

Βάσεις Δεδομένων

Εξαμηνιαία Εργασία

Ακαδημαϊκό Έτος 2023-2024

Γιώργος Οικονόμου (Α.Μ. 03121882)

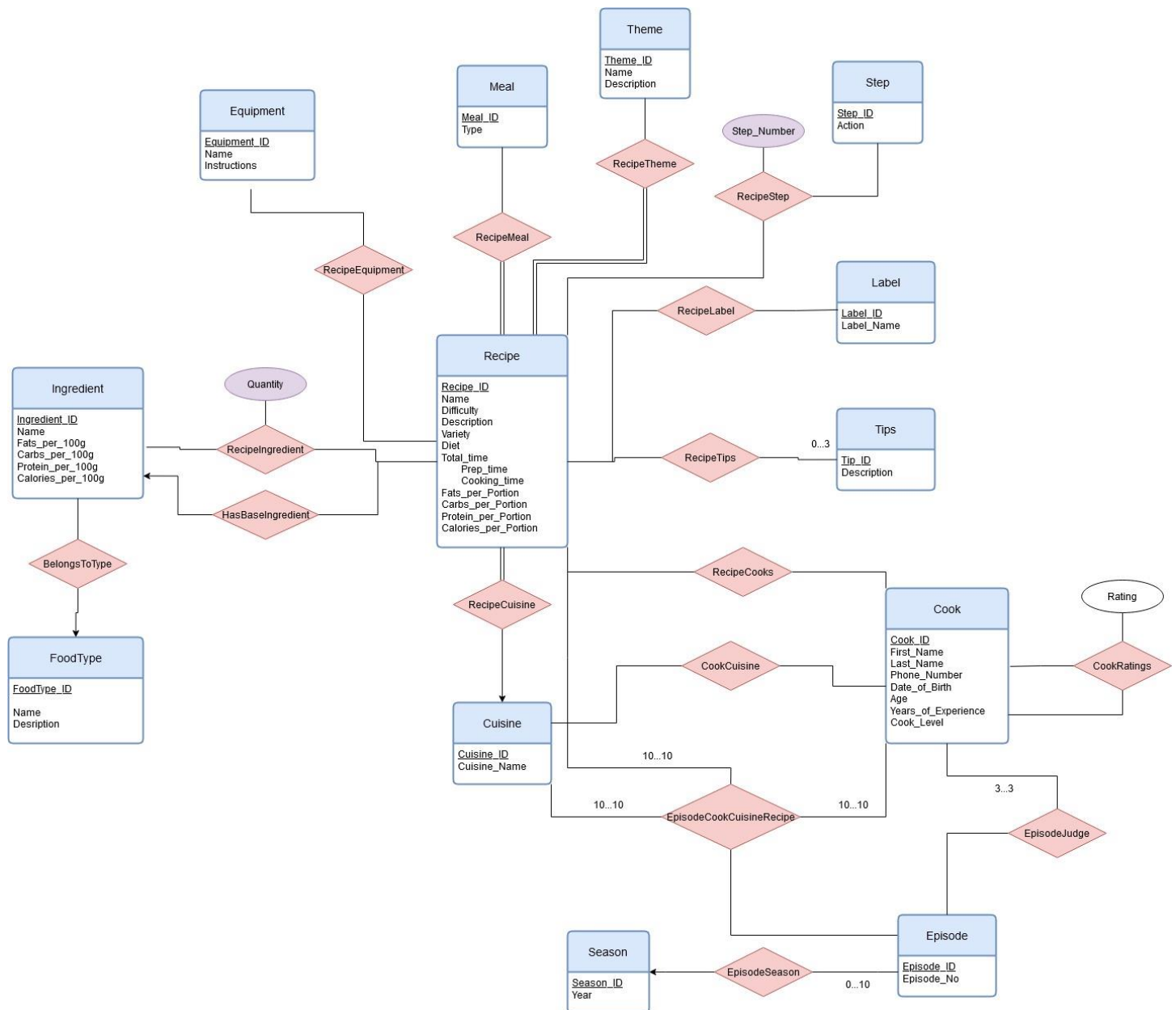
Παύλος Λιβιτσάνος (Α.Μ. 03120024)

Github repository link: <https://github.com/pavblos/NTUACHEF/tree/main>

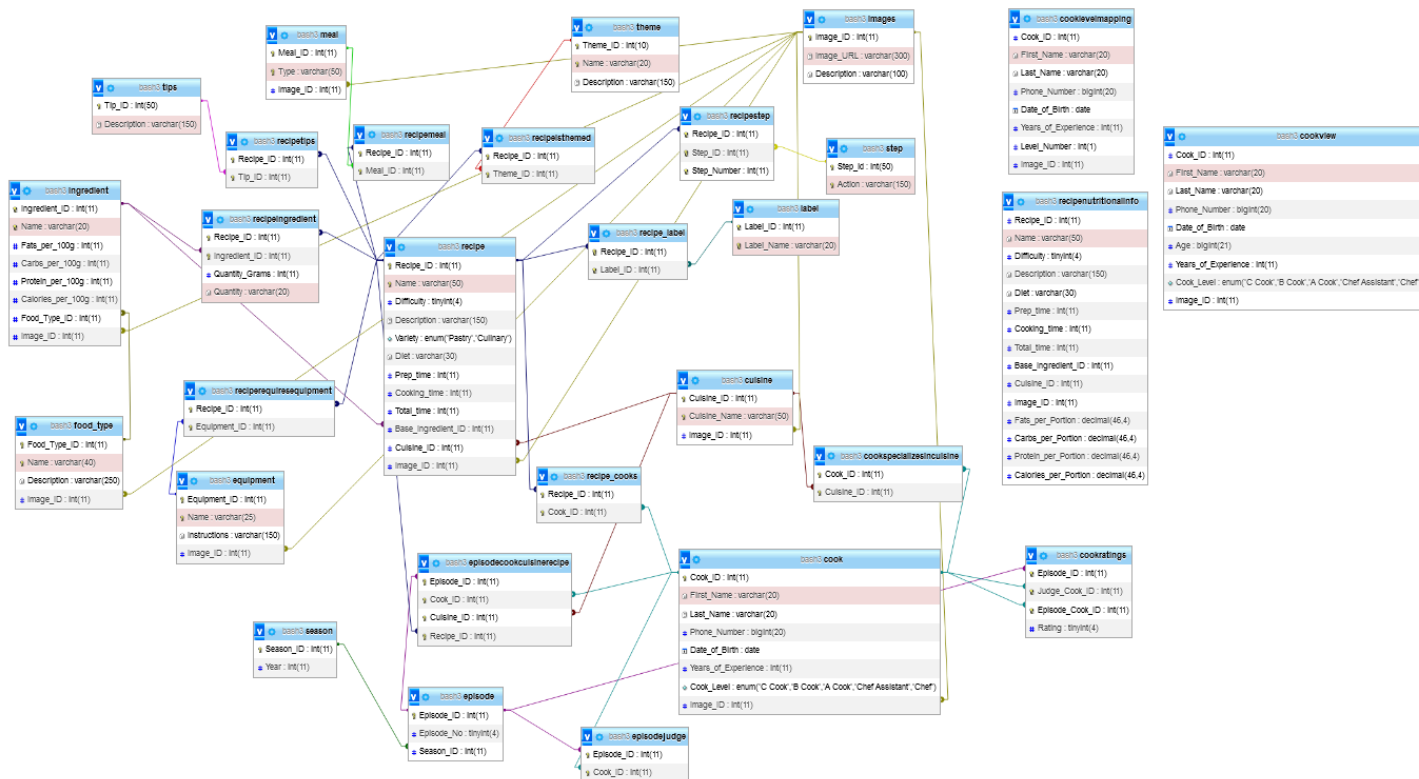


NtuaChef

1.Entity – Relationship Diagram



Χαρακτηρίζεται από ID, όνομα, οδηγίες χρήσης.



5. Meal

Τύπος γεύματος. Χαρακτηρίζεται από ID και όνομα τύπου.

6. Theme

Χαρακτηρίζεται από ID, όνομα, σύντομη περιγραφή.

7. Step

Περιέχει ID και περιγραφή του βήματος (Action).

8. Label

Χαρακτηρίζεται από ID και όνομα.

9. Tip

Χαρακτηρίζεται από ID και όνομα.

10. Cuisine

Εθνικές κουζίνες στις οποίες κατηγοριοποιούνται οι συνταγές. Χαρακτηρίζονται από ID και όνομα.

11. Cook

Η δεύτερη κύρια οντότητα της βάσης μας. Χαρακτηρίζεται από ID, κύριο όνομα, επίθετο, αριθμό τηλεφώνου, ημερομηνία γέννησης και ηλικία, χρόνια επαγγελματικής εμπειρίας και επίπεδο κατάρτισης.

12. Season

Οι κύκλοι κατά τους οποίους διαδραματίζονται τα επεισόδια του διαγωνισμού. Χαρακτηρίζονται από ID και έτος.

13. Episode

Τα επεισόδια στα οποία διαδραματίζεται ο διαγωνισμός. Χαρακτηρίζονται από ID και αριθμό επεισοδίου.

Οι σχέσεις που διέπουν τη βάση μας είναι οι εξής:

1. RecipeIngredient

Αναθέτει υλικά σε συνταγές. Η σχέση περιέχει ως attribute την ποσότητα (σαφώς ή λιγότερο σαφώς ορισμένη) του υλικού που απαιτεί η συνταγή.

2. HasBaseIngredient

Αναθέτει ακριβώς 1 βασικό υλικό σε κάθε συνταγή.

3. BelongsToType

Συνδέει κάθε υλικό με τη μοναδική διατροφική κατηγορία στην οποία ανήκει (FoodType).

4. RecipeEquipment

Αναθέτει εξοπλισμό σε συνταγές.

5. RecipeMeal

Αναθέτει τύπο γεύματος σε κάθε συνταγή.

6. RecipeTheme

Αναθέτει θεματική ενότητα σε κάθε συνταγή.

7. RecipeStep

Αναθέτει βήματα στις συνταγές. Η σχέση περιέχει ως attribute τον αριθμό του βήματος, καθώς αυτά θα πρέπει να εκτελεστούν σειριακά.

8. RecipeLabel

Αναθέτει ετικέτες στις συνταγές.

9. RecipeTips

Αναθέτει έως 3 χρηστικά tips στις συνταγές.

10. RecipeCuisine

Αναθέτει ακριβώς μία κουζίνα σε κάθε συνταγή.

11. RecipeCooks

Συνδέει τις συνταγές με τους μάγειρες που μπορούν να τις εκτελέσουν.

12. CookCuisine

Συνδέει τους μάγειρες με τις κουζίνες στις οποίες εξειδικεύονται.

13. CookRatings

Συνδέει μάγειρες με μάγειρες (κριτές με διαγωνιζόμενους στον διαγωνισμό) και έχει ως attribute ακέραι βαθμολογία μεταξύ 1 και 5.

14. EpisodeJudge

Συνδέει μάγειρες που θα διατελέσουν τον ρόλο κριτών με επεισόδια.

15. EpisodeSeason

Συνδέει επεισόδια με τη σεζόν στην οποία ανήκουν.

16. EpisodeCookCuisineRecipe

Η πιο σύνθετη σχέση της βάσης μας. Συνδέει 4 οντότητες μεταξύ τους: επεισόδιο, μάγειρα, κουζίνα, συνταγή. Αποτελεί την ουσία της έννοιας του επεισοδίου, καθώς αναθέτει ακριβώς 10 μάγειρες σε ακριβώς 10 ξεχωριστές συνταγές που ανήκουν σε ακριβώς 10 ξεχωριστές κουζίνες.

4. Indexing:

Αναφέρουμε ότι η SQL δίνει indexes αυτόματα σε όλα τα primary keys και τα attributes με το διακριτικό UNIQUE.

Προσθέτουμε indexes σε άλλα columns που χρησιμοποιούμε συχνά (κυρίως σε foreign keys) τα οποία βρίσκονται στο αρχείο Indexes.sql.

5. Constraints

1. Κάθε συνταγή έχει βαθμό δυσκολίας ακέραιο στο διάστημα [1,5]

2. Τα tips για κάθε συνταγή μπορούν να είναι από 0 έως 3.

3. Το RecipeNo πρέπει να είναι σειριακό για δεδομένη συνταγή στη σχέση RecipeStep. Η επιβολή γίνεται μέσω κατάλληλου trigger.

4. Σε κάθε σεζόν έχουμε το πολύ 10 επεισόδια, των οποίων το StepNo είναι σειριακό. Ξανά, η επιβολή γίνεται μέσω trigger.

5. Σε ένα επεισόδιο υπάρχουν ακριβώς 3 κριτές. Αυτό διασφαλίζεται μέσω triggers που απαγορεύουν τη διαγραφή και την προσθήκη judge εφόσον έχουμε 3.
6. Κάθε rating είναι ακέραιος μεταξύ 1 και 5.
7. Κάθε ένας από τους 10 μάγειρες σε κάθε επεισόδιο
8. Οι συνταγές κάθε επεισοδίου πρέπει να ανήκουν σε αντίστοιχη κουζίνα του επεισοδίου και τόσο οι πρώτες όσο και οι τελευταίες μοναδικές.
9. Θεωρήσαμε ότι ο μάγειρας πρέπει να έχει specialization στην κουζίνα που εκπροσωπεί.
10. Στη σχέση EpisodeCuisineRecipe υπάρχουν το πολύ 10 entries το οποίο διασφαλίζει
11. Κάθε κουζίνα στο EpisodeCook είναι μοναδική και κάθε Recipe μοναδικό.
12. Ο JudgeCook στο CookRating πρέπει να είναι κριτής στο επεισόδιο και ο Episodicook να είναι διαγωνιζόμενος μάγειρας στο επεισόδιο.
13. Μάγειρας, κριτής, συνταγή, κουζίνα δεν μπορούν να συμμετέχουν πάνω από 3 συνεχόμενες φορές σε επεισόδια μιας σεζόν. Βλ. σχετικό trigger.

Τα σχετικά triggers επισυνάπτονται στο αρχείο triggers.sql.

Υποθέσαμε ότι η ποσότητα (Quantity) που δίνεται ως attribute στη σχέση RecipeIngredient λαμβάνεται υπ'όψιν στον υπολογισμό των διατροφικών πληροφοριών για κάποια συνταγή μόνο αν έχει δοθεί σε γραμμάρια. Σε αυτήν την περίπτωση, δίνεται στο attribute Quantity_grams. Ειδιάλλως, λιγότερο σαφής ποσότητα δίνεται στο attribute Quantity και δε λαμβάνεται υπ'όψιν στον υπολογισμό. Ο υπολογισμός γίνεται μέσω του view RecipeNutritionalInfo.

6. DDL

```
CREATE TABLE IF NOT EXISTS `Images` (  
  `Image_ID` int AUTO_INCREMENT,  
  `Image_URL` varchar(300) DEFAULT 'http://ikonomoug.com/ntuachef.png',  
  `Description` varchar(100),  
  PRIMARY KEY(`Image_ID`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;  
  
CREATE TABLE IF NOT EXISTS `Food_Type` (  
  `Food_Type_ID` int AUTO_INCREMENT,  
  `Name` varchar(40) NOT NULL UNIQUE,  
  `Description` varchar(250) NOT NULL,  
  `Image_ID` int DEFAULT 1,  
  PRIMARY KEY(`Food_Type_ID`),  
  FOREIGN KEY (`Image_ID`) REFERENCES `Images` (`Image_ID`) ON DELETE SET NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;  
  
CREATE TABLE IF NOT EXISTS `Equipment` (  
  `Equipment_ID` int AUTO_INCREMENT,  
  `Name` varchar(25) NOT NULL UNIQUE,  
  `Instructions` varchar(150),  
  `Image_ID` int DEFAULT 1,  
  PRIMARY KEY(`Equipment_ID`),  
  FOREIGN KEY (`Image_ID`) REFERENCES `Images` (`Image_ID`) ON DELETE SET NULL
```

```

)ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

CREATE TABLE IF NOT EXISTS `Label` (
  `Label_ID` int NOT NULL AUTO_INCREMENT,
  `Label_Name` varchar(20) NOT NULL UNIQUE,
  PRIMARY KEY (`Label_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

CREATE TABLE IF NOT EXISTS `Meal` (
  `Meal_ID` int AUTO_INCREMENT,
  `Type` varchar(50) NOT NULL UNIQUE,
  `Image_ID` int DEFAULT 1,
  PRIMARY KEY(`Meal_ID`),
  FOREIGN KEY (`Image_ID`) REFERENCES `Images`(`Image_ID`) ON DELETE SET NULL
)ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

CREATE TABLE IF NOT EXISTS `Cuisine` (
  `Cuisine_ID` int AUTO_INCREMENT,
  `Cuisine_Name` varchar(50) NOT NULL UNIQUE,
  `Image_ID` int DEFAULT 1,
  PRIMARY KEY (`Cuisine_ID`),
  FOREIGN KEY (`Image_ID`) REFERENCES `Images`(`Image_ID`) ON DELETE SET NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

CREATE TABLE IF NOT EXISTS `Theme` (
  `Theme_ID` int(10) NOT NULL AUTO_INCREMENT,
  `Name` varchar(20) NOT NULL UNIQUE,
  `Description` varchar(150),
  PRIMARY KEY (`Theme_ID`)
)ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

CREATE TABLE IF NOT EXISTS `Tips` (
  `Tip_ID` int(50) NOT NULL AUTO_INCREMENT,
  `Description` varchar(150) NOT NULL,
  PRIMARY KEY (`Tip_ID`)
)ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

CREATE TABLE IF NOT EXISTS `Step` (
  `Step_id` int(50) NOT NULL AUTO_INCREMENT,
  `Action` varchar(150) NOT NULL UNIQUE,
  PRIMARY KEY (`Step_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

CREATE TABLE IF NOT EXISTS `Ingredient` (
  `Ingredient_ID` int AUTO_INCREMENT,

```

```

`Name` varchar(20) NOT NULL UNIQUE,
`Fats_per_100g` int NOT NULL,
`Carbs_per_100g` int NOT NULL,
`Protein_per_100g` int NOT NULL,
`Calories_per_100g` int NOT NULL,
`Food_Type_ID` int NOT NULL,
`Image_ID` int DEFAULT 1,
PRIMARY KEY (`Ingredient_ID`),
FOREIGN KEY (`Food_Type_ID`) REFERENCES `Food_Type`(`Food_Type_ID`) ON DELETE CASCADE,
FOREIGN KEY (`Image_ID`) REFERENCES `Images`(`Image_ID`) ON DELETE SET NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

```

CREATE TABLE IF NOT EXISTS `Recipe` (
  `Recipe_ID` int NOT NULL AUTO_INCREMENT,
  `Name` varchar(50) NOT NULL UNIQUE,
  `Difficulty` tinyint NOT NULL CHECK (`Difficulty` >= 1 AND `Difficulty` <= 5),
  `Description` varchar(150) NOT NULL,
  `Variety` enum('Pastry', 'Culinary') NOT NULL,
  `Diet` varchar(30),

  `Prep_time` int NOT NULL,
  `Cooking_time` int NOT NULL,
  `Total_time` int GENERATED ALWAYS AS (Prep_time + Cooking_time) STORED,

  `Base_Ingredient_ID` int NOT NULL,
  `Cuisine_ID` int NOT NULL,
  `Image_ID` int DEFAULT 1,

  PRIMARY KEY (`Recipe_ID`),
  FOREIGN KEY (`Base_Ingredient_ID`) REFERENCES `Ingredient`(`Ingredient_ID`),
  FOREIGN KEY (`Cuisine_ID`) REFERENCES `Cuisine`(`Cuisine_ID`),
  FOREIGN KEY (`Image_ID`) REFERENCES `Images`(`Image_ID`) ON DELETE SET NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

```

CREATE TABLE IF NOT EXISTS `Recipe_Label` (
  `Recipe_ID` int,
  `Label_ID` int,
  PRIMARY KEY (`Recipe_ID`, `Label_ID`),
  FOREIGN KEY (`Recipe_ID`) REFERENCES `Recipe`(`Recipe_ID`),
  FOREIGN KEY (`Label_ID`) REFERENCES `Label`(`Label_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

```

CREATE TABLE IF NOT EXISTS `RecipeMeal` (
  `Recipe_ID` int,
  `Meal_ID` int,
  PRIMARY KEY (`Recipe_ID`, `Meal_ID`),
  FOREIGN KEY (`Recipe_ID`) REFERENCES `Recipe`(`Recipe_ID`),
  FOREIGN KEY (`Meal_ID`) REFERENCES `Meal`(`Meal_ID`)

```



```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
CREATE TABLE IF NOT EXISTS `RecipeIngredient` (  
  `Recipe_ID` int NOT NULL,  
  `Ingredient_ID` int NOT NULL,  
  `Quantity_Grams` int,  
  `Quantity` varchar(20),  
  PRIMARY KEY (`Recipe_ID`, `Ingredient_ID`),  
  FOREIGN KEY (`Recipe_ID`) REFERENCES `Recipe` (`Recipe_ID`),  
  FOREIGN KEY (`Ingredient_ID`) REFERENCES `Ingredient` (`Ingredient_ID`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
CREATE TABLE IF NOT EXISTS `RecipeRequiresEquipment` (  
  `Recipe_ID` int NOT NULL,  
  `Equipment_ID` int NOT NULL,  
  PRIMARY KEY (`Recipe_ID`, `Equipment_ID`),  
  FOREIGN KEY (`Recipe_ID`) REFERENCES `Recipe` (`Recipe_ID`),  
  FOREIGN KEY (`Equipment_ID`) REFERENCES `Equipment` (`Equipment_ID`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
CREATE TABLE IF NOT EXISTS `RecipeTips` (  
  `Recipe_ID` int NOT NULL,  
  `Tip_ID` int NOT NULL,  
  PRIMARY KEY (`Recipe_ID`, `Tip_ID`),  
  FOREIGN KEY (`Recipe_ID`) REFERENCES `Recipe` (`Recipe_ID`) ON DELETE CASCADE,  
  FOREIGN KEY (`Tip_ID`) REFERENCES `Tips` (`Tip_ID`) ON DELETE CASCADE  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
CREATE TABLE IF NOT EXISTS `RecipeIsThemed` (  
  `Recipe_ID` int,  
  `Theme_ID` int,  
  PRIMARY KEY (`Recipe_ID`, `Theme_ID`),  
  FOREIGN KEY (`Recipe_ID`) REFERENCES `Recipe` (`Recipe_ID`) ON DELETE CASCADE,  
  FOREIGN KEY (`Theme_ID`) REFERENCES `Theme` (`Theme_ID`) ON DELETE CASCADE  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
CREATE TABLE IF NOT EXISTS `RecipeStep` (  
  `Recipe_ID` int,  
  `Step_ID` int,  
  `Step_Number` int NOT NULL,  
  PRIMARY KEY (`Recipe_ID`, `Step_ID`),  
  UNIQUE (`Recipe_ID`, `Step_Number`),  
  FOREIGN KEY (`Recipe_ID`) REFERENCES `Recipe` (`Recipe_ID`) ON DELETE CASCADE,  
  FOREIGN KEY (`Step_ID`) REFERENCES `Step` (`Step_ID`) ON DELETE CASCADE  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```

CREATE TABLE IF NOT EXISTS `Cook` (
  `Cook_ID` int NOT NULL AUTO_INCREMENT,
  `First_Name` varchar(20) NOT NULL,
  `Last_Name` varchar(20) NOT NULL,
  `Phone_Number` bigint NOT NULL,
  `Date_of_Birth` date NOT NULL,
  `Years_of_Experience` int NOT NULL,
  `Cook_Level` enum('C Cook', 'B Cook', 'A Cook', 'Chef Assistant', 'Chef') NOT NULL,
  `Image_ID` int DEFAULT 1,
  PRIMARY KEY (`Cook_ID`),
  FOREIGN KEY (`Image_ID`) REFERENCES `Images`(`Image_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

```

CREATE TABLE IF NOT EXISTS `Recipe_Cooks` (
  `Recipe_ID` int,
  `Cook_ID` int,
  PRIMARY KEY (`Recipe_ID`, `Cook_ID`),
  FOREIGN KEY (`Recipe_ID`) REFERENCES `Recipe`(`Recipe_ID`) ON DELETE CASCADE,
  FOREIGN KEY (`Cook_ID`) REFERENCES `Cook`(`Cook_ID`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

```

CREATE TABLE IF NOT EXISTS `CookSpecializesInCuisine` (
  `Cook_ID` int,
  `Cuisine_ID` int,
  PRIMARY KEY (`Cook_ID`, `Cuisine_ID`),
  FOREIGN KEY (`Cook_ID`) REFERENCES `Cook`(`Cook_ID`) ON DELETE CASCADE,
  FOREIGN KEY (`Cuisine_ID`) REFERENCES `Cuisine`(`Cuisine_ID`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

```

CREATE TABLE IF NOT EXISTS `Season` (
  `Season_ID` int AUTO_INCREMENT,
  `Year` int,
  PRIMARY KEY (`Season_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

```

CREATE TABLE IF NOT EXISTS `Episode` (
  `Episode_ID` int AUTO_INCREMENT,
  `Episode_No` tinyint NOT NULL CHECK (`Episode_No` >= 1 AND `Episode_No` <= 10),
  `Season_ID` int,
  PRIMARY KEY (`Episode_ID`),
  FOREIGN KEY (`Season_ID`) REFERENCES `Season`(`Season_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

```

CREATE TABLE IF NOT EXISTS `EpisodeJudge` (
  `Episode_ID` int,
  `Cook_ID` int,
  PRIMARY KEY (`Episode_ID`, `Cook_ID`),
  FOREIGN KEY (`Episode_ID`) REFERENCES `Episode`(`Episode_ID`) ON DELETE CASCADE,
  FOREIGN KEY (`Cook_ID`) REFERENCES `Cook`(`Cook_ID`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

```

CREATE TABLE IF NOT EXISTS `EpisodeCookCuisineRecipe` (
  `Episode_ID` int,
  `Cook_ID` int,
  `Cuisine_ID` int,
  `Recipe_ID` int,
  PRIMARY KEY (`Episode_ID`, `Cook_ID`),
  FOREIGN KEY (`Episode_ID`) REFERENCES `Episode`(`Episode_ID`) ON DELETE CASCADE,
  FOREIGN KEY (`Cook_ID`) REFERENCES `Cook`(`Cook_ID`) ON DELETE CASCADE,
  FOREIGN KEY (`Cuisine_ID`) REFERENCES `Cuisine`(`Cuisine_ID`) ON DELETE CASCADE,
  FOREIGN KEY (`Recipe_ID`) REFERENCES `Recipe`(`Recipe_ID`) ON DELETE CASCADE,
  UNIQUE (`Episode_ID`, `Recipe_ID`),
  UNIQUE (`Episode_ID`, `Cuisine_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

CREATE TABLE IF NOT EXISTS `CookRatings` (
  `Episode_ID` int,
  `Judge_Cook_ID` int,
  `Episode_Cook_ID` int,
  `Rating` tinyint NOT NULL CHECK (`Rating` BETWEEN 1 AND 5),
  PRIMARY KEY (`Episode_ID`, `Judge_Cook_ID`, `Episode_Cook_ID`),
  FOREIGN KEY (`Episode_ID`) REFERENCES `Episode`(`Episode_ID`),
  FOREIGN KEY (`Judge_Cook_ID`) REFERENCES `Cook`(`Cook_ID`),
  FOREIGN KEY (`Episode_Cook_ID`) REFERENCES `Cook`(`Cook_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

CREATE TABLE IF NOT EXISTS `Cook` (
  `Cook_ID` int NOT NULL AUTO_INCREMENT,
  `First_Name` varchar(20) NOT NULL,
  `Last_Name` varchar(20) NOT NULL,
  `Phone_Number` bigint NOT NULL,
  `Date_of_Birth` date NOT NULL,
  `Years_of_Experience` int NOT NULL,
  `Cook_Level` enum('C Cook', 'B Cook', 'A Cook', 'Chef Assistant', 'Chef') NOT NULL,
  `Image_ID` int DEFAULT 1,
  PRIMARY KEY (`Cook_ID`),
  FOREIGN KEY (`Image_ID`) REFERENCES `Images`(`Image_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

CREATE TABLE IF NOT EXISTS `Recipe_Cooks` (
  `Recipe_ID` int,
  `Cook_ID` int,
  PRIMARY KEY (`Recipe_ID`, `Cook_ID`),
  FOREIGN KEY (`Recipe_ID`) REFERENCES `Recipe`(`Recipe_ID`) ON DELETE CASCADE,
  FOREIGN KEY (`Cook_ID`) REFERENCES `Cook`(`Cook_ID`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

CREATE TABLE IF NOT EXISTS `CookSpecializesInCuisine` (

```

```

`Cook_ID` int,
`Cuisine_ID` int,
PRIMARY KEY (`Cook_ID`, `Cuisine_ID`),
FOREIGN KEY (`Cook_ID`) REFERENCES `Cook`(`Cook_ID`) ON DELETE CASCADE,
FOREIGN KEY (`Cuisine_ID`) REFERENCES `Cuisine`(`Cuisine_ID`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

CREATE TABLE IF NOT EXISTS `Season` (
  `Season_ID` int AUTO_INCREMENT,
  `Year` int,
  PRIMARY KEY (`Season_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

CREATE TABLE IF NOT EXISTS `Episode` (
  `Episode_ID` int AUTO_INCREMENT,
  `Episode_No` tinyint NOT NULL CHECK (`Episode_No` >= 1 AND `Episode_No` <= 10),
  `Season_ID` int,
  PRIMARY KEY (`Episode_ID`),
  FOREIGN KEY (`Season_ID`) REFERENCES `Season`(`Season_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

CREATE TABLE IF NOT EXISTS `EpisodeJudge` (
  `Episode_ID` int,
  `Cook_ID` int,
  PRIMARY KEY (`Episode_ID`, `Cook_ID`),
  FOREIGN KEY (`Episode_ID`) REFERENCES `Episode`(`Episode_ID`) ON DELETE CASCADE,
  FOREIGN KEY (`Cook_ID`) REFERENCES `Cook`(`Cook_ID`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

CREATE TABLE IF NOT EXISTS `EpisodeCookCuisineRecipe` (
  `Episode_ID` int,
  `Cook_ID` int,
  `Cuisine_ID` int,
  `Recipe_ID` int,
  PRIMARY KEY (`Episode_ID`, `Cook_ID`),
  FOREIGN KEY (`Episode_ID`) REFERENCES `Episode`(`Episode_ID`) ON DELETE CASCADE,
  FOREIGN KEY (`Cook_ID`) REFERENCES `Cook`(`Cook_ID`) ON DELETE CASCADE,
  FOREIGN KEY (`Cuisine_ID`) REFERENCES `Cuisine`(`Cuisine_ID`) ON DELETE CASCADE,
  FOREIGN KEY (`Recipe_ID`) REFERENCES `Recipe`(`Recipe_ID`) ON DELETE CASCADE,
  UNIQUE (`Episode_ID`, `Recipe_ID`),
  UNIQUE (`Episode_ID`, `Cuisine_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

CREATE TABLE IF NOT EXISTS `CookRatings` (
  `Episode_ID` int,
  `Judge_Cook_ID` int,
  `Episode_Cook_ID` int,
  `Rating` tinyint NOT NULL CHECK (`Rating` BETWEEN 1 AND 5),
  PRIMARY KEY (`Episode_ID`, `Judge_Cook_ID`, `Episode_Cook_ID`),

```

```
FOREIGN KEY (`Episode_ID`) REFERENCES `Episode`(`Episode_ID`),  
FOREIGN KEY (`Judge_Cook_ID`) REFERENCES `Cook`(`Cook_ID`),  
FOREIGN KEY (`Episode_Cook_ID`) REFERENCES `Cook`(`Cook_ID`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Οδηγίες Εκτέλεσης:

Χρησιμοποιήσαμε server version 11.3.2-MariaDB μέσω XAMPP.

Αρχικά, ξεκινήστε τη βάση με XAMPP, συνδεθείτε με MySQL και δημιουργήστε τη βάση με την εξής εντολή (προσοχή στο όνομα):

```
CREATE DATABASE NTUACHEF;
```

```
USE NTUACHEF
```

Έπειτα, εκτελέστε την ακόλουθη εντολή για να φορτώσει πλήρως η βάση:

```
SOURCE C:/path_to_folder/Everything.sql;
```

Για να γίνει η κλήρωση για τη δημιουργία νέου επεισοδίου για δεδομένη σεζόν, δεδομένων όλων των περιορισμών, ορίζεται η συνάρτηση `generate_new_episode(season_year)` στο αρχείο `generate_episode_for_season.py`, η οποία παίρνει ως παράμετρο το έτος της αναφερόμενης σεζόν. Χρειάζεται τις βιβλιοθήκες `random` και `mysql.connector`.

Για δημιουργία administrator και παραδείγματος χρήστη μάγειρα, εκτελέστε τον sql κώδικα που βρίσκεται στο αρχείο `AdminUserCreation.sql`. Για να αλλάξει τις πληροφορίες του ο μάγειρας θα πρέπει να κάνει την εξής κλήση:

```
CALL UpdateCookDetails(3, 'NewFirstName', 'NewLastName', 1234567890);
```

Για την ανακήρυξη του νικητή ενός επεισοδίου τρέξτε την ακόλουθη εντολή:

```
CALL FindEpisodeWinner(Episode.ID);
```

, όπου `Episode.ID` <-- επεισόδιο του οποίου θέλω να βρω τον νικητή.

Τα queries βρίσκονται στο αρχείο `queries.sql`.