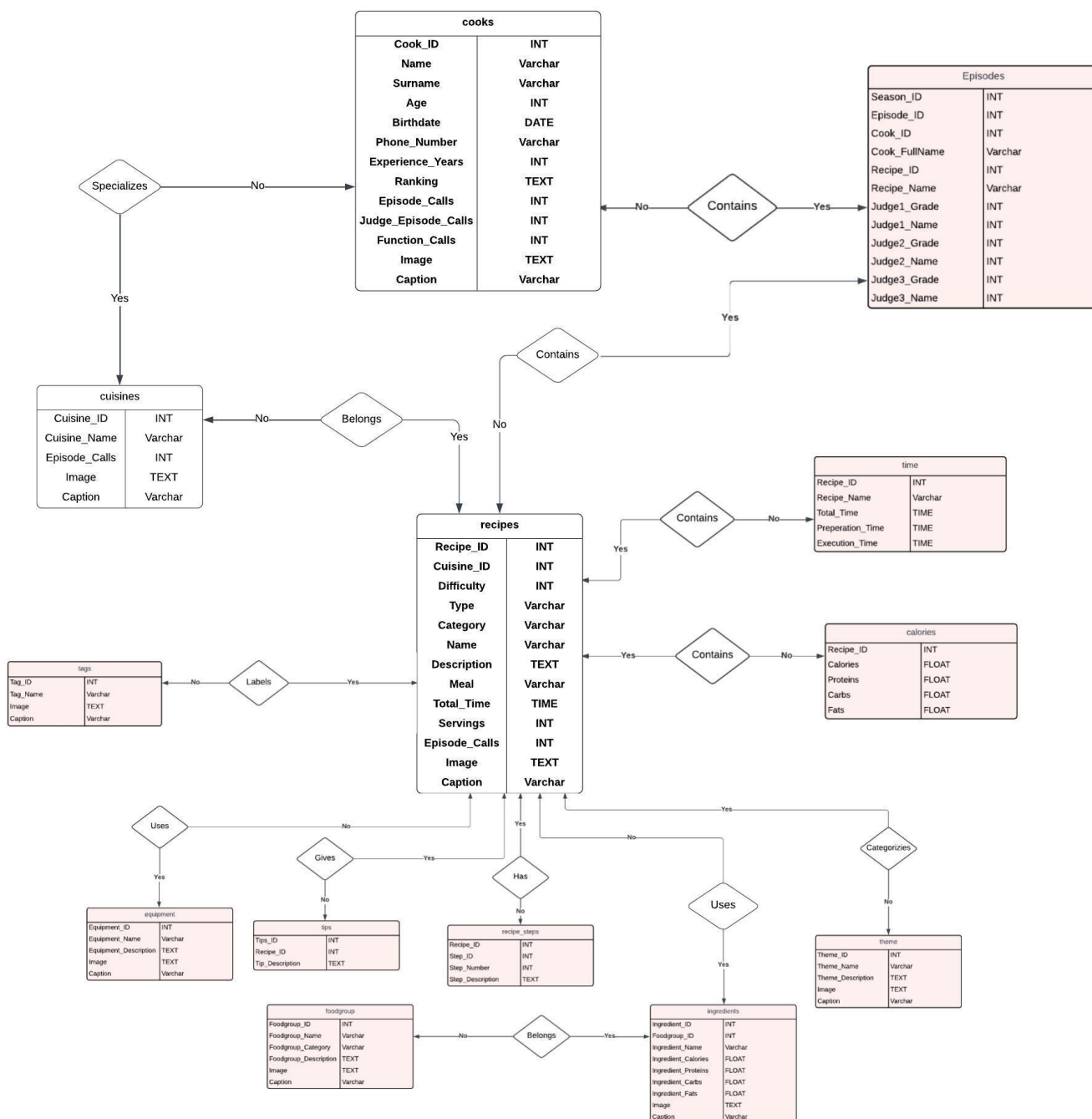




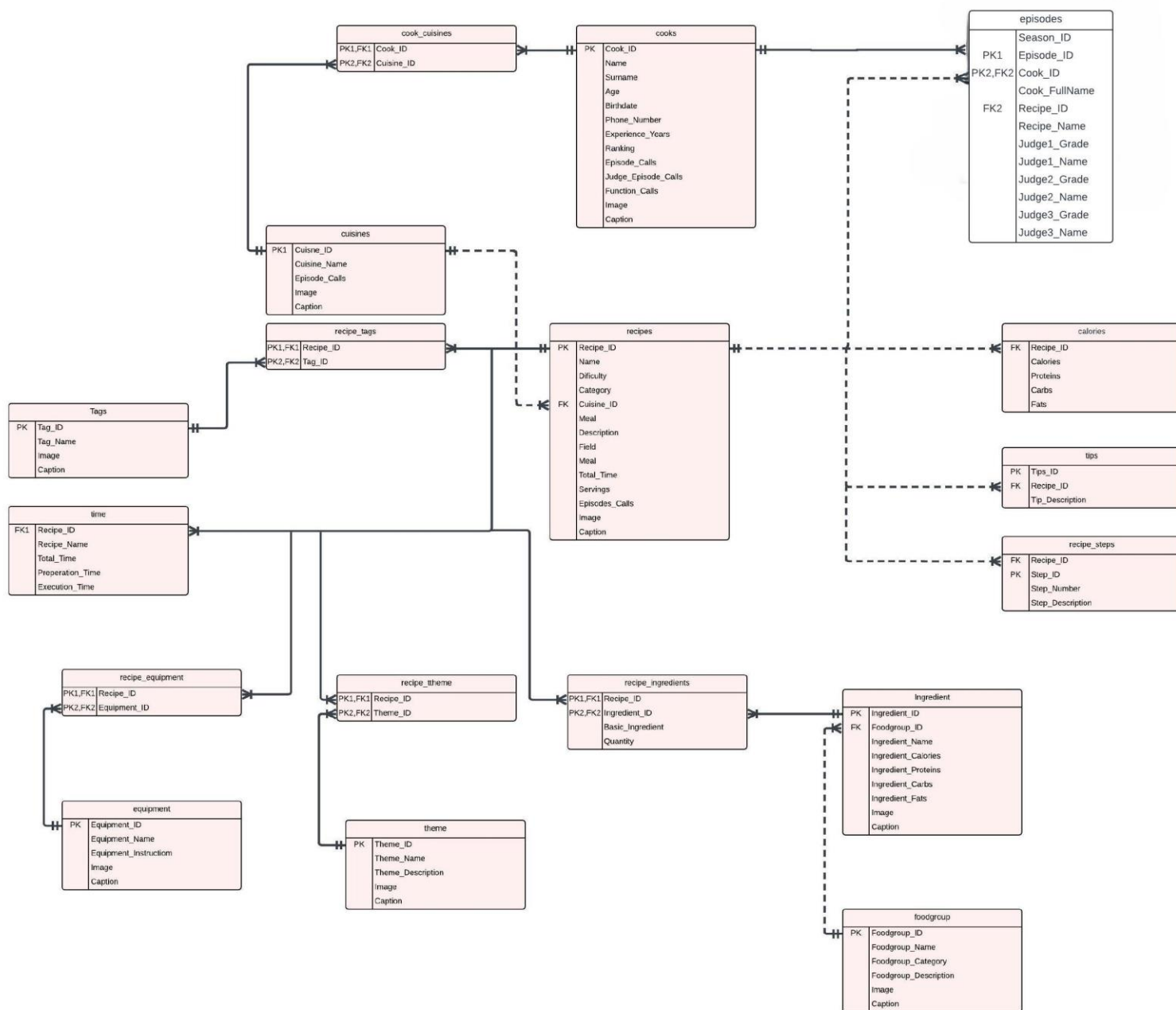
Εξαμηνιαία Εργασία	2023-2024
AM	03121914(Γεώργιος Φλώρος)
AM	03121827(Γεώργιος Ισόπουλος)
AM	03121838(Νικόλαος Μπαρκονίκος)

## Εξαμηνιαία Εργασία : Databases

1) Ακολουθεί το **Entity Relations** (ER) της Βάσης Δεδομένων μας:



Παραπάνω βλέπουμε το σχήμα σε διάγραμμα οντοτήτων - συσχετίσεων (Entity-Relationship Diagram).



Η Βάση Δεδομένων (MasterChef) που φτιάξαμε αποτελείται από 18 πίνακες (12 βασικούς πίνακες και 6 μεταβατικούς).

- Ο βασικότερος πίνακας μας είναι ο πίνακας **recipes**, ο οποίος περιέχει τα απαραίτητα στοιχεία/columns για την ανάλυση της κάθε συνταγή (**Name, Surname, Category, Difficulty** κ.λπ.).
- Έχουμε φτιάξει ορισμένους μεταβατικούς πίνακες (**Recipe\_Equipment, Recipe\_Tags, Recipe\_Steps, Recipe\_Ingredients, Recipe\_Theme**), οι οποίοι συνδέουν τους πίνακες **Equipment, Tags, Steps, Ingredients, Theme**, με τον **Recipes**, ώστε να επιτύχουμε σχέση Many-To-Many μεταξύ τους.
- Η Βάση Δεδομένων διαθέτει επίσης και πίνακες όπως **Foodgroup, Cooks**, οι οποίοι συνδέονται με τους πίνακες **Ingredients** (One-To-Many) και **Cuisines, Episodes** αντίστοιχα.

Γενικά η Βάση Δεδομένων μας επιτεύχθηκε, δημιουργώντας πίνακες σύμφωνα με τις υπογραμμισμένες λέξεις της εκφώνησης, ενώ οι σχέσεις μεταξύ τους προέκυψαν με την λογική, και πολλών trial-and-errors που έγιναν κατά το χτίσιμο της Βάσης μας (πχ. Σχέση Cooks με Cuisines Many-To-Many) και κατά προσπάθεια επίλυσης του 3<sup>ου</sup> θέματος της εκφώνησης (Ερωτήματα πάνω στην DB).

Για την βέλτιστη επίλυση των ερωτημάτων του 3<sup>ου</sup> θέματος δημιουργήσαμε τα κατάλληλα ευρετήρια, με σκοπό να ελαχιστοποιήσουμε τον χρόνο αναμονής επιστροφής των αποτελεσμάτων από τα procedures που φτιάξαμε για κάθε Question. Βέβαια, δεν είναι σωστή πρακτική να θέσουμε όλα μας τα columns ως ευρετήρια, καθώς επιβραδύνουν λειτουργίες εγγραφής, ενημέρωσης και διαγραφής. Τα ευρετήρια που φτιάξαμε είναι τα εξής:

- Episode\_Calls του recipes (αντιστοίχιση της κάθε συνταγής με επεισόδιο. Χρησιμοποιείται σε όλα τα Procedures/Questions)
- Difficulty του recipes (αναζήτηση της δυσκολίας της κάθε συνταγής [Question 3.12])
- Name του recipes (χρησιμοποιείται για την δημιουργία του Episodes table, άρα έμμεσα σε όλα σχεδόν τα Procedures/Questions)
- Carbs του calories (εύρεση των υδατανθράκων ανά συνταγή [Δυναμικοί υπολογισμοί με βάση τα συστατικά της κάθε συνταγής, [Question 3.9])
- Age του Cooks (αναζήτηση της ηλικίας του κάθε μάγειρα [Question 3.9])
- Cuisine\_Name του Cuisines (εύρεση του ονόματος της κάθε κουζίνας [Question 3.10, 3.2, 3.1])
- Season\_ID του episodes (Χρησιμοποιείται σχεδόν όλα τα Procedures/Questions, αφού ισχύει ότι One Season = One Year)
- Recipe\_Name του episodes (Χρησιμοποιείται σχεδόν όλα τα Procedures/Questions)
- Judge1\_Name του episodes (Χρησιμοποιείται σχεδόν όλα τα Procedures/Questions)
- Judge2\_Name του episodes (Χρησιμοποιείται σχεδόν όλα τα Procedures/Questions)
- Judge3\_Name του episodes (Χρησιμοποιείται σχεδόν όλα τα Procedures/Questions)
- Foodgroup\_Name του foodgroup (Αντιστοίχιση της κάθε συνταγής με μία κατηγορία φαγητού βάση του Basic\_Ingredient της [Question 3.15])
- Tag\_Name του tags (Εύρεση της ετικέτας που έχουμε δώσει σε Recipe [Question 3.6])
- Theme\_Name του theme (Εύρεση της θεματικής ενότητας της κάθε συνταγής [Question 3.14])

Τα υπόλοιπα Indexes αποτελούνται από ευρετήρια τύπου B+ tree, δηλαδή Primary και Foreign Keys.

- Recipe\_ID, Cuisine\_ID, Cook\_ID, Cuisine\_ID, Equipment\_ID, Foodgroup\_ID, Ingredients\_ID, Tags\_ID, Theme\_ID, Tip\_ID.

Ορισμένα από τα παραπάνω indexes ίσως να μην είναι κατάλληλα, καθώς παρόλο που ζητούνται για την καλύτερη απόδοση ενός Procedure/Question, χρησιμοποιούνται σπανίως.

- 2) Τα [DDL](#) και [DML](#) αρχεία βρίσκονται στο GitHub Repository, που συντελούν την βάση δεδομένων μας.
- 3) Η διαδικασία εγκατάστασης της βάσης καθώς και της εφαρμογής μας βρίσκεται στο αρχείο [manual.pdf](#)
- 4) [Link για το GitHub repository της εργασίας.](#)

### Πληροφορίες για 3.6, 3.8:

Μέσα στο εναλλακτικό query plan χρησιμοποιήσαμε force index. Με αυτόν τον τρόπο ορίζουμε στην βάση ποιον index να χρησιμοποιήσει για να εκτελέσει το συγκεκριμένο query. Συνήθως αυτή η υλοποίηση υπάρχει σε queries που μπορεί να θεωρηθούν αρκετά κοστοβόρα. Στην δικιά μας περίπτωση είδαμε ποια είναι τα possible keys σε κάποιο SELECT και ανάλογα ορίσαμε το FORCE INDEX ώστε μην χρειάζεται να διαλέξει μόνη της η βάση αλλά και να ορίσουμε το βέλτιστο.

Παρατηρώντας το trace structure των queries που φτιάξαμε μπορούμε να δούμε τις ενέργειες του query optimizer στην προσπάθεια του να μειώσει το κόστος. Σε κάθε SELECT θα έχουμε 2 βήματα στην JSON : join\_preparation, join\_optimization. Το βήμα join\_preparation κάνει απλά αλλαγές στο query για να το διαβάσει διαφορετικά, το βήμα join\_optimization κάνει την περισσότερη δουλειά για να βελτιστοποιήσει το query.