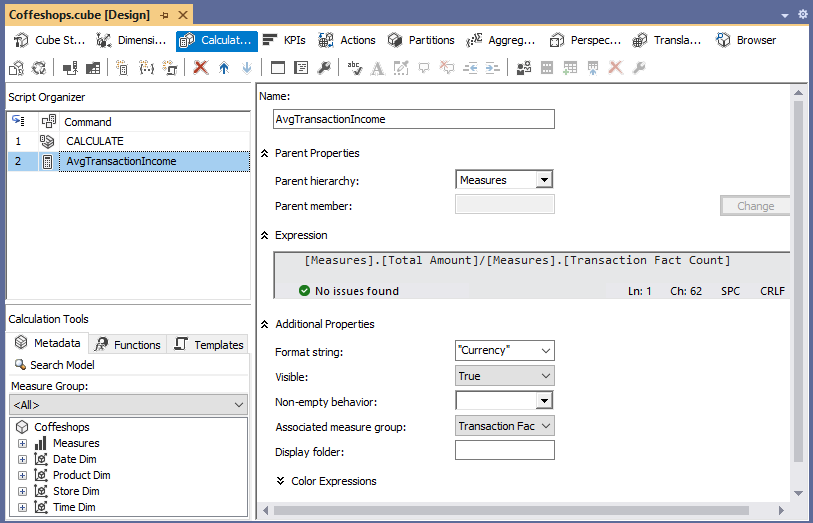
Προσθέτω τα attributes σε όλους τους πίνακες διάστασης για να μπορώ να τα χρησιμοποιήσω στις μετρικές εάν χρειαστεί και ξανακάνω process τον κύβο.

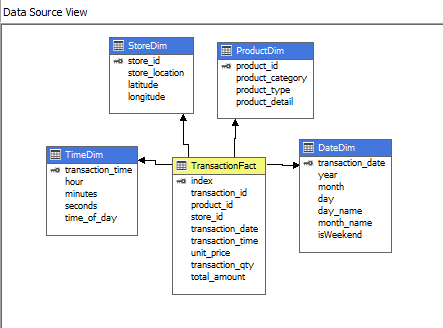


Μπορούμε μέσα από τον browser να περιηγηθούμε στον κύβο εκτελώντας διάφορα περίπλοκους συνδυασμούς των μετρικών και των διαστάσεων. Για παράδειγμα τα έσοδα για τις κατηγορίες των προϊόντων τον μήνα Ιανουάριο στο κατάστημα της Astoria τις ώρες 12.00 με 19.59.



Τέλος μπορούμε να δημιουργήσουμε και κάποιες μετρικές όπως την μέση τιμή συναλλαγής.





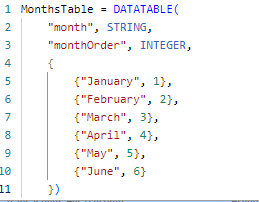
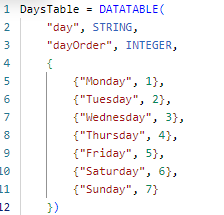
# Visualizations

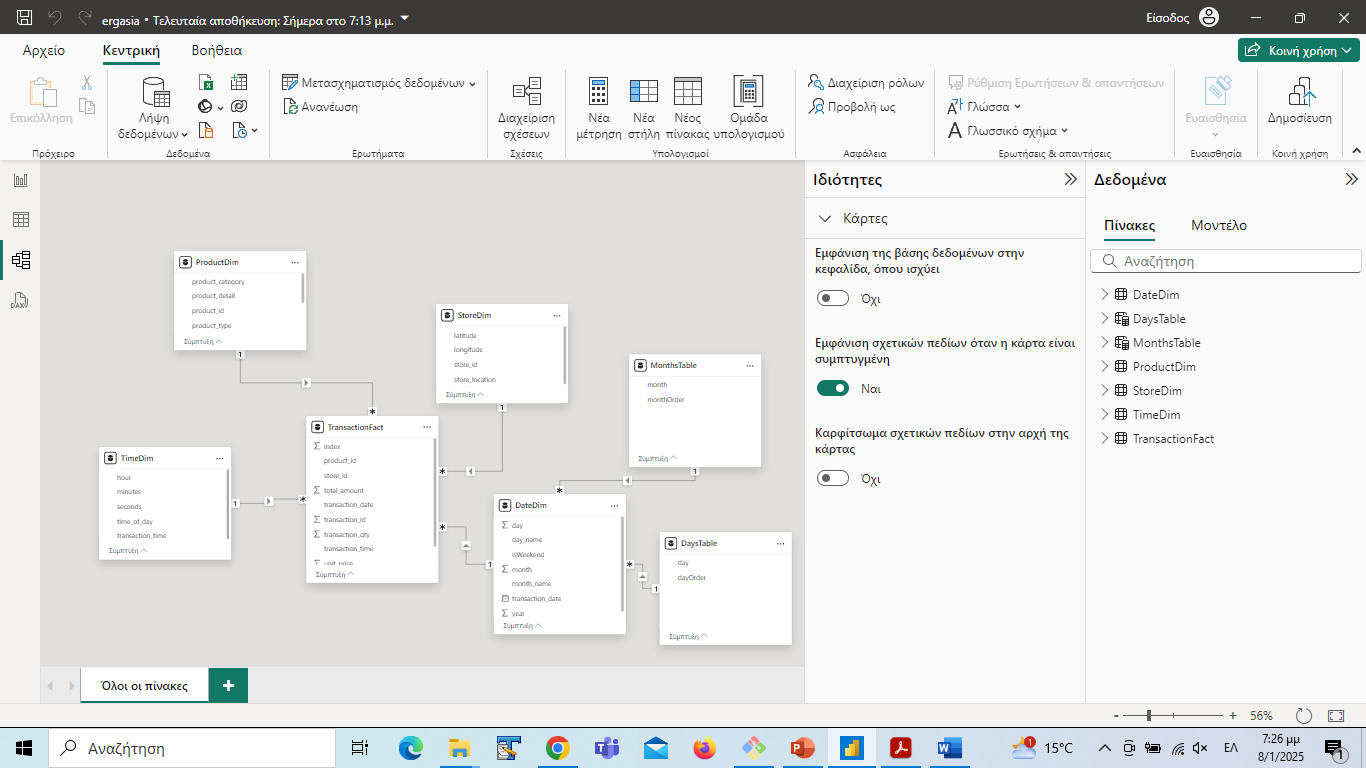
Για τις οπτικοποιήσεις χρησιμοποίησα το Power Βi Desktop. Η σύνδεση έγινε απευθείας με την τοπική βάση του SQL Server.

Δημιουργήθηκαν συνολικά 6 διαγράμματα και 3 φίλτρα. Φυσικά, έγιναν πολλές αλλαγές στο panel των ρυθμίσεων για κάθε διάγραμμα. Τα διαγράμματα είναι:

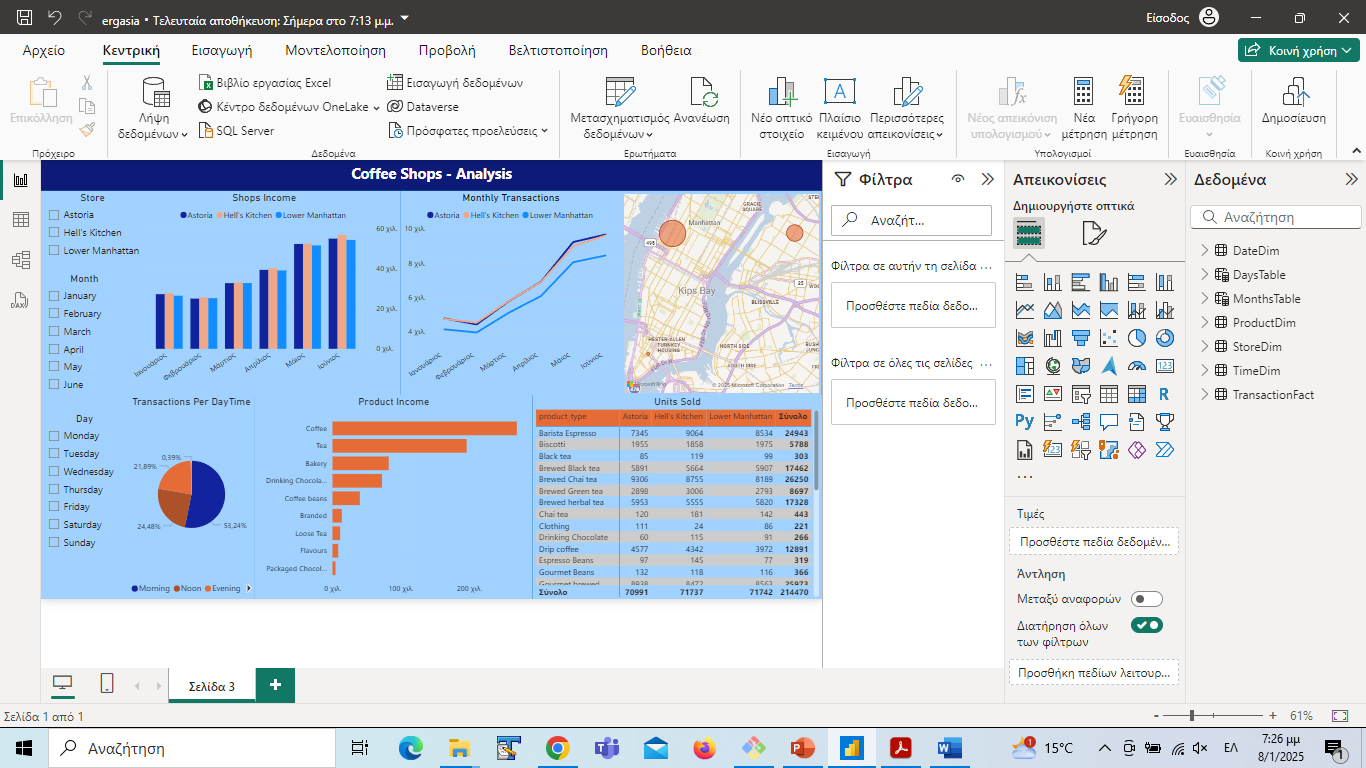
* Column Chart για κάθε κατάστημα το οποίο περιλαμβάνει τα συνολικά του έσοδα ανά μήνα και ανά μέρα του μήνα (υποστηρίζει δυνατότητα drill down).
* Line Chart για κάθε κατάστημα το οποίο περιλαμβάνει τον συνολικό αριθμό συναλλαγών ανά μήνα και ανά μέρα του μήνα (υποστηρίζει δυνατότητα drill down).
* Χάρτης που περιέχει τις τοποθεσίες των καταστημάτων μέσω των συντεταγμένων τους και το μέγεθος της κάθε φυσαλίδας αντιστοιχεί στα συνολικά έσοδα του καταστήματος.
* Pie chart με τον αριθμό των συναλλαγών που έγιναν ανάλογα την ώρα της μέρας.
* Bar Chart με τα έσοδα ανά προϊόν (κατηγορία ή τύπος) για όλα τα καταστήματα (υποστηρίζει δυνατότητα drill down).
* Matrix με τις ποσότητες ανά προϊόν που πωλήθηκαν (κατηγορία ή τύπος) για όλα τα καταστήματα (υποστηρίζει δυνατότητα drill down).
* Φίλτρο 1: Επιλογή Καταστήματος
* Φίλτρο 2: Επιλογή Μήνα
* Φίλτρο 3: Επιλογή Ημέρας της εβδομάδας

Για την σωστή - ταξινομημένη προβολή των ημερών και μηνών στα φίλτρα δημιουργήθηκαν 2 νέοι πίνακες στο power bi οι οποίοι συνδέθηκαν με τον πίνακα DateDim που περιείχε την αταξινόμητη σειρά.

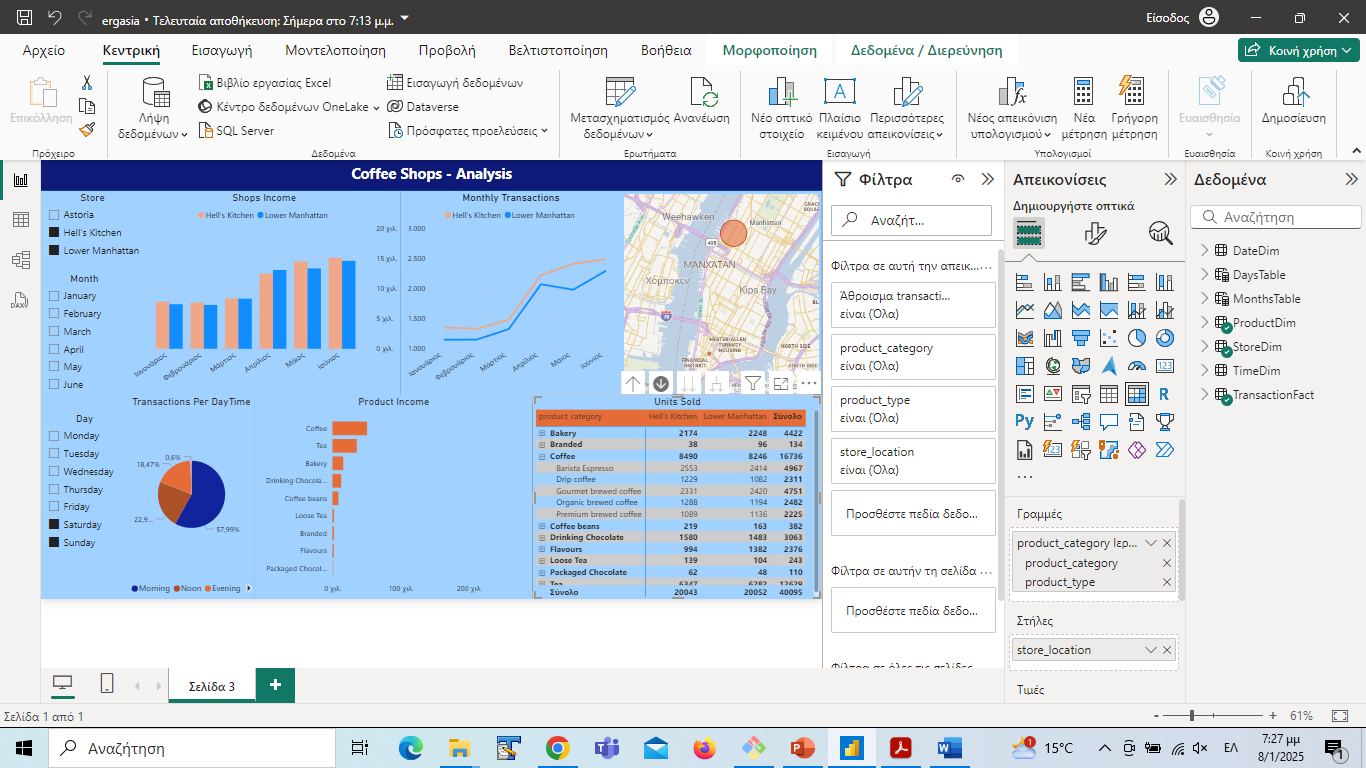




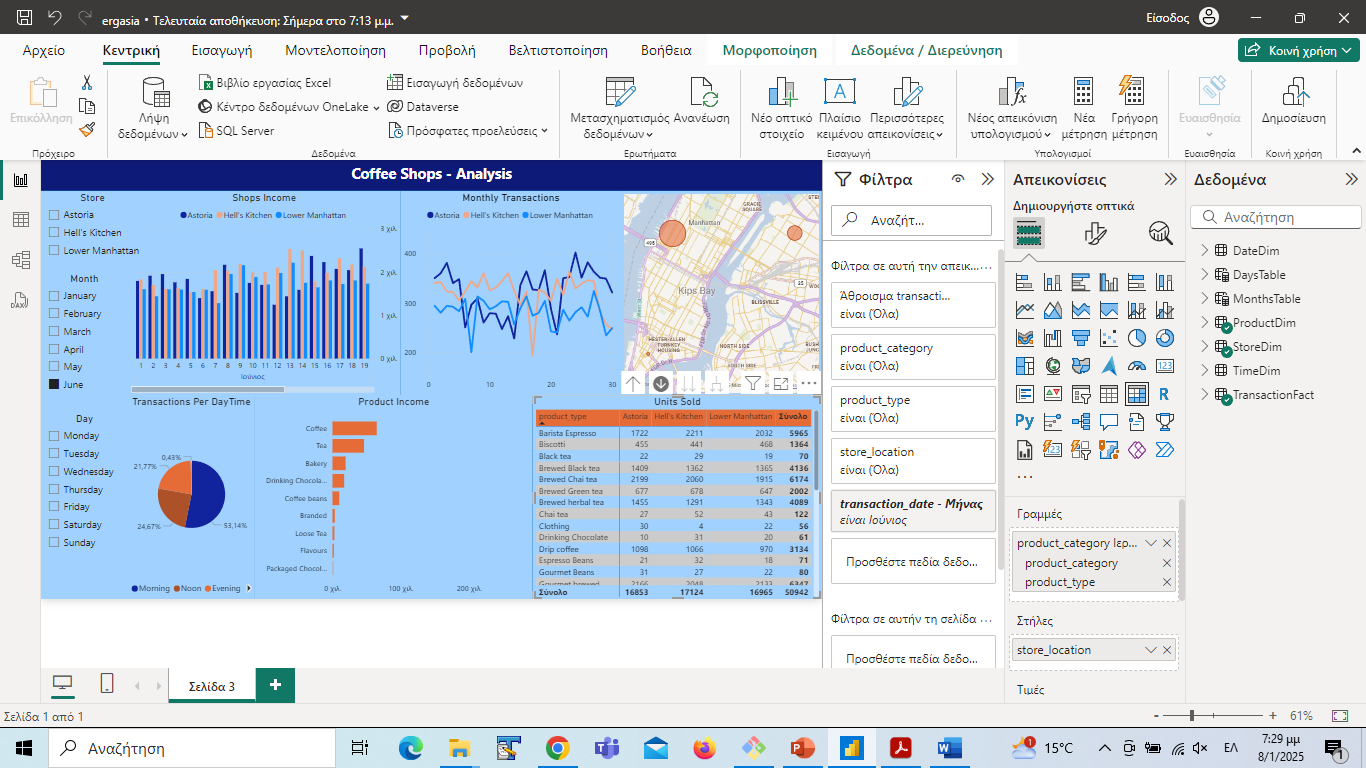
Στην συνέχεια βλέπουμε το τελικό αποτέλεσμα της οπτικοποίησης και μερικά παραδείγματα με τη χρήση των φίλτρων.



Παράδειγμα 1: Επιλογή δύο καταστημάτων και ημέρες Σάββατο και Κυριακή. Επίσης ανοίγουμε την κατηγορία του coffe στο matrix.



Παράδειγμα 2: Χρήση drill down για τον μήνα Ιούνιο, επιλογή morning από το Pie Chart και χρήση της ιεραρχίας από το matrix για όλους τους τύπους των προϊόντων.



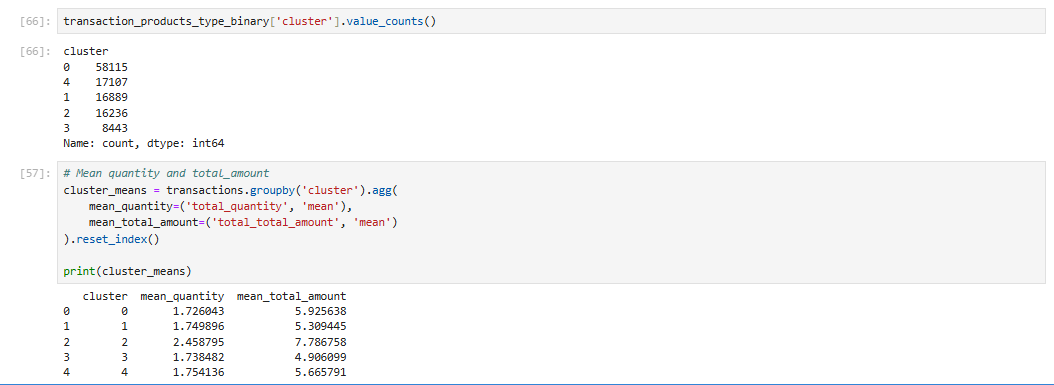
# Data mining

Τα 2 μοντέλα data mining που χρησιμοποίησα ήταν ένα clustering στις συναλλαγές εστιάζοντας στα προϊόντα που περιλαμβάνει η κάθε συναλλαγή και association rules σχετικά με το πια ζευγάρια προϊόντων αγοράζονται συχνότερα μαζί. Και για τις 2 μεθόδους χρησιμοποιήθηκε python σε jupyter notebook.

Για να ξεκινήσω το clustering χρειάζεται ένας πίνακας ο οποίος να έχει στις γραμμές τις συναλλαγές και στις στήλες τα προϊόντα. Οι τιμές των κελιών θα είναι 1 εάν υπάρχει το αντίστοιχο προϊόν στην συναλλαγή, ανεξάρτητα από το πόσες φορές υπάρχει, και 0 εάν δεν υπάρχει το προϊόν. Επίσης πρέπει να ορίσω ποια κατηγορία προϊόντων θα χρησιμοποιήσω ανάμεσα σε product category, type, description. Επέλεξα την type καθώς είναι η μεσαία στην ιεραρχία και δοκιμαστικά μου έδωσε τα καλύτερα αποτελέσματα.



Στην συνέχεια εφαρμόζω k-means στον παραπάνω πίνακα. Με δοκιμές, ο καλύτερος αριθμός για τα clusters ήταν 5. Τέλος υπολογίζω διάφορες μετρικές για τα clusters.



Τα τελικά clusters ονομάζονται

* Cluster 0: "Warm Beverage Enjoyers"
* Cluster 1: "Gourmet Coffee Lovers"
* Cluster 2: "Espresso Fans"
* Cluster 3: "Clasic Coffee Lovers"
* Cluster 4: "Chai Tea Enthusiasts"

Ξεκινώντας τα association rules πρέπει αρχικά θα φιλτράρω τις συναλλαγές που περιλάμβαναν τουλάχιστον 2 διαφορετικά προϊόντα και στη συνέχεια θα χρησιμοποιήσω τον πίνακα με τα μηδενικά και άσσους από το clustering.



Για να βρεθούν οι κανόνες συσχέτισης θα χρησιμοποιήσω τον αλγόριθμο apriori. Ως προϋποθέσεις για να οριστεί ο κανόνας, να υπάρχει το ζευγάρι των προϊόντων σε τουλάχιστον 4% των συναλλαγών και το lift (πόσο πιο συχνά εμφανίζονται τα προϊόντα μαζί από ότι θα αναμένονταν ανεξάρτητα) να είναι πάνω από 1. Τέλος παίρνω τους 5 κανόνες με το μεγαλύτερο confidence (πιθανότητα να αγοραστεί το 2ο προϊόν εφόσον αγοραστεί το 1ο).