

Σχολή Ηλεκτρολόγων Μηχανικών &

Μηχανικών Υπολογιστών ΕΜΠ

ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

Εξαμηνιαία εργασία(Project)

Ανάλυση του καλαθιού αγορών - Shopping basket analysis

Ομάδα 80

Φοιτητές

Καραβαγγέλης Αθανάσιος 03117022

Μαντζούτας Ανδρέας 03117108

Καραμπίνας Παναγιώτης 03116170

22 Ιουνίου 2020

ΕΙΣΑΓΩΓΗ

«Μία μεγάλη αλυσίδα Super Market αποφάσισε να εκδώσει μία κάρτα πιστότητας πελατών ώστε να συγκεντρώσει πληροφορίες για τους πελάτες της και να μπορέσει να πάρει στοχευμένες αποφάσεις marketing. Μας ανέθεσε λοιπόν την υλοποίηση ενός ολοκληρωμένου συστήματος αποθήκευσης, διαχείρισης και ανάλυσης των πληροφοριών που συγκεντρώνονται.»

Για το σκοπό αυτό αναπτύξαμε μία εφαρμογή που απευθύνεται τόσο στους πελάτες όσο και στο προσωπικό της αλυσίδας που θα τους παρέχει συγκεκριμένες χρήσιμες πληροφορίες.

Ο χρήστης της εφαρμογής μας έχει τη δυνατότητα να εισάγει , να επεξεργάζεται και να διαγράφει όπως εκείνος κρίνει τα εξής μέσα από την εφαρμογή μας: προϊόντα , καταστήματα , πελάτες . Ακόμα μπορεί να βλέπει τα υπάρχοντα δεδομένα κάθε κατηγορίας όπως πελάτες ,προϊόντα, καταστήματα, συναλλαγές ,κατηγορίες προϊόντων και ιστορικά τιμών προϊόντων μεταξύ άλλων.

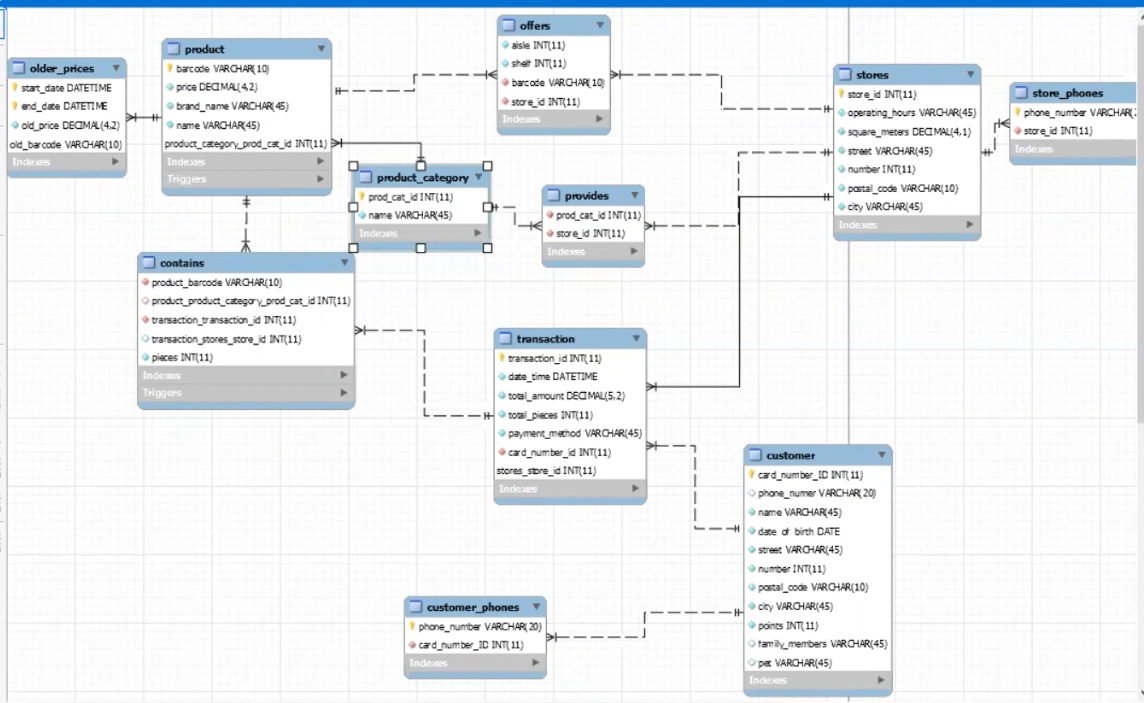
Έπειτα ο χρήστης έχει τη δυνατότητα να βλέπει σημαντικές πληροφορίες για την αλυσίδα SuperMarket πάνω στην οποία βασίζεται η εφαρμογή .Συγκεκριμένα, μπορεί να αναζητήσει με διάφορα κριτήρια συναλλαγές που έχουν γίνει στα καταστήματα και να παρατηρήσει στοιχεία αναφορικά με τους πελάτες και τις αγορές που κάνουν . Ακόμη , μπορεί να λαμβάνει πληροφορίες σχετικά με την δημοφιλία των προϊόντων , των κατηγοριών προϊόντων καθώς και των καταστημάτων.

Γενικά υπάρχουν διάφορες επιλογές που επικουρούν τόσο έναν πελάτη να κατευθύνει καλύτερα τις αγορές του αλλά και το προσωπικό και τη διοίκηση του καταστήματος να λαμβάνει σημαντικά στοιχεία τα οποία συνεισφέρουν προς την καλύτερη ανάπτυξη και οργάνωση της αλυσίδας.

Παράλληλα ,για το user interface(UI) μας έχουν χρησιμοποιηθεί dropdown lists ,radio buttons και διάφορες υποδείξεις ώστε να είναι όσο το δυνατόν πιο φιλικό προς το χρήστη. Ακόμη σε περίπτωση λαθών αποστέλλονται κατάλληλα μηνύματα που ενημερώνουν το χρήστη.

1.ΣΧΕΣΙΑΚΟ ΔΙΑΓΡΑΜΜΑ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Το σχεσιακό διάγραμμα της βάσης δεδομένων που υλοποιήσαμε είναι το ακόλουθο:



Στη συνέχεια, θα προβούμε στον σχολιασμό του.

a. Περιορισμοί που έχουν οριστεί, ανά κατηγορία, με σύντομη αιτιολόγηση για την επιλογή

**Περιορισμοί ακεραιότητας:**

-**Περιορισμός <not null>:**

Χρησιμοποιήσαμε αυτό τον περιορισμό σε κάθε PK καθώς όπως γνωρίζουμε τα PK δεν μπορούν να έχουν μηδενικές τιμές. Επίσης είναι φανερό πως στην βάση μας σχεδόν όλα τα πεδία τα έχουμε ορίσει ως not null, διότι απαιτούνταν σε κάθε περίπτωση όσα στοιχεία ζητάμε να μην απουσιάζουν για να δουλεύει ορθά η βάση και να βγάζει μέσω των queries σαφή αποτελέσματα.

**-Περιορισμός <unique>:**

Χρησιμοποιήσαμε αυτό τον περιορισμό στο "name" του product\_category, καθώς θέλαμε να τονίσουμε πως δεν υπάρχουν δύο κατηγορίες με το ίδιο όνομα. Ομοίως για το product βάλαμε τον περιορισμό unique στο "name" και στο "barcode" για να δείξουμε και πάλι την μοναδικότητα των δύο αυτών χαρακτηριστικών του προϊόντος.

-**Περιορισμός check\_brand :**

Στην οντότητα product έχουμε oρίσει ένα check constraint που περιορίζει το brand\_name σε ‘Yes’ ή ‘No’.

-**Περιορισμός check\_date :**

Στην οντότητα older\_prices έχουμε ορίσει ένα check constraint που περιορίζει το start\_date να είναι < end\_date.

-**Περιορισμός check\_method :**

Στην οντότητα transaction έχουμε ορίσει ένα check constraint που περιορίζει το payment\_method σε 2 συγκεκριμένες τιμές : «Πιστωτική Κάρτα» ή «Μετρητά».

**Κλειδιά(PK):**

-**customer**(Card\_Number\_ID): χρησιμοποιήθηκε για να προσδώσει μοναδικότητα σε κάθε εγγραφή πελάτη, επιπρόσθετα διαφοροποιεί τους πελάτες στην περίπτωση συνωνυμίας.

-**transaction**(transaction\_id,card\_number\_id,stores\_store\_id): κωδικοποιεί την κάθε συναλλαγή μοναδικά ώστε να αντιστοιχίζεται σε έναν πελάτη, επίσης παίρνει και το PK του customer καθώς το transaction είναι weak entity και πρέπει να πλαισιώνεται από το card\_number\_id. Τέλος το transaction λαμβάνει και το PK του stores αφού η σχέση που συνδέει τις δύο οντότητες είναι 1 προς Ν(stores,transaction αντίστοιχα).

-**product**(barcode): ομοίως διαφοροποιεί το κάθε προϊόν.

-**stores**(store\_id): ομοίως διαφοροποιεί το κάθε κατάστημα.

-**older\_prices**(start\_date,end\_date,old\_barcode): Θεωρήσαμε πως θα ήταν βολικό να ορίσουμε και το end\_date ως PK καθώς όταν ανοίγει ένα νέο κατάστημα πολλά έχουν το ίδιο start\_date, επομένως θέτοντας και το end\_date ως PK μπορούμε να ξεχωρίσουμε ευκολότερα τις εγγραφές, επίσης ως weak entity το older\_prices θα έχει ως PK και το old\_barcode που είναι το primary key της product.

-**product\_category**(prod\_cat\_id): διαφοροποιεί την κάθε κατηγορία προϊόντων.

-**provides**(prod\_cat\_id,store\_id): η σχέση που περιγράφεται από αυτό το table συνδέει τις οντότητες

product\_category και stores, είναι επόμενο το primary key του να περιλαμβάνει(και στην συγκεκριμένη

περίπτωση να αποτελείται) από τα primary keys των προαναφερθέντων οντοτήτων.

-**offers**(barcode,store\_id): ομοίως με την περίπτωση της provides θα έχει ως primary key τα primary keys

των product και stores αντίστοιχα.

-**contains**(product\_barcode,transaction\_transaction\_id): ομοίως με την περίπτωση της provides θα έχει ως primary key τα primary keys

των product και transaction αντίστοιχα.

-**store\_phones**(phone\_number,store\_id): χρησιμοποιούμαι δύο primary key, καθώς είναι πιθανό το κάθε

κατάστημα να έχει περισσότερα από ένα τηλέφωνα.

-**customer\_phones**(phone\_number,card\_number\_ID): ομοίως με store\_phones.

**Ακεραιότητα πεδίου τιμών:**

Το μεγαλύτερο μέρος των πεδίων της βάσης που αναπτύξαμε είναι τύπου int και varchar καθώς όπως είναι λογικό σε πεδία που απαιτούνταν δεδομένα αριθμητικά τα ορίσαμε ως int, επίσης τα πεδία που απαιτούσαν δυνητικά αλφαριθμητικά ή απλά αριθμητικά δεδομένα τα ορίσαμε ως varchar. Ωστόσο κάποια πεδία έχριζαν ειδικής μεταχείρισης και αναφέρονται παρακάτω:

-price(product) είναι τύπου decimal(4,2) για λόγους ακρίβειας καθώς γνωρίζουμε πως κανένα προϊόν δεν έχει τιμή μεγαλύτερη ή ίση του 100 και μπορεί δυνητικά (λόγω κάποιας έκπτωσης πιθανώς) να αποκτήσει ψηφία δεξιά από την υποδιαστολή.

-total\_amount(transaction) ομοίως με το price είναι τύπου decimal(5,2) για λόγους που προαναφέρθηκαν.

-date\_time(transaction) έχει τύπο datetime καθώς θέλουμε να αποθηκεύουμε ακριβή ημερομηνία και ώρα για διάφορες πρακτικές εφαρμογές τους (πχ συναλλαγές ανά ώρα κ.ά.)

-date\_of\_birth(costumer) έχει τύπο date καθώς ενδιαφερόμαστε για ακρίβεια ημέρας.

-square\_meters(stores) έχει τύπο decimal(4,1) γιατί θέλουμε να μετρήσουμε με ακρίβεια dm.

-start\_date,end\_date(older\_prices) έχει τύπο datetime.

-price(older\_prices) έχει τύπο decimal(4,2)

**Αναφορική ακεραιότητα(FK):**

-Στο πεδίο transaction\_transaction\_id του table contains με αναφορά στο table transaction (ON DELETE CASCADE ON UPDATE CASCADE). Με την διαγραφή κάποιου transaction\_id θα διαγράφετε η συναλλαγή από την "απόδειξη" που έχουμε ορίσει (δηλαδή στο contains).

-Στο πεδίο product\_barcode και product\_product\_category\_prod\_cat\_id του contains με αναφορά στο table product(ON DELETE CASCADE ON UPDATE CASCADE). Διαγράφοντας το προϊόν που ανήκει σε μια κατηγορία προϊόντων θα διαγράφεται αυτόματα και η συναλλαγή από την "απόδειξη" που έχουμε ορίσει(δηλαδή στο contains)

-Στο πεδίο card\_number\_id του table customer\_phones με αναφορά στο (ON DELETE CASCADE ON UPDATE CASCADE). Με την διαγραφή ενός πελάτη θα διαγράφεται και το τηλέφωνο του(ή τα τηλέφωνα του).

-Στο πεδίο barcode του table offers με αναφορά στο table product (ON DELETE CASCADE ON UPDATE CASCADE). Δεν θα υπάρχει σχέση offers που να περιέχει προϊόν που έχει διαγραφεί.

-Στο πεδίο store\_id του table offers με αναφορά στο table stores (ON DELETE CASCADE ON UPDATE CASCADE). Δεν θα υπάρχει σχέση offers που να περιέχει κατάστημα που έχει διαγραφεί.

-Στο πεδίο old\_barcode του table older\_prices με αναφορά στο table product (ON DELETE CASCADE ON UPDATE CASCADE).

Δεν θα υπάρχουν καταχωρήσεις ιστορικού τιμής για ένα προϊόν που έχει διαγραφεί.

-Στο table older\_prices ελέγχεται αν η τιμή start\_date είναι μικρότερη της τιμής end\_date, καθώς αν δεν ίσχυε αυτό θα υπήρχε λογικό πρόβλημα στην βάση.

-Στο πεδίο product\_category\_prod\_cat\_id του table product με αναφορά στο table product\_category (ON DELETE CASCADE ON UPDATE CASCADE). Δεν θα υπάρχει καταχώρηση προϊόντος που να μην ανήκει σε μια κατηγορία προϊόντος.

-Στο table product ελέγχουμε αν η καταχώρηση brand\_name είναι "Yes" ή "No" καθώς αν δεν είναι θα υπάρξει πρόβλημα ακεραιότητας.

-Στο πεδίο prod\_cat\_id του table provides με αναφορά στο table product\_category (ON DELETE CASCADE ON UPDATE CASCADE). Δεν θα υπάρχει καταχώρηση στην provides με κατηγορία προϊόντων που έχει διαγραφεί.

-Στο πεδίο store\_id του table provides με αναφορά στο table stores (ON DELETE CASCADE ON UPDATE CASCADE). Δεν θα υπάρχει καταχώρηση στην provides με κατάστημα που έχει διαγραφεί.

-Στο πεδίο store\_id του table store\_phones με αναφορά στο table stores (ON DELETE CASCADE ON UPDATE CASCADE). Αν διαγραφεί ενα κατάστημα θα διαγράφονται και τα τηλέφωνα που σχετίζονται με αυτό.

-Στο πεδίο card\_number\_id του table transaction με αναφορά στο table customer (ON UPDATE CASCADE). Αν ενημέρωσουμε το card\_number\_id κάποιου πελάτη τότε θα ενημερωθεί και η καταχώρηση που σχετίζεται με αυτόν το πελάτη στο transaction.

-Στο πεδίο stores\_store\_id του table transaction με αναφορά στο table store (ON UPDATE CASCADE). Αν ενημέρωσουμε το store\_id κάποιου καταστήματος τότε θα ενημερωθεί και η καταχώρηση που σχετίζεται με αυτόν το κατάστημα στο transaction.

-Στο table transaction αν η καταχώρηση payment\_method είναι 'Πιστωτική Κάρτα' ή 'Μετρητά' καθώς αν δεν είναι θα υπάρξει πρόβλημα ακεραιότητας.

**Περιορισμοί οριζομένοι από τον χρήστη(Triggers):**

-Με κάθε εισαγωγή επιπλέον προϊόντων στην απόδειξη μας(δηλαδή την contains) αυξάνεται το σύνολο των προϊόντων του αντίστοιχου transaction και(σε ένα επόμενο trigger) αυξάνεται η συνολική τιμή του transaction.

-Με κάθε αλλαγή τιμής προϊόντος αλλάζει το ιστορικό της τιμής του εκάστοτε προϊόντος και θέτει ως start\_date το αμέσως προηγούμενο end\_date, ωστόσο εάν δεν υπήρχε αυτό το προϊόν στις καταχωρήσεις μας αυτόματα τίθεται μια στάνταρ ημερομηνία ως start\_date και ως end\_date το παρόν.

-Με κάθε εισαγωγή νέου προϊόντος προστίθεται το συγκεκριμένο προϊόν σε κάθε κατάστημα σε ένα τυχαίο ράφι.

b. Ευρετήρια που έχουν ορισθεί μαζί με σύντομη αιτιολόγηση

Δημιουργήσαμε indexes στα εξής πεδία για λόγους ταχύτερης υλοποίησης των queries που μας δόθηκαν(καθώς χρησιμοποιείται μεγάλος αριθμός where, order by κτλ στα queries). Επίσης οι στήλες που κάναμε index δεν αλλάζουν πολλές φορές καθώς αυτό θα ήταν επιπλέον βάρος για το πρόγραμμα να σηκώσει, διότι θα χρειαζόταν να ανανεώθουν εκ νέου οι indexes στους οποίους γίνονται αλλαγές.

Τα index που δημιουργήσαμε είναι τα εξής:

-Στο πεδίο `product\_barcode` και `product\_product\_category\_prod\_cat\_id` του table contains.

-Στο πεδίο `transaction\_stores\_store\_id` και `transaction\_transaction\_id`του table contains.

-Στο πεδίο `transaction\_transaction\_id`του table contains.

-Στο πεδίο `product\_barcode`(3) του table contains.

-Στο πεδίο 'card\_number\_ID` του table customer\_phones.

-Στο πεδίο `store\_id` του table offers.

-Στο πεδίο `barcode` του table offers.

-Στο πεδίο `aisle`,`shelf` του table offers.

-Στο πεδίο `old\_barcode` του table older\_prices.

-Στο πεδίο `old\_barcode`(3) του table older\_prices.

-Στο πεδίο `product\_category\_prod\_cat\_id` του table product.

-Στο πεδίο `prod\_cat\_id` του table provides.

-Στο πεδίο `store\_id` του table provides.

-Στο πεδίο `store\_id` του table store\_phones.

-Στο πεδίο `card\_number\_id` του table transaction.

-Στο πεδίο `stores\_store\_id` του table transaction.

c. Το σύστημα και οι γλώσσες προγραμματισμού που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογήςμας



* Για την υλοποίηση της βάσης δεδομένων χρησιμοποιήσαμε την έκδοση 8.0 του MySQL.
* Για την επικοινωνία client – server χρησιμοποιήσαμε Apache Web Server.
* Για το serverside της εφαρμογής χρησιμοποιήσαμε PHP.
* Για το clientside της εφαρμογής χρησιμοποιήσαμε κυρίως HTML, αλλά και κάποια στοιχεία javascript, κυρίως για τα διαγράμματα .
* Για τον καθορισμό των αισθητικών στοιχείων της εφαρμογής (όπως πχ για τα χρώματα του background, για τα περιγράμματα, για το styling των κουμπιών κλπ) χρησιμοποιήσαμε CSS.

d. Αναλυτικά βήματα για το πώς θα μπορούσε να εγκαταστήσει κάποιος την εφαρμογή μας από την αρχή

Για να εγκαταστήσει κάποιος από την αρχή την εφαρμογή μας θα πρέπει να διαθέτει MySQLServer καθώς και ApacheWebServer με SQL και PHPμέσω του XAMPP. Αυτά μπορούν να βρεθούν στα παρακάτω links :

* Apache Web Server μέσωτουXAMPP

<https://www.apachefriends.org/index.html>

* MySQL Server 8.0.19:

<https://downloads.mysql.com/archives/get/p/25/file/mysql-installer-web-community-8.0.19.0.msi>

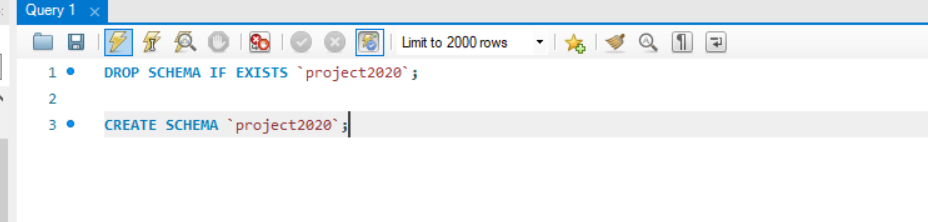
Στις επιλογές που μας δίνονται στον Installer του MySQLServer επιλέγουμε προϊόντα από τις κατηγορίες ως εξής:

* MySQLServer->MySQLServer 8.0 ->MySQLServer 8.0.19 – X64 και
* Applications -> MySQL Workbench -> MySQL Workbench 8.0 -> MySQL Workbench 8.0.19 – Χ64

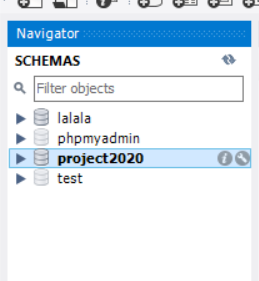
Ενώ στον Installer του XAMPPεπιλέγουμε να κατεβάσουμε : Apache , SQL και PHP.

Όταν έχουμε τελειώσει με αυτά, πρέπει να ανοίξουμε το MySqlWorkbench και να δημιουργήσουμε ένα νέο project, αφού πρώτα ορίσουμε τα στοιχεία του root user που μας ζητάει.

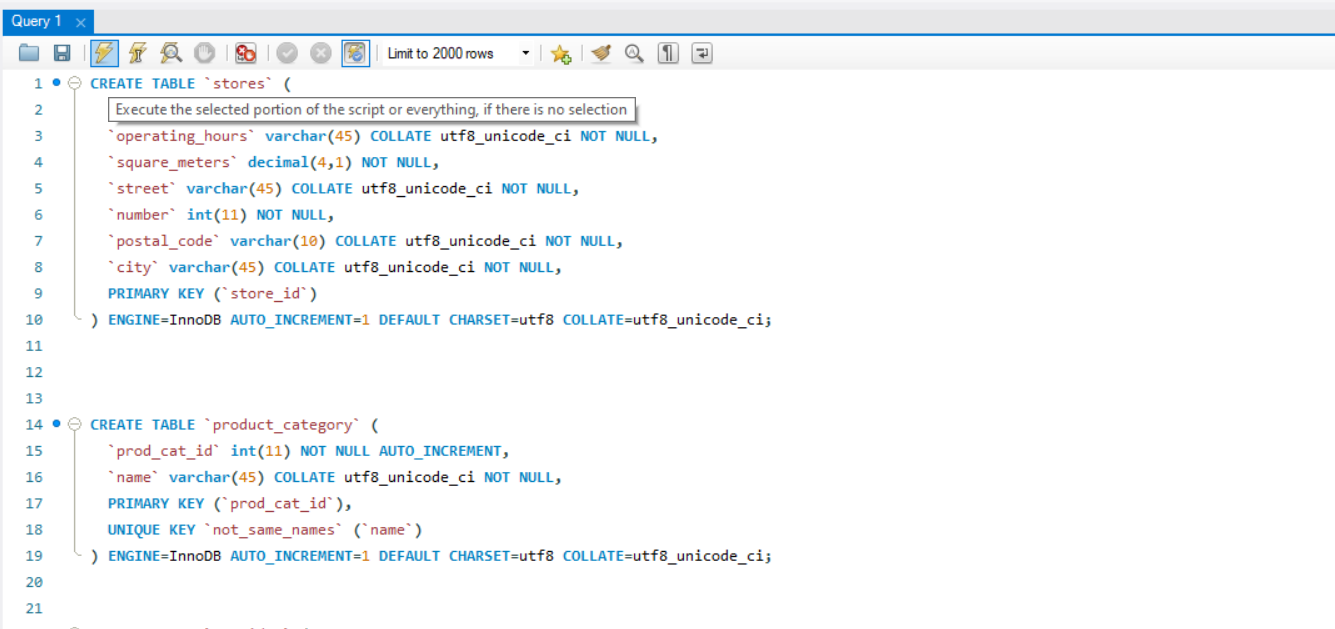
Έπειτα, αφού δημιουργήσουμε το project, πληκτρολογούμε τις παρακάτω εντολές και τις τρέχουμε, ώστε να δημιουργηθεί η βάση. Για να τις τρέξουμε, πατάμε το σημαδάκι του κεραυνού που δείχνει ο κέρσορας. Τέλος, είναι σημαντικό το όνομα που θα πληκτρολογήσετε, να είναι το ίδιο ακριβώς με αυτό που φαίνεται παρακάτω.



Στη συνέχεια, επιλέγουμε το schema που δημιουργήσαμε, κάνοντας διπλό κλικ πάνω του, όπως φαίνεται παρακάτω.



Παίρνουμε αντιγραφή τον κώδικα που παρατίθεται στο αρχείο με όνομα sql\_code.sql, και τον εκτελούμε με τον ίδιο τρόπο.



Τέλος, για το User Interface της εφαρμογής, αρκεί να αντιγράψετε τον φάκελο με όνομα project2020, που έχουμε επισυνάψει, στον φάκελο XAMPP htdocs folder και για να συνδεθείτε, να πληκτρολογήσετε σε κάποιον browser τη διεύθυνση [**http://localhost/project2020/main/index.php**](http://localhost/project2020/main/index.php) .

Να σημειώσουμε ότι για να λειτουργήσει σωστά το User Interface, θα πρέπει πρώτα να έχουν εισαχθεί μερικά δεδομένα στη βάση μέσω του MySQLWorkbench(τουλάχιστον 1 σε κάθε table). Για διευκόλυνση, έχουμε συμπεριλάβει στον επισυναπτόμενο φάκελο τα δεδομένα που χρησιμοποιήσαμε στη μορφή .csv αρχείου και αρκεί να κάνετε import μέσω του MySQLWorkbench το κάθε αρχείο στον πίνακα που υποδεικνύει το όνομά του.