# **ΕΡΓΑΣΙΑ ΜΑΘΗΜΑΤΟΣ**

# “Προγραμματισμός και Συστήματα στον Παγκόσμιο Ιστό”

# 2022-2023

**Ομάδα: Α.Μ.**

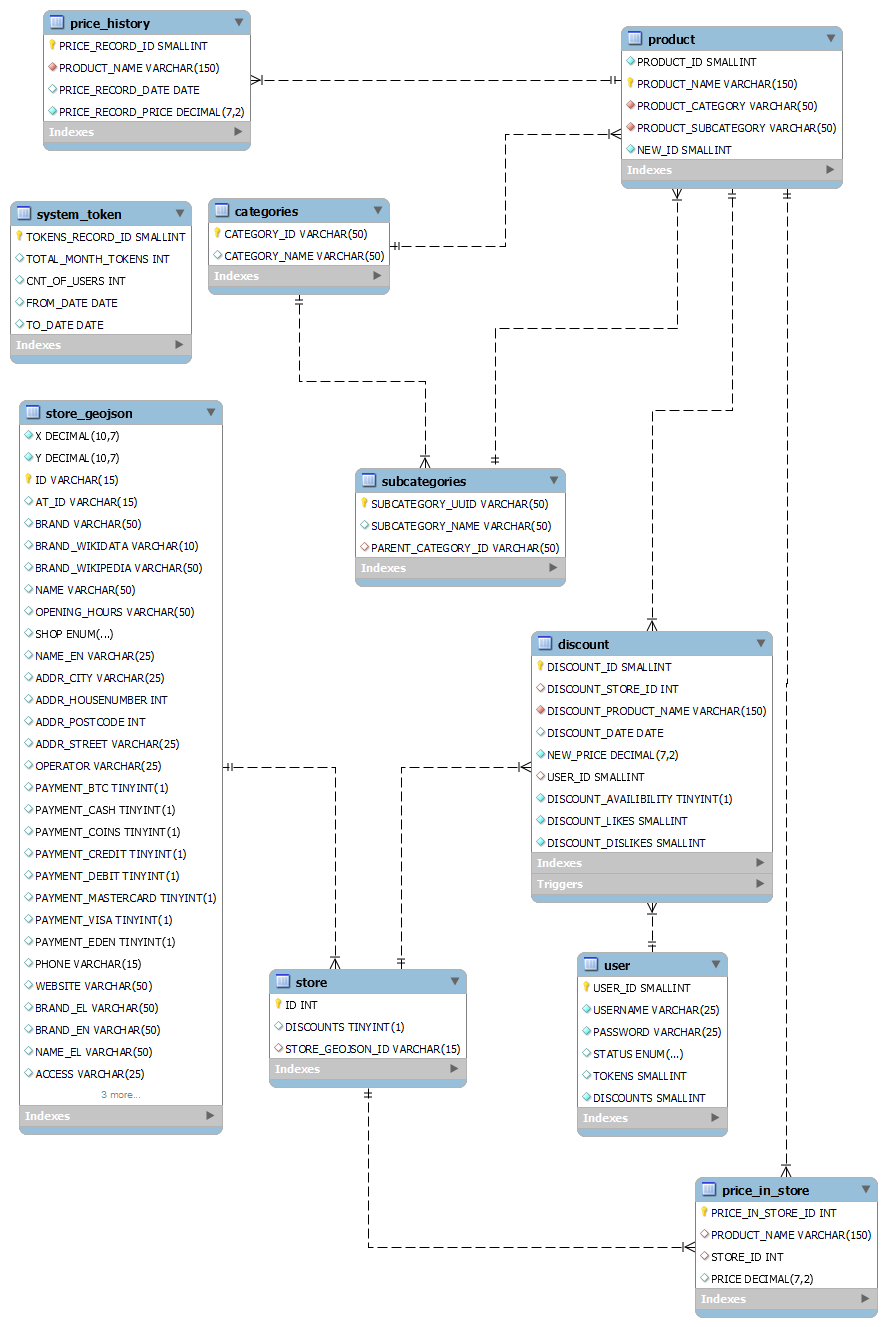
Οδυσσέας Ζάχος 1072640

Βλάσιος Παναγιώτης Παναγιώτου 1067517

Κωνσταντίνος Παρασκευόπουλος 1072608

**Σημείωση:** μαζί με το αρχείο της αναφοράς υποβλήθηκαν επίσης και οι κώδικες της Βάσης Δεδομένων, ορισμένα βοηθητικά αρχεία καθώς και το αρχείο web\_db.sql το οποίο προέρχεται από το Export που κάναμε από το phpMyAdmin. Επίσης υποβάλλονται τα αρχεία παραμετροποίησης του Server.

**ΜΕΡΟΣ 1Ο ΣΧΕΔΙΑΣΜΟΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ (ER Diagram)**



Επεξήγηση: Για την Βάση Δεδομένων κατασκευάσαμε 10 πίνακες για ανάκληση και αποθήκευση δεδομένων. Παρακάτω εξηγούμε το σκεπτικό μας πίσω από την δημιουργία κάθε πίνακα.

**USER:** Κατασκευάσαμε τον πίνακα User για αποθήκευση των στοιχείων του χρήστη. Τέτοια στοιχεία αποτελούν το Username, το Password, το είδος του χρήστη (User/Admin) καθώς και τα token του που θεωρούμε και ως μέσο αξιολόγησής του.

**STORE\_GEOJSON:** Στον συγκεκριμένο πίνακα αποθηκεύονται τα στοιχεία που λαμβάνουμε από το GeoJSON αρχείο που μας δίνεται για τον σχεδιασμό των καταστημάτων στο χάρτη. Φτιάξαμε τον συγκεκριμένο πίνακα ώστε με απλά SELECT QUERIES να μας επιστρέφονται ορισμένα καταστήματα και να μπορούμε να τα σχεδιάζουμε στον χάρτη (παράδειγμα: Χάρτης με φίλτρο ανά κατηγορία προσφορών)

**CATEGORIES:**  Στον συγκεκριμένο πίνακα αποθηκεύονται τα ονόματα και τα IDs των κατηγοριών όπως αυτά μας δόθηκαν από τα αντίστοιχα JSON αρχεία.

**SUBCATEGORIES:** Στον συγκεκριμένο πίνακα αποθηκεύονται τα ονόματα και τα UUIDs των υποκατηγοριών καθώς και τα IDs των κατηγοριών στις οποίες ανήκουν οι υποκατηγορίες, όπως αυτά μας δόθηκαν από τα αντίστοιχα JSON αρχεία.

**PRODUCT:** Στον πίνακα αυτό αποθηκεύονται τα στοιχεία των προϊόντων όπως αυτά μας δόθηκαν από τα αντίστοιχα JSON αρχεία. Ο πίνακας αυτός γεμίζει αυτόματα με εγγραφές μέσω PHP και των JSON αρχείων που μας έχουν δοθεί.

**PRICE\_HISTORY:** Στον πίνακα αυτό αποθηκεύεται το ιστορικό μέσης τιμής ενός προϊόντος ανά ημέρα. Ο πίνακας αυτός γεμίζει αυτόματα με εγγραφές μέσω PHP και των JSON αρχείων που μας έχουν δοθεί.

**STORE:** Στον πίνακα αυτό αποθηκεύουμε μερικά επιπλέον στοιχεία όπως το αν υπάρχουν διαθέσιμες προσφορές στο κατάστημα ενώ αντιστοιχούμε και ένα νέο ID στα καταστήματα για ευχρηστία καθώς το ID του GeoJSON είναι αρκετά δύσχρηστο λόγω του μεγέθους του.

**PRICE\_IN\_STORE:** Στον συγκεκριμένο πίνακα αποθηκεύονται οι τιμές των προϊόντων ανά κατάστημα (λόγω έλλειψης δεδομένων αποφασίσαμε ότι θα θεωρήσουμε ότι για κάθε κατάστημα η αρχική τιμή θα είναι ίδια και ίση με την τελευταία εγγραφή της μέσης ημερήσιας τιμής για κάθε προϊόν του πίνακα Price\_History). Ο πίνακας αυτός γεμίζει αυτόματα με εγγραφές μέσω PHP και των JSON αρχείων που μας έχουν δοθεί.

**DISCOUNT:** Στον συγκεκριμένο πίνακα αποθηκεύονται στοιχεία σχετικά με προσφορές που έχουν υποβληθεί από τους Χρήστες, όπως το κατάστημα στο οποίο υπάρχει η προσφορά, το όνομα του προϊόντος, η νέα τιμή του, το ID του χρήστη που κατέθεσε την προσφορά, η διαθεσιμότητα της προσφοράς (μια προσφορά μπορεί μετά από ένα χρονικό διάστημα να μην είναι πλέον διαθέσιμη -μετά από 2 εβδομάδες θεωρείται μη διαθέσιμη κάτι το οποίο γίνεται αυτόματα μέσω του Event AVAILABILITY- είτε να έχει εξαντληθεί, όπου ο χρήστης έχει τη δυνατότητα να το δηλώσει μέσω ειδικού Button σε αντίστοιχη σελίδα) και τέλος τα Likes/Dislikes της προσφοράς.

Στη συνέχεια σχεδιάσαμε ορισμένα Trigger τα οποία αυτόματα επηρεάζουν τιμές ή δημιουργούν εγγραφές έπειτα από συγκεκριμένα INSERTs ή UPDATEs σε συγκεκριμένους πίνακες της Βάσης.

**discount\_likes\_gives\_tokens** και **discount\_dislikes\_subtracts\_tokens:** Τα συγκεκριμένα Triggers έπειτα από Update του αριθμού των Likes ή Dislikes αντίστοιχα σε μία προσφορά αυξάνουν ή μειώνουν αντίστοιχα τον αριθμό των Tokens του Χρήστη που υπέβαλλε την προσφορά.

**update\_product\_price\_after\_discount\_placement** και **update\_product\_price\_after\_discount\_delete:** Τα συγκεκριμένα Trigger θέτουν την νέα τιμή ενός προϊόντος στο αντίστοιχο κατάστημα έπειτα από υποβολή προσφοράς ή την επαναφέρουν στο προηγούμενο επίπεδο μετά από διαγραφή προσφοράς ενώ ελέγχουν και τον συνολικό αριθμό προσφορών στο αντίστοιχο κατάστημα και αντίστοιχα θέτουν το πεδίο DISCOUNTS του πίνακα Store στο 1 ή στο 0 ώστε να γνωρίζουμε αν υπάρχουν διαθέσιμες προσφορές στο κατάστημα. Επίσης σε περίπτωση ύπαρξης προσφοράς στο συγκεκριμένο προϊόν και κατάστημα κατά πόσο «καλύτερη» είναι η νέα προσφορά και αποδίδουν τον αντίστοιχο αριθμό Token στον χρήστη που την υπέβαλε.

**ΜΕΡΟΣ 2Ο Ρυθμίσεις Server (Διαχείριση cache):**

**TTL:** Θέσαμε TTL μέσω ενός .htaccess αρχείου το οποίο βρίσκεται στον φάκελο με τα html αρχεία του site. Πιο συγκεκριμένα ορίσαμε τα παρακάτω commands:

1. **Header set Cache-Control "max-age=172800, public"**
2. **<FilesMatch ".(ico|pdf|flv|jpg|jpeg|png|gif|js|css|swf)$"> Header set Cache-Control "max-age=86400, public" </FilesMatch>**

Τα οποία ορίζουν συγκεκριμένους χρόνους παραμονής για ορισμένους τύπους αρχείων στην cache (σε seconds).

Πιο συγκεκριμένα, ορίσαμε καθολικά για όλους τους τύπους αρχείων χρόνο παραμονής 48 ωρών στην μνήμη cache απ΄την τελευταία τους ανάκτηση και 24 ώρες αντίστοιχα σε συγκεκριμένους τύπους αρχείων όπως πχ pdf,jpg,png κ.α

H πηγές πληροφορίας στις οποίες βασίστηκαν οι παραμετροποιήσεις μας προέρχονται απ’το διαδίκτυο και απ’τoυς εξωτερικούς συνδέσμους των διαφανειών “3. HTTP Υποστήριξη για Κρυφές Μνήμες και Αντίγραφα” του eclass: “[Does the number of .htaccess rules impact performance and scalability? - StrategiQ](https://strategiq.co/does-the-number-of-htaccess-rules-impact-performance-and-scalability/)”.