Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Βάσεις Δεδομένων

Αναφορά Εξαμηνιαίας εργασίας

Δέσποινα – Χριστίνα Μαρκάτου (03121433)

Γιώργος Παρασκευάς Παπανδρικόπουλος (03121102)

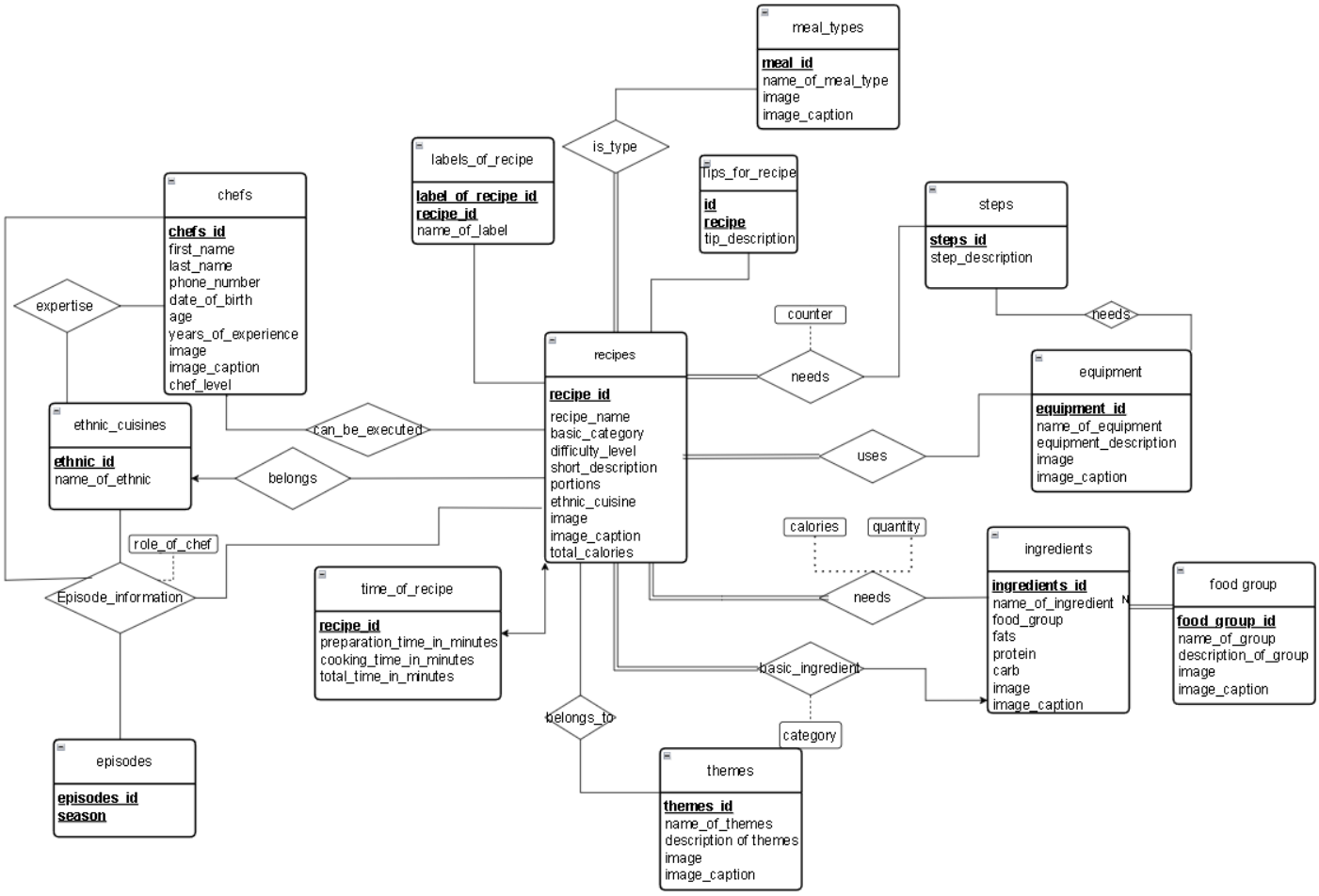
Χαράλαμπος Σπυρόπουλος (03121122)

Περιεχόμενα:

1. Διαγράμματα Βάσης Δεδομένων

1.1 Διάγραμμα Οντοτήτων-Συσχετίσεων (Entity-Relationship Diagram)

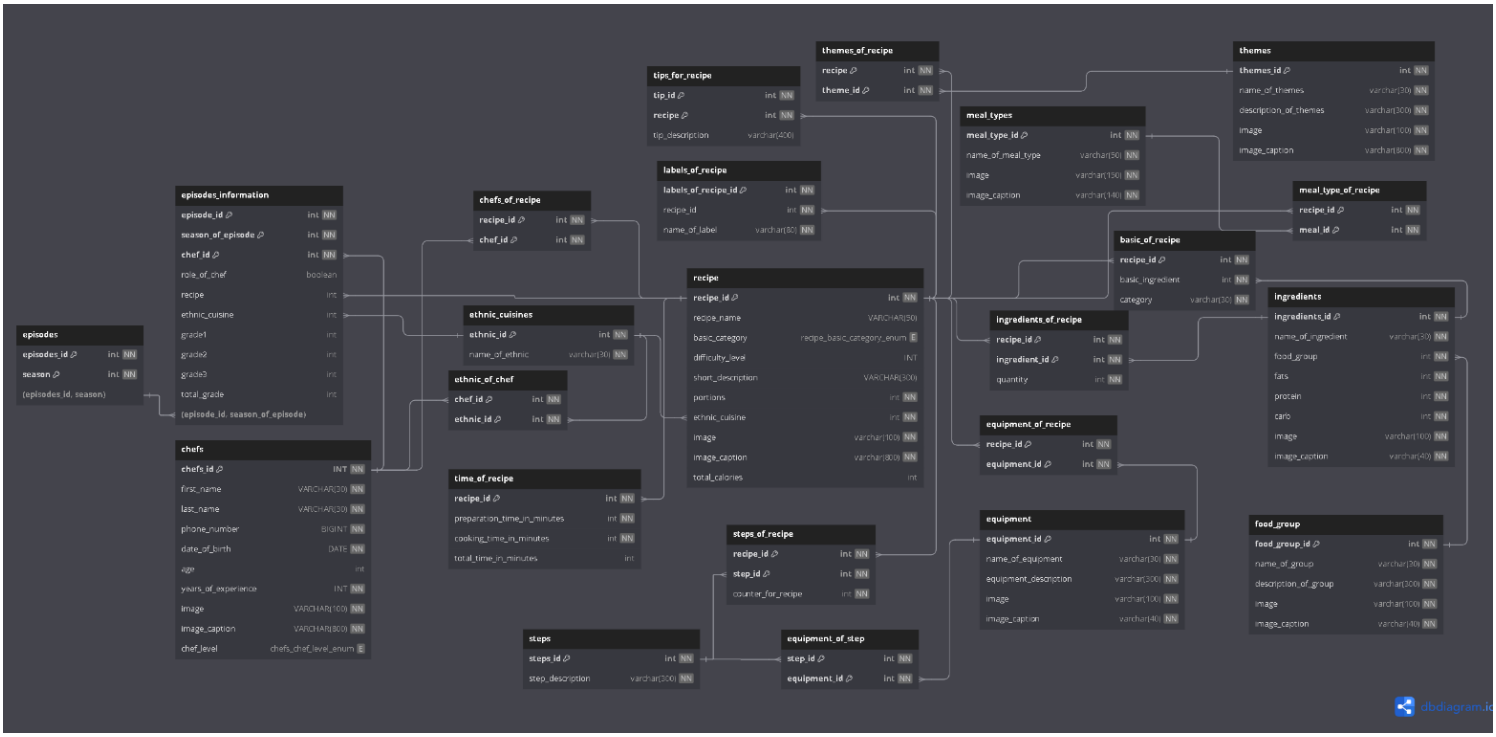
Παρακάτω φαίνεται το ER διάγραμμα της βάσης δεδομένων που υλοποιήσαμε, σύμφωνα με τις προδιαγραφές που δόθηκαν:



Το διάγραμμα δημιουργήθηκε χρησιμοποιώντας το εργαλείο draw-io και εμφανίζει τις οντότητες (entities) της βάσης μας, τα χαρακτηριστικά τους (attributes), καθώς και τις σχέσεις (relationships) που τις συνδέουν (τα attributes που χαρακτηρίζουν με μοναδικό τρόπο τα entities συμβολίζονται με έντονο χρώμα και είναι υπογραμμισμένα).

1.2 Σχεσιακό διάγραμμα (Relational Diagram)

Παρακάτω φαίνεται το αντίστοιχο σχεσιακό διάγραμμα της βάσης δεδομένων:



Το relational model δημιουργήθηκε στην εφαρμογή bddiagram.io εισάγοντας το ddl script, που δημιουργήσαμε στο MySQL Workbench. Στο διάγραμμα εμφανίζονται πίνακες που αντιστοιχούν στα entities του ERD, καθώς και επιπλέον πίνακες που στο ERD απεικονίζονταν ως σχέσεις. Επιπλέον φαίνονται οι τύποι δεδομένων των μεταβλητών και αποτυπώνεται μία πιο ξεκάθαρη εικόνα ως προς των τρόπο που συνδέονται οι πίνακες με την χρήση primary και foreign keys.

2. Ευρετήρια

Η δημιουργία των indexes έχει ως στόχο να επιταχύνει την αναζήτηση και την ανάκτηση δεδομένων, αφού δεν θα είναι πλέον απαραίτητη η πλήρης σάρωση του πίνακα. Γενικά επιλέγουμε να δημιουργήσουμε indexes για τις στήλες που χρησιμοποιούνται συχνά στις εντολές ORDER BY και GROUP BY, καθώς και σε ερωτήματα που συνδυάζουν δεδομένα από πολλούς πίνακες, όπως στις εντολές JOIN. Επίσης, επιλέγουμε στήλες που μπορούν να πάρουν πολλές διαφορετικές τιμές και όχι για παράδειγμα στήλες που αντιστοιχούν σε boolean τιμές. Πρέπει όμως να λάβουμε υπόψιν ότι παρόλο που τα indexes επιταχύνουν την ανάκτηση δεδομένων, μπορούν να επιβραδύνουν τις λειτουργίες INSERT, UPDATE και DELETE, διότι το index πρέπει να ενημερώνεται κάθε φορά

που αλλάζουν τα δεδομένα. Επίσης για πολύ μικρούς πίνακες, το κόστος της δημιουργίας και συντήρησης του index μπορεί να μην αντισταθμίζει το όφελος.

Έτσι, σύμφωνα με τις παραπάνω παρατηρήσεις, καθώς και τις ερωτήσεις που καλούμαστε να κάνουμε στην βάση, δημιουργήσαμε τα παρακάτω ευρετήρια:

• chefs_id του table chefs, για γρήγορη εύρεση των chefs

• recipe_id του table recipe, για γρήγορη εύρεση των recipes

• ingredients_id του table ingredients, για γρήγορη εύρεση των ingredients

• recipe_name του table recipe, διότι η αναζήτηση των συνταγών γίνεται μέσω του ονόματός τους

• (first_name, last_name) του table chefs, διότι η αναζήτηση των μαγείρων γίνεται μέσω του ονόματός τους

• chef_id του table chefs_of_recipe, για γρήγορη εύρεση των συνταγών που μπορεί να εκτελέσει κάποιος μάγειρας

• recipe_id του table ingredients_of_recipe, για γρήγορη εύρεση των υλικών που απαιτούνται για την εκτέλεση μίας συνταγής

• chef_id του table ethnics_of_chef, για γρήγορη εύρεση των εθνικών κουζινών που αντιστοιχούν σε κάποιον μάγειρα

• chef_id του table episodes_information, διότι σε πολλά ερωτήματα πρέπει να εντοπίσουμε τους μάγειρες που συμμετέχουν σε κάποιο επεισόδιο

• ethnic_cuisine του table episodes_information, διότι σε πολλά ερωτήματα πρέπει να εντοπίσουμε τις εθνικές κουζίνες που εμφανίζονται σε κάποιο επεισόδιο

• recipe του table episodes_information, διότι σε πολλά ερωτήματα πρέπει να εντοπίσουμε τις συνταγές που εκτελούνται σε κάποιο επεισόδιο

• season_of_episode του table episodes_information, διότι σε πολλά ερωτήματα πρέπει να χωρίσουμε τα αποτελέσματα με βάση τη σεζόν στην οποία ανήκουν

• recipe_id του table equipment_of_recipe, για γρήγορη εύρεση του εξοπλισμού που απαιτείται για την εκτέλεση κάποιας συνταγής

• recipe_id του table meal_type_of_recipe, για γρήγορη εύρεση των τύπων γεύματος που αντιστοιχούν σε κάποια συνταγή

• recipe_id του table steps_of_recipe, για γρήγορη εύρεση των βημάτων που πρέπει να ακολουθηθούν για την εκτέλεση μίας συνταγής

• themes_id του table themes, για γρήγορη εύρεση των θεματικών ενοτήτων

• recipe του table themes_of_recipe, για γρήγορη εύρεση των θεματικών ενοτήτων στις οποίες εντάσσεται μία συνταγή

• (recipe_id, name_of_label) του table labels_of_recipe, για γρήγορη εύρεση των ετικετών που αντιστοιχούν σε μία συνταγή, καθώς και το όνομά τους

Στα ευρετήρια συμπεριλάβαμε και κάποια που αναφέρονται στο primary key ενός table για λόγους πληρότητας, όμως για τα primary keys δημιουργείται έτσι και αλλιώς ευρετήριο κατά τον ορισμό τους. Υπάρχει πιθανότητα με την πάροδο του χρόνου να παρατηρηθούν διαφορετικές απαιτήσεις όσον αφορά στην συχνότητα εμφάνισης των διάφορων ερωτήματων, οπότε και να χρειαστεί να αλλάξουμε τα indexes με βάση αυτές.

3. Υλοποίηση εφαρμογής

Για τον σχεδιασμό και την υλοποίηση της βάσης δεδομένων χρησιμοποιήθηκε το σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων mySQL.

3.1 DDL script

Ακολουθεί ο κώδικας σε sql του DDL script για την δημιουργία των πινάκων της βάσης μας, μαζί με τους περιορισμούς ακεραιότητας στα διάφορα χαρακτηριστικά των πινάκων. Για την δημιουργία της βάσης έχουμε χρησιμοποιήσει και διάφορα triggers και procedures, για να ελέγχουμε την σωστή εισαγωγή δεδομένων στην βάση μας, καθώς και την σωστή τελική κατάσταση της βάσης. Αυτά τα μέρη του κώδικα περιέχονται στο git repo μας στους ακόλουθους συνδέσμους:

https://github.com/Madeperf/DataBaseProject/blob/main/DDLscript.sql

https://github.com/Madeperf/DataBaseProject/blob/main/Procedures.sql

```
DROP SCHEMA if exists `masterchef`;

CREATE SCHEMA `masterchef`;
use masterchef;

-- Απενεργοποίηση των περιορισμών ξένων κλειδιών
SET foreign_key_checks = 0;

-- Διαγραφή όλων των πινάκων
DROP TABLE IF EXISTS ethnic_cuisines;
DROP TABLE IF EXISTS recipe;
DROP TABLE IF EXISTS tips_for_recipe;
DROP TABLE IF EXISTS meal_types;
DROP TABLE IF EXISTS meal_type_of_recipe;
DROP TABLE IF EXISTS steps;
DROP TABLE IF EXISTS steps_of_recipe;
DROP TABLE IF EXISTS equipment;
DROP TABLE IF EXISTS equipment_of_recipe;
DROP TABLE IF EXISTS equipment_of_step;
DROP TABLE IF EXISTS food_group;
DROP TABLE IF EXISTS ingredients;
DROP TABLE IF EXISTS ingredients_of_recipe;
DROP TABLE IF EXISTS time_of_recipe;
DROP TABLE IF EXISTS themes;
```

```
DROP TABLE IF EXISTS themes_of_recipe;
DROP TABLE IF EXISTS chefs;
DROP TABLE IF EXISTS chefs_of_recipe;
DROP TABLE IF EXISTS ethnic_of_chef;
DROP TABLE IF EXISTS labels_of_recipe;
DROP TABLE IF EXISTS basic_of_recipe;
DROP TABLE IF EXISTS episodes;
DROP TABLE IF EXISTS episodes_information;

SET foreign_key_checks = 1;

create table ethnic_cuisines(
ethnic_id  int UNSIGNED NOT NULL AUTO_INCREMENT primary key ,
name_of_ethnic varchar(30) UNIQUE NOT NULL
);

create table recipe(
recipe_id int UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY ,
recipe_name VARCHAR(50) ,
basic_category ENUM("savory","pastry"),
difficulty_level INT UNSIGNED check(difficulty_level BETWEEN 1 AND 5),
short_description VARCHAR(300),
portions int NOT NULL ,
ethnic_cuisine int unsigned NOT NULL,
image varchar(100) not null,
image_caption varchar(800) not null,
total_calories int,
CONSTRAINT FK_recipe_ethnic foreign key (ethnic_cuisine) references ethnic_cuisines(ethnic_id)
ON DELETE RESTRICT ON UPDATE CASCADE
);


create table tips_for_recipe(
tip_id int NOT NULL,
recipe int Unsigned NOT NULL,
tip_description varchar(400),
primary key( tip_id,recipe) ,
CONSTRAINT tips_for_recipe_recipe_id foreign key(recipe) references recipe(recipe_id)
ON DELETE RESTRICT ON UPDATE CASCADE
);


create table meal_types (
meal_type_id int primary key NOT NULL,
name_of_meal_type varchar(50) NOT NULL,
image varchar(150) not null ,
image_caption varchar(140) not null
);

create table meal_type_of_recipe(
recipe_id int unsigned NOT NULL,
```

```
meal_id int NOT NULL,
primary key(recipe_id,meal_id),
CONSTRAINT meal_type_of_recipe_recipe_id foreign key(recipe_id) references recipe(recipe_id)
ON DELETE RESTRICT ON UPDATE CASCADE,
CONSTRAINT meal_type_of_recipe_meal_id foreign key(meal_id) references
meal_types(meal_type_id)
ON DELETE RESTRICT ON UPDATE CASCADE
);


create table steps (
steps_id int UNSIGNED NOT NULL AUTO_INCREMENT primary key ,
step_description varchar(300) NOT NULL
);

create table steps_of_recipe(
recipe_id int unsigned NOT NULL ,
step_id int unsigned NOT NULL,
counter_for_recipe int NOT NULL,
primary key(recipe_id,step_id),
CONSTRAINT steps_of_recipe_recipe_id foreign key(recipe_id) references recipe(recipe_id)
ON DELETE RESTRICT ON UPDATE CASCADE,
CONSTRAINT steps_of_recipe_step_id foreign key(step_id) references steps(steps_id)
ON DELETE RESTRICT ON UPDATE CASCADE
);

create table equipment (
equipment_id int UNSIGNED NOT NULL AUTO_INCREMENT primary key ,
name_of_equipment varchar(30) NOT NULL,
equipment_description varchar(300) NOT NULL,
image varchar(100) not null ,
image_caption varchar(40) not null
);

create table equipment_of_recipe(
recipe_id int unsigned NOT NULL,
equipment_id int unsigned NOT NULL,
primary key(recipe_id,equipment_id),
CONSTRAINT equipment_of_recipe_recipe_id foreign key(recipe_id) references recipe(recipe_id)
ON DELETE RESTRICT ON UPDATE CASCADE,
CONSTRAINT equipment_of_recipe_equipment_id foreign key(equipment_id) references
equipment(equipment_id)
ON DELETE RESTRICT ON UPDATE CASCADE);


create table equipment_of_step(
step_id int unsigned  NOT NULL,
equipment_id int unsigned NOT NULL,
primary key(step_id,equipment_id),
CONSTRAINT equipment_of_step_step_id foreign key(step_id) references steps(steps_id)
ON DELETE RESTRICT ON UPDATE CASCADE,
```

```
CONSTRAINT equipment_of_step_equipment_id foreign key(equipment_id) references
equipment(equipment_id)
ON DELETE RESTRICT ON UPDATE CASCADE
);

create table food_group (
food_group_id int primary key NOT NULL,
name_of_group varchar(30) NOT NULL,
description_of_group varchar(300) NOT NULL,
image varchar(100) not null,
image_caption varchar(40) not null
);

create table ingredients (
ingredients_id int UNSIGNED NOT NULL AUTO_INCREMENT primary key ,
name_of_ingredient varchar(30) NOT NULL,
food_group int NOT NULL,
fats int NOT NULL,
protein int NOT NULL,
carb int NOT NULL,
image varchar(100) not null,
image_caption varchar(40) not null,
CONSTRAINT ingredients_food_group foreign key (food_group) references
food_group(food_group_id)
ON DELETE RESTRICT ON UPDATE CASCADE
);

create table ingredients_of_recipe(
recipe_id int unsigned NOT NULL,
ingredient_id int unsigned NOT NULL,
quantity int NOT NULL,
primary key(recipe_id,ingredient_id),
CONSTRAINT ingredients_of_recipe_recipe_id foreign key(recipe_id) references recipe(recipe_id)
ON DELETE RESTRICT ON UPDATE CASCADE,
CONSTRAINT ingredients_of_recipe_ingredient_id foreign key(ingredient_id) references
ingredients(ingredients_id)
ON DELETE RESTRICT ON UPDATE CASCADE
);


create table time_of_recipe (
recipe_id int unsigned  NOT NULL,
preparation_time_in_minutes int  NOT NULL,
cooking_time_in_minutes int NOT NULL,
total_time_in_minutes int  ,
primary key (recipe_id),
CONSTRAINT time_of_recipe_recipe_id foreign key(recipe_id) references recipe(recipe_id)
ON DELETE RESTRICT ON UPDATE CASCADE
);

create table themes(
```

```
themes_id int UNSIGNED NOT NULL AUTO_INCREMENT primary key ,
name_of_themes varchar(30) NOT NULL,
description_of_themes varchar(300) NOT NULL,
image varchar(100) not null,
image_caption varchar(800) not null
);

create table themes_of_recipe(
recipe int unsigned NOT NULL,
theme_id int unsigned  NOT NULL,
primary key(recipe,theme_id),
CONSTRAINT themes_of_recipe_recipe_id foreign key(recipe) references recipe(recipe_id)
ON DELETE RESTRICT ON UPDATE CASCADE,
CONSTRAINT themes_of_recipe_theme_id  foreign key(theme_id) references themes(themes_id)
ON DELETE RESTRICT ON UPDATE CASCADE
);


CREATE TABLE chefs (
   chefs_id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
   first_name VARCHAR(30) NOT NULL,
   last_name VARCHAR(30) NOT NULL,
   phone_number BIGINT UNIQUE NOT NULL CHECK (phone_number BETWEEN 6900000000
AND 6999999999),
   date_of_birth DATE NOT NULL,
   age int unsigned,
   years_of_experience INT NOT NULL,
   image VARCHAR(100) NOT NULL ,
   image_caption VARCHAR(800) NOT NULL,
   chef_level ENUM('G CHEF ', 'B CHEF', 'A CHEF', 'ASSISTANT CHEF', 'MASTER CHEF')
);



create table chefs_of_recipe(
recipe_id int unsigned NOT NULL ,
chef_id int unsigned  NOT NULL,
primary key(recipe_id,chef_id),
CONSTRAINT chefs_of_recipe_recipe_id foreign key(recipe_id) references recipe(recipe_id),
CONSTRAINT chefs_of_recipe_chef_id foreign key(chef_id) references chefs(chefs_id)
);

create table ethnic_of_chef(
chef_id int unsigned NOT NULL,
ethnic_id int unsigned NOT NULL,
primary key(ethnic_id,chef_id),
CONSTRAINT ethnic_of_chef_ethnic_id foreign key(ethnic_id) references ethnic_cuisines(ethnic_id)
ON DELETE RESTRICT ON UPDATE CASCADE,
CONSTRAINT ethnic_of_chef_chef_id foreign key(chef_id) references chefs(chefs_id)
ON DELETE RESTRICT ON UPDATE CASCADE
);
```

```
create table labels_of_recipe(
labels_of_recipe_id int UNSIGNED NOT NULL AUTO_INCREMENT ,
recipe_id int unsigned NOT NULL,
name_of_label varchar(80) NOT NULL ,
primary key(labels_of_recipe_id) ,
CONSTRAINT labels_of_recipe_recipe_id foreign key(recipe_id) references recipe(recipe_id)
ON DELETE RESTRICT ON UPDATE CASCADE,
UNIQUE KEY (recipe_id, name_of_label)
) ;

create table basic_of_recipe(
recipe_id int unsigned  NOT NULL,
basic_ingredient int unsigned  NOT NULL,
category varchar(30) NOT NULL,
primary key(recipe_id) ,
CONSTRAINT basic_of_recipe_recipe_id foreign key(recipe_id) references recipe(recipe_id)
ON DELETE RESTRICT ON UPDATE CASCADE,
CONSTRAINT basic_of_recipe_basic_ingredient foreign key(basic_ingredient) references
ingredients(ingredients_id)
ON DELETE RESTRICT ON UPDATE CASCADE
);

create table episodes(
episodes_id int UNSIGNED NOT NULL AUTO_INCREMENT,
season int UNSIGNED NOT NULL,
primary key(episodes_id,season)
);

create table episodes_information(
episode_id int unsigned NOT NULL,
season_of_episode int unsigned not null,
chef_id int unsigned NOT NULL,
role_of_chef boolean, -- role of chef = 1 μαγειρας / role of chef = 0 κριτης
recipe int unsigned,
ethnic_cuisine int unsigned,
grade1 int check(grade1 BETWEEN 1 AND 5),
grade2 int  check(grade2 BETWEEN 1 AND 5),
grade3 int  check(grade3 BETWEEN 1 AND 5),
total_grade int UNSIGNED,
primary key (episode_id,season_of_episode,chef_id),
CONSTRAINT episodes_information_episode_id foreign key (episode_id,season_of_episode)
references  episodes(episodes_id,season)
ON DELETE RESTRICT ON UPDATE CASCADE,
CONSTRAINT episodes_information_chef_id foreign key (chef_id) references chefs(chefs_id)
ON DELETE RESTRICT ON UPDATE CASCADE,
CONSTRAINT episodes_information_recipe foreign key (recipe) references recipe(recipe_id)
ON DELETE RESTRICT ON UPDATE CASCADE,
CONSTRAINT episodes_information_ethnic_cuisine foreign key (ethnic_cuisine) references
ethnic_cuisines(ethnic_id)
ON DELETE RESTRICT ON UPDATE CASCADE
);
```

Στον κώδικα μας, αφού δημιουργήσουμε το σχήμα με όνομα “masterchef” για την βάση μας, με την χρήση της εντολής create table δημιουργούμε τους πίνακες που θα χρησιμοποιήσουμε. Η σειρά δημιουργίας των πινάκων μας είναι τέτοια έτσι ώστε τυχών πίνακες που περιέχουν ξένα κλειδιά να δημιουργούνται αφού έχουν δημιουργηθεί οι πίνακες στους οποίους αναφέρονται.

Στα αρχεία ddl.sql και procedures.sql που βρίσκονται στο github repository, περιέχονται και τα απαραίτητα triggers που εξασφαλίζουν την ορθότητα της βάσης (π.χ. η ηλικία ενός μάγειρα να είναι μεγαλύτερη από τα χρόνια εμπειρίας του, να μην εμφανίζεται η ίδια εθνική κουζίνα σε περισσότερα από τρία διαδοχικά επεισόδια κ.τ.λ.), καθώς και τα απαραίτητα procedures (π.χ. για την ανάδειξη του νικητή του διαγωνισμού).

3.2 DML script

Με το DML script, μέσω της εντολής insert, αποθηκεύουμε δεδομένα στην βάση μας. Ενδεικτικά έχουμε δημιουργήσει 100 συνταγές, 100 μάγειρες, 50 θεματικές ενότητες (themes), 200 ετικέτες (labels), 100 συστατικά, 100 κομμάτια εξοπλισμού (equipment), 20 εθνικές κουζίνες, 50 επεισόδια και 5 σεζόν. Το DML script βρίσκεται στον ακόλουθο σύνδεσμο: https://github.com/Madeperf/DataBaseProject/blob/main/Fake_data.sql

Τα mock data που εισαγάγαμε στη βάση δεδομένων, κατασκευάστηκαν με χρήση του διαδικτυακού εργαλείου Mockaroo (https://www.mockaroo.com/).

4. Οδηγίες Εγκατάστασης

Για να δημιουργήσετε και να χρησιμοποιήσετε την βάση δεδομένων που έχουμε υλοποιήσει, θα πρέπει αρχικά να έχετε εγκαταστήσει την κατάλληλη έκδοση του MySQL ανάλογα με το λειτουργικό σας σύστημα (Windows, macOS, Linux) από την επίσημη ιστοσελίδα της MySQL. Κατά τη διάρκεια της εγκατάστασης, θα σας ζητηθεί να ορίσετε τον root κωδικό πρόσβασης. Μετά την εγκατάσταση, πρέπει να εκκινήσετε τον MySQL Server και να πλοηγηθείτε στο directory C:\Program Files\MySQL\MySQL Server 8.1\bin (το directory ίσως διαφέρει ανάλογα με το λειτουργικό σας σύστημα) πληκτρολογώντας στο τερματικό:

```
cd C:\Program Files\MySQL\MySQL Server 8.1\bin
```

Στην συνέχεια πρέπει να συνδεθείτε ως root χρήστης πληκτρολογώντας στο τερματικό:

```
mysql -u root -p
```

Θα σας ζητηθεί να εισαγάγετε τον root κωδικό πρόσβασης που ορίσατε κατά την εγκατάσταση. Μετά τη σύνδεση, θα πρέπει να εκτελέστε με τη σειρά τα scripts DDLscript.sql, Fake_data.sql και Procedures.sql τα οποία κατασκευάζουν τους πίνακες με τα ευρετήρια, τα απαραίτητα procedures και εισάγουν τα δεδομένα στη βάση. Για να το κάνετε αυτό πρέπει να εκτελέσετε στο τερματικό την εντολή:

```
source <path>
```

όπου το path θα το αντικαταστήσετε με το μονοπάτι που αντιστοιχεί σε κάθε αρχείο sql. Αυτό προυποθέτει να έχετε κατεβάσει τα αντίστοιχα αρχεία από το github repository.

4.1 Authentication

Ως root χρήστης έχετε όλα τα permissions για τροποποίηση της βάσης δεδομένων. Μέσω του ddl script έχουν δημιουργηθεί επίσης ένας χρήστης admin, αλλά και ένας χρήστης chef, που αντιστοιχεί στον chef με chef_id=50. Για κάθε χρήστη, το σύστημα εξακριβώνει την ταυτότητά του κατά την είσοδο στην εφαρμογή (μέσω username / password). Συγκεκριμένα έχουν οριστεί:

• admin
username: admin
password: admin

• chef user
username: chef
password: chef

Ο admin μπορεί να καταχωρίζει και να τροποποιεί όλα τα απαιτούμενα στοιχεία. Μπορεί ακόμη να δημιουργήσει αντίγραφο ασφαλείας για όλη τη βάση (backup) και να επαναφέρει το σύστημα από αυτό (restore). Ο χρήστης chef έχει την δυνατότητα να επεξεργαστεί τα στοιχεία των συνταγών που του αντιστοιχούν, καθώς και να προσθέσει νέα συνταγή. Επίσης μπορεί να επεξεργαστεί τα προσωπικά του στοιχεία. Οι δύο αυτές λειτουργίες γίνονται με κλήση των procedures InsertOrUpdateRecipe και UpdateChefInfo αντίστοιχα με τις κατάλληλες παραμέτρους. Σε περίπτωση που θέλαμε να φτιάξουμε και άλλους chef users, που αντιστοιχούν στα υπόλοιπα chef_ids της βάσης μας, θα έπρεπε να κάνουμε την ίδια διαδικασία για κάθε id (να φτιάξουμε user με κατάλληλο username και password, να φτιάξουμε τα απαραίτητα views και να δώσουμε τα permissions που του αντιστοιχούν). Μπορείτε τώρα να αρχίσετε να χρησιμοποιείτε τη βάση δεδομένων είτε ως admin, είτε ως chef user και να εκτελέσετε τα ερωτήματα που επιθυμείτε. Για να αποσυνδεθείτε πληκτρολογήστε στο τερματικό exit.

5. Ανάλυση λογικής queries

1) Σε αυτό το query για να υπολογίσουμε τον μέσο όρο των αξιολογήσεων κάθε σεφ και κάθε εθνικής κουζίνας ακολουθήσαμε 2 φορές την ίδια διαδικασία. Συγκεκριμένα δημιουργήσαμε 2 φορές μια απλή δομή select, η οποία επιστρέφει ένα πίνακα που περιέχει τον μέσο όσο των συνολικών βαθμών (με την χρήση της εντολής avg) και το αναγνωριστικό του μάγειρα ή της εθνικής κουζίνας. Παρακάτω παρατίθεται ενδεικτικά ένα απόσπασμα του αποτελέσματος.

ethnic_cuisine_id	average_total_grade_of_ethnic_cuisine
1	8.5714
2	9.5714
3	9.0526
4	7.8824
5	8.7273
6	9.1667
7	9.1111
8	9.9231
9	9.6250
10	9.6875
11	9.8571

2) Σε αυτό το query ορίσαμε αυθαίρετα μια τυχαία εθνική κουζίνα και μια σεζόν ώστε να εμφανιστούν ενδεικτικά αποτελέσματα. Αρχικά βρήκαμε τους μάγειρες που ανήκουν σε αυτήν την εθνική κουζίνα. Έπειτα δημιουργήσαμε έναν μεγάλο πίνακα, ο οποίος περιείχε χαρακτηριστικά από τους πίνακες ethnic_of_chef, ethnic_cuisines, episode_information και recipe, οι οποίοι ενώθηκαν με πολλά join στα κοινά τους χαρακτηριστικά, οπότε βρήκαμε μάγειρες που ανήκουν στην εθνική κουζίνα Mexican και εμφανίζονται σε κάποιο επεισόδιο της σεζόν 1. Παρακάτω παρατίθεται ενδεικτικά ένα απόσπασμα του αποτελέσματος.

chefs_id	first_name	last_name	phone_number	date_of_birth	age	years_of_experience	image	image_caption	chef_level
6	Lorene	Beevens	6945333818	1955-12-11	68	4	http://dummyimage.com/147x100.png/cc0000/...	Artistic chef who treats plating as an art form	MASTER CHEF
17	Moreen	Philpart	6929674980	1965-12-25	58	3	http://dummyimage.com/156x100.png/ff4444/f...	Inventive chef constantly coming up with new a...	A CHEF
18	Sheila	Constantine	6955849790	1957-04-10	67	15	http://dummyimage.com/140x100.png/ff4444/f...	Skillful chef who can handle the demands of a b...	G CHEF
19	Clayborne	Apfel	6935937021	1957-02-01	67	10	http://dummyimage.com/215x100.png/ff4444/f...	Adventurous chef who loves to experiment with...	G CHEF
21	Waylon	Braund	6964980970	1990-04-26	34	16	http://dummyimage.com/248x100.png/dddddd/...	Artistic chef who treats cooking as a form of sel...	A CHEF
26	Ivan	Matuszyk	6981042864	1963-04-26	61	10	http://dummyimage.com/159x100.png/cc0000/...	Seasoned chef who has perfected their craft ov...	MASTER CHEF
35	Amitie	Robley	6988559328	1974-01-21	50	10	http://dummyimage.com/204x100.png/5fa2dd/...	Collaborative chef who enjoys working with loca...	G CHEF
44	Gilbertina	Sapshed	6900903013	1989-04-03	35	2	http://dummyimage.com/142x100.png/ff4444/f...	Charming chef who charms diners with their per...	B CHEF
60	Sharleen	Vickery	6951446657	1959-03-30	65	4	http://dummyimage.com/216x100.png/cc0000/...	Creative chef who is always pushing the bound...	B CHEF
64	Harmon	Brahan	6932737899	1994-02-16	30	17	http://dummyimage.com/249x100.png/dddddd/...	Dedicated chef who puts their heart and soul in...	A CHEF

3) Το αποτέλεσμα που δίνει αυτό το query είναι ένας πίνακας που περιέχει τα χαρακτηριστικά chef_name, chef_age ,recipes_count. Για να γίνει αυτό χρησιμοποιήσαμε και πάλι την εντολή count αλλά και τις εντολές sum για να βρούμε το άθροισμα των συνταγών και την εντολή concat ώστε να εκτυπώσουμε αποτέλεσμα με συγκεκριμένη μορφή. Παρακάτω παρατίθεται ενδεικτικά ένα απόσπασμα του αποτελέσματος.

chef_id	chef_name	chef_age	recipes_count
65	Curtice Coiley	28	5
84	Regan Custard	19	5
97	Appolonia Batcock	18	4
5	Theresina Favey	27	4
7	Frederique Hopkyns	23	4
87	Tiffie Necrews	21	4
24	Dael Zanotti	23	4
63	Jorgan Sarch	21	4
69	Bessie Boler	25	3
9	Stoddard Walne	28	3
14	Gabi Mangeon	27	3

4) Η διαφορά αυτού του query από τα προηγούμενα είναι πως δεν θέλουμε να βρούμε εγγραφές οι οποίες θα έχουν ένα χαρακτηριστικό αλλά εγγραφές που δεν θα έχουν αυτό το

χαρακτηριστικό. Για αυτό τον σκοπό χρησιμοποιούμε την εντολή where not exists μέσα στο query μας. Παρακάτω παρατίθεται ενδεικτικά ένα απόσπασμα του αποτελέσματος.

chefs_id	first_name	last_name
25	Adriane	Catteroll
76	Anson	Castaneda
12	Bennett	Skrzynski
96	Burg	Edwick
89	Cammy	Foort
81	Cecile	Mackness
49	Clarabelle	Milverton
19	Clayborne	Apfel
86	Collete	Patillo
66	Durand	Gascoine
14	Gabi	Mangeon
95	Georgie	Rankine
39	Gianna	McNickle
29	Guillermo	Castro
57	Gunar	Bitten
37	Issi	Filippo
46	Joeann	Agdahl

5) Η ιδιαιτερότητα αυτού του query έγκειται στο γεγονός ότι κάνουμε ένωση ενός πίνακα με τον εαυτό του. Με αυτό τον τρόπο βρίσκουμε 2 κριτές του episode_information οι οποίοι στην ίδια σεζόν(έστω στην δεύτερη) έχουν τον ίδιο αριθμό συμμετοχών(>3) ως κριτές. Για να μην μετρήσουμε 2 φορές το κάθε συνδυασμό κριτών έχουμε θέσει τον περιορισμό το id του πρώτου να είναι πάντα μικρότερο του δεύτερου.

judge1_id	judge2_id	appearance_count
1	2	6

6) Αυτό το query εντοπίζει και μετρά τα πιο συχνά ζευγάρια ετικετών που εμφανίζονται μαζί σε συνταγές. Συγκεκριμένα, χρησιμοποιεί δύο ψευδώνυμα για τον πίνακα labels_of_recipe (l1,l2), τα οποία συνδέει με βάση το recipe_id, επιτρέποντας τον εντοπισμό ζευγαριών ετικετών που ανήκουν στην ίδια συνταγή. Αφού εξασφαλίσουμε ότι κάθε ζευγάρι εμφανίζεται μία φορά, στην συνέχεια φροντίζουμε να συνδέσουμε το query με τον πίνακα των επεισοδίων episodes_infromation(ei) με βάση το recipe_id, για να εξασφαλιστεί η εγκυρότητα των συνταγών. Έπειτα, μέσω της εντολής GROUP_BY και της συνάρτησης COUNT*, επιτυγχάνουμε την ομαδοποίηση και την μέτρηση αντίστοιχα. Στην συνέχεια ,ταξινομούμε και περιορίζουμε τα αποτελέσματα.

label1	label2	pair_count
hot-meal	snack	150
brunch	hot-meal	120
hot-meal	quick-lunch	100

Εναλλακτικό Query Plan: Στο σημείο αυτό χρησιμοποιούμε το ίδιο query με την διαφορά ότι προσθέτουμε την εντολή FORCE INDEX, που δηλώνει το index που μας βοηθά στον υπολογισμό του ερωτήματος. Εκτελούμε το query και παίρνουμε το ίδιο αποτέλεσμα:

	label1	label2	pair_count
▶	hot-meal	snack	150
	brunch	hot-meal	120
	hot-meal	quick-lunch	100

Όταν εκτελέσουμε το EXPLAIN για τα παραπάνω queries, θα πάρουμε έναν πίνακα που περιγράφει τον τρόπο με τον οποίο το MySQL διαχειρίζεται το query. Τα κύρια πεδία που πρέπει να προσέξουμε είναι:

- table: Ο πίνακας που εξετάζεται σε αυτό το βήμα.
- type: Ο τύπος της ένωσης (JOIN) που χρησιμοποιείται.
- possible_keys: Τα ευρετήρια που θα μπορούσαν να χρησιμοποιηθούν για αυτό το query.
- key: Το ευρετήριο που χρησιμοποιείται τελικά.
- key_len: Το μήκος του χρησιμοποιούμενου ευρετηρίου.
- ref: Οι στήλες ή οι σταθερές που χρησιμοποιούνται με το ευρετήριο.
- rows: Ο εκτιμώμενος αριθμός των γραμμών που θα διαβαστούν από τον πίνακα.
- filtered: Το ποσοστό των γραμμών που αναμένεται να περάσουν το φίλτρο.
- Extra: Πρόσθετες πληροφορίες, όπως αν χρησιμοποιείται προσωρινός πίνακας ή αρχειοθέτηση.

Προκύπτουν λοιπόν τα παρακάτω αποτελέσματα για το απλό query και το query με force indexes αντίστοιχα:

	id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
▶	1	SIMPLE	l1	index	recipe_id,idx_labels_recipe	recipe_id	326	NULL	200	Using index; Using temporary; Using filesort
	1	SIMPLE	l2	ref	recipe_id,idx_labels_recipe	recipe_id	4	masterchef.l1.recipe_id	1	Using where; Using index
	1	SIMPLE	ei	ref	idx_episodes_recipe	idx_episodes_recipe	5	masterchef.l1.recipe_id	1	Using index

	id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
▶	1	SIMPLE	l1	index	idx_labels_recipe	idx_labels_recipe	326	NULL	200	Using index; Using temporary; Using filesort
	1	SIMPLE	l2	ref	idx_labels_recipe	idx_labels_recipe	4	masterchef.l1.recipe_id	1	Using where; Using index
	1	SIMPLE	ei	ref	idx_episodes_recipe	idx_episodes_recipe	5	masterchef.l1.recipe_id	1	Using index

Και στα δύο queries, ο αριθμός των γραμμών που θα διαβαστούν είναι ο ίδιος (200 για τον πίνακα l1, 1 για τους πίνακες l2 και ei). Το απλό query μπορεί να χρησιμοποιήσει περισσότερα ευρετήρια, ενώ το query με force index χρησιμοποιεί μόνο συγκεκριμένα ευρετήρια, όμως φαίνεται να έχουν παρόμοια αποδοτικότητα. Για να έχουμε μία πιο ακριβή εικόνα σχετικά με το χρόνο εκτέλεσης του κάθε query εκτελούμε την εντολή *set profiling = 1*; και στην συνέχεια εκτελούμε τα δύο queries. Αν χρησιμοποιήσουμε την εντολή *show profiles;* παίρνουμε τοN παρακάτω πίνακα.

	Query_ID	Duration	Query
▶	1	0.01217240	SELECT l1.name_of_label AS label1, l2.na...
	2	0.00250850	SELECT l1.name_of_label AS label1, l2.na...

Παρατηρούμε δηλαδή ότι με force index το query εκτελείται γρηγορότερα.

7) Αυτό το SQL ερώτημα εντοπίζει τους σεφ που έχουν συμμετάσχει τουλάχιστον 5 λιγότερες φορές από τον μάγειρα με τις περισσότερες συμμετοχές σε επεισόδια. Για τους σεφ όπου ο ρόλος είναι 1 επιλέγεται το chef_id και ο αριθμός συμμετοχών count από τον πίνακα

επεισοδίων. Το υποερώτημα subquery υπολογίζει τον μέγιστο αριθμό συμμετοχών MAX(participation_count) όλων των σεφ αφαιρώντας 5. Στην συνέχεια, μέσω του φίλτρου HAVING φιλτράρονται τα αποτελέσματα ώστε να περιλαμβάνει μόνο τους σεφ των οποίων ο αριθμός συμμετοχών είναι μικρότερος η ίσος με τον μέγιστο αριθμό συμμετοχών μείον 5. Βρίσκουμε (με το δεύτερο query που έχουμε φτιάξει) ότι ο μάγειρας με το μεγαλύτερο αριθμό συμμετοχών είναι αυτός που αντιστοιχεί στο chef_id=4 με 25 εμφανίσεις. Παρακάτω παρατίθεται ενδεικτικά ένα απόσπασμα του αποτελέσματος.

chef_id	participation_count
1	15
5	15
6	10
7	10
8	5
9	5
10	5
14	5
16	5
17	5
19	5
20	5
21	15

8) Αυτό το ερώτημα βρίσκει τα επεισόδια που χρησιμοποίησαν τον μέγιστο αριθμό εξοπλισμού και τα επιστρέφει ταξινομημένα κατά episode_id. Ειδικότερα, ενώνει τον πίνακα episodes_infromation (ei) με τον πίνακα equipment_of_recipe (eor) με γνώμονα το recipe και το recipe_id αντίστοιχα. Ακόμα, ομαδοποίει τα αποτελέσματα ανά episode_id και μετρά τον αριθμό εξοπλισμού που χρησιμοποιήθηκε σε κάθε επεισόδιο. Το φίλτρο HAVING διασφαλίζει ότι μόνο τα επεισόδια μέγιστου αριθμού εξοπλισμού συμπεριλαμβάνονται στο αποτέλεσμα. Αυτό επιτυγχάνεται με ένα υποερώτημα που υπολογίζει τον μέγιστο αριθμό εξοπλισμού που χρησιμοποιήθηκε σε οποιοδήποτε επεισόδιο. Τέλος, τα αποτελέσματα ταξινομούνται κατά episode_id.

episode_id	equipment_count
1	160

Ενναλακτικό Query Plan: Στο σημείο αυτό βάζουμε force index .Το idx_recipe_id για τον πίνακα recipe_id επιτυγχάνει την αναζήτηση και την ένωση βάση συνταγής, ενώ το idx_episodes_recipe για τον πίνακα episodes_infromation για την στήλη recipe επιτυχάνει επίσης την αναζήτηση και την ένωση βάση συνταγής.

episode_id	equipment_count
1	160

Η χρήση της force index επιβάλλει μάλιστα στο ερώτημα να χρησιμοποιήσει τα ανώτερα ευρετήρια. Τρέχοντας την εντολή explain για τα δύο queries λαμβάνουμε τα παρακάτω αποτελέσματα:

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	PRIMARY	eor	index	PRIMARY,idx_recipe_id	PRIMARY	8	NULL	302	Using index; Using temporary; Using filesort
1	PRIMARY	ei	ref	idx_episodes_recipe	idx_episodes_recipe	5	masterchef.eor.recipe_id	1	Using index
2	SUBQUERY	<derived3>	ALL	NULL	NULL	NULL	NULL	302	
3	DERIVED	eor_inner	index	PRIMARY,idx_recipe_id	PRIMARY	8	NULL	302	Using index; Using temporary; Using filesort
3	DERIVED	ei_inner	ref	idx_episodes_recipe	idx_episodes_recipe	5	masterchef.eor_inner.recipe_id	1	Using index

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	PRIMARY	eor	index	idx_recipe_id	idx_recipe_id	4	NULL	302	Using index; Using temporary; Using filesort
1	PRIMARY	ei	ref	idx_episodes_recipe	idx_episodes_recipe	5	masterchef.eor.recipe_id	1	Using index
2	SUBQUERY	<derived3>	ALL	NULL	NULL	NULL	NULL	302	
3	DERIVED	eor_inner	index	idx_recipe_id	idx_recipe_id	4	NULL	302	Using index; Using temporary; Using filesort
3	DERIVED	ei_inner	ref	idx_episodes_recipe	idx_episodes_recipe	5	masterchef.eor_inner.recipe_id	1	Using index

Από την ανάλυση των EXPLAIN αποτελεσμάτων, παρατηρούμε ότι και στα δύο queries, τα indexes που χρησιμοποιούνται είναι τα ίδια. Αν εκτελέσουμε τις εντολές *set profiling = 1*; και *show profiles;* παίρνουμε τα παρακάτω αποτελέσματα σχετικά με τον χρόνο εκτέλεσης.

	7	0.00258010	SELECT	ei.episode_id,	COUNT(eor.equipm...
	8	0.00349000	SELECT	ei.episode_id,	COUNT(eor.equipm...

Παρατηρούμε δηλαδή ότι το απλό query εκτελείται γρηγορότερα. Το force index άρα μας εξυπηρετεί κυρίως, ώστε να επιλέξουμε ένα συγκεκριμένο μονοπάτι, αλλά από άποψη χρόνου δεν προσφέρει πάντα βελτίωση.

9) Αυτό το query, αρχικά συνδέει τον πίνακα episodes_infromation (ei) με το ingredients_of_recipe (ir) με βάση το recipe_id, καθώς και τον ingredients_of_recipe (ir) με τον ingredients(i) με βάση το ingredient_id. Στην συνέχεια υπολογίζουμε τον μέσο όρο υδατανθράκων για κάθε σεζόν. Τέλος, τα αποτελέσματα ομαδοποιούνται και ταξινομούνται κατά season_of_episode και έτσι λαμβάνουμε το επιθυμητό αποτέλεσμα.

year	avg_carbohydrates
1	2789.45457338
2	2789.45457338
3	2789.45457338
4	2789.45457338
5	2789.45457338

10) Η ιδιαιτερότητα της λύσης που έχουμε δώσει σε αυτό το query είναι πως με την εντολή with δημιουργούμε έναν προσωρινό πίνακα τον οποίον στην συνέχεια χρησιμοποιούμε εντός της with ώστε να δημιουργήσουμε έναν νέο πίνακα. Αυτός ο πίνακας περιέχει την συχνότητα εμφάνισης των εθνικών κουζινών. Κάνοντας join για 2 στοιχεία αυτού του πίνακα βρίσκουμε τις εθνικές κουζίνες με ίδια συχνότητα εμφάνισης σε διάστημα 2 συνεχόμενων ετών. Δίνεται ενδεικτικά ένα απόσπασμα του αποτελέσματος.

	period	ethnic_cuisine	ethnic_cuisine	total_freq
▶	1-2	1	17	12
	2-3	1	4	12
	3-4	3	12	12
	3-4	12	17	12
	4-5	4	17	12
	1-2	3	9	12
	2-3	4	12	12
	3-4	9	17	12
	4-5	1	17	12
	1-2	9	12	12

11) Στο query αυτό υπολογίσαμε το άθροισμα των βαθμολογιών που έχουν δώσει όλοι οι κριτές σε κάθε μάγειρα και αποθηκεύσαμε τις πέντε μεγαλύτερες βαθμολογίες. Για να βρούμε ποιος κριτής έδωσε τον κάθε βαθμό στους μάγειρες ακολουθούμε τον επόμενο κώδικα. Όταν το role_of-chef=0,δηλαδή ο συγκεκριμένος chef είναι κριτής, θέτουμε 0 τον i-οστο βαθμό και NULL τους άλλους 2. Για παράδειγμα αν κάποιος κριτής είναι ο δεύτερος κριτής ενός επεισοδίου θα έχει τις τιμές (grade1,grade2,grade3)=(NULL,0,NULL). Όσον αφορά τον sql κώδικα χρησιμοποιούμε την εντολή case για να βρούμε τους κριτές.

	total_score	contestant_id	judge_id
▶	19	45	53
	18	1	65
	18	9	53
	17	4	1
	17	88	1

12) Η υλοποίηση αυτού του query γίνεται σε τρία μέρη. Αρχικά, μέσω της εντολής AVG εντός ενός with υπολογίζουμε έναν πίνακα με την μέση δυσκολία των επεισοδίων ανά έτος. Στην συνέχεια, βρίσκουμε το επεισόδιο με την μέγιστη μέση δυσκολία και αποθηκεύουμε την τιμή της δυσκολίας στο max-average-difficulty. Τέλος, με βάση την τιμή που έχουμε βρει βρίσκουμε από στον πίνακα με τις μέσες δυσκολίες τα χαρακτηριστικά του επεισοδίου που αντιστοιχεί.

	season_of_episode	episode_id	avg_difficulty
▶	1	8	3.9000
	2	8	3.9000
	3	8	3.9000
	4	8	3.9000
	5	8	3.9000

13) Το ιδιαίτερο χαρακτηριστικό αυτού του query είναι πως πρέπει να αντιστοιχίσουμε τις επιμέρους τιμές επαγγελματικής κατάρτισης σε ακέραιους αριθμούς ώστε να κάνουμε την πρόσθεση των ικανοτήτων του κάθε chef. Πέρα από αυτό, χρησιμοποιούμε και πάλι την εντολή with και την εντολή min ώστε να υπολογίσουμε την ελάχιστη συνολική τιμή επαγγελματικής κατάρτισης ανά επεισόδιο.

	season_of_episode	episode_id	total_rank
▶	2	2	32
	2	7	32

14) Η απάντηση σε αυτό το query όπως και στο query 12 δίνεται σε 3 μέρη. Με παρόμοια λογική στην αρχή υπολογίζουμε τις εμφανίσεις της κάθε θεματικής ενότητας στον διαγωνισμό, στην συνέχεια τον μέγιστο αριθμό εμφανίσεων και εν τέλει τον θεματική ενότητα στην οποία αντιστοιχεί αυτή η μέγιστη τιμή.

	name_of_themes	theme_count
▶	Mexican fiesta	65

15) Αυτό το query είναι παρόμοιο με το query 4 αφού πάλι θέλουμε να βρούμε εγγραφές που να μην πληρούν κάποιο κριτήριο. Για τον σκοπό αυτό χρησιμοποιούμε την εντολή where <attribute> not in και βρίσκουμε την ομάδα τροφίμων που θέλουμε στο κατά τα άλλα εύκολο query.

	name_of_group
▶	Fruits and vegetables

6. Σύνδεσμος github repository

Ο συνολικός κώδικας της εφαρμογής μας περιέχεται στο παρακάτω github repository:

https://github.com/Madeperf/DataBaseProject