



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2023-2024

ΕΞΑΜΗΝΟ 6ο

ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

ΥΠΕΥΘΥΝΟΙ ΚΑΘΗΓΗΤΕΣ: Δ. ΤΣΟΥΜΑΚΟΣ, ΜΑΡΙΟΣ ΚΟΝΙΑΡΗΣ (Ε.ΔΙ.Π.)

Βάσεις Δεδομένων

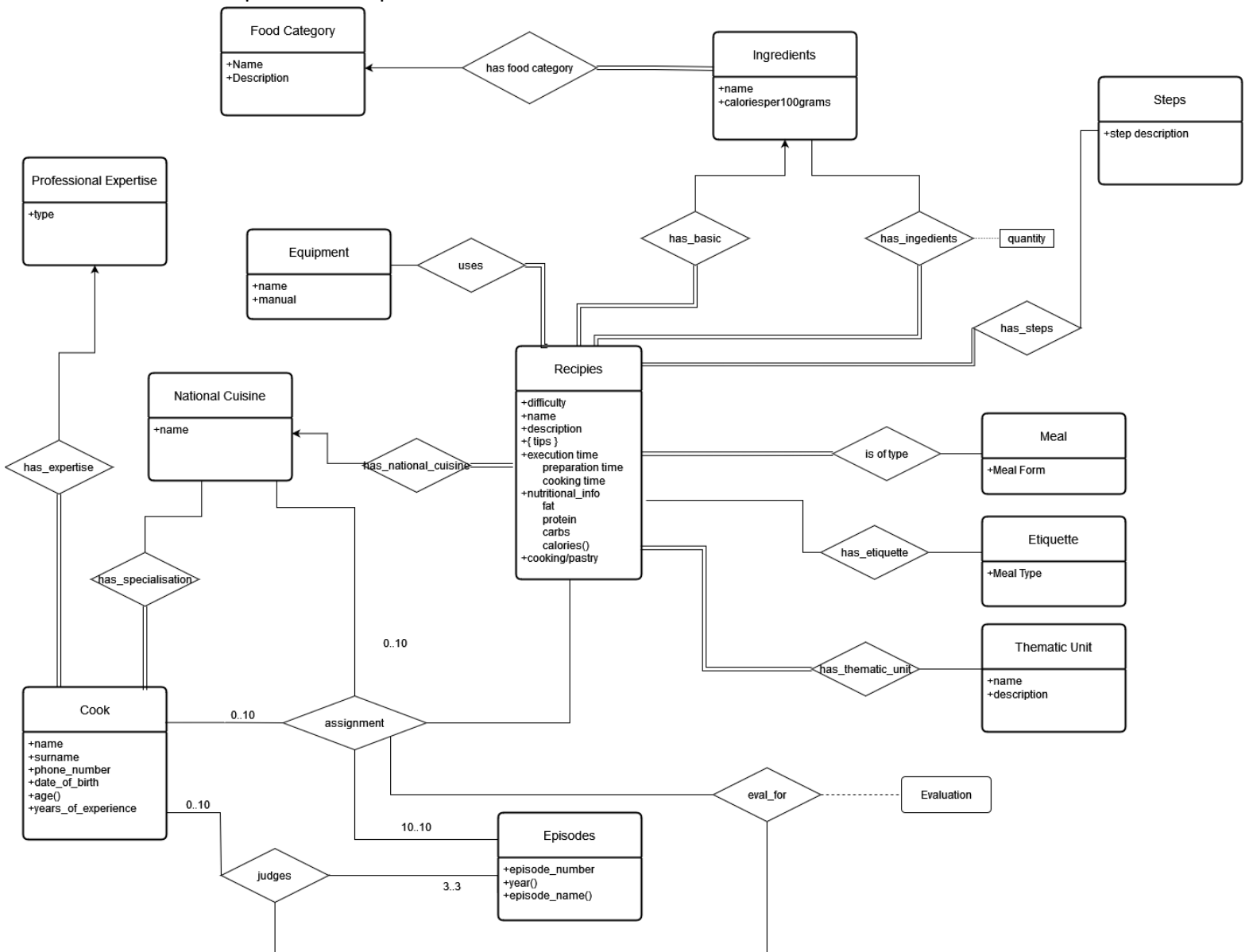
Εξαμηνιαία Εργασία

ΣΥΓΓΡΑΦΕΙΣ	Κάκος Σωτήριος (ΑΜ: 03121110) Κριθαρίδης Κωνσταντίνος (ΑΜ: 03121045) Μηνάγιας Δημήτριος (ΑΜ: 03122813)
ΠΕΡΙΕΧΟΜΕΝΑ	1. Διάγραμμα Οντοτήτων - Συσχετίσεων 2. Σχεσιακό Σχήμα 3. Ευρετήρια 4. DDL και DML 5. Winner 6. Authentication 7. Οδηγίες εγκατάστασης

1. Διάγραμμα Οντοτήτων - Συσχετίσεων

Ζητούμενο της εργασίας είναι η σχεδίαση και υλοποίηση κατάλληλης βάσης δεδομένων, η οποία θα αξιοποιηθεί από δημοφιλή διαγωνισμό μαγειρικής για την αποθήκευση και διαχείριση σχετικών πληροφοριών.

Η πρώτη φάση της ανάπτυξης της βάσης ήταν η εξόρυξη των απαιτήσεων και των σχετικών προδιαγραφών που θα έπρεπε αυτή να πληροί. Ο εννοιολογικός σχεδιασμός της βάσης στηρίχθηκε γύρω από τις κύριες κατηγορίες πληροφοριών που θέλαμε να αποθηκεύουμε, δηλαδή αυτές σχετικά με τις συνταγές, τους μάγειρες και τα επεισόδιά του διαγωνισμού. Προσπαθώντας να κάνουμε μία αφαιρετική αλλά πλήρη περιγραφή του μικρόκοσμού μας καταλήξαμε διαισθητικά στο διάγραμμα οντοτήτων συσχετίσεων που παρατίθεται παρακάτω.



Ο διαγωνισμός διαδραματίζεται σε επεισόδια σε καθένα από τα οποία επιλέγονται με τυχαίο τρόπο 10 εθνικές κουζίνες, 10 μάγειρες αντιπρόσωποι από κάθε εθνική κουζίνα, 3 μάγειρες-κριτές και 1 συνταγή από κάθε εθνική κουζίνα που ανατίθεται σε 1 μάγειρα. Στην υλοποίησή μας χρησιμοποιούμε δύο συσχετίσεις (relations) εκ των οποίων η μία (Judges) συνδέει τους μάγειρες με τα επεισόδια για την αποθήκευση των κριτών ανά επεισόδιο και η δεύτερη (Assignment) συσχετίζει τα επεισόδια με τους μάγειρες, τις εθνικές κουζίνες και τις συνταγές. Ένας περιορισμός του E-R μοντέλου είναι ότι δεν μπορεί να εκφράσει συσχετίσεις μεταξύ συσχετίσεων.

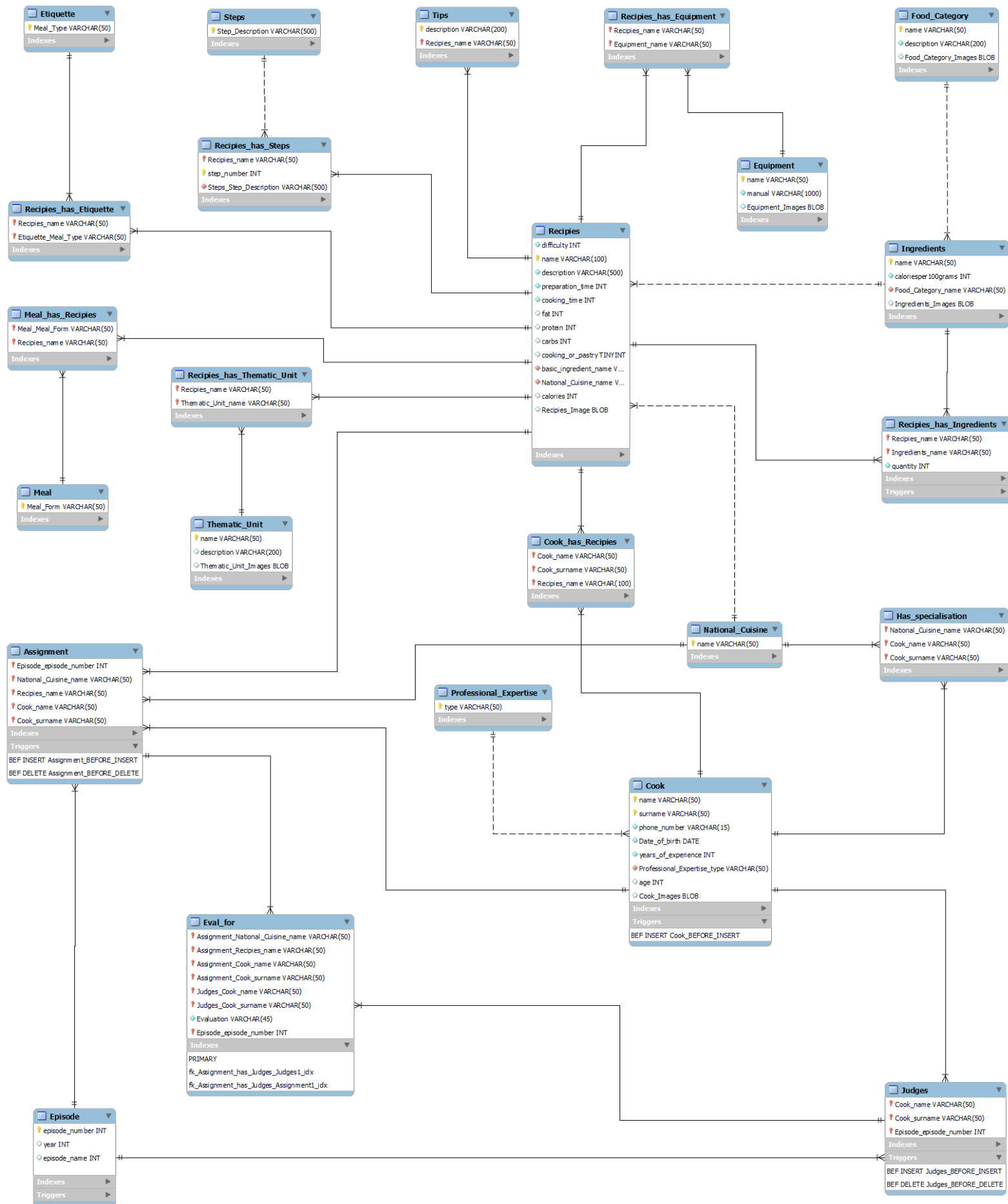
Μία τέτοια δομή είναι αναγκαία στην περίπτωση μας προκειμένου να διατηρούμε την πληροφορία της βαθμολογίας του κάθε κριτή σε κάθε μάγειρα ανά επεισόδιο. Ο καλύτερος τρόπος μοντελοποίησης αυτής της κατάστασης είναι η χρήση συνάθροισης. Δεν προσθέσαμε στο ER τα ορθογώνια για τις 2 συναθροίσεις διότι θα γινόταν δυσνόητη η απεικόνιση του μοντέλου, οπότε απεικονίσαμε τη συσχέτιση Eval_for να ενώνει τις συσχετίσεις Assignment και Judges, προσθέτοντας το επιπλέον attribute evaluation.

2. Σχεσιακό Σχήμα

Ολοκληρώνοντας τον εννοιολογικό σχεδιασμό της βάσης προχωρήσαμε στον λογικό σχεδιασμό μετατρέποντας το E-R διάγραμμα σε σχεσιακό σχήμα (relational schema). Ακολουθήσαμε τις μεθόδους που διδαχθήκαμε στο μάθημα, τροποποιώντας την υλοποίησή μας όπου αυτό κρινόταν απαραίτητο, για την αποφυγή πλεονάζουσας πληροφορίας.

Όπως προαναφέρθηκε, κύριοι πίνακες της βάσης είναι αυτός των συνταγών και των μαγείρων. Κάθε άλλη οντότητα που σχετιζόταν με αυτές μετατράπηκε είτε σε επιπλέον attribute αυτών είτε σε καινούριο πίνακα με foreign keys προς αυτούς. Επίσης οι μη δυαδικές μας σχέσεις μετατράπηκαν σε δυαδικές με χρήση βοηθητικών πινάκων. Όσον αφορά την υλοποίηση των περιορισμών των ελάχιστης και μέγιστης πληθυκότητας κάθε οντότητας που συμμετείχε σε σχέση, αυτό έγινε στο επίπεδο εισαγωγής των δεδομένων.

Το σχεσιακό μας μοντέλο συνοψίζεται στο παρακάτω σχήμα:



Τα ευρετήρια που δημιουργούνται αυτόματα λόγω των primary και foreign keys είναι:

1. Στον πίνακα Recipies, έχουμε το name, δηλαδή το όνομα της συνταγής στην οποία αναφερόμαστε (Primary Key).
2. Στον ίδιο πίνακα υπάρχει το attribute basic_ingredient που είναι foreign key στον πίνακα Ingredients και έχουμε ένα attribute National_Cuisine_name που είναι foreign key στον πίνακα National_Cuisine.
3. Ο πίνακας Ingredients έχει primary key το name του.
4. Ο πίνακας που περιέχει τα υλικά της κάθε συνταγής (Recipies_has_Ingredients) έχει foreign keys στον πίνακα των συνταγών και των υλικών.
5. Ο πίνακας Equipment έχει primary key το όνομα name του εξοπλισμού και ο Recipies_has_Equipment έχει foreign keys στα πίνακα του εξοπλισμού και των συνταγών.
6. Ο πίνακας των συμβουλών Tips έχει foreign key στον πίνακα των συνταγών.
7. Ο πίνακας Recipies_has_Steps στον οποίον αποθηκεύουμε τα βήματα κάθε συνταγής έχει primary key το ζεύγος Recipies_name και step_number και foreign keys στον πίνακα των συνταγών των βημάτων Steps.
8. Ο πίνακας των ετικετών etiquette έχει primary key το όνομα της ετικέτας Meal_Type.
9. Ο πίνακας Recipies_has_Etiquette έχει primary key το ζεύγος Recipies_name και Etiquette_Meal_Type και foreign key προς τον πίνακα των συνταγών και των ετικετών αντίστοιχα.
10. Ο πίνακας των μορφών γευμάτων Meal έχει primary key το Meal_Form και ο πίνακας Meal_has_Recipies έχει primary key το ζεύγος Recipies_name, Meal_meal_form και foreign keys προς τον πίνακα των συνταγών και των γευμάτων αντίστοιχα.
11. Ο πίνακας των θεματικών ενοτήτων Thematic_Unit έχει primary key το όνομα name και ο πίνακας Recipies_has_Thematic_Unit έχει primary key το ζεύγος Recipies_name, Thematic_Unit_name και foreign keys στον πίνακα των συνταγών και των θεματικών ενοτήτων αντίστοιχα.
12. Ο πίνακας των εθνικών κουζίνας National_Cuisine έχει primary key το όνομα name.
13. Ο πίνακας της εξειδίκευσης των μαγείρων Has_specialisation έχει primary key την τριάδα National_Cuisine_name, Cook_name, Cook_surname και foreign keys προς τον πίνακα των Εθνικών κουζινών και των μαγείρων αντίστοιχα.
14. Ο πίνακας των μαγείρων Cook έχει primary key το ζεύγος name, surname και foreign key στον πίνακα της επαγγελματικής κατάρτισης Professional_Expertise.
15. Ο πίνακας της επαγγελματικής κατάρτισης Professional_Expertise έχει primary key το είδος type.
16. Ο πίνακας των επεισοδίων έχει primary key τον αριθμό των επεισοδίων episode_number.

17. Ο πίνακας Assignment έχει primary key το σύνολο Αριθμός Επεισοδίου Episode_episode_number, το όνομα εθνικής κουζίνας National_Cuisine_name, το όνομα της συνταγής Recipies_name, το ονοματεπώνυμο του μάγειρα, Cook_name, Cook_surname με foreign keys προς τους αντίστοιχους πίνακες.
18. Ο κύριος πίνακας της βάσης eval_for έχει primary key τον συνδυασμό εθνικής κουζίνας, όνομα συνταγής, μάγειρα, κριτή και αριθμό επεισοδίου με foreign keys προς τα αντίστοιχα tables.

Πέρα από αυτά τα ευρετήρια που δημιουργούνται αυτόματα λόγω των primary και foreign κλειδιών, υλοποιήσαμε και ένα ακόμα ευρετήριο για την βελτιστοποίηση της απόδοσης των ερωτημάτων επί της βάσης. Πιο συγκεκριμένα έχουμε:

19. Την ηλικία του μάγειρα age.

4. DDL και DML

Έχοντας έτοιμο το σχεσιακό μοντέλο, αξιοποιήσαμε το εργαλείο MySQLWorkbench για την ανάπτυξη του κώδικα της εφαρμογής σε MySQL. Πιο συγκεκριμένα, εκτός από τους ορισμούς των tables και των αντίστοιχων κλειδιών και ευρετηρίων τους, που θα περιγράψουμε πολύ αναλυτικά στην συνέχεια, δημιουργήσαμε τα διάφορα triggers που ελέγχουν τους περιορισμούς των δεδομένων της βάσης ή υπολογίζουν δυναμικά άλλα δεδομένα.

1. Ο υπολογισμός των θερμίδων της κάθε συνταγής υπολογίζεται από τις ποσότητες των υλικών που χρησιμοποιεί και τις θερμίδες αυτών ανά 100 γραμμάρια. Κάθε φορά που προσθέτουμε στην βάση μας πληροφορία της μορφής (όνομα συνταγής, υλικό) στον πίνακα Recipes_has_ingredients, ένα trigger (after insert) αναλαμβάνει να υπολογίσει τον καινούριο αριθμών θερμίδων της συνταγής. Έτσι όταν θα έχουμε εισάγει όλα τα υλικά της συνταγής, θα έχουμε και τον τελικό αριθμό των θερμίδων της.
2. Κάθε φορά που εισάγουμε έναν νέο μάγειρα, η ηλικία του υπολογίζεται ως η διαφορά του timestamp την στιγμή της εισαγωγής και της ημερομηνίας γέννησής του (trigger before insert).
3. Στην βάση μας αρκεί να φορτώνουμε τους αριθμούς των επεισοδίων που θα θέλουμε να έχουμε και ένα trigger before insert στον πίνακα των επεισοδίων αναλαμβάνει να υπολογίσει τον αριθμό χρονιάς και σχετικό αριθμό επεισοδίου.
4. Κάθε φορά που προσθέτουμε κάποιον κριτή σε κάποιο επεισόδιο είμαστε σίγουροι ότι αυτό το επεισόδιο δεν θα έχει πάνω από 3 κριτές, χάρη στην δημιουργία ενός trigger (before insert) που ελέγχει αν θα τηρείται ο αντίστοιχος περιορισμός. Αντίστοιχα υπάρχει trigger για να μην επιτρέπει την μη αποδεκτή διαγραφή κριτή.
5. Ο πίνακας όπου αποθηκεύουμε τις αναθέσεις μαγείρων, εθνικών κουζινών και συνταγών ανά επεισόδιο (Assignment) διαθέτει αντίστοιχα triggers με τον πίνακα

των κριτών για να διασφαλίζεται ότι τα επεισόδια κάθε χρονιάς θα είναι πράγματι 10.

Στο σημείο αυτό η βάση που δημιουργήσαμε είναι κατάλληλη για να πληρωθεί με δεδομένα (data loading). Παρόλα αυτά, ο φυσικός σχεδιασμός της αποφασίσαμε να περιλαμβάνει και stored procedures τα οποία θα δημιουργούν τις αναθέσεις του διαγωνισμού, παράγοντας ουσιαστικά τα «επεισόδια»!

Με την βοήθεια LLMs παράξαμε τα δεδομένα σχετικά με τις συνταγές και διασφαλίσουμε πως αυτά δεν είναι απλώς fake dump data αλλά αληθινές συνταγές που κάποιος θα μπορούσε να μαγειρέψει. Τα περιεχόμενα των αρχείων csv φορτώθηκαν στην βάση μέσω ενός Python script που δημιουργήσαμε.

Καλώντας την ρουτίνα populate_all δημιουργούνται 10 χρονιές του διαγωνισμού, με 10 επεισόδια η καθεμία, οι αναθέσεις των μαγείρων και των κριτών και η βαθμολόγηση των πιάτων. Η ρουτίνα αυτή καλεί τις παρακάτω ρουτίνες με την εξής σειρά:

1. Πρώτη καλείται η ρουτίνα for_loop_judges που καλεί 100 φορές (όσες και ο αριθμός των επεισοδίων) την ρουτίνα create_judges. Με την σειρά της η ρουτίνα αυτή διαλέγει κάποιον τυχαίο μάγειρα, ελέγχει αν αυτός δεν έχει χρησιμοποιηθεί ως κριτής και στα δύο προηγούμενα επεισόδια (κλήση ρουτίνας check_judge_not_3_in_a_row) και αν αυτός περάσει τον έλεγχο, τον εισάγει στον πίνακα των κριτών. Έτσι εισάγονται 3 κριτές ανά επεισόδιο (300 κριτές συνολικά).
2. Δεύτερη καλείται η ρουτίνα create_assignments που είναι και η πιο χρονοβόρα. Η ρουτίνα αυτή θα δημιουργήσει τις 10 αναθέσεις εθνικών κουζίνας σε μάγειρες και μετά σε συνταγές (συνολικά 1000 αναθέσεις). Κάθε φορά διαλέγεται τυχαία μία εθνική κουζίνα και από αυτή ένας μάγειρας στην οποία έχει ειδικευση και μία συνταγή από αυτή την κουζίνα που ο μάγειρας ξέρει να μαγειρεύει. Καλούμε την ρουτίνα check_valid_assignment_insert που ελέγχει ότι η ανάθεση αυτή είναι έγκυρη (ο μάγειρας πρέπει να έχει εξειδίκευση στην εθνική κουζίνα, η συνταγή να ανήκει στην εθνική κουζίνα, και η εθνική κουζίνα, ο μάγειρας και η συνταγή να μην έχουν ξαναεμφανιστεί στο ίδιο επεισόδιο). Έπειτα ελέγχεται αν ο μάγειρας δεν έχει χρησιμοποιηθεί 3 φορές συνεχόμενα (κλήση ρουτίνας check_cook_not_3_in_a_row) και αντίστοιχα για την συνταγή (check_recipe_not_3_in_a_row) και για την εθνική κουζίνα (check_national_cuisine_not_3_in_a_row).
3. Τελευταία καλείται η ρουτίνα populate_eval_for που αναλαμβάνει να κάνει το inner join μεταξύ του πίνακα των assignments και των κριτών ανά επεισόδιο καθώς και να αναθέσει τυχαία τις βαθμολογίες τους σε κάθε προσπάθεια.

5. Winner

Μία από τις πιο χρήσιμες πληροφορίες που μπορεί κανείς να εξάγει από την βάση δεδομένων του διαγωνισμού, είναι οι μάγειρες-νικητές του κάθε επεισοδίου. Ως νικητής ορίζεται ο μάγειρας με την υψηλότερη βαθμολογία, ενώ σε περίπτωση ισοβαθμίας κερδίζει ο μάγειρας με την υψηλότερη επαγγελματική κατάρτιση. Η διάταξη της επαγγελματικής κατάρτισης με βάση την εκφώνηση είναι: Γ΄ μάγειρας < Β΄ μάγειρας < Α΄ μάγειρας < Sous chef < Chef. Σε περίπτωση δεύτερης ισοβαθμίας ο μάγειρας επιλέγεται τυχαία. Τρέχοντας τον κώδικα του επερωτήματος Winner, υπολογίζεται η μέση βαθμολογία του κάθε μάγειρα ανά επεισόδιο και στην συνέχεια εφαρμόζονται τα κριτήρια που προαναφέρθηκαν. Το αποτέλεσμα είναι ένας πίνακας 100 εγγραφών με τον αριθμό επεισοδίου, το ονοματεπώνυμο του νικητή, η μέση βαθμολογία του και η επαγγελματική του κατάρτιση.

6. Authentication

Πρόσβαση στην βάση δεδομένων πρέπει να έχουν τόσο οι διαχειριστές του διαγωνισμού όσο και οι διαγωνιζόμενοι. Δημιουργούμε έναν χρήστη “admin” στον οποίο δίνουμε όλα τα δικαιώματα επί της βάσης, όπως τροποποίηση όλων των δεδομένων, δημιουργία αντιγράφου ασφαλείας και επαναφορά του συστήματος. Η είσοδος στην βάση γίνεται μετά την εισαγωγή του συνθηματικού “secure_password”.

Για την υλοποίηση των λογαριασμών των μαγείρων, θα έπρεπε με την βοήθεια ενός trigger after insert, κάθε φορά που εισάγεται στην βάση ένας μάγειρας να δημιουργείται ένας αντίστοιχος λογαριασμός. Αυτός θα είχε πρόσβαση σε μονάχα 2 views, ένα που θα περιείχε τις προσωπικές του πληροφορίες και ένας δεύτερο με τις πληροφορίες των συνταγών που του έχουν ανατεθεί στον διαγωνισμό. Αυτή η δυναμική δημιουργία χρηστών δεν συνιστά καλή πρακτική και κάνει την βάση μας ευάλωτη σε SQL injections, έτσι το εργαλείο του MYSQLWorkbench εμπόδιζε αυτή την υλοποίηση. Εάν η εφαρμογή μας συνοδευόταν από UI τότε το authentication θα γινόταν σε αυτό το επίπεδο και το πρόβλημά μας θα πρακκάμπτοταν. Επί της παρούσης αρκεστήκαμε στην δημιουργία ενός μονάχα μάγειρα χρήστη με τα αντίστοιχα views και δικαιώματα αλλαγών των δεδομένων που περιέχουν. Αξίζει να σημειωθεί πως έχουμε φροντίσει τα views να είναι updatable, δηλαδή να μην χρησιμοποιούν aggregate functions, group by ή having clauses, ενώσεις ή τομές, subqueries στο select και στο from και natural joins.

7. Οδηγίες εγκατάστασης

1. Κάνετε clone το GitHub repository της εργασίας.

<https://github.com/konstantinosk31/Databases-I-NTUA>

2. Ανοίξτε τον MySQL Server. (πχ από το XAMPP)

3. Εκτελέστε την εντολή source C:/Path_to_Repository/sql/ddl.sql;

4. Εκτελέστε την εντολή source C:/Path_to_Repository/sql/dml.sql;

5. Εκτελέστε την εντολή source C:/Path_to_Repository/users/users.sql;

6. Εκτελέστε την εντολή source C:/Path_to_Repository/users/views.sql;

7. Η βάση μας είναι έτοιμη, με exit μπορείτε στην συνέχεια να συνδεθείτε ως κάποιος χρήστης, είτε διαχειριστής (admin), είτε μάγειρας (Cook).

Να σημειωθεί ότι δεν επισυνάψαμε τον κώδικα του DDL ή του DML, καθώς πρόκειται για αρχεία των 1000+ γραμμών και των 10000+ γραμμών αντίστοιχα.