**ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ**

**ΑΝΑΦΟΡΑ**

**LINK :** **https://github.com/Silia030303/Data\_Base\_NTUA/tree/main**

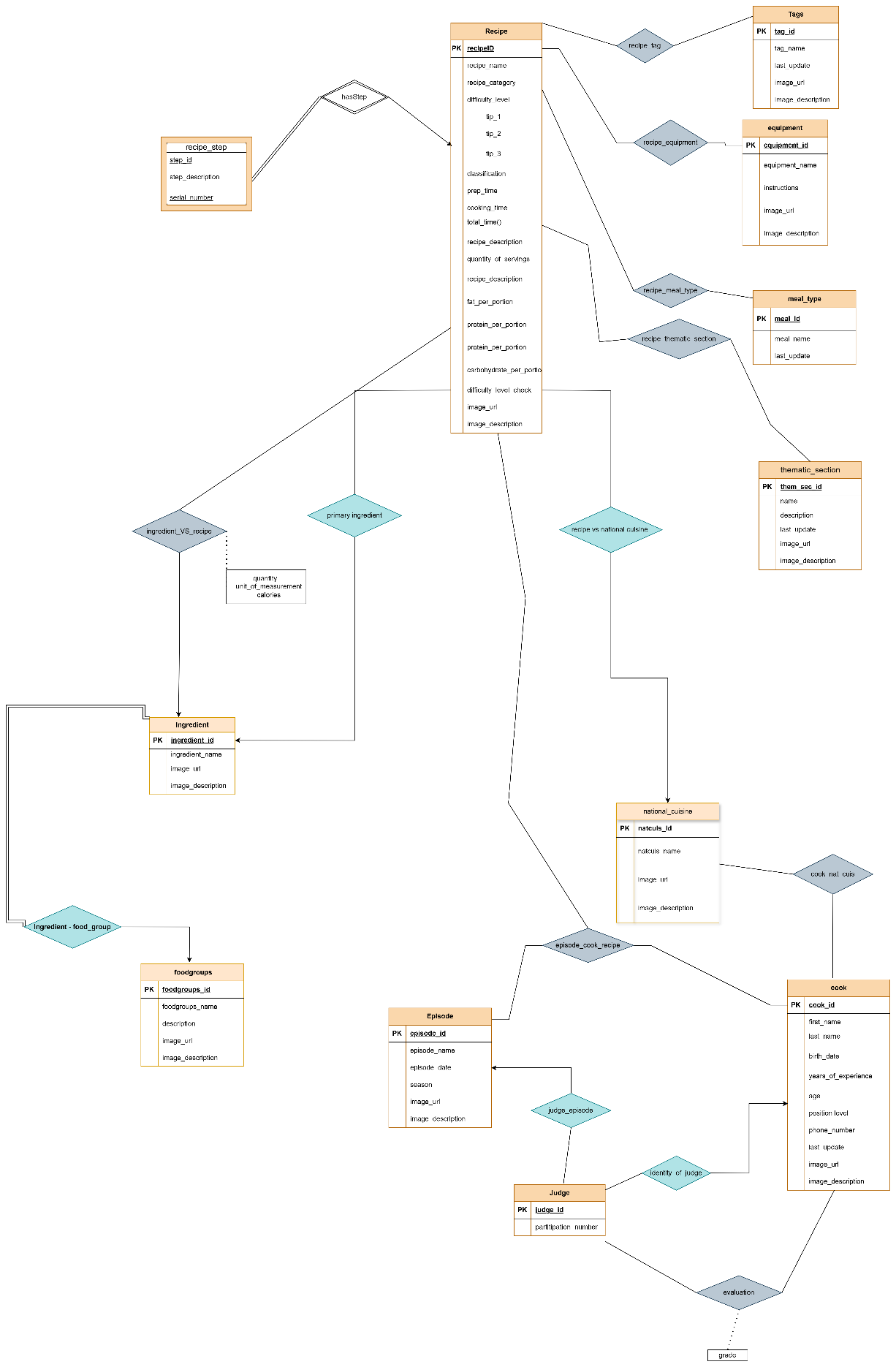
****

**Κοντοθανάση Σωτηρία : 09120080**

**Ρουσσέας Παναγιώτης : 09120113**

**Σκάγκου Ευγενία: 09120014**

**ER Μοντέλο**

****

**Λίγα Λόγια για τον κώδικα :**

Σημείωση: Στην βάση όλες οι σημαντικές οντότητες περιλαμβάνουν δύο attributes

• image\_url

• image\_description

Αυτές δεν περιλαμβάνονται στο ER, γιατί θεωρήσαμε ότι θα απομάκρυναν από την σημαντική πληροφορία που έχει να δώσει και θα μεγάλωναν χωρίς λόγο τους πίνακες.

Ξεκινάμε από τον πίνακα recipes που είναι βασικός και περιέχει όλες τις μοναδικές για την συνταγή πληροφορίες όπως ο χρόνος προετοιμασίας ή οι μερίδες. Μερικά ενδιαφέροντα attributes είναι το total\_time που είναι derived για αυτό εμφανίζεται με παρένθεση δίπλα και το classification (‘Κάθε συνταγή, χαρακτηρίζεται με βάση το βασικό υλικό της’) η οποία στην βάση θα εισάγεται αφού έχει μπει η συνταγή πέρνοντας υπόψιν το foodgroup στο οποίο ανήκει το primary ingredient. Επίσης ενδιαφέρον είναι και το tips , όπου επιλέξαμε να είναι attribute και όχι πίνακας καθώς υπήρχε το όριο των 3 tips. Από την άλλη τα tags που θα μπορούσαν να είναι άπειρα είναι πίνακας με πολλά προς πολλά σχέση με το recipes. Αντίστοιχοι είναι οι πίνακες για το equipment, meal\_type (μορφές γεύματος), thematic\_section(θεματικές ενότητες) και ingredient . Ταυτόχρονα υπάρχουν και σχέσεις πολλά προς ένα όπως η recipe\_vs\_national\_cuisine καθώς κάθε συνταγή θα έχει μόνο μία εθνική κουζίνα και ένα κύριο συστατικό (primary ingredient).

Να σημειωθεί πως αυτές οι σχέσεις στο σχεσιακό μοντέλο εξαφανίζονται και προσθέτονται τα primary keys των ‘προς ένα’ πινάκων στα attributes των ‘προς πολλά’ πινάκων (πχ natcuis\_id). Στο recipe έχουμε ακόμα ένα weak πίνακα τον recipe\_steps ο οποίος δεν έχει λόγο ύπαρξης χωρίς τις συνταγές σε αντίθεση με τις εθνικές κουζίνες πχ που προσδιορίζουν και τους μάγειρες.

Οι μάγειρες τώρα έχουν μια πολλά προς πολλά σχέση με τις εθνικές κουζίνες καθώς μπορούν να είναι ειδικοί σε πολλές και συμμετέχουν σε τρεις ακόμα σχέσεις καθώς μπορούν

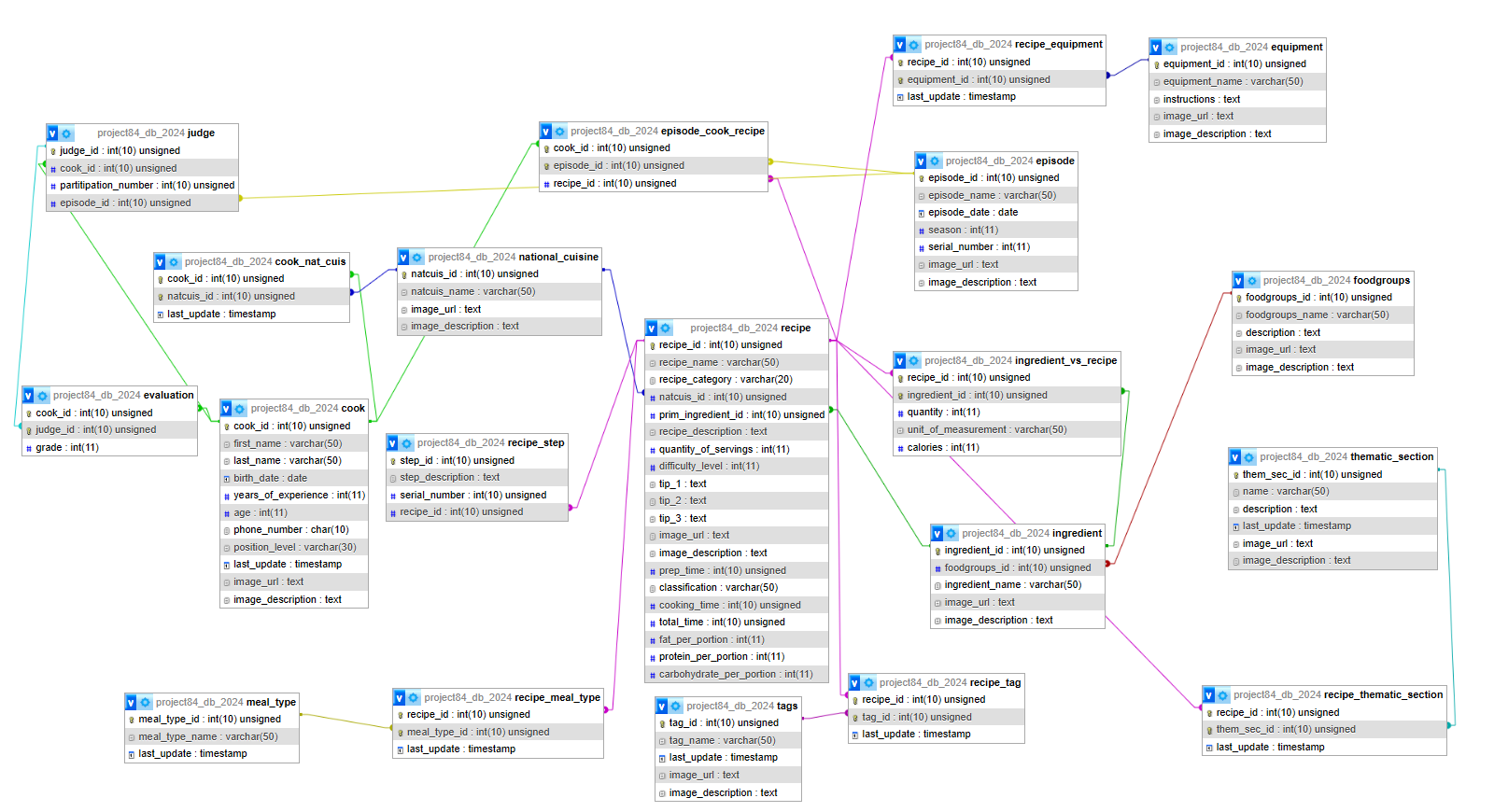
1) Να συμμετέχουν σε κάποιο επεισόδιο ως διαγωνιζόμενος (episode\_cook\_recipe)

2) Να συμμετέχουν σε κάποιο επεισόδιο ως κριτής (identity\_of\_judge)

3) Να βαθμολογήσουν και να βαθμολογηθούν (evaluation)

Τα επεισόδια έχουν και ημερομηνία και σχεζόν αλλά θεωρούμε ότι ένας χρόνος ταυτίζεται με μία σεζόν (πχ 2024 = σεζον 1). Παρατηρούμε επίσης ότι ένα επεισόδιο μπορεί να έχει πολλούς judges αλλά ένας judge μόνο ένα επεισόδιο καθώς ο judge έχει ως primary key (cook\_id, episode\_id). Επιλέον, η σχέση episode\_cook\_recipe είναι η μόνη σχέση ανάμεσα σε τρεις πίνακες αλλά ήταν απολύτως απαραίτητη καθώς σε κάθε επεισόδιο ένας μάγειρας μαγειρεύει μία συνταγή και δεν συνδέονται οι πληροφορίες αυτές με άλλο τρόπο.

Τέλος, τα foodgroups συνδέονται με διπλή γραμμή με τα ingredients καθώς ένα ingredient πρέπει να ανήκει σε ένα και μόνο foodgroup .

**Σχεσιακό διάγραμμα της ΒΔ**

**Εγκατάστασή Εφαρμογής – Ως Root**

**Βήμα 1 :**

Κατέβασε ένα δεν έχεις κάποιο RDBMS , εμείς συστήνουμε τον XAMPP

**Βήμα 2 :**

Πήγαινε στο link στο github :

[**https://****github.com/Silia030303/Data\_Base\_NTUA**](https://github.com/Silia030303/Data_Base_NTUA)

**Βήμα 3 :**

Τρέξε στο terminal σου τον κώδικα του φακέλου : **masterc****hef.sql**

**Βήμα 4:**

Τρέξε στο terminal σου τον κώδικα του φακέλου : **Triggers.sql**

**Βήμα 5 : Εισαγωγή Δεδομένων στην Βάση**

Τρέξε στο terminal σου τον κώδικα του φακέλου : **dml\_link.sql**

Ύστερα για να περάσεις τα random δεδομένα θα πρέπει να ακολουθήσεις τα εξής βήματα :

* Κατέβασε VisualStudio (ή κάποιο άλλο εργαλείο ώστε να τρέχεις python ): [Download Visual Studio Code - Mac, Linux, Windows](https://code.visualstudio.com/Download)
* Κατέβασε την python: [Download Python | Python.org](https://www.python.org/downloads/)
* Κατέβασε τις παρακάτω βιβλιοθήκες πληκτρολογώντας στον terminal σου τις ακόλουθες εντολές:

**pip install mysql-connector-python**

* Κατέβασε τα παρακάτω 3 αρχεία από το github repository και τρέξε τους με την σειρά που δίνετε :
  1. **episode\_cook\_recipe inserts.py**
  2. **judge inserts.py**
  3. **evaluation inserts.py**

Τώρα έχουμε γεμίσει την βάση μας με δεδομένα.

**Βήμα 6 :**

Τρέξε στο terminal σου τον κώδικα του φακέλου : **Indexes.sql**

**Είσαι έτοιμος !!**

**Ως User**

Αφού έχουμε φτιάξει την Βάση Δεδομένων θέλουμε να συνδεθούμε στην ως απλοί χρήστες και όχι ως διαχειριστές .

Γράφουμε στον terminal μας τις εξης εντολές :

1. **cd C:\xampp\mysql\bin**
2. **mysql -u (το username που σας έχει δοθεί ) -p ( ο κωδικός που σας έχει δοθεί )**

Τώρα έχουμε συνδεθεί στο RDBMS ως Users:

Έπειτα για να συνδεθούμε στην Βάση πληκτρολογούμε : **USE project84\_DB\_2024;**

**Λίγα Λογία για τον κώδικα :**

Ο κώδικας στο αρχείο **mastercef.sql** δημιουργεί μια βάση δεδομένων **project84\_DB\_2024** με αρκετούς πίνακες, σχέσεις και προβολές (views) που σχετίζονται με μια εφαρμογή που διαχειρίζεται συνταγές, μάγειρες, αξιολογήσεις, εξοπλισμό και άλλα στοιχεία μαγειρικής. Ακολουθεί μια περιγραφή των κύριων στοιχείων του κώδικα:

1. **Δημιουργία της βάσης δεδομένων**

DROP DATABASE IF EXISTS project84\_DB\_2024;

CREATE database project84\_DB\_2024;

USE project84\_DB\_2024;

Αρχικά, η βάση δεδομένων **project84\_DB\_2024** διαγράφεται εάν υπάρχει, και στη συνέχεια δημιουργείται εκ νέου και επιλέγεται για χρήση.

1. **Δημιουργία πινάκων**

* **Πίνακες βασικών δεδομένων**
  + **equipment:** Αποθηκεύει πληροφορίες για τον εξοπλισμό που χρησιμοποιείται στις συνταγές.
  + **foodgroups**: Περιέχει τις κατηγορίες τροφίμων.
  + **national\_cuisine**: Αποθηκεύει πληροφορίες για τις εθνικές κουζίνες.
  + **meal\_type**: Κατηγοριοποιεί τις συνταγές με βάση τον τύπο γεύματος (π.χ., πρωινό, μεσημεριανό).
  + **ingredient**: Περιέχει τα συστατικά των συνταγών, με αναφορές στις κατηγορίες τροφίμων (foodgroups).
  + **recipe**: Περιέχει τις συνταγές, συμπεριλαμβανομένων λεπτομερειών όπως η κατηγορία, η κουζίνα, τα κύρια συστατικά, οι χρόνοι προετοιμασίας και μαγειρέματος, και οι διατροφικές πληροφορίες.
  + **tags:** Περιέχει ετικέτες για τις συνταγές.
  + **episode**: Αποθηκεύει τα επεισόδια ενός μαγειρικού διαγωνισμού.
  + **cook:** Περιέχει πληροφορίες για τους μάγειρες.
  + **judge**: Αποθηκεύει τους κριτές, οι οποίοι είναι επίσης μάγειρες, και τα επεισόδια στα οποία συμμετέχουν ως κριτές.
  + **recipe\_step:** Περιέχει τα βήματα των συνταγών.
  + **thematic\_section**: Περιέχει θεματικές ενότητες στις οποίες μπορεί να ανήκουν οι συνταγές.
* **Πίνακες σχέσεων**
  + **recipe\_equipment:** Συνδέει συνταγές με τον εξοπλισμό που χρησιμοποιείται.
  + **ingredient\_VS\_recipe:** Συνδέει συνταγές με τα συστατικά τους, μαζί με τις ποσότητες και τις θερμίδες.
  + **recipe\_tag:** Συνδέει συνταγές με ετικέτες.
  + **recipe\_meal\_type:** Συνδέει συνταγές με τύπους γεύματος.
  + **episode\_cook\_recipe:** Συνδέει μάγειρες, επεισόδια και συνταγές.
  + **cook\_natcuis:** Συνδέει μάγειρες με εθνικές κουζίνες.
  + **recipe\_thematic\_section:** Συνδέει συνταγές με θεματικές ενότητες.
* **Πίνακας αξιολογήσεων**
  + **evaluation: Αποθηκεύει τις αξιολογήσεις που δίνουν οι κριτές στις συνταγές των μαγείρων.**

1. **Δημιουργία προβολών (Views)**

* **recipe\_nutritional\_info\_vw :** Αυτή η προβολή παρέχει πληροφορίες για τις θερμίδες ανά μερίδα των συνταγών.
* **winner\_vw :** Αυτή η προβολή υπολογίζει τον νικητή κάθε επεισοδίου βασιζόμενη στη βαθμολογία των μαγείρων.

**ΚΏΔΙΚΑΣ : mastercef.sql**

**-**-DDL

DROP DATABASE IF EXISTS project84\_DB\_2024;

CREATE database project84\_DB\_2024;

use project84\_DB\_2024;

--DROP TABLE IF EXISTS recipe;

-- DELETE FROM table\_name;

CREATE TABLE equipment(

equipment\_id INT(10) unsigned AUTO\_INCREMENT NOT NULL,

equipment\_name VARCHAR(50) NOT NULL ,

instructions text DEFAULT NULL ,

image\_url text DEFAULT NULL,

image\_description text DEFAULT NULL,

PRIMARY KEY(equipment\_id)

);

CREATE TABLE foodgroups(

foodgroups\_id INT(10) unsigned AUTO\_INCREMENT NOT NULL,

foodgroups\_name VARCHAR(50) NOT NULL,

description text DEFAULT NULL,

image\_url text DEFAULT NULL,

image\_description text DEFAULT NULL,

PRIMARY KEY(foodgroups\_id)

);

CREATE TABLE national\_cuisine(

natcuis\_id INT(10) unsigned AUTO\_INCREMENT NOT NULL,

natcuis\_name VARCHAR(50) NOT NULL,

image\_url text DEFAULT NULL,

image\_description text DEFAULT NULL,

PRIMARY KEY(natcuis\_id)

);

CREATE TABLE meal\_type(

meal\_type\_id INT(10) unsigned AUTO\_INCREMENT NOT NULL,

meal\_type\_name VARCHAR(50) NOT NULL,

last\_update timestamp NOT NULL DEFAULT current\_timestamp() ON UPDATE current\_timestamp(),

PRIMARY KEY(meal\_type\_id)

);

CREATE TABLE ingredient(

ingredient\_id INT(10) unsigned AUTO\_INCREMENT NOT NULL,

foodgroups\_id INT(10) unsigned NOT NULL,

ingredient\_name VARCHAR(50) NOT NULL,

image\_url text DEFAULT NULL,

image\_description text DEFAULT NULL,

PRIMARY KEY(ingredient\_id),

FOREIGN KEY(foodgroups\_id) REFERENCES foodgroups(foodgroups\_id)

);

CREATE TABLE recipe(

recipe\_id INT(10) unsigned AUTO\_INCREMENT NOT NULL,

recipe\_name VARCHAR(50) NOT NULL,

recipe\_category VARCHAR(20) NOT NULL,

CONSTRAINT Check\_YourCategory CHECK (recipe\_category IN ('main course', 'dessert')) ,

natcuis\_id INT(10) unsigned NOT NULL ,

prim\_ingredient\_id INT(10) unsigned NOT NULL,

recipe\_description text DEFAULT NULL ,

quantity\_of\_servings INT NOT NULL,

difficulty\_level INT NOT NULL,

CONSTRAINT difficulty\_level\_check CHECK (difficulty\_level IN (1,2,3,4,5)),

tip\_1 text DEFAULT NULL,

tip\_2 text DEFAULT NULL,

tip\_3 text DEFAULT NULL,

image\_url text DEFAULT NULL,

image\_description text DEFAULT NULL,

prep\_time INT unsigned NOT NULL,

classification VARCHAR(50),

cooking\_time INT unsigned NOT NULL,

total\_time INT unsigned AS (prep\_time + cooking\_time) STORED,

fat\_per\_portion INT NOT NULL,

protein\_per\_portion INT NOT NULL,

carbohydrate\_per\_portion INT NOT NULL,

KEY idx\_total\_time (total\_time),

FOREIGN KEY(natcuis\_id) REFERENCES national\_cuisine(natcuis\_id),

FOREIGN KEY(prim\_ingredient\_id) REFERENCES ingredient(ingredient\_id),

PRIMARY KEY(recipe\_id)

);

CREATE TABLE recipe\_equipment(

recipe\_id INT unsigned NOT NULL,

equipment\_id INT unsigned NOT NULL,

last\_update timestamp NOT NULL DEFAULT current\_timestamp() ON UPDATE current\_timestamp(),

FOREIGN KEY(recipe\_id) REFERENCES recipe(recipe\_id),

FOREIGN KEY(equipment\_id) REFERENCES equipment(equipment\_id),

PRIMARY KEY(recipe\_id,equipment\_id)

);

CREATE TABLE ingredient\_VS\_recipe(

recipe\_id INT(10) unsigned NOT NULL,

ingredient\_id INT(10) unsigned NOT NULL,

PRIMARY KEY (recipe\_id, ingredient\_id),

FOREIGN KEY (recipe\_id) REFERENCES recipe(recipe\_id),

FOREIGN KEY (ingredient\_id) REFERENCES ingredient(ingredient\_id),

quantity INT NOT NULL CHECK (quantity >= 0),

unit\_of\_measurement VARCHAR(50) DEFAULT NULL ,

calories INT NOT NULL CHECK (calories >= 0)

);

CREATE TABLE tags(

tag\_id INT(10) unsigned AUTO\_INCREMENT NOT NULL,

tag\_name VARCHAR(50) NOT NULL,

last\_update timestamp NOT NULL DEFAULT current\_timestamp() ON UPDATE current\_timestamp(),

image\_url text DEFAULT NULL,

image\_description text DEFAULT NULL,

PRIMARY KEY(tag\_id)

);

CREATE TABLE recipe\_tag(

recipe\_id INT unsigned NOT NULL,

tag\_id INT unsigned NOT NULL,

last\_update timestamp NOT NULL DEFAULT current\_timestamp() ON UPDATE current\_timestamp(),

FOREIGN KEY(recipe\_id) REFERENCES recipe(recipe\_id),

FOREIGN KEY(tag\_id) REFERENCES tags(tag\_id),

PRIMARY KEY(recipe\_id,tag\_id)

);

CREATE TABLE recipe\_meal\_type (

recipe\_id INT unsigned NOT NULL,

meal\_type\_id INT unsigned NOT NULL,

FOREIGN KEY(recipe\_id) REFERENCES recipe(recipe\_id),

FOREIGN KEY(meal\_type\_id) REFERENCES meal\_type(meal\_type\_id),

last\_update timestamp NOT NULL DEFAULT current\_timestamp() ON UPDATE current\_timestamp(),

PRIMARY KEY(recipe\_id,meal\_type\_id)

);

CREATE TABLE episode(

episode\_id INT(10) unsigned AUTO\_INCREMENT NOT NULL,

episode\_name VARCHAR(50) NOT NULL,

episode\_date date NOT NULL,

season INT NOT NULL CHECK (season >= 0),

serial\_number INT NOT NULL CHECK (serial\_number >= 1),

image\_url text DEFAULT NULL,

image\_description text DEFAULT NULL,

primary key(episode\_id)

);

CREATE TABLE cook(

cook\_id INT(10) unsigned NOT NULL AUTO\_INCREMENT,

first\_name VARCHAR(50) NOT NULL,

last\_name VARCHAR(50) NOT NULL,

birth\_date DATE NOT NULL,

years\_of\_experience INT NOT NULL,

age INT AS (YEAR(CURDATE()) - YEAR(birth\_date) - (DATE\_FORMAT(CURDATE(), '%m%d') < DATE\_FORMAT(birth\_date, '%m%d'))),

phone\_number CHAR(10) NOT NULL,

position\_level VARCHAR(30) CHECK (position\_level IN ('cook A','cook B','cook C','chef assistant','chef')),

last\_update TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP(),

image\_url text DEFAULT NULL,

image\_description text DEFAULT NULL,

PRIMARY KEY(cook\_id)

);

CREATE TABLE judge(

judge\_id INT(10) unsigned NOT NULL AUTO\_INCREMENT,

cook\_id INT(10) unsigned NOT NULL,

participation\_number INT unsigned DEFAULT NULL,

episode\_id INT(10) unsigned NOT NULL,

PRIMARY KEY(judge\_id),

FOREIGN KEY (cook\_id) REFERENCES cook(cook\_id),

FOREIGN KEY(episode\_id) REFERENCES episode(episode\_id)

);

CREATE TABLE recipe\_step (

step\_id INT unsigned NOT NULL AUTO\_INCREMENT,

step\_description text DEFAULT NULL ,

serial\_number INT unsigned NOT NULL,

/\*add trigger (?)\*/

recipe\_id INT(10) unsigned NOT NULL,

FOREIGN KEY(recipe\_id) REFERENCES recipe(recipe\_id),

PRIMARY KEY(step\_id)

);

CREATE TABLE thematic\_section(

them\_sec\_id INT(10) unsigned AUTO\_INCREMENT NOT NULL,

name VARCHAR(50) NOT NULL,

description text DEFAULT NULL,

last\_update timestamp NOT NULL DEFAULT current\_timestamp() ON UPDATE current\_timestamp(),

image\_url text DEFAULT NULL,

image\_description text DEFAULT NULL,

PRIMARY KEY(them\_sec\_id)

);

CREATE TABLE recipe\_thematic\_section(

recipe\_id INT(10) unsigned NOT NULL,

them\_sec\_id INT(10) unsigned NOT NULL,

last\_update timestamp NOT NULL DEFAULT current\_timestamp() ON UPDATE current\_timestamp(),

PRIMARY KEY(recipe\_id,them\_sec\_id),

FOREIGN KEY(them\_sec\_id) REFERENCES thematic\_section(them\_sec\_id),

FOREIGN KEY(recipe\_id) REFERENCES recipe(recipe\_id)

);

CREATE TABLE evaluation (

cook\_id INT unsigned NOT NULL,

judge\_id INT unsigned NOT NULL,

grade INT NOT NULL,

CONSTRAINT grade\_check CHECK (grade IN (1,2,3,4,5)),

FOREIGN KEY(cook\_id) REFERENCES cook(cook\_id),

FOREIGN KEY(judge\_id) REFERENCES judge(judge\_id),

PRIMARY KEY(cook\_id,judge\_id)

);

CREATE TABLE episode\_cook\_recipe(

cook\_id INT(10) unsigned NOT NULL,

episode\_id INT(10) unsigned NOT NULL,

recipe\_id INT(10) unsigned NOT NULL,

FOREIGN KEY(cook\_id) REFERENCES cook(cook\_id),

FOREIGN KEY(episode\_id) REFERENCES episode(episode\_id),

FOREIGN KEY(recipe\_id) REFERENCES recipe(recipe\_id),

PRIMARY KEY(cook\_id,episode\_id)

);

CREATE TABLE cook\_nat\_cuis(

cook\_id INT unsigned NOT NULL,

natcuis\_id INT unsigned NOT NULL,

last\_update timestamp NOT NULL DEFAULT current\_timestamp() ON UPDATE current\_timestamp(),

FOREIGN KEY(cook\_id) REFERENCES cook(cook\_id),

FOREIGN KEY(natcuis\_id) REFERENCES national\_cuisine(natcuis\_id),

PRIMARY KEY(cook\_id,natcuis\_id)

);

-- recipe\_nutritional\_info view

CREATE VIEW recipe\_nutritional\_info\_vw AS

SELECT

table1.recipe\_id,

table1.recipe\_name,

table1.quantity\_of\_servings,

table1.fat\_per\_portion,

table1.protein\_per\_portion,

table1.carbohydrate\_per\_portion,

CASE

WHEN table1.quantity\_of\_servings = 0 THEN NULL

ELSE table1.total\_calories / table1.quantity\_of\_servings

END AS calories\_per\_portion

FROM

(SELECT

r.recipe\_id,

r.recipe\_name,

r.quantity\_of\_servings,

r.fat\_per\_portion,

r.protein\_per\_portion,

r.carbohydrate\_per\_portion,

SUM(ir.calories) AS total\_calories

FROM

recipe r

JOIN

ingredient\_VS\_recipe ir

ON

r.recipe\_id = ir.recipe\_id

GROUP BY

r.recipe\_id) AS table1;

-- winner view

CREATE VIEW winner\_vw AS

WITH RankedCooks AS (

SELECT

ep.episode\_name,

c.last\_name,

CASE c.position\_level

WHEN 'cook A' THEN 1

WHEN 'cook B' THEN 2

WHEN 'cook C' THEN 3

WHEN 'chef assistant' THEN 4

WHEN 'chef' THEN 5

END as position\_level,

SUM(e.grade) AS score,

ROW\_NUMBER() OVER (

PARTITION BY ecr.episode\_id

ORDER BY

SUM(e.grade) DESC,

CASE c.position\_level

WHEN 'cook A' THEN 1

WHEN 'cook B' THEN 2

WHEN 'cook C' THEN 3

WHEN 'chef assistant' THEN 4

WHEN 'chef' THEN 5

END DESC,

RAND()

) AS rank

FROM

episode\_cook\_recipe ecr

JOIN

cook c ON c.cook\_id = ecr.cook\_id

JOIN

evaluation e ON e.cook\_id = c.cook\_id

JOIN

episode ep ON ep.episode\_id = ecr.episode\_id

GROUP BY

ecr.episode\_id,

c.cook\_id

)

SELECT

episode\_name,

last\_name,

position\_level,

score

FROM

RankedCooks

WHERE

rank = 1;

--end

Ο κώδικας στο αρχείο **Triggers.sql** περιλαμβάνει τη δημιουργία triggers (ενεργοποιητών) στη βάση δεδομένων **project84\_DB\_2024**. Αυτοί οι triggers εκτελούν συγκεκριμένες ενέργειες πριν την εισαγωγή ή την ενημέρωση δεδομένων σε ορισμένους πίνακες, ώστε να εξασφαλιστεί η ακεραιότητα των δεδομένων και να εφαρμοστούν επιχειρηματικοί κανόνες. Ακολουθεί μια περιγραφή των triggers που δημιουργήθηκαν:

1. **Triggers για τον πίνακα : ingredient\_vs\_recipe**
   1. **check\_unit\_of\_measurement\_insert:**

Αυτός ο trigger ελέγχει αν η τιμή της στήλης unit\_of\_measurement είναι μία από τις επιτρεπόμενες τιμές πριν την εισαγωγή δεδομένων στον πίνακα ingredient\_vs\_recipe. Εάν δεν είναι, απορρίπτει την εισαγωγή και εμφανίζει μήνυμα σφάλματος.

* 1. **check\_unit\_of\_measurement\_update:**

Παρόμοιος με τον προηγούμενο, αυτός ο trigger ελέγχει τις επιτρεπόμενες τιμές της unit\_of\_measurement πριν την ενημέρωση δεδομένων στον πίνακα ingredient\_vs\_recipe.

1. **Triggers για τον πίνακα : foodgroups**
   1. **check\_foodgroups\_update:**

Αυτός ο trigger ελέγχει αν το foodgroups\_name είναι μία από τις καθορισμένες επιτρεπόμενες τιμές πριν την εισαγωγή δεδομένων στον πίνακα foodgroups.

1. **Triggers για τον πίνακα: ingredient:**
   1. **check\_foodgroups\_update\_in\_ingridient:**

Αυτός ο trigger ελέγχει αν το foodgroups\_id είναι ένα από τα καθορισμένα επιτρεπόμενα IDs πριν την εισαγωγή δεδομένων στον πίνακα ingredient.

1. **Triggers για τον πίνακα: episode\_cook\_recipe**
   1. **before\_episode\_cook\_recipe\_insert**

Αυτός ο trigger ελέγχει αν ο αριθμός των συνταγών και μαγείρων ανά επεισόδιο δεν υπερβαίνει τις 10 πριν την εισαγωγή στον πίνακα episode\_cook\_recipe.

* 1. **before\_episode\_cook\_insert**

Αυτός ο trigger ελέγχει αν ένας μάγειρας δεν συμμετέχει σε τρία συνεχόμενα επεισόδια πριν την εισαγωγή στον πίνακα episode\_cook\_recipe**.**

1. **Triggers για τον πίνακα recipe\_step**
   1. **check\_sequential\_steps:**

Αυτός ο trigger ελέγχει αν κάθε βήμα μιας συνταγής έχει το αμέσως προηγούμενο βήμα πριν την εισαγωγή του στον πίνακα recipe\_step**.**

* 1. **check\_step\_serial\_number**

Αυτός ο trigger ελέγχει αν υπάρχει ήδη βήμα με τον ίδιο σειριακό αριθμό για μια συγκεκριμένη συνταγή πριν την εισαγωγή του στον πίνακα recipe\_step.

**ΚΏΔΙΚΑΣ : Triggers.sql**

DELIMITER //

CREATE TRIGGER check\_unit\_of\_measurement\_insert

BEFORE INSERT ON ingredient\_VS\_recipe

FOR EACH ROW

BEGIN

IF NEW.unit\_of\_measurement NOT IN (

'g',

'kg',

'mg',

'oz',

'lb',

'cup',

'half cup',

'quarter cup',

'three-quarters cup',

'tbsp',

'tsp',

'ml',

'L',

'pint',

'gallon',

'fl. oz',

'piece',

'half piece',

'slice',

'wedge',

'clove',

'head',

'shot',

'pack'

) THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Invalid value for unit\_of\_measurement. Allowed values are: g, kg, mg, oz, lb, cup, half cup, quarter cup, three-quarters cup, tbsp, tsp, ml, L, pint, gallon, fl. oz, piece, half piece, slice, wedge, clove, head, shot, pack';

END IF;

END //

-- Trigger για την ενημέρωση δεδομένων στο unit\_of\_measurement

CREATE TRIGGER check\_unit\_of\_measurement\_update

BEFORE UPDATE ON ingredient\_VS\_recipe

FOR EACH ROW

BEGIN

IF NEW.unit\_of\_measurement NOT IN (

'g',

'kg',

'mg',

'oz',

'lb',

'cup',

'half cup',

'quarter cup',

'three-quarters cup',

'tbsp',

'tsp',

'ml',

'L',

'pint',

'gallon',

'fl. oz',

'piece',

'half piece',

'slice',

'wedge',

'clove',

'head',

'shot',

'pack'

) THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Invalid value for unit\_of\_measurement. Allowed values are: g, kg, mg, oz, lb, cup, half cup, quarter cup, three-quarters cup, tbsp, tsp, ml, L, pint, gallon, fl. oz, piece, half piece, slice, wedge, clove, head, shot, pack';

END IF;

END //

-- Trigger για την ενημέρωση δεδομένων στο foodgroups

CREATE TRIGGER check\_foodgroups\_insert

BEFORE INSERT ON foodgroups

FOR EACH ROW

BEGIN

IF NEW.foodgroups\_name NOT IN (

'Aromatic Herbs and Essential Oils',

'Coffee, Tea, and Their Products',

'Preserved Foods',

'Sweeteners',

'Fats and Oils',

'Milk, Eggs, and Their Products',

'Meat and Meat Products',

'Fish and Fish Products' ,

'Cereals and Their Products',

'Various Plant-based Foods',

'Products with Sweeteners',

'Various Beverages',

'other foodgroup') THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Invalid value for unit\_of\_foodgroups. Allowed values are

Aromatic Herbs and Essential Oils,

Coffee, Tea, and Their Products,

Preserved Foods,

Sweeteners,

Fats and Oils,

Milk, Eggs, and Their Products,

Meat and Meat Products,

Fish and Fish Products ,

Cereals and Their Products,

Various Plant-based Foods,

Products with Sweeteners,

Various Beverages' ;

END IF;

END; //

-- Trigger για την ενημέρωση δεδομένων στο foodgroups

CREATE TRIGGER check\_foodgroups\_insert\_in\_ingridient

BEFORE INSERT ON ingredient

FOR EACH ROW

BEGIN

IF NEW.foodgroups\_id NOT IN (1,2,3,4,5,6,7,8,9,10,11,12,13) THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Invalid value for unit\_of\_measurement\_quantity. Allowed values are:

1 = Aromatic Herbs and Essential Oils,

2 = Coffee, Tea, and Their Products,

3 = Preserved Foods,

4 = Sweeteners,

5 = Fats and Oils,

6 = Milk, Eggs, and Their Products,

7 = Meat and Meat Products,

8= Fish and Fish Products ,

9 = Cereals and Their Products,

10 = Various Plant-based Foods,

11 = Products with Sweeteners,

12 = Various Beverages,

13 = other foodgroup';

END IF;

END; //

-- trigger for only 10 cooks and recipes in each episode

CREATE TRIGGER before\_episode\_cook\_recipe\_insert

BEFORE INSERT ON episode\_cook\_recipe

FOR EACH ROW

BEGIN

DECLARE count\_episodes INT;

-- Count the number of tuples with the same episode\_id

SELECT COUNT(\*) INTO count\_episodes

FROM episode\_cook\_recipe

WHERE episode\_id = NEW.episode\_id;

-- If the count is already 10, prevent insertion

IF count\_episodes >= 10 THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Cannot insert more than 10 tuples with the same episode\_id';

END IF;

END; //

DELIMITER ;

--trigger for nonconsecutive cooks

DELIMITER //

CREATE TRIGGER before\_episode\_cook\_insert

BEFORE INSERT ON episode\_cook\_recipe

FOR EACH ROW

BEGIN

DECLARE episode\_serial\_number INT;

DECLARE prev\_episode\_count INT;

-- Get the serial number of the episode being inserted

SELECT serial\_number INTO episode\_serial\_number

FROM episode

WHERE episode\_id = NEW.episode\_id;

-- Only perform the check if the episode serial number is greater than 2

IF episode\_serial\_number > 2 THEN

-- Check if the cook was in the previous two episodes

SELECT COUNT(\*)

INTO prev\_episode\_count

FROM episode\_cook\_recipe ecr

JOIN episode e ON ecr.episode\_id = e.episode\_id

WHERE ecr.cook\_id = NEW.cook\_id

AND e.serial\_number IN (episode\_serial\_number - 1, episode\_serial\_number - 2);

-- If the cook was in both previous episodes, prevent the insert

IF prev\_episode\_count = 2 THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Cook cannot participate in three consecutive episodes.';

END IF;

END IF;

END;

//

DELIMITER ;

--trigger for no 3 consecutive recipes

DELIMITER //

CREATE TRIGGER before\_episode\_recipe\_insert

BEFORE INSERT ON episode\_cook\_recipe

FOR EACH ROW

BEGIN

DECLARE episode\_serial\_number INT;

DECLARE prev\_episode\_count INT;

-- Get the serial number of the episode being inserted

SELECT serial\_number INTO episode\_serial\_number

FROM episode

WHERE episode\_id = NEW.episode\_id;

-- Only perform the check if the episode serial number is greater than 2

IF episode\_serial\_number > 2 THEN

-- Check if the cook was in the previous two episodes

SELECT COUNT(\*)

INTO prev\_episode\_count

FROM episode\_cook\_recipe ecr

JOIN episode e ON ecr.episode\_id = e.episode\_id

WHERE ecr.cook\_id = NEW.cook\_id

AND e.serial\_number IN (episode\_serial\_number - 1, episode\_serial\_number - 2);

-- If the cook was in both previous episodes, prevent the insert

IF prev\_episode\_count = 2 THEN

SIGNAL SQLSTATE '45000';

SET MESSAGE\_TEXT = 'recipe cannot participate in three consecutive episodes.';

END IF;

END IF;

END;

//

DELIMITER ;

--trigger for no consecutive natcuis

DELIMITER //

CREATE TRIGGER before\_episode\_cook\_insert\_natcuis

BEFORE INSERT ON episode\_cook\_recipe

FOR EACH ROW

BEGIN

DECLARE episode\_serial\_number INT;

DECLARE prev\_episode\_count INT;

-- Get the serial number of the episode being inserted

SELECT serial\_number INTO episode\_serial\_number

FROM episode

WHERE episode\_id = NEW.episode\_id;

-- Only perform the check if the episode serial number is greater than 2

IF episode\_serial\_number > 2 THEN

-- Check if the natcuis was in the previous two episodes

SELECT COUNT(\*)

INTO prev\_episode\_count

FROM episode\_cook\_recipe ecr

JOIN episode e ON ecr.episode\_id = e.episode\_id

JOIN recipe r ON r.recipe\_id = ecr.recipe\_id

JOIN national\_cuisine nc ON nc.natcuis\_id= r.natcuis\_id

WHERE ecr.recipe\_id = NEW.recipe\_id

AND e.serial\_number IN (episode\_serial\_number - 1, episode\_serial\_number - 2)

GROUP BY nc.nat\_cuis\_id;

-- If the natcuis was in both previous episodes, prevent the insert

IF prev\_episode\_count = 2 THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'natcuis cannot participate in three consecutive episodes.';

END IF;

END IF;

END;

//

DELIMITER ;

--no consecutive 3 judges :

DELIMITER //

CREATE TRIGGER before\_judge

BEFORE INSERT ON judge

FOR EACH ROW

BEGIN

DECLARE episode\_serial\_number INT;

DECLARE prev\_episode\_count INT;

-- Get the serial number of the episode being inserted

SELECT serial\_number INTO episode\_serial\_number

FROM episode

WHERE episode\_id = NEW.episode\_id;

-- Only perform the check if the episode serial number is greater than 2

IF episode\_serial\_number > 2 THEN

-- Check if the judge was in the previous two episodes

SELECT COUNT(\*)

INTO prev\_episode\_count

FROM judge j

JOIN episode\_cook\_recipe ecr ON j.episode\_id = ecr.episode\_id

JOIN episode e ON ecr.episode\_id = e.episode\_id

WHERE j.judge\_id = NEW.judge\_id

AND e.serial\_number IN (episode\_serial\_number - 1, episode\_serial\_number - 2);

-- If the judge was in both previous episodes, prevent the insert

IF prev\_episode\_count = 2 THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'judge cannot participate in three consecutive episodes.';

END IF;

END IF;

END;

//

DELIMITER ;

--trigger for steps

DELIMITER $$

CREATE TRIGGER check\_sequential\_steps

BEFORE INSERT ON recipe\_step

FOR EACH ROW

BEGIN

DECLARE prev\_step\_count INT;

IF NEW.serial\_number > 1 THEN

SELECT COUNT(\*) INTO prev\_step\_count FROM recipe\_step WHERE recipe\_id = NEW.recipe\_id AND serial\_number = NEW.serial\_number - 1;

IF prev\_step\_count = 0 THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Cannot insert step without previous step.';

END IF;

END IF;

END$$

DELIMITER ;

DELIMITER $$

CREATE TRIGGER check\_step\_serial\_number

BEFORE INSERT ON recipe\_step

FOR EACH ROW

BEGIN

DECLARE existing\_serial\_count INT;

SELECT COUNT(\*) INTO existing\_serial\_count FROM recipe\_step WHERE recipe\_id = NEW.recipe\_id AND serial\_number = NEW.serial\_number;

IF existing\_serial\_count > 0 THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'A step with the same serial number already exists for this recipe.';

END IF;

END$$

DELIMITER ;

--end

Ο κώδικας στο αρχείο **dml\_final.sql** περιέχει εντολές που χρησιμοποιούνται για την εισαγωγή, δεδομένων στους πίνακες της βάσης δεδομένων. Αυτές οι εντολές εκτελούνται για να προσαρμοστεί το περιεχόμενο της βάσης δεδομένων σύμφωνα με τις ανάγκες της εφαρμογής.

Οι βασικές εντολές DML είναι:

1. **INSERT**: Εισάγει νέες εγγραφές σε έναν πίνακα της βάσης δεδομένων.
2. **UPDATE**: Ενημερώνει τιμές σε υπάρχουσες εγγραφές ενός πίνακα.
3. **DELETE**: Διαγράφει εγγραφές από έναν πίνακα.

**ΚΏΔΙΚΑΣ : dml\_final.sql**

INSERT INTO national\_cuisine (natcuis\_name, image\_url)

VALUES

('Albanian cuisine','https://irp.cdn-website.com/560b6587/dms3rep/multi/image20-f1109880.png');

INSERT INTO national\_cuisine (natcuis\_name) VALUES

('Afghan cuisine'),

('American cuisine'),

('Argentine cuisine'),

('British cuisine'),

('Cambodian cuisine'),

('Colombian cuisine'),

('Costa Rican cuisine'),

('Croatian cuisine'),

('Peruvian cuisine'),

('Polish cuisine'),

('Portuguese cuisine'),

('Ecuadorian cuisine'),

('Egyptian cuisine'),

('Estonian cuisine'),

('Ethiopian cuisine'),

('German cuisine'),

('Greek cuisine'),

('Hungarian cuisine'),

('Indian cuisine'),

('Italian cuisine'),

('Jamaican cuisine'),

('Romanian cuisine'),

('Russian cuisine'),

('Spanish cuisine'),

('Palau cuisine'),

('Mexican cuisine'),

('Latvian cuisine'),

('Ukrainian cuisine'),

('Liberian cuisine'),

('South African cuisine')

;

INSERT INTO cook (first\_name, last\_name, birth\_date, years\_of\_experience, phone\_number, position\_level, image\_url)

VALUES ('Gordon', 'Ramsey', '1966-11-08', 30, '1234567890', 'chef', 'https://hips.hearstapps.com/hmg-prod/images/chef-host-gordon-ramsay-in-the-semi-finale-pt-2-3-chef-news-photo-1699984599.jpg?crop=0.788xw:1.00xh;0.212xw,0&resize=1200:\*');

INSERT INTO cook (first\_name, last\_name, birth\_date, years\_of\_experience, phone\_number, position\_level)

VALUES

('Jamie', 'Oliver', '1975-05-27', 25, '9876543210', 'chef'),

('Bobby', 'Flay', '1964-12-10', 35, '5551234567', 'chef'),

('Nigella', 'Lawson', '1960-01-06', 40, '1112223333', 'chef'),

('Anthony', 'Bourdain', '1956-06-25', 45, '4445556666', 'chef'),

('Julia', 'Child', '1912-08-15', 50, '7778889999', 'chef'),

('Alain', 'Ducasse', '1956-09-13', 40, '2223334444', 'chef'),

('Marco', 'Pierre White', '1961-12-11', 35, '9998887777', 'chef'),

('Thomas', 'Keller', '1955-10-14', 45, '6667778888', 'chef'),

('Massimo', 'Bottura', '1962-09-30', 30, '3334445555', 'chef'),

('Wolfgang', 'Puck', '1949-07-08', 50, '8889990000', 'chef'),

('Gino', 'D Acampo', '1976-07-17', 20, '0009998887', 'cook A'),

('Curtis', 'Stone', '1975-11-04', 15, '9876543210', 'cook B'),

('Giada', 'De Laurentiis', '1970-08-22', 20, '5551234567', 'cook C'),

('Emeril', 'Lagasse', '1959-10-15', 30, '1112223333', 'cook A'),

('Rick', 'Bayless', '1953-11-23', 40, '4445556666', 'cook B'),

('Guy', 'Fieri', '1968-01-22', 25, '7778889999', 'cook C'),

('Bourdain', 'Ramsay', '1970-05-25', 15, '2223334444', 'cook A'),

('Heston', 'Blumenthal', '1966-05-27', 25, '9998887777', 'cook B'),

('José', 'Andrés', '1969-07-13', 20, '6667778888', 'cook C'),

('Yotam', 'Ottolenghi', '1968-12-14', 25, '3334445555', 'cook A'),

('Laura', 'Gomez', '1990-03-15', 5, '1234567890', 'chef assistant'),

('Daniel', 'Smith', '1988-06-20', 8, '9876543210', 'chef assistant'),

('Rachel', 'Johnson', '1992-11-10', 3, '5551234567', 'chef assistant'),

('Christopher', 'Martinez', '1985-01-06', 10, '1112223333', 'chef assistant'),

('Maria', 'Lee', '1991-04-25', 4, '4445556666', 'chef assistant'),

('Alexander', 'Brown', '1989-09-15', 6, '7778889999', 'chef assistant'),

('Emma', 'Davis', '1987-07-20', 7, '2223334444', 'chef assistant'),

('Ryan', 'Chen', '1993-05-18', 2, '9998887777', 'chef assistant'),

('Sara', 'Lee', '1980-09-12', 10, '1234567890', 'cook B'),

('Peter', 'Pan', '1982-07-08', 8, '9876543210', 'cook B'),

('Emily', 'Ross', '1975-03-21', 12, '5551234567', 'cook B'),

('Paula', 'Deen', '1947-01-19', 40, '1234567890', 'cook A'),

('Rachael', 'Ray', '1968-08-25', 25, '9876543210', 'cook C'),

('Thomas', 'Keller', '1955-10-14', 45, '5551234567', 'cook C'),

('Alice', 'Waters', '1944-04-28', 50, '1112223333', 'chef'),

('Masaharu', 'Morimoto', '1955-05-26', 35, '4445556666', 'chef'),

('David', 'Cook', '1987-11-30', 6, '1112223333', 'cook B'),

('Michelle', 'Nguyen', '1984-05-17', 9, '4445556666', 'cook B'),

('James', 'Wong', '1981-12-25', 11, '7778889999', 'cook B'),

('Amanda', 'Scott', '1978-08-18', 10, '2223334444', 'cook B'),

('Kevin', 'Ng', '1985-02-14', 7, '9998887777', 'cook B'),

('Amit', 'Patel', '1988-03-15', 8, '1234567890', 'cook B'),

('Priya', 'Sharma', '1990-06-20', 5, '9876543210', 'cook B'),

('Raj', 'Kumar', '1992-11-10', 3, '5551234567', 'chef assistant'),

('Anjali', 'Mehta', '1985-01-06', 10, '1112223333', 'chef'),

('Rahul', 'Singh', '1991-04-25', 4, '4445556666', 'chef assistant'),

('Sneha', 'Gupta', '1989-09-15', 6, '7778889999', 'chef assistant'),

('Vikram', 'Shah', '1987-07-20', 7, '2223334444', 'chef'),

('Neha', 'Verma', '1993-05-18', 2, '9998887777', 'chef assistant'),

('Gabriel', 'Silva', '1987-10-25', 12, '1234567891', 'cook B'),

('Luana', 'Costa', '1995-03-12', 3, '9876543211', 'chef assistant'),

('Pedro', 'Santos', '1990-08-30', 8, '5551234568', 'chef'),

('Mariana', 'Oliveira', '1988-05-20', 6, '1112223334', 'chef assistant'),

('Lucas', 'Rodrigues', '1992-01-15', 5, '4445556667', 'chef'),

('Juliana', 'Almeida', '1986-11-07', 9, '7778889991', 'chef assistant'),

('Bruno', 'Ferreira', '1994-07-18', 4, '2223334445', 'cook A'),

('Carolina', 'Nunes', '1991-09-22', 7, '9998887778', 'chef assistant'),

('Juan', 'Santos', '1985-06-12', 10, '1234567892', 'chef assistant'),

('Maria', 'Garcia', '1990-09-18', 7, '9876543212', 'cook B'),

('Jose', 'Reyes', '1988-04-30', 9, '5551234569', 'chef assistant'),

('Ana', 'Torres', '1993-01-25', 6, '1112223335', 'cook C'),

('Chinedu', 'Okoye', '1986-05-12', 9, '1234567893', 'chef assistant'),

('Ngozi', 'Eze', '1991-08-18', 6, '9876543213', 'chef'),

('Obinna', 'Onuoha', '1989-03-30', 8, '5551234570', 'chef assistant'),

('Chiamaka', 'Nwosu', '1994-02-25', 5, '1112223336', 'cook A'),

('Oluwaseun', 'Adeleke', '1992-07-15', 7, '4445556669', 'chef assistant'),

('Adeola', 'Olawale', '1988-01-07', 10, '7778889993', 'cook B'),

('Maria', 'Georgiou', '1991-04-18', 6, '9876543215', 'cook C'),

('Yannis', 'Andreadis', '1989-12-30', 8, '5551234572', 'chef assistant'),

('Petros','Kokkinos','2002-06-21', 4, '9998887782', 'cook A') ,

('Alexander','Koulakov','2002-05-23', 3, '9998387582', 'cook B') ;

INSERT INTO equipment (equipment\_name, instructions, image\_url)

VALUES

('Chef Knife', 'Use for chopping vegetables and meat. Keep it sharp and handle it with care.', 'https://cdn.shopify.com/s/files/1/0372/6232/7941/products/Wu\_estof\_Classic\_8-inch\_Chef\_Bobbi\_Lin\_0348x\_58ab572f-cc6f-40b4-9597-451149205c05.jpg?v=1713982031');

INSERT INTO equipment (equipment\_name, instructions)

VALUES

('Cutting Board', 'Place it on a flat surface to chop vegetables or meat. Wash it thoroughly after each use.'),

('Mixing Bowl', 'Use for mixing ingredients like flour, eggs, and sugar. Comes in various sizes.'),

('Saucepan', 'Use for cooking sauces, soups, and boiling water. Comes with a lid.'),

('Frying Pan', 'Use for frying and sautéing food. Ensure it is properly heated before adding ingredients.'),

('Oven', 'Use for baking and roasting. Preheat to the desired temperature before placing food inside.'),

('Blender', 'Use for blending ingredients to make smoothies, soups, and sauces. Make sure to secure the lid properly.'),

('Food Processor', 'Use for chopping, slicing, and shredding vegetables and fruits. Comes with various attachments.'),

('Measuring Cups and Spoons', 'Use for precise measurement of ingredients like flour, sugar, and spices.'),

('Mixing Spoon', 'Use for stirring and mixing ingredients in pots and bowls.'),

('Whisk', 'Use for whisking eggs, whipping cream, and mixing ingredients to incorporate air.'),

('Spatula', 'Use for flipping pancakes, burgers, and eggs. Also useful for scraping bowls and pans.'),

('Tongs', 'Use for flipping meats and vegetables on the grill or in the pan. Also handy for serving.'),

('Colander', 'Use for draining pasta and rinsing fruits and vegetables. Make sure it has sturdy handles.'),

('Grater', 'Use for grating cheese, vegetables, and spices. Be careful of your fingers when using.'),

('Peeler', 'Use for peeling fruits and vegetables like potatoes and carrots. Keep fingers away from blade.'),

('Rolling Pin', 'Use for rolling out dough for pies, cookies, and pastries. Keep it clean and dry after use.'),

('Mixer', 'Use for beating, creaming, and whipping ingredients. Ensure attachments are securely in place.'),

('Strainer', 'Use for straining liquids and removing solids from soups and sauces. Choose a fine mesh strainer for best results.'),

('Pastry Brush', 'Use for brushing egg wash, butter, or glaze onto pastries and meats. Clean thoroughly after each use.'),

('Scale', 'Use for measuring ingredients by weight for precise baking and cooking. Calibrate regularly for accuracy.'),

('Mortar and Pestle', 'Use for grinding spices, herbs, and seeds. Clean thoroughly after each use.'),

('Baking Sheet', 'Use for baking cookies, pastries, and roasting vegetables. Line with parchment paper for easy cleanup.'),

('Cake Pan', 'Use for baking cakes and brownies. Grease and flour the pan before adding batter.'),

('Pie Dish', 'Use for baking pies and quiches. Ensure the dough is evenly distributed and trim excess.'),

('Cooling Rack', 'Place baked goods on it to cool evenly and prevent them from getting soggy. Clean and dry after each use.'),

('Muffin Tin', 'Use for baking muffins, cupcakes, and mini quiches. Grease or line with paper liners before use.'),

('Sieve', 'Use for sifting flour, cocoa powder, and powdered sugar. Shake gently to sift ingredients evenly.'),

('Basting Brush', 'Use for applying marinades and sauces to meats and vegetables while cooking. Clean thoroughly after use.'),

('Juicer', 'Use for extracting juice from fruits and vegetables. Clean the juicer components immediately after use to prevent staining.'),

('Salad Spinner', 'Use for washing and drying salad greens and herbs. Spin the basket to remove excess water.'),

('Kitchen Timer', 'Use for timing cooking and baking processes. Set the timer to the desired duration and place it where you can easily see it.'),

('Pizza Stone', 'Use for baking homemade pizzas. Preheat the stone in the oven before placing the pizza on it.'),

('Immersion Blender', 'Use for blending soups, sauces, and smoothies directly in the pot or container. Ensure the blender is submerged.'),

('Kitchen Shears', 'Use for cutting herbs, trimming meat, and opening food packaging. Clean and dry thoroughly after each use.'),

('Microplane Zester', 'Use for grating citrus zest, garlic, ginger, and hard cheeses. Hold the zester at a slight angle for best results.'),

('Electric Kettle', 'Use for quickly boiling water for tea, coffee, and cooking. Fill with the desired amount of water and plug it in.'),

('Kitchen Scale', 'Use for accurately measuring ingredients by weight. Place the ingredients on the scale and read the weight.'),

('Pastry Blender', 'Use for cutting butter or shortening into flour for pie crusts and biscuits. Move the blender in a chopping motion.'),

('Meat Thermometer', 'Use for checking the internal temperature of meats to ensure they are cooked to the desired level of doneness.'),

('Ice Cream Scoop', 'Use for scooping ice cream, sorbet, and other frozen desserts. Dip the scoop in warm water to ease scooping.'),

('Kitchen Torch', 'Use for caramelizing sugar, toasting meringue, and browning dishes. Hold the torch about 6 inches away from the food.'),

('Wok', 'Use for stir-frying, deep-frying, and steaming. Preheat the wok over high heat before adding ingredients.');

INSERT INTO thematic\_section (name, description,image\_url)

VALUES

('Village Recipes', 'Traditional recipes from the countryside, passed down through generations.','https://irepo.primecp.com/2015/05/220374/Country-Cookin-Recipes\_ExtraLarge1000\_ID-994664.jpg?v=994664');

INSERT INTO thematic\_section (name, description)

VALUES

('Risotto Recipes', 'Delicious recipes for creamy and flavorful risottos, perfect for any occasion.'),

('Easter Desserts', 'Sweets and treats specially made for Easter celebrations.'),

('Mediterranean Dishes', 'Authentic dishes inspired by the flavors of the Mediterranean region.'),

('Comfort Food Classics', 'Classic comfort food recipes that warm the soul and satisfy the palate.'),

('Summer Grilling', 'Recipes perfect for outdoor grilling during the summer months.'),

('Vegetarian Delights', 'Delicious and nutritious vegetarian recipes for every meal.'),

('Seafood Extravaganza', 'Fresh and flavorful seafood recipes that are sure to impress.'),

('Quick and Easy Meals', 'Simple and speedy recipes for busy weeknights.'),

('International Cuisine', 'Explore the flavors of the world with these diverse recipes.'),

('Healthy Living', 'Nutritious recipes to support a healthy and balanced lifestyle.'),

('Slow Cooker Wonders', 'Hearty and flavorful recipes cooked low and slow in the crockpot.'),

('Family Favorites', 'Recipes that bring comfort and joy to the whole family.'),

('Holiday Feasts', 'Special recipes to celebrate festive occasions and gatherings.'),

('Plant-Based Power', 'Wholesome and delicious recipes focused on plant-based ingredients.'),

('Weekend Brunch', 'Elevate your brunch game with these mouthwatering recipes.'),

('Budget-Friendly Eats', 'Tasty meals that will not break the bank.'),

('One-PoWonders', 'Simplify mealtime with these easy and delicious one-pot recipes.'),

('Healthy Breakfasts', 'Start your day right with nutritious and energizing breakfast recipes.'),

('Gourmet Desserts', 'Indulgent and decadent desserts for special occasions or a sweet treat.'),

('Homemade Breads', 'Master the art of bread-making with these homemade bread recipes.'),

('Global Street Food', 'Explore the vibrant and diverse world of street food from around the globe.'),

('Seasonal Harvest', 'Celebrate the flavors of each season with recipes featuring fresh, seasonal produce.'),

('Culinary Adventures', 'Embark on culinary journeys with these adventurous and exotic recipes.'),

('Classic Cocktails', 'Mix up classic cocktails and elevate your home bartending skills.');

INSERT INTO foodgroups (foodgroups\_name, description,image\_url) VALUES

('Aromatic Herbs and Essential Oils', 'Includes herbs and oils used for flavoring and aroma purposes.','https://img.freepik.com/premium-photo/bottles-essential-oils-table-with-aromatic-herbs-black-stones\_100787-3014.jpg');

INSERT INTO foodgroups (foodgroups\_name, description) VALUES

('Coffee, Tea, and Their Products', 'Covers coffee, tea, and related products.'),

('Preserved Foods', 'Includes preserved or canned foods for longer shelf life.'),

('Sweeteners', 'Encompasses various sweetening agents like sugar, honey, etc.'),

('Fats and Oils', 'Includes fats and oils used in cooking and food preparation.'),

('Milk, Eggs, and Their Products', 'Covers dairy products like milk, eggs, and their derivatives.'),

('Meat and Meat Products', 'Encompasses various types of meat and processed meat products.'),

('Fish and Fish Products', 'Includes different kinds of fish and fish-based products.'),

('Cereals and Their Products', 'Covers grains and cereal-based products like bread, pasta, etc.'),

('Various Plant-based Foods', 'Encompasses a variety of plant-based foods like fruits, vegetables, nuts, etc.'),

('Products with Sweeteners', 'Includes products that contain added sweetening agents.'),

('Various Beverages', 'Covers a range of different beverages including soft drinks, juices, etc.'),

('other foodgroup', 'description');

INSERT INTO ingredient (foodgroups\_id, ingredient\_name)

VALUES

(1, 'Basil'), -- Aromatic Herbs and Essential Oils

(2, 'Green Tea'), -- Coffee, Tea, and Their Products

(3, 'Pickles'), -- Preserved Foods

(4, 'Honey'), -- Sweeteners

(5, 'Olive Oil'), -- Fats and Oils

(6, 'Milk'), -- Milk, Eggs, and Their Products

(7, 'Chicken Breast'), -- Meat and Meat Products

(8, 'Salmon'), -- Fish and Fish Products

(9, 'Wheat Bread'), -- Cereals and Their Products

(10, 'Apple'), -- Various Plant-based Foods

(11, 'Chocolate Bar'), -- Products with Sweeteners

(12, 'Orange Juice'), -- Various Beverages

(1, 'Rosemary'), -- Aromatic Herbs and Essential Oils

(2, 'Coffee Beans'), -- Coffee, Tea, and Their Products

(3, 'Olives'), -- Preserved Foods

(4, 'Maple Syrup'), -- Sweeteners

(5, 'Butter'), -- Fats and Oils

(6, 'Cheese'), -- Milk, Eggs, and Their Products

(7, 'Bacon'), -- Meat and Meat Products

(8, 'Tuna'), -- Fish and Fish Products

(9, 'Rice'), -- Cereals and Their Products

(10, 'Carrot'), -- Various Plant-based Foods

(11, 'Candy'), -- Products with Sweeteners

(12, 'Cola'), -- Various Beverages

(1, 'Thyme'), -- Aromatic Herbs and Essential Oils

(2, 'Black Tea'), -- Coffee, Tea, and Their Products

(3, 'Canned Tomatoes'), -- Preserved Foods

(4, 'Agave Syrup'), -- Sweeteners

(5, 'Coconut Oil'), -- Fats and Oils

(6, 'Yogurt'), -- Milk, Eggs, and Their Products

(7, 'Ham'), -- Meat and Meat Products

(8, 'Cod'), -- Fish and Fish Products

(9, 'Pasta'), -- Cereals and Their Products

(10, 'Banana'), -- Various Plant-based Foods

(11, 'Cookies'), -- Products with Sweeteners

(12, 'Lemonade'), -- Various Beverages

(1, 'Oregano'), -- Aromatic Herbs and Essential Oils

(2, 'Herbal Tea'), -- Coffee, Tea, and Their Products

(3, 'Jam'), -- Preserved Foods

(4, 'Molasses'), -- Sweeteners

(5, 'Ghee'), -- Fats and Oils

(6, 'Eggs'), -- Milk, Eggs, and Their Products

(7, 'Sausage'), -- Meat and Meat Products

(8, 'Shrimp'), -- Fish and Fish Products

(9, 'Oatmeal'), -- Cereals and Their Products

(10, 'Broccoli'), -- Various Plant-based Foods

(11, 'Cake'), -- Products with Sweeteners

(12, 'Iced Tea'),

(7, 'Beef'), -- Meat and Meat Products

(10, 'Lentil'), -- Various Plant-based Foods

(10, 'Curry'), -- Various Plant-based Foods

(10, 'Blueberry'), -- Various Plant-based Foods

(9, 'Flour'), -- Cereals and Their Products

(10, 'Mango'), -- Various Plant-based Foods

(10, 'Pepper'), -- Various Plant-based Foods

(10, 'Chickpea'), -- Various Plant-based Foods

(8, 'Clam'), -- Fish and Fish Products

(10, 'Eggplant'), -- Various Plant-based Foods

(10, 'Lettuce'), -- Various Plant-based Foods

(7, 'Pork'), -- Meat and Meat Products

(10, 'Spinach'), -- Various Plant-based Foods

(7, 'Lamb'), -- Meat and Meat Products

(10, 'Pumpkin');

INSERT INTO recipe (

recipe\_name,

recipe\_category,

natcuis\_id,

prim\_ingredient\_id,

recipe\_description,

quantity\_of\_servings,

difficulty\_level,

prep\_time,

cooking\_time,

tip\_1,

tip\_2,

tip\_3,

fat\_per\_portion,

protein\_per\_portion,

carbohydrate\_per\_portion

) VALUES

('Grilled Salmon', 'main course', 1, 8, 'Delicious grilled salmon seasoned with herbs', 4, 3, 20, 15, 'Make sure to preheat the grill before cooking.', 'Marinate the salmon for at least 30 minutes before grilling.', NULL, 10, 25, 5),

('Chocolate Cake', 'dessert', 2, 11, 'Decadent chocolate cake topped with chocolate ganache', 8, 4, 30, 40, 'Use high-quality cocoa powder for a richer flavor.', 'Allow the cake to cool completely before adding the ganache.', 'Garnish with fresh berries before serving.', 20, 6, 50),

('Chicken Alfredo Pasta', 'main course', 3, 7, 'Creamy chicken Alfredo pasta with Parmesan cheese', 6, 3, 25, 20, 'Use fresh Parmesan cheese for best flavor.', 'Cook the pasta al dente for optimal texture.', 'Garnish with chopped parsley before serving.', 15, 30, 45),

('Apple Pie', 'dessert', 4, 10, 'Classic homemade apple pie with cinnamon-spiced apples', 8, 4, 40, 50, 'Make sure to use a mix of sweet and tart apples for the filling.', 'Brush the top crust with egg wash for a golden finish.', 'Serve warm with vanilla ice cream.', 15, 2, 55),

('Vegetable Stir-Fry', 'main course', 5, 22, 'Healthy vegetable stir-fry with tofu in a savory sauce', 4, 2, 15, 15, 'Cut vegetables uniformly for even cooking.', 'Add tofu towards the end to prevent overcooking.', 'Serve hot over steamed rice.', 5, 10, 30),

('Vanilla Cupcakes', 'dessert', 6, 6, 'Soft and fluffy vanilla cupcakes with buttercream frosting', 12, 3, 30, 20, 'Make sure all ingredients are at room temperature for best results.', 'Use a piping bag to frost the cupcakes for a professional look.', 'Decorate with sprinkles or edible flowers.', 12, 2, 40),

('Spaghetti Bolognese', 'main course', 7, 33, 'Classic Italian spaghetti with rich Bolognese sauce', 6, 3, 20, 30, 'Cook pasta until al dente for the perfect texture.', 'Simmer the sauce on low heat for at least 1 hour to develop flavors.', 'Serve with freshly grated Parmesan cheese.', 18, 20, 55),

('New York Cheesecake', 'dessert', 8, 18, 'Creamy and decadent New York-style cheesecake with graham cracker crust', 10, 5, 45, 50, 'Ensure all ingredients are at room temperature for a smooth batter.', 'Bake the cheesecake in a water bath to prevent cracking.', 'Chill the cheesecake for at least 4 hours before serving.', 25, 7, 50),

('Beef Tacos', 'main course', 9, 7, 'Tender beef tacos with homemade salsa and guacamole', 4, 2, 25, 20, 'Season the beef with Mexican spices for authentic flavor.', 'Warm the tortillas before assembling the tacos for better texture.', 'Top with fresh cilantro and a squeeze of lime juice.', 20, 30, 40),

('Tiramisu', 'dessert', 10, 14, 'Classic Italian dessert made with layers of coffee-soaked ladyfingers and mascarpone cream', 8, 4, 30, 0, 'Use espresso or strong coffee for dipping the ladyfingers.', 'Dust the top with cocoa powder just before serving.', 'Chill the tiramisu for at least 4 hours to set.', 15, 8, 45),

('Chicken Curry', 'main course', 11, 7, 'Flavorful chicken curry with aromatic spices and coconut milk', 6, 3, 25, 30, 'Toast whole spices before grinding for enhanced flavor.', 'Simmer the curry on low heat to allow flavors to develop.', 'Garnish with fresh cilantro and a squeeze of lime juice.', 18, 25, 35),

('Lemon Bars', 'dessert', 12, 10, 'Tangy and sweet lemon bars with buttery shortbread crust', 12, 2, 20, 35, 'Zest the lemons before juicing for maximum flavor.', 'Chill the bars before cutting into squares for cleaner edges.', 'Dust with powdered sugar just before serving.', 10, 2, 50),

('Mushroom Risotto', 'main course', 13, 22, 'Creamy mushroom risotto cooked with Arborio rice and Parmesan cheese', 4, 4, 30, 25, 'Use a combination of mushrooms for depth of flavor.', 'Add hot broth gradually and stir frequently for creamy texture.', 'Finish with a drizzle of truffle oil for a luxurious touch.', 15, 10, 60),

('Chocolate Chip Cookies', 'dessert', 14, 11, 'Classic chocolate chip cookies with chewy centers and crispy edges', 24, 2, 15, 12, 'Chill the cookie dough for at least 30 minutes before baking for thicker cookies.', 'Use high-quality chocolate chips for the best flavor.', 'Bake until the edges are golden brown for the perfect texture.', 10, 3, 25),

('Grilled Chicken Caesar Salad', 'main course', 15, 7, 'Grilled chicken served on a bed of crisp romaine lettuce with Caesar dressing', 4, 2, 20, 15, 'Marinate the chicken in Caesar dressing for extra flavor.', 'Grill the chicken until charred and cooked through.', 'Toss the lettuce with dressing just before serving to prevent wilting.', 12, 25, 5),

('Panna Cotta', 'dessert', 16, 6, 'Creamy Italian dessert made with sweetened cream and gelatin', 6, 3, 20, 10, 'Use vanilla bean for a more intense flavor or vanilla extract as a substitute.', 'Bloom gelatin in cold water before adding to the warm cream mixture.', 'Serve with fresh berries or a fruit compote on top.', 18, 4, 20),

('Tomato Basil Soup', 'main course', 17, 1, 'A rich and creamy tomato basil soup perfect for a light meal', 4, 2, 15, 30, 'Use ripe tomatoes for a better flavor.', 'Garnish with fresh basil leaves and a drizzle of olive oil.', NULL, 6, 2, 25),

('Green Tea Cheesecake', 'dessert', 18, 2, 'A smooth and creamy cheesecake infused with green tea flavor', 8, 3, 30, 50, 'Use matcha powder for a vibrant green color and distinct taste.', 'Chill for at least 4 hours before serving.', 'Serve with whipped cream and a dusting of matcha powder.', 20, 6, 45),

('Olive Tapenade', 'main course', 19, 15, 'A savory olive tapenade perfect for spreading on bread or crackers', 6, 2, 10, 0, 'Use a mix of green and black olives for depth of flavor.', 'Serve with fresh baguette slices.', NULL, 12, 1, 5),

('Maple Syrup Pancakes', 'dessert', 20, 16, 'Fluffy pancakes drizzled with rich maple syrup', 4, 1, 10, 10, 'Make sure not to overmix the batter.', 'Serve hot with butter and additional syrup.', NULL, 8, 4, 35),

('Garlic Butter Shrimp', 'main course', 21, 44, 'Juicy shrimp cooked in a garlic butter sauce', 4, 2, 10, 5, 'Use fresh garlic for the best flavor.', 'Serve over rice or pasta.', NULL, 20, 10, 2),

('Cheese Omelette', 'main course', 22, 18, 'A classic cheese omelette perfect for breakfast', 2, 1, 5, 5, 'Use a non-stick pan for easy flipping.', 'Add a splash of milk for a fluffier texture.', NULL, 10, 6, 1),

('Bacon Wrapped Asparagus', 'main course', 23, 19, 'Crispy bacon wrapped around tender asparagus spears', 4, 2, 10, 15, 'Pre-cook the bacon slightly to reduce cooking time.', 'Serve immediately for the best texture.', NULL, 25, 4, 6),

('Tuna Salad', 'main course', 24, 20, 'A light and healthy tuna salad with a lemon dressing', 4, 1, 10, 0, 'Use canned tuna in water for a lighter option.', 'Add diced avocado for extra creaminess.', NULL, 5, 20, 2),

('Rice Pudding', 'dessert', 25, 21, 'A creamy and comforting rice pudding with a hint of cinnamon', 4, 2, 20, 30, 'Use Arborio rice for a creamier texture.', 'Garnish with a sprinkle of ground cinnamon.', NULL, 5, 4, 45),

('Carrot Soup', 'main course', 26, 22, 'A smooth and flavorful carrot soup with a hint of ginger', 4, 1, 10, 20, 'Use fresh ginger for a more vibrant flavor.', 'Serve with a dollop of sour cream.', NULL, 3, 2, 15),

('Candy Apples', 'dessert', 27, 23, 'Crisp apples coated in a shiny candy shell', 8, 3, 15, 5, 'Use a candy thermometer to ensure the correct temperature.', 'Allow to cool completely before serving.', NULL, 0, 1, 75),

('Greek Salad', 'main course', 5, 5, 'A refreshing salad with tomatoes, cucumbers, olives, and feta cheese', 4, 1, 15, 0, NULL, NULL, NULL, 12, 6, 10),

('Beef Stroganoff', 'main course', 15, 49, 'Tender beef in a creamy mushroom sauce, served over egg noodles', 6, 3, 20, 30, NULL, NULL, NULL, 25, 30, 45),

('Lentil Soup', 'main course', 23, 50, 'Hearty lentil soup with vegetables and spices', 6, 2, 15, 30, NULL, NULL, NULL, 6, 10, 25),

('French Toast', 'main course', 12, 6, 'Golden-brown French toast served with maple syrup', 4, 1, 10, 10, NULL, NULL, NULL, 15, 6, 30),

('Caprese Salad', 'main course', 8, 1, 'Simple salad with tomatoes, mozzarella, and basil drizzled with balsamic glaze', 4, 1, 10, 0, NULL, NULL, NULL, 8, 5, 6),

('Spicy Tuna Roll', 'main course', 16, 20, 'Sushi roll filled with spicy tuna and cucumber', 8, 4, 25, 0, NULL, NULL, NULL, 5, 10, 30),

('Vegetable Curry', 'main course', 30, 51, 'A flavorful vegetable curry with coconut milk and spices', 4, 3, 20, 25, NULL, NULL, NULL, 8, 5, 35),

('Blueberry Muffins', 'dessert', 19, 52, 'Soft and moist blueberry muffins with a hint of lemon', 12, 2, 20, 25, NULL, NULL, NULL, 10, 4, 40),

('Quiche Lorraine', 'main course', 24, 42, 'Savory quiche with bacon, cheese, and onions in a flaky crust', 6, 3, 30, 45, NULL, NULL, NULL, 20, 15, 25),

('Pad Thai', 'main course', 28, 21, 'Stir-fried rice noodles with shrimp, tofu, peanuts, and bean sprouts', 4, 3, 20, 15, NULL, NULL, NULL, 15, 20, 45),

('Pancakes', 'main course', 11, 53, 'Fluffy pancakes served with butter and syrup', 4, 1, 10, 10, NULL, NULL, NULL, 8, 4, 35),

('Roast Chicken', 'main course', 22, 7, 'Juicy roast chicken with crispy skin and herbs', 6, 2, 15, 60, NULL, NULL, NULL, 25, 35, 2),

('Mango Smoothie', 'dessert', 27, 54, 'A refreshing mango smoothie with yogurt and honey', 2, 1, 5, 0, NULL, NULL, NULL, 1, 5, 25),

('Stuffed Peppers', 'main course', 29, 55, 'Bell peppers stuffed with a mixture of rice, ground beef, and spices', 4, 3, 20, 30, NULL, NULL, NULL, 10, 20, 40),

('Falafel Wrap', 'main course', 13, 56, 'Crispy falafel wrapped in a pita with hummus and vegetables', 4, 2, 15, 10, NULL, NULL, NULL, 8, 12, 30),

('Clam Chowder', 'main course', 31, 57, 'Creamy New England clam chowder with potatoes and bacon', 6, 3, 20, 30, NULL, NULL, NULL, 25, 15, 35),

('Eggplant Parmesan', 'main course', 28, 58, 'Baked eggplant slices layered with marinara sauce and cheese', 6, 3, 30, 45, NULL, NULL, NULL, 18, 12, 40),

('Banana Bread', 'dessert', 14, 34, 'Moist banana bread with walnuts and a hint of cinnamon', 10, 2, 15, 60, NULL, NULL, NULL, 12, 5, 50),

('Caesar Salad', 'main course', 20, 59, 'Classic Caesar salad with romaine lettuce, croutons, and Parmesan', 4, 1, 10, 0, NULL, NULL, NULL, 10, 5, 10),

('Teriyaki Chicken', 'main course', 17, 7, 'Grilled chicken glazed with a sweet and savory teriyaki sauce', 4, 2, 15, 20, NULL, NULL, NULL, 10, 25, 15),

('Shrimp Scampi', 'main course', 2, 44, 'Shrimp cooked in a garlic butter sauce, served over pasta', 4, 3, 10, 10, NULL, NULL, NULL, 15, 20, 40),

('Chocolate Mousse', 'dessert', 7, 11, 'Light and airy chocolate mousse topped with whipped cream', 6, 2, 15, 0, NULL, NULL, NULL, 18, 4, 25),

('Minestrone Soup', 'main course', 21, 22, 'Hearty Italian vegetable soup with beans and pasta', 6, 2, 20, 30, NULL, NULL, NULL, 5, 10, 30),

('Pork Chops', 'main course', 25, 60, 'Juicy pork chops cooked with a savory apple glaze', 4, 3, 15, 20, NULL, NULL, NULL, 20, 25, 15),

('Greek Yogurt Parfait', 'dessert', 10, 30, 'Layers of Greek yogurt, honey, and fresh berries', 2, 1, 5, 0, NULL, NULL, NULL, 8, 10, 20),

('Spinach and Feta Pie', 'main course', 26, 61, 'Savory pie with spinach, feta cheese, and flaky pastry', 6, 3, 30, 45, NULL, NULL, NULL, 12, 8, 35),

('Lamb Kebabs', 'main course', 31, 62, 'Grilled lamb kebabs marinated with herbs and spices', 4, 3, 20, 15, NULL, NULL, NULL, 25, 30, 10),

('Pumpkin Soup', 'main course', 18, 63, 'Creamy pumpkin soup with a hint of nutmeg', 4, 1, 15, 20, NULL, NULL, NULL, 8, 4, 20),

('Cheese Pizza', 'main course', 30, 18, 'Classic cheese pizza with homemade tomato sauce', 8, 2, 20, 15, NULL, NULL, NULL, 12, 15, 40),

('Mango Sorbet', 'dessert', 5, 54, 'Refreshing mango sorbet made with fresh mangoes', 4, 2, 10, 0, NULL, NULL, NULL, 0, 1, 30);

---inserting the primary ingredients of all the recipes

INSERT INTO ingredient\_VS\_recipe (recipe\_id, ingredient\_id, quantity, unit\_of\_measurement, calories)

VALUES

(1, 8, 500, 'g', 250), -- Grilled Salmon

(2, 11, 200, 'g', 400), -- Chocolate Cake

(3, 7, 300, 'g', 400), -- Chicken Alfredo Pasta

(4, 10, 6, 'piece', 300), -- Apple Pie

(5, 22, 200, 'g', 150), -- Vegetable Stir-Fry

(6, 6, 2, 'cup', 300), -- Vanilla Cupcakes

(7, 33, 500, 'g', 600), -- Spaghetti Bolognese

(8, 18, 800, 'g', 800), -- New York Cheesecake

(9, 7, 400, 'g', 500), -- Beef Tacos

(10, 14, 250, 'g', 360), -- Tiramisu

(11, 7, 500, 'g', 550), -- Chicken Curry

(12, 10, 3, 'piece', 200), -- Lemon Bars

(13, 22, 300, 'g', 200), -- Mushroom Risotto

(14, 11, 150, 'g', 300), -- Chocolate Chip Cookies

(15, 7, 2, 'piece', 250), -- Grilled Chicken Caesar Salad

(16, 6, 500, 'ml', 400), -- Panna Cotta

(17, 1, 500, 'ml', 150), -- Tomato Basil Soup

(18, 2, 5, 'tsp', 20), -- Green Tea Cheesecake

(19, 15, 200, 'g', 300), -- Olive Tapenade

(20, 16, 3, 'tbsp', 150), -- Maple Syrup Pancakes

(21, 44, 250, 'g', 300), -- Garlic Butter Shrimp

(22, 18, 3, 'slice', 200), -- Cheese Omelette

(23, 19, 6, 'piece', 500), -- Bacon Wrapped Asparagus

(24, 20, 200, 'g', 150), -- Tuna Salad

(25, 21, 200, 'g', 250), -- Rice Pudding

(26, 22, 500, 'ml', 150), -- Carrot Soup

(27, 23, 8, 'piece', 500), -- Candy Apples

(28, 5, 2, 'tbsp', 180), -- Caprese Salad

(29, 49, 200, 'g', 350), -- Spicy Tuna Roll

(30, 50, 300, 'g', 250), -- Vegetable Curry

(31, 6, 2, 'cup', 120), -- Blueberry Muffins

(32, 1, 100, 'g', 50), -- Quiche Lorraine

(33, 20, 250, 'g', 300), -- Pad Thai

(34, 51, 100, 'g', 200), -- Cheese Pizza

(35, 52, 500, 'g', 180), -- Mango Sorbet

(36, 42, 4, 'piece', 400), -- Stuffed Peppers

(37, 21, 200, 'g', 220), -- Falafel Wrap

(38, 53, 4, 'cup', 400), -- Clam Chowder

(39, 7, 300, 'g', 450), -- Eggplant Parmesan

(40, 54, 4, 'slice', 320), -- Banana Bread

(41, 55, 3, 'cup', 150), -- Caesar Salad

(42, 56, 300, 'g', 400), -- Teriyaki Chicken

(43, 57, 200, 'g', 180), -- Shrimp Scampi

(44, 58, 4, 'cup', 500), -- Chocolate Mousse

(45, 34, 500, 'ml', 300), -- Minestrone Soup

(46, 59, 4, 'piece', 500), -- Pork Chops

(47, 7, 2, 'cup', 200), -- Greek Yogurt Parfait

(48, 44, 6, 'piece', 450), -- Spinach and Feta Pie

(49, 11, 100, 'g', 400), -- Lamb Kebabs

(50, 22, 300, 'g', 150), -- Pumpkin Soup

(51, 60, 4, 'piece', 500), -- Roast Pork

(52, 30, 3, 'cup', 250), -- Yogurt and Honey Parfait

(53, 61, 200, 'g', 150), -- Spinach Lasagna

(54, 62, 400, 'g', 600), -- Grilled Lamb Chops

(55, 63, 300, 'g', 200), -- Pumpkin Pie

(56, 18, 4, 'slice', 400), -- Cheese Platter

(57, 54, 3, 'cup', 300); -- Mango Smoothie

INSERT INTO ingredient\_VS\_recipe (recipe\_id, ingredient\_id, quantity, unit\_of\_measurement, calories)

VALUES

-- Grilled Salmon

(1, 5, 2, 'tbsp', 180), -- Olive Oil

(1, 13, 1, 'tsp', 2), -- Rosemary

-- Chocolate Cake

(2, 17, 100, 'g', 300), -- Butter

(2, 53, 250, 'g', 910), -- Flour

-- Chicken Alfredo Pasta

(3, 33, 250, 'g', 310), -- Pasta

(3, 42, 1, 'cup', 150), -- Eggs

-- Apple Pie

(4, 53, 200, 'g', 728), -- Flour

(4, 17, 100, 'g', 720), -- Butter

-- Vegetable Stir-Fry

(5, 50, 150, 'g', 140), -- Lentil

(5, 46, 100, 'g', 50), -- Broccoli

-- Vanilla Cupcakes

(6, 17, 150, 'g', 1080), -- Butter

(6, 53, 200, 'g', 728), -- Flour

-- Spaghetti Bolognese

(7, 49, 300, 'g', 750), -- Beef

(7, 27, 400, 'g', 80), -- Canned Tomatoes

-- New York Cheesecake

(8, 6, 2, 'cup', 300), -- Milk

(8, 17, 100, 'g', 720), -- Butter

-- Beef Tacos

(9, 49, 300, 'g', 750), -- Beef

(9, 59, 100, 'g', 10), -- Lettuce

-- Tiramisu

(10, 6, 200, 'ml', 120), -- Milk

(10, 11, 100, 'g', 200), -- Chocolate Bar

-- Chicken Curry

(11, 51, 200, 'g', 150), -- Curry

(11, 42, 1, 'cup', 150), -- Eggs

-- Lemon Bars

(12, 36, 100, 'ml', 30), -- Lemonade

(12, 17, 100, 'g', 720), -- Butter

-- Mushroom Risotto

(13, 21, 200, 'g', 220), -- Rice

(13, 46, 100, 'g', 50), -- Broccoli

-- Chocolate Chip Cookies

(14, 53, 200, 'g', 728), -- Flour

(14, 17, 100, 'g', 720), -- Butter

-- Grilled Chicken Caesar Salad

(15, 59, 200, 'g', 20), -- Lettuce

(15, 18, 100, 'g', 400), -- Cheese

-- Panna Cotta

(16, 4, 3, 'tbsp', 180), -- Honey

(16, 18, 200, 'g', 160), -- Cheese

-- Tomato Basil Soup

(17, 27, 300, 'g', 60), -- Canned Tomatoes

(17, 17, 100, 'g', 720), -- Butter

-- Green Tea Cheesecake

(18, 6, 200, 'ml', 120), -- Milk

(18, 18, 200, 'g', 160), -- Cheese

-- Olive Tapenade

(19, 5, 2, 'tbsp', 180), -- Olive Oil

(19, 13, 1, 'tsp', 2), -- Rosemary

-- Maple Syrup Pancakes

(20, 53, 200, 'g', 728), -- Flour

(20, 17, 100, 'g', 720), -- Butter

-- Garlic Butter Shrimp

(21, 17, 100, 'g', 720), -- Butter

(21, 5, 2, 'tbsp', 180), -- Olive Oil

-- Cheese Omelette

(22, 42, 2, 'piece', 150), -- Eggs

(22, 17, 50, 'g', 360), -- Butter

-- Bacon Wrapped Asparagus

(23, 46, 200, 'g', 100), -- Broccoli

(23, 5, 2, 'tbsp', 180), -- Olive Oil

-- Tuna Salad

(24, 59, 100, 'g', 10), -- Lettuce

(24, 5, 2, 'tbsp', 180), -- Olive Oil

-- Rice Pudding

(25, 6, 500, 'ml', 400), -- Milk

(25, 4, 3, 'tbsp', 180), -- Honey

-- Carrot Soup

(26, 27, 300, 'g', 60), -- Canned Tomatoes

(26, 17, 100, 'g', 720), -- Butter

-- Candy Apples

(27, 10, 6, 'piece', 300), -- Apple

(27, 4, 3, 'tbsp', 180), -- Honey

(28, 1, 100, 'g', 20), -- Basil

(28, 18, 150, 'g', 120), -- Cheese

(29, 20, 100, 'g', 75), -- Tuna

(29, 21, 200, 'g', 220), -- Rice

(30, 22, 200, 'g', 150), -- Carrot

(30, 27, 300, 'g', 60), -- Canned Tomatoes

(31, 52, 100, 'g', 60), -- Blueberry

(31, 53, 200, 'g', 728), -- Flour

(32, 18, 150, 'g', 120), -- Cheese

(32, 42, 2, 'piece', 150), -- Eggs

(33, 21, 200, 'g', 220), -- Rice

(33, 5, 2, 'tbsp', 180), -- Olive Oil

(34, 18, 200, 'g', 160), -- Cheese

(34, 53, 250, 'g', 910), -- Flour

(35, 54, 400, 'g', 240), -- Mango

(35, 4, 3, 'tbsp', 180), -- Honey

(36, 46, 200, 'g', 100), -- Broccoli

(36, 18, 150, 'g', 120), -- Cheese

(37, 56, 150, 'g', 180), -- Chickpea

(37, 5, 2, 'tbsp', 180), -- Olive Oil

(38, 57, 300, 'g', 150), -- Clam

(38, 6, 500, 'ml', 400), -- Milk

(39, 58, 200, 'g', 60), -- Eggplant

(39, 18, 150, 'g', 120), -- Cheese

(40, 34, 2, 'piece', 150), -- Banana

(40, 53, 200, 'g', 728), -- Flour

(41, 59, 200, 'g', 20), -- Lettuce

(41, 18, 100, 'g', 80), -- Cheese

(42, 7, 400, 'g', 500), -- Chicken Breast

(42, 5, 2, 'tbsp', 180), -- Olive Oil

(43, 44, 150, 'g', 180), -- Shrimp

(43, 17, 100, 'g', 720), -- Butter

(44, 11, 200, 'g', 400), -- Chocolate Bar

(44, 17, 100, 'g', 720), -- Butter

(45, 27, 300, 'g', 60), -- Canned Tomatoes

(45, 22, 200, 'g', 150), -- Carrot

(46, 60, 300, 'g', 450), -- Pork

(46, 5, 2, 'tbsp', 180), -- Olive Oil

(47, 30, 300, 'g', 150), -- Yogurt

(47, 34, 2, 'piece', 150), -- Banana

(48, 61, 300, 'g', 90), -- Spinach

(48, 18, 150, 'g', 120), -- Cheese

(49, 62, 300, 'g', 600), -- Lamb

(49, 17, 100, 'g', 720), -- Butter

(50, 63, 400, 'g', 120), -- Pumpkin

(50, 17, 100, 'g', 720), -- Butter

(51, 5, 2, 'tbsp', 180), -- Olive Oil

(51, 13, 1, 'tsp', 2), -- Rosemary

(52, 4, 3, 'tbsp', 180), -- Honey

(52, 34, 2, 'piece', 150), -- Banana

(53, 33, 250, 'g', 310), -- Pasta

(53, 18, 150, 'g', 120), -- Cheese

(54, 5, 2, 'tbsp', 180), -- Olive Oil

(54, 13, 1, 'tsp', 2), -- Rosemary

(55, 53, 200, 'g', 728), -- Flour

(55, 17, 100, 'g', 720), -- Butter

(56, 1, 100, 'g', 20), -- Basil

(56, 15, 50, 'g', 75), -- Olives

(57, 6, 500, 'ml', 400), -- Milk

(57, 4, 3, 'tbsp', 180); -- Honey

UPDATE recipe r

JOIN ingredient i ON r.prim\_ingredient\_id = i.ingredient\_id

JOIN foodgroups f ON i.foodgroups\_id = f.foodgroups\_id

SET r.classification = CASE

WHEN f.foodgroups\_name = 'Various Plant-based Foods' THEN 'Vegetarian'

WHEN f.foodgroups\_name = 'Fish and Fish Products' THEN 'Seafood'

WHEN f.foodgroups\_name = 'Meat and Meat Products' THEN 'Meat'

WHEN f.foodgroups\_name = 'Aromatic Herbs and Essential Oils' THEN 'Plant-based'

WHEN f.foodgroups\_name = 'Coffee, Tea, and Their Products' THEN 'Beverage'

WHEN f.foodgroups\_name = 'Preserved Foods' THEN 'Preserved'

WHEN f.foodgroups\_name = 'Sweeteners' THEN 'Sweetener'

WHEN f.foodgroups\_name = 'Fats and Oils' THEN 'Fat/Oil'

WHEN f.foodgroups\_name = 'Milk, Eggs, and Their Products' THEN 'Dairy/Egg'

WHEN f.foodgroups\_name = 'Cereals and Their Products' THEN 'Cereal'

WHEN f.foodgroups\_name = 'Products with Sweeteners' THEN 'Sweetened Product'

WHEN f.foodgroups\_name = 'Various Beverages' THEN 'Beverage'

ELSE 'Other'

END;

INSERT INTO episode (episode\_name, episode\_date, season, serial\_number)

VALUES

('Season Premiere', '2024-01-10', 1,1),

('The Great Bake-Off', '2024-01-17', 1,2),

('International Flavors', '2024-01-24', 1,3),

('Family Favorites', '2024-01-31', 1,4),

('Pressure Cooker Challenge', '2024-02-07', 1,5),

('Sweet Treats', '2024-02-14', 1, 6),

('Restaurant Wars', '2024-02-21', 1,7),

('Ultimate BBQ', '2024-02-28', 1, 8),

('Mystery Box Madness', '2024-03-06', 1,9),

('Grand Finale', '2024-03-13', 1,10);

INSERT INTO episode (episode\_name, episode\_date, season, serial\_number)

VALUES

('Season Premiere: New Challenges', '2025-01-10', 2,1),

('Around the World in 80 Dishes', '2025-01-17', 2, 2),

('Comfort Food Classics', '2025-01-24', 2,3),

('Healthy Eating Habits', '2025-01-31', 2,4),

('Dessert Extravaganza', '2025-02-07', 2,5),

('Gourmet Grilling', '2025-02-14', 2,6),

('Baking Bonanza', '2025-02-21', 2,7),

('Culinary Showdown', '2025-02-28', 2,8),

('Mystery Box Mayhem', '2025-03-06', 2,9),

('Season Finale: The Ultimate Cook-Off', '2025-03-13', 2,10);

INSERT INTO episode (episode\_name, episode\_date, season, serial\_number)

VALUES

('Season Premiere: New Beginnings', '2026-01-10', 3,1),

('Spicy Sensations', '2026-01-17', 3,2),

('Vegetarian Delights', '2026-01-24', 3,3),

('Seafood Spectacular', '2026-01-31', 3,4),

('Global Fusion', '2026-02-07', 3,5),

('Burger Battle', '2026-02-14', 3,6),

('Pasta Party', '2026-02-21', 3,7),

('Tropical Temptations', '2026-02-28', 3,8),

('Sizzling Steaks', '2026-03-06', 3,9),

('Season Finale: The Ultimate Showdown', '2026-03-13', 3,10);

/\*

---------------------------------------------------NEW DATA: ------------------------------------------------------------

INSERT INTO judge (cook\_id, participation\_number, episode\_id)

VALUES

-- Episode 1 judges

(12, 3, 1), -- Gino D'Acampo

(23, 1, 1), -- Curtis Stone

(41, 1, 1), -- Giada De Laurentiis

-- Episode 2 judges

(15, 1, 2), -- Emeril Lagasse

(21, 1, 2), -- Rick Bayless

(12, 3, 2), -- Gino D'Acampo

-- Episode 3 judges

(18, 1, 3), -- Bourdain Ramsay

(29, 2, 3), -- Ryan Chen

(20, 1, 3), -- José Andrés

-- Episode 4 judges

(21, 2, 4), -- Yotam Ottolenghi

(22, 1, 4), -- Laura Gomez

(23, 1, 4), -- Daniel Smith

-- Episode 5 judges

(24, 1, 5), -- Rachel Johnson

(21, 2, 5), -- Yotam Ottolenghi

(26, 1, 5), -- Maria Lee

-- Episode 6 judges

(41, 1, 6), -- Alexander Brown

(28, 1, 6), -- Emma Davis

(29, 2, 6), -- Ryan Chen

-- Episode 7 judges

(30, 1, 7), -- Sara Lee

(31, 1, 7), -- Peter Pan

(32, 2, 7), -- Emily Ross

-- Episode 8 judges

(32, 2, 8), -- Emily Ross

(21, 1, 8), -- Rachael Ray

(35, 1, 8), -- Thomas Keller

-- Episode 9 judges

(36, 1, 9), -- Alice Waters

(41, 2, 9), -- Amanda Scott

(38, 1, 9), -- David Cook

-- Episode 10 judges

(39, 1, 10), -- Michelle Nguyen

(12, 3, 10), -- Gino D'Acampo

(41, 2, 10); -- Amanda Scott

INSERT INTO judge (cook\_id, participation\_number, episode\_id)

VALUES

-- Episode 11 judges

(71, 1, 11),

(70, 1, 11),

(69, 1, 11),

-- Episode 12 judges

(70, 1, 12),

(14, 1, 12),

(71, 1, 12),

-- Episode 13 judges

(69, 1, 13),

(14, 2, 13),

(67, 1, 13),

-- Episode 14 judges

(70, 2, 14),

(28, 1, 14),

(68, 2, 14),

-- Episode 15 judges

(13, 1, 15),

(62, 2, 15),

(61, 1, 15),

-- Episode 16 judges

(12, 1, 16),

(59, 1, 16), -- Emma Davis

(58, 2, 16), -- Ryan Chen

-- Episode 17 judges

(61, 2, 17),

(58, 2, 17),

(55, 1, 17),

-- Episode 18 judges

(53, 1, 18),

(71, 1, 18),

(51, 1, 18),

-- Episode 19 judges

(27, 1, 19),

(41, 2, 19),

(70, 1, 19),

-- Episode 20 judges

(39, 1, 20),

(61, 1, 20),

(71, 1, 20);

INSERT INTO judge (cook\_id, participation\_number, episode\_id)

VALUES

-- Season 3

-- Episode 21 judges

(1, 1, 21),

(2, 1, 21),

(3, 1, 21),

-- Episode 22 judges

(4, 1, 22),

(5, 1, 22),

(6, 1, 22),

-- Episode 23 judges

(7, 1, 23),

(8, 2, 23),

(9, 1, 23),

-- Episode 24 judges

(10, 2, 24),

(21, 1, 24),

(9, 2, 24),

-- Episode 25 judges

(13, 1, 25),

(21, 2, 25),

(15, 1, 25),

-- Episode 26 judges

(16, 1, 26),

(7, 1, 26),

(18, 2, 26),

-- Episode 27 judges

(9, 2, 27),

(20, 2, 27),

(21, 1, 27),

-- Episode 28 judges

(1, 1, 28),

(3, 1, 28),

(5, 1, 28),

-- Episode 29 judges

(1, 1, 29),

(7, 2, 29),

(9, 1, 29),

-- Episode 30 judges

(1, 1, 30),

(5, 1, 30),

(21, 1, 30);

\*/

/\*

---------------------------------------------------------------------------------------------------------------

INSERT INTO evaluation(cook\_id, judge\_id, grade) VALUES

-- Evaluations for Episode 1

(12, 1, 4), -- Gino DAcampo

(13, 2, 3), -- Curtis Stone

(14, 3, 5), -- Giada De Laurentiis

-- Evaluations for Episode 2

(15, 4, 3), -- Emeril Lagasse

(16, 5, 4), -- Rick Bayless

(17, 6, 5), -- Guy Fieri

-- Evaluations for Episode 3

(18, 7, 5), -- Bourdain Ramsay

(19, 8, 4), -- Heston Blumenthal

(20, 9, 3), -- Jose Andres

-- Evaluations for Episode 4

(21, 10, 4), -- Yotam Ottolenghi

(22, 11, 5), -- Laura Gomez

(23, 12, 3), -- Daniel Smith

-- Evaluations for Episode 5

(24, 13, 5), -- Rachel Johnson

(25, 14, 4), -- Christopher Martinez

(26, 15, 3), -- Maria Lee

-- Evaluations for Episode 6

(27, 16, 3), -- Alexander Brown

(28, 17, 4), -- Emma Davis

(29, 18, 5), -- Ryan Chen

-- Evaluations for Episode 7

(30, 19,4), -- Sara Lee

(31, 20, 5), -- Peter Pan

(32, 21, 3), -- Emily Ross

-- Evaluations for Episode 8

(33, 22, 5), -- Paula Deen

(34, 23, 4), -- Rachael Ray

(35, 24, 3); -- Thomas Keller

INSERT INTO evaluation(cook\_id, judge\_id, grade) VALUES

(12, 2, 5),

(12, 3, 1),

(31, 2, 2),

(31, 3, 1); -- no sure if this is valid with all the constraints

\*/

/\*

-- Inserting data for Episode 1

INSERT INTO episode\_cook\_recipe (cook\_id, episode\_id, recipe\_id) VALUES

(12, 1, 1), -- Gino D'Acampo - Grilled Salmon

(13, 1, 2), -- Curtis Stone - Chocolate Cake

(14, 1, 3), -- Giada De Laurentiis - Chicken Alfredo Pasta

-- Inserting data for Episode 2

(15, 2, 4), -- Emeril Lagasse - Apple Pie

(16, 2, 5), -- Rick Bayless - Vegetable Stir-Fry

(17, 2, 6), -- Guy Fieri - Vanilla Cupcakes

-- Inserting data for Episode 3

(18, 3, 1), -- Bourdain Ramsay - Grilled Salmon

(19, 3, 2), -- Heston Blumenthal - Chocolate Cake

(20, 3, 3), -- Jos? Andr?s - Chicken Alfredo Pasta

-- Inserting data for Episode 4

(21, 4, 4), -- Yotam Ottolenghi - Apple Pie

(22, 4, 5), -- Laura Gomez - Vegetable Stir-Fry

(23, 4, 6), -- Daniel Smith - Vanilla Cupcakes

-- Inserting data for Episode 5

(24, 5, 1), -- Rachel Johnson - Grilled Salmon

(25, 5, 2), -- Christopher Martinez - Chocolate Cake

(26, 5, 3), -- Maria Lee - Chicken Alfredo Pasta

-- Inserting data for Episode 6

(27, 6, 4), -- Alexander Brown - Apple Pie

(28, 6, 5), -- Emma Davis - Vegetable Stir-Fry

(29, 6, 6), -- Ryan Chen - Vanilla Cupcakes

-- Inserting data for Episode 7

(30, 7, 1), -- Sara Lee - Grilled Salmon

(31, 7, 2), -- Peter Pan - Chocolate Cake

(32, 7, 3), -- Emily Ross - Chicken Alfredo Pasta

-- Inserting data for Episode 8

(33, 8, 4), -- Paula Deen - Apple Pie

(34, 8, 5), -- Rachael Ray - Vegetable Stir-Fry

(35, 8, 6); -- Thomas Keller - Vanilla Cupcakes

-- Inserting data for Episode 1 (Season 2)

INSERT INTO episode\_cook\_recipe (cook\_id, episode\_id, recipe\_id) VALUES

(42, 11, 1), -- Amit Patel - Grilled Salmon

(43, 11, 2), -- Priya Sharma - Chocolate Cake

(44, 11, 3); -- Raj Kumar - Chicken Alfredo Pasta

-- Inserting data for Episode 2 (Season 2)

INSERT INTO episode\_cook\_recipe (cook\_id, episode\_id, recipe\_id) VALUES

(45, 12, 4), -- Anjali Mehta - Apple Pie

(46, 12, 5), -- Rahul Singh - Vegetable Stir-Fry

(47, 12, 6); -- Sneha Gupta - Vanilla Cupcakes

-- Inserting data for Episode 3 (Season 2)

INSERT INTO episode\_cook\_recipe (cook\_id, episode\_id, recipe\_id) VALUES

(48, 13, 1), -- Vikram Shah - Grilled Salmon

(49, 13, 2), -- Neha Verma - Chocolate Cake

(50, 13, 3); -- Gabriel Silva - Chicken Alfredo Pasta

-- Inserting data for Episode 4 (Season 2)

INSERT INTO episode\_cook\_recipe (cook\_id, episode\_id, recipe\_id) VALUES

(51, 14, 4), -- Luana Costa - Apple Pie

(52, 14, 5), -- Pedro Santos - Vegetable Stir-Fry

(53, 14, 6); -- Mariana Oliveira - Vanilla Cupcakes

-- Inserting data for Episode 5 (Season 2)

INSERT INTO episode\_cook\_recipe (cook\_id, episode\_id, recipe\_id) VALUES

(54, 15, 1), -- Lucas Rodrigues - Grilled Salmon

(55, 15, 2), -- Juliana Almeida - Chocolate Cake

(56, 15, 3); -- Bruno Ferreira - Chicken Alfredo Pasta

-- Inserting data for Episode 6 (Season 2)

INSERT INTO episode\_cook\_recipe (cook\_id, episode\_id, recipe\_id) VALUES

(57, 16, 4), -- Carolina Nunes - Apple Pie

(58, 16, 5), -- Juan Santos - Vegetable Stir-Fry

(59, 16, 6); -- Maria Garcia - Vanilla Cupcakes

-- Inserting data for Episode 7 (Season 2)

INSERT INTO episode\_cook\_recipe (cook\_id, episode\_id, recipe\_id) VALUES

(60, 17, 1), -- Jose Reyes - Grilled Salmon

(61, 17, 2), -- Ana Torres - Chocolate Cake

(62, 17, 3); -- Chinedu Okoye - Chicken Alfredo Pasta

-- Inserting data for Episode 8 (Season 2)

INSERT INTO episode\_cook\_recipe (cook\_id, episode\_id, recipe\_id) VALUES

(63, 18, 4), -- Ngozi Eze - Apple Pie

(64, 18, 5), -- Obinna Onuoha - Vegetable Stir-Fry

(65, 18, 6); -- Chiamaka Nwosu - Vanilla Cupcakes

-- Inserting data for Episode 1 (Season 3)

INSERT INTO episode\_cook\_recipe (cook\_id, episode\_id, recipe\_id) VALUES

(66, 19, 1), -- Oluwaseun Adeleke - Grilled Salmon

(67, 19, 2), -- Adeola Olawale - Chocolate Cake

(68, 19, 3); -- Maria Georgiou - Chicken Alfredo Pasta

-- Inserting data for Episode 2 (Season 3)

INSERT INTO episode\_cook\_recipe (cook\_id, episode\_id, recipe\_id) VALUES

(69, 20, 4), -- Yannis Andreadis - Apple Pie

(70, 20, 5), -- Petros Kokkinos - Vegetable Stir-Fry

(71, 20, 6); -- Alexander Koulakov - Vanilla Cupcakes

INSERT INTO episode\_cook\_recipe (cook\_id, episode\_id, recipe\_id) VALUES

(31, 21, 1), -- Yannis Andreadis - Grilled Salmon

(12, 21, 2), -- Petros Kokkinos - Chocolate Cake

(45, 21, 3);

\*/

INSERT INTO cook\_nat\_cuis (cook\_id, natcuis\_id) VALUES

(1, 1),

(1, 4),

(2, 2),

(3, 3),

(4, 4),

(5, 5),

(6, 6),

(7, 7),

(8, 8),

(9, 9),

(10, 10),

(10, 30),

(11, 11),

(12, 12),

(13, 13),

(14, 14),

(15, 15),

(15 ,17),

(16, 16),

(17, 17),

(18, 18),

(19, 19),

(20, 20),

(21, 21),

(22, 22),

(23, 23),

(24, 24),

(25, 25),

(26, 26),

(27, 27),

(28, 28),

(29, 29),

(30, 30),

(31, 31),

(32, 1),

(33, 2),

(34, 3),

(35, 4),

(36, 5),

(37, 6),

(38, 7),

(39, 8),

(40, 9),

(41,10),

(42,11),

(43,12),

(44,13),

(45,14),

(46,15),

(47,16),

(48,17),

(49,18),

(50,19),

(51,20),

(52,21),

(53,22),

(54,23),

(55,24),

(56,25),

(57,26),

(58,27),

(59,28),

(59,15),

(60,29),

(61,30),

(62,31),

(63,1),

(63,2),

(64,5),

(65,12),

(66,17),

(67,9),

(68,23),

(69,24),

(70,28),

(71,31),

(72,1),

(72,16),

(72,18);

-- Example insertions for recipe categories

INSERT INTO tags (tag\_name) VALUES

('brunch'),

('quick-lunch'),

('cold dish'),

('appetizer'),

('soup'),

('salad'),

('main course'),

('side dish'),

('dessert'),

('snack'),

('beverage'),

('breakfast'),

('lunch'),

('dinner'),

('snack'),

('vegetarian'),

('vegan'),

('gluten-free'),

('dairy-free'),

('low-carb'),

('high-protein'),

('paleo'),

('easy'),

('quick'),

('healthy'),

('low-calorie'),

('family-friendly'),

('one-pot'),

('budget-friendly'),

('comfort food'),

('grilled'),

('baked'),

('fried'),

('roasted'),

('steamed'),

('spicy'),

('sweet'),

('savory'),

('glazed'),

('creamy'),

('light'),

('hearty');

-- Grilled Salmon

INSERT INTO recipe\_tag (recipe\_id, tag\_id)

SELECT r.recipe\_id, t.tag\_id FROM recipe r

JOIN tags t ON t.tag\_name IN ('main course', 'grilled')

WHERE r.recipe\_name = 'Grilled Salmon';

-- Chocolate Cake (without the "chocolate" tag)

INSERT INTO recipe\_tag (recipe\_id, tag\_id)

SELECT r.recipe\_id, t.tag\_id FROM recipe r

JOIN tags t ON t.tag\_name IN ('dessert', 'baked', 'sweet')

WHERE r.recipe\_name = 'Chocolate Cake';

-- Chicken Alfredo Pasta

INSERT INTO recipe\_tag (recipe\_id, tag\_id)

SELECT r.recipe\_id, t.tag\_id FROM recipe r

JOIN tags t ON t.tag\_name IN ('main course', 'pasta', 'creamy')

WHERE r.recipe\_name = 'Chicken Alfredo Pasta';

-- Apple Pie

INSERT INTO recipe\_tag (recipe\_id, tag\_id)

SELECT r.recipe\_id, t.tag\_id FROM recipe r

JOIN tags t ON t.tag\_name IN ('dessert', 'baked', 'sweet')

WHERE r.recipe\_name = 'Apple Pie';

-- Vegetable Stir-Fry

INSERT INTO recipe\_tag (recipe\_id, tag\_id)

SELECT r.recipe\_id, t.tag\_id FROM recipe r

JOIN tags t ON t.tag\_name IN ('main course', 'vegetarian', 'quick', 'easy')

WHERE r.recipe\_name = 'Vegetable Stir-Fry';

-- Vanilla Cupcakes

INSERT INTO recipe\_tag (recipe\_id, tag\_id)

SELECT r.recipe\_id, t.tag\_id FROM recipe r

JOIN tags t ON t.tag\_name IN ('dessert', 'baked', 'sweet')

WHERE r.recipe\_name = 'Vanilla Cupcakes';

-- Spaghetti Bolognese

INSERT INTO recipe\_tag (recipe\_id, tag\_id)

SELECT r.recipe\_id, t.tag\_id FROM recipe r

JOIN tags t ON t.tag\_name IN ('main course', 'pasta', 'comfort food')

WHERE r.recipe\_name = 'Spaghetti Bolognese';

-- New York Cheesecake

INSERT INTO recipe\_tag (recipe\_id, tag\_id)

SELECT r.recipe\_id, t.tag\_id FROM recipe r

JOIN tags t ON t.tag\_name IN ('dessert', 'baked', 'sweet')

WHERE r.recipe\_name = 'New York Cheesecake';

-- Beef Tacos

INSERT INTO recipe\_tag (recipe\_id, tag\_id)

SELECT r.recipe\_id, t.tag\_id FROM recipe r

JOIN tags t ON t.tag\_name IN ('main course', 'quick', 'easy', 'spicy')

WHERE r.recipe\_name = 'Beef Tacos';

-- Tiramisu

INSERT INTO recipe\_tag (recipe\_id, tag\_id)

SELECT r.recipe\_id, t.tag\_id FROM recipe r

JOIN tags t ON t.tag\_name IN ('dessert', 'easy', 'sweet')

WHERE r.recipe\_name = 'Tiramisu';

-- Grilled Salmon

INSERT INTO recipe\_equipment (recipe\_id, equipment\_id)

SELECT r.recipe\_id, e.equipment\_id FROM recipe r

JOIN equipment e ON e.equipment\_name IN ('Chef Knife', 'Kitchen Timer', 'Baking Sheet')

WHERE r.recipe\_name = 'Grilled Salmon';

-- Chocolate Cake

INSERT INTO recipe\_equipment (recipe\_id, equipment\_id)

SELECT r.recipe\_id, e.equipment\_id FROM recipe r

JOIN equipment e ON e.equipment\_name IN ('Mixer', 'Sieve', 'Rolling Pin', 'Baking Sheet')

WHERE r.recipe\_name = 'Chocolate Cake';

-- Chicken Alfredo Pasta

INSERT INTO recipe\_equipment (recipe\_id, equipment\_id)

SELECT r.recipe\_id, e.equipment\_id FROM recipe r

JOIN equipment e ON e.equipment\_name IN ('Chef Knife', 'Cutting Board', 'Saucepan', 'Mixing Spoon')

WHERE r.recipe\_name = 'Chicken Alfredo Pasta';

-- Apple Pie

INSERT INTO recipe\_equipment (recipe\_id, equipment\_id)

SELECT r.recipe\_id, e.equipment\_id FROM recipe r

JOIN equipment e ON e.equipment\_name IN ('Chef Knife', 'Cutting Board', 'Pie Dish', 'Rolling Pin')

WHERE r.recipe\_name = 'Apple Pie';

-- Vegetable Stir-Fry

INSERT INTO recipe\_equipment (recipe\_id, equipment\_id)

SELECT r.recipe\_id, e.equipment\_id FROM recipe r

JOIN equipment e ON e.equipment\_name IN ('Wok', 'Chef Knife', 'Cutting Board', 'Mixing Spoon')

WHERE r.recipe\_name = 'Vegetable Stir-Fry';

-- Vanilla Cupcakes

INSERT INTO recipe\_equipment (recipe\_id, equipment\_id)

SELECT r.recipe\_id, e.equipment\_id FROM recipe r

JOIN equipment e ON e.equipment\_name IN ('Mixer', 'Mixing Bowl', 'Muffin Tin', 'Pastry Brush')

WHERE r.recipe\_name = 'Vanilla Cupcakes';

-- Spaghetti Bolognese

INSERT INTO recipe\_equipment (recipe\_id, equipment\_id)

SELECT r.recipe\_id, e.equipment\_id FROM recipe r

JOIN equipment e ON e.equipment\_name IN ('Chef Knife', 'Cutting Board', 'Saucepan', 'Mixing Spoon')

WHERE r.recipe\_name = 'Spaghetti Bolognese';

-- New York Cheesecake

INSERT INTO recipe\_equipment (recipe\_id, equipment\_id)

SELECT r.recipe\_id, e.equipment\_id FROM recipe r

JOIN equipment e ON e.equipment\_name IN ('Mixer', 'Mixing Bowl', 'Springform Pan', 'Spatula')

WHERE r.recipe\_name = 'New York Cheesecake';

-- Beef Tacos

INSERT INTO recipe\_equipment (recipe\_id, equipment\_id)

SELECT r.recipe\_id, e.equipment\_id FROM recipe r

JOIN equipment e ON e.equipment\_name IN ('Skillet', 'Chef Knife', 'Cutting Board', 'Mixing Spoon')

WHERE r.recipe\_name = 'Beef Tacos';

-- Tiramisu

INSERT INTO recipe\_equipment (recipe\_id, equipment\_id)

SELECT r.recipe\_id, e.equipment\_id FROM recipe r

JOIN equipment e ON e.equipment\_name IN ('Mixing Bowl', 'Whisk', 'Square Baking Dish', 'Spatula')

WHERE r.recipe\_name = 'Tiramisu';

-- Chicken Curry

INSERT INTO recipe\_equipment (recipe\_id, equipment\_id)

SELECT r.recipe\_id, e.equipment\_id FROM recipe r

JOIN equipment e ON e.equipment\_name IN ('Chef Knife', 'Cutting Board', 'Saucepan', 'Mixing Spoon')

WHERE r.recipe\_name = 'Chicken Curry';

-- Lemon Bars

INSERT INTO recipe\_equipment (recipe\_id, equipment\_id)

SELECT r.recipe\_id, e.equipment\_id FROM recipe r

JOIN equipment e ON e.equipment\_name IN ('Mixing Bowl', 'Baking Dish', 'Spatula', 'Chef Knife')

WHERE r.recipe\_name = 'Lemon Bars';

-- Mushroom Risotto

INSERT INTO recipe\_equipment (recipe\_id, equipment\_id)

SELECT r.recipe\_id, e.equipment\_id FROM recipe r

JOIN equipment e ON e.equipment\_name IN ('Saucepan', 'Mixing Spoon', 'Chef Knife', 'Cutting Board')

WHERE r.recipe\_name = 'Mushroom Risotto';

-- Chocolate Chip Cookies

INSERT INTO recipe\_equipment (recipe\_id, equipment\_id)

SELECT r.recipe\_id, e.equipment\_id FROM recipe r

JOIN equipment e ON e.equipment\_name IN ('Mixer', 'Mixing Bowl', 'Baking Sheet', 'Cookie Scoop')

WHERE r.recipe\_name = 'Chocolate Chip Cookies';

-- Grilled Chicken Caesar Salad

INSERT INTO recipe\_equipment (recipe\_id, equipment\_id)

SELECT r.recipe\_id, e.equipment\_id FROM recipe r

JOIN equipment e ON e.equipment\_name IN ('Grill', 'Chef Knife', 'Cutting Board', 'Mixing Bowl')

WHERE r.recipe\_name = 'Grilled Chicken Caesar Salad';

-- Panna Cotta

INSERT INTO recipe\_equipment (recipe\_id, equipment\_id)

SELECT r.recipe\_id, e.equipment\_id FROM recipe r

JOIN equipment e ON e.equipment\_name IN ('Saucepan', 'Mixing Bowl', 'Ramekins', 'Whisk')

WHERE r.recipe\_name = 'Panna Cotta';

-- Grilled Salmon

INSERT INTO recipe\_thematic\_section (recipe\_id, them\_sec\_id)

SELECT r.recipe\_id, ts.them\_sec\_id

FROM recipe r

JOIN thematic\_section ts ON ts.name IN ('Seafood Extravaganza', 'Summer Grilling')

WHERE r.recipe\_name = 'Grilled Salmon';

-- Chocolate Cake

INSERT INTO recipe\_thematic\_section (recipe\_id, them\_sec\_id)

SELECT r.recipe\_id, ts.them\_sec\_id

FROM recipe r

JOIN thematic\_section ts ON ts.name IN ('Gourmet Desserts', 'Comfort Food Classics')

WHERE r.recipe\_name = 'Chocolate Cake';

-- Chicken Alfredo Pasta

INSERT INTO recipe\_thematic\_section (recipe\_id, them\_sec\_id)

SELECT r.recipe\_id, ts.them\_sec\_id

FROM recipe r

JOIN thematic\_section ts ON ts.name IN ('Family Favorites', 'Quick and Easy Meals', 'Pasta')

WHERE r.recipe\_name = 'Chicken Alfredo Pasta';

-- Apple Pie

INSERT INTO recipe\_thematic\_section (recipe\_id, them\_sec\_id)

SELECT r.recipe\_id, ts.them\_sec\_id

FROM recipe r

JOIN thematic\_section ts ON ts.name IN ('Seasonal Harvest', 'Holiday Feasts', 'Comfort Food Classics')

WHERE r.recipe\_name = 'Apple Pie';

-- Vegetable Stir-Fry

INSERT INTO recipe\_thematic\_section (recipe\_id, them\_sec\_id)

SELECT r.recipe\_id, ts.them\_sec\_id

FROM recipe r

JOIN thematic\_section ts ON ts.name IN ('Vegetarian Delights', 'Quick and Easy Meals')

WHERE r.recipe\_name = 'Vegetable Stir-Fry';

-- Vanilla Cupcakes

INSERT INTO recipe\_thematic\_section (recipe\_id, them\_sec\_id)

SELECT r.recipe\_id, ts.them\_sec\_id

FROM recipe r

JOIN thematic\_section ts ON ts.name IN ('Gourmet Desserts', 'Comfort Food Classics')

WHERE r.recipe\_name = 'Vanilla Cupcakes';

-- Spaghetti Bolognese

INSERT INTO recipe\_thematic\_section (recipe\_id, them\_sec\_id)

SELECT r.recipe\_id, ts.them\_sec\_id

FROM recipe r

JOIN thematic\_section ts ON ts.name IN ('Family Favorites', 'Comfort Food Classics', 'Italian Cuisine')

WHERE r.recipe\_name = 'Spaghetti Bolognese';

-- New York Cheesecake

INSERT INTO recipe\_thematic\_section (recipe\_id, them\_sec\_id)

SELECT r.recipe\_id, ts.them\_sec\_id

FROM recipe r

JOIN thematic\_section ts ON ts.name IN ('Gourmet Desserts', 'Comfort Food Classics')

WHERE r.recipe\_name = 'New York Cheesecake';

-- Beef Tacos

INSERT INTO recipe\_thematic\_section (recipe\_id, them\_sec\_id)

SELECT r.recipe\_id, ts.them\_sec\_id

FROM recipe r

JOIN thematic\_section ts ON ts.name IN ('International Cuisine', 'Comfort Food Classics')

WHERE r.recipe\_name = 'Beef Tacos';

-- Tiramisu

INSERT INTO recipe\_thematic\_section (recipe\_id, them\_sec\_id)

SELECT r.recipe\_id, ts.them\_sec\_id

FROM recipe r

JOIN thematic\_section ts ON ts.name IN ('Italian Cuisine', 'Gourmet Desserts')

WHERE r.recipe\_name = 'Tiramisu';

-- Chicken Curry

INSERT INTO recipe\_thematic\_section (recipe\_id, them\_sec\_id)

SELECT r.recipe\_id, ts.them\_sec\_id

FROM recipe r

JOIN thematic\_section ts ON ts.name IN ('International Cuisine', 'Comfort Food Classics')

WHERE r.recipe\_name = 'Chicken Curry';

-- Lemon Bars

INSERT INTO recipe\_thematic\_section (recipe\_id, them\_sec\_id)

SELECT r.recipe\_id, ts.them\_sec\_id

FROM recipe r

JOIN thematic\_section ts ON ts.name IN ('Desserts', 'Summer Grilling')

WHERE r.recipe\_name = 'Lemon Bars';

-- Mushroom Risotto

INSERT INTO recipe\_thematic\_section (recipe\_id, them\_sec\_id)

SELECT r.recipe\_id, ts.them\_sec\_id

FROM recipe r

JOIN thematic\_section ts ON ts.name IN ('Italian Cuisine', 'Comfort Food Classics', 'Risotto Recipes')

WHERE r.recipe\_name = 'Mushroom Risotto';

-- Chocolate Chip Cookies

INSERT INTO recipe\_thematic\_section (recipe\_id, them\_sec\_id)

SELECT r.recipe\_id, ts.them\_sec\_id

FROM recipe r

JOIN thematic\_section ts ON ts.name IN ('Desserts', 'Quick and Easy Meals')

WHERE r.recipe\_name = 'Chocolate Chip Cookies';

-- Grilled Chicken Caesar Salad

INSERT INTO recipe\_thematic\_section (recipe\_id, them\_sec\_id)

SELECT r.recipe\_id, ts.them\_sec\_id

FROM recipe r

JOIN thematic\_section ts ON ts.name IN ('Main Course', 'Salads', 'Summer Grilling')

WHERE r.recipe\_name = 'Grilled Chicken Caesar Salad';

-- Panna Cotta

INSERT INTO recipe\_thematic\_section (recipe\_id, them\_sec\_id)

SELECT r.recipe\_id, ts.them\_sec\_id

FROM recipe r

JOIN thematic\_section ts ON ts.name IN ('Desserts', 'Italian Cuisine')

WHERE r.recipe\_name = 'Panna Cotta';

-- it says zero rows affected here (?)

--These are some less important tables and so there are inserts for the first 20 recipes only

INSERT INTO meal\_type (meal\_type\_name) VALUES

('Breakfast'),

('Brunch'),

('Lunch'),

('Afternoon Snack'),

('Dinner'),

('Midnight Snack');

INSERT INTO recipe\_meal\_type (recipe\_id, meal\_type\_id) VALUES

-- Grilled Salmon

(1, 5), -- Dinner

-- Chocolate Cake

(2, 4), -- Afternoon Snack

-- Chicken Alfredo Pasta

(3, 5), -- Dinner

-- Apple Pie

(4, 4), -- Afternoon Snack

-- Vegetable Stir-Fry

(5, 5), -- Dinner

-- Vanilla Cupcakes

(6, 4), -- Afternoon Snack

-- Spaghetti Bolognese

(7, 5), -- Dinner

(7, 3), -- Lunch

-- New York Cheesecake

(8, 4), -- Afternoon Snack

-- Beef Tacos

(9, 3), -- Lunch

-- Tiramisu

(10, 6), -- Midnight Snack

-- Chicken Curry

(11, 5), -- Dinner

(11, 3), -- Lunch

-- Lemon Bars

(12, 4), -- Afternoon Snack

-- Mushroom Risotto

(13, 5), -- Dinner

(13, 3), -- Lunch

-- Chocolate Chip Cookies

(14, 4), -- Afternoon Snack

(14, 6), -- Midnight Snack

-- Grilled Chicken Caesar Salad

(15, 2), -- Brunch

-- Panna Cotta

(16, 4), -- Afternoon Snack

-- Tomato Basil Soup

(17, 3), -- Lunch

-- Green Tea Cheesecake

(18, 4), -- Afternoon Snack

(18, 6), -- Midnight Snack

-- Olive Tapenade

(19, 3), -- Lunch

-- Maple Syrup Pancakes

(20, 1), -- Breakfast

(20, 2); -- Brunch

INSERT INTO recipe\_step (step\_description, serial\_number, recipe\_id) VALUES

-- Grilled Salmon

('Preheat the grill to medium-high heat.', 1, 1),

('Marinate the salmon with herbs and let it sit for 30 minutes.', 2, 1),

('Grill the salmon for 15 minutes, turning halfway through.', 3, 1),

-- Chocolate Cake

('Preheat the oven to 350°F (175°C) and grease the cake pans.', 1, 2),

('Mix the dry ingredients and the wet ingredients separately, then combine.', 2, 2),

('Pour the batter into the pans and bake for 30-40 minutes.', 3, 2),

('Allow the cake to cool before applying the ganache.', 4, 2),

-- Chicken Alfredo Pasta

('Cook the pasta according to package instructions until al dente.', 1, 3),

('In a pan, cook the chicken pieces until browned and cooked through.', 2, 3),

('Prepare the Alfredo sauce by melting butter, adding cream and Parmesan, then mix with the pasta and chicken.', 3, 3),

-- Apple Pie

('Preheat the oven to 375°F (190°C) and prepare the pie crust.', 1, 4),

('Mix the sliced apples with cinnamon, sugar, and lemon juice.', 2, 4),

('Fill the crust with the apple mixture, cover with top crust, and bake for 40-50 minutes.', 3, 4),

-- Vegetable Stir-Fry

('Heat oil in a wok over high heat.', 1, 5),

('Add vegetables and stir-fry for 10 minutes.', 2, 5),

('Add tofu and sauce, then cook for another 5 minutes.', 3, 5),

-- Vanilla Cupcakes

('Preheat the oven to 350°F (175°C) and line a cupcake pan with liners.', 1, 6),

('Mix the cupcake batter and fill the liners halfway.', 2, 6),

('Bake for 20 minutes, then cool before frosting.', 3, 6),

-- Spaghetti Bolognese

('Cook the spaghetti according to package instructions.', 1, 7),

('In a separate pot, cook ground beef until browned, then add onions, garlic, and tomatoes.', 2, 7),

('Simmer the sauce for 1 hour and serve over the cooked spaghetti.', 3, 7),

-- New York Cheesecake

('Preheat the oven to 325°F (163°C) and prepare the crust.', 1, 8),

('Mix the cheesecake batter and pour it into the crust.', 2, 8),

('Bake the cheesecake in a water bath for 50 minutes.', 3, 8),

('Chill the cheesecake for at least 4 hours before serving.', 4, 8),

-- Beef Tacos

('Cook the beef with spices until browned.', 1, 9),

('Warm the tortillas in a pan.', 2, 9),

('Assemble the tacos with beef, salsa, and guacamole.', 3, 9),

-- Tiramisu

('Prepare the espresso and dip the ladyfingers in it.', 1, 10),

('Layer the ladyfingers and mascarpone mixture in a dish.', 2, 10),

('Chill the tiramisu for at least 4 hours before serving.', 3, 10),

-- Chicken Curry

('Cook onions, garlic, and ginger in a pot until softened.', 1, 11),

('Add chicken and spices, cook until chicken is browned.', 2, 11),

('Add coconut milk and simmer for 30 minutes.', 3, 11),

-- Lemon Bars

('Preheat the oven to 350°F (175°C) and prepare the crust.', 1, 12),

('Mix the lemon filling and pour over the pre-baked crust.', 2, 12),

('Bake for 35 minutes and chill before cutting into bars.', 3, 12),

-- Mushroom Risotto

('Cook the mushrooms in a pan until browned.', 1, 13),

('In a separate pot, cook onions and rice, gradually adding broth.', 2, 13),

('Stir in the cooked mushrooms and Parmesan cheese.', 3, 13),

-- Chocolate Chip Cookies

('Preheat the oven to 350°F (175°C) and prepare the cookie dough.', 1, 14),

('Chill the dough for 30 minutes.', 2, 14),

('Scoop dough onto a baking sheet and bake for 12 minutes.', 3, 14),

-- Grilled Chicken Caesar Salad

('Marinate the chicken in Caesar dressing.', 1, 15),

('Grill the chicken until cooked through and slice.', 2, 15),

('Toss the lettuce with dressing and top with chicken.', 3, 15),

-- Panna Cotta

('Bloom gelatin in cold water.', 1, 16),

('Heat cream and sugar until warm, then add gelatin.', 2, 16),

('Pour into molds and chill for 4 hours.', 3, 16),

-- Tomato Basil Soup

('Cook onions and garlic in a pot until softened.', 1, 17),

('Add tomatoes and basil, simmer for 30 minutes.', 2, 17),

('Blend the soup until smooth.', 3, 17),

-- Green Tea Cheesecake

('Preheat the oven to 325°F (163°C) and prepare the crust.', 1, 18),

('Mix the cheesecake batter with matcha and pour into the crust.', 2, 18),

('Bake the cheesecake for 50 minutes, then chill for 4 hours.', 3, 18),

-- Olive Tapenade

('Combine olives, capers, and garlic in a food processor.', 1, 19),

('Blend until smooth, adding olive oil as needed.', 2, 19),

('Serve with bread or crackers.', 3, 19),

-- Maple Syrup Pancakes

('Mix the pancake batter until just combined.', 1, 20),

('Cook the pancakes on a hot griddle until bubbles form.', 2, 20),

('Serve with butter and maple syrup.', 3, 20);

**Εξήγηση τυχαίων insert σε πίνακες episode\_cook\_recipe , judge , evalution με**

**χρήση python:**

**episode\_cook\_recipe inserts:**

Αρχικα η python συνδεεται στην βαση. Ορίζουμε την συναρτηση pick\_episode\_data

η οποια διαλεγει 10 διαφορετικά natcuis\_id και με βάση αυτά 10 διαφορετικά

cook\_id (με χρήση set αφου κάθε μάγειρας μπορεί να ανήκει σε πολλές κουζίνες και να ξαναεμφανιστεί) και recipe\_id , το καθένα από τα οποία αντιστοιχεί σε 1 εθνική κουζίνα (μέσω dictionaries). Στη συνέχεια για κάθε episode\_id απο

1 ως 30 κάνουμε 10 insert στον πίνακα episode\_cook\_recipe. Με χρήση if και

except , εαν η κουζίνα που επιλέχθηκε υπάρχει ήδη στο επισόδιο ή ενεργοποιείται κάποιο trigger λόγω του περιορισμού μη 3 συνεχόμενων cooks,

recipes, nat\_cuis , δεν γίνεται το insert και ξανακαλείται η συνάρτηση pick\_episode\_data για νέα δεδομένα.

**ΚΏΔΙΚΑΣ :**

import mysql.connector # type: ignore

import random

def pick\_episode\_data(conn):

cursor1 = conn.cursor()

cursor2 = conn.cursor()

cursor3 = conn.cursor()

cursor1.execute("SELECT natcuis\_id FROM national\_cuisine")

results1 = cursor1.fetchall()

random\_national\_cuisines\_ids = random.sample(results1, 10)

picked\_cooks = set()

cuisine\_cook\_map = {}

cuisine\_recipe\_map = {}

natcuis\_cook\_ids\_map = {}

for national\_cuisine\_id in random\_national\_cuisines\_ids:

natcuis\_id = national\_cuisine\_id[0]

cursor2.execute("SELECT cook\_id FROM cook\_nat\_cuis WHERE natcuis\_id = %s", (natcuis\_id,))

results2 = cursor2.fetchall()

cook\_ids = [result[0] for result in results2 if result[0] not in picked\_cooks]

natcuis\_cook\_ids\_map[natcuis\_id] = cook\_ids

if cook\_ids:

random\_cook\_id = random.choice(cook\_ids)

picked\_cooks.add(random\_cook\_id)

cuisine\_cook\_map[natcuis\_id] = random\_cook\_id

else:

print(f"National Cuisine ID: {natcuis\_id} has no available unique cooks")

cursor3.execute("SELECT recipe\_id, recipe\_name FROM recipe WHERE natcuis\_id = %s", (natcuis\_id,))

results3 = cursor3.fetchall()

if results3:

random\_recipe\_id = random.choice(results3)

cuisine\_recipe\_map[natcuis\_id] = random\_recipe\_id

else:

print(f"National Cuisine ID: {natcuis\_id} has no available recipes")

cursor1.close()

cursor2.close()

cursor3.close()

return cuisine\_cook\_map, cuisine\_recipe\_map

conn = mysql.connector.connect(

host="localhost",

user="root",

password="",

database="project84\_DB\_2024"

)

max\_retries = 15 # Maximum number of retries

insert\_counts = {} # Dictionary to store the number of successful inserts for each episode\_id

for episode\_id in range(1, 31):

insert\_counts[episode\_id] = 0 # Initialize insert count for each episode\_id

retries = 0

picked\_national\_cuisines = set() # Track picked national cuisines for each episode

while retries < max\_retries and insert\_counts[episode\_id] < 10:

try:

# Pick new data until we have enough unique entries

cuisine\_cook\_map, cuisine\_recipe\_map = pick\_episode\_data(conn)

cursor = conn.cursor()

success = False # Flag to indicate a successful insert

for natcuis\_id, cook\_id in cuisine\_cook\_map.items():

if natcuis\_id in picked\_national\_cuisines:

continue # Skip already picked national cuisines

recipe = cuisine\_recipe\_map.get(natcuis\_id)

if recipe:

recipe\_id = recipe[0]

insert\_query = "INSERT INTO episode\_cook\_recipe (cook\_id, episode\_id, recipe\_id) VALUES (%s, %s, %s)"

try:

cursor.execute(insert\_query, (cook\_id, episode\_id, recipe\_id))

conn.commit()

print(f"Inserted: Episode ID: {episode\_id}, National Cuisine ID: {natcuis\_id}, Cook ID: {cook\_id}, Recipe ID: {recipe\_id}")

insert\_counts[episode\_id] += 1 # Update insert count for the episode\_id

picked\_national\_cuisines.add(natcuis\_id) # Add to picked national cuisines

success = True # Mark the insert as successful

if insert\_counts[episode\_id] >= 10:

break # Exit the inner for-loop if reached 10 inserts

except mysql.connector.IntegrityError as e:

if e.sqlstate == '45000': # Custom error code for SIGNAL SQLSTATE

print(f"Insert failed for Episode ID {episode\_id}: {e.msg}. Retrying...")

else:

print(f"Insert failed for Episode ID {episode\_id}: {e.msg}. Retrying...")

else:

print(f"No recipe found for National Cuisine ID {natcuis\_id}")

cursor.close()

if success:

retries = 0 # Reset retries after a successful insert

else:

retries += 1 # Only increment retries if no successful insert

except mysql.connector.Error as err:

print(f"Error: {err}")

conn.rollback()

retries += 1 # Increment retries on error

conn.close()

**judge inserts:**

Αφού έχει γεμίσει ο πίνακας episode\_cook\_recipe , βρίσκουμε τους μάγειρες

που θα συμμετάσχουν σε ένα συγκεκριμένο επισόδιο και διαλέγουμε 3 διαφορετικές με την συνάρτηση pick\_judge. Στη συνέχεια με όμοιο τρόπο γεμίζουμε τον πίνακα judge , λαμβάνοντας υπόψιν περιορισμούς και triggers και δημιουργώντας νέα δεδομένα εαν ένα insert αποτύχει.

**ΚΏΔΙΚΑΣ :**

# judges

import mysql.connector # type: ignore

import random

def pick\_judge(conn, episode\_id):

cursor1 = conn.cursor() # for cooks in episodes

cursor2 = conn.cursor() # for all cooks

cursor1.execute("SELECT cook\_id FROM episode\_cook\_recipe WHERE episode\_id = %s", (episode\_id,))

existing\_cook\_ids = {row[0] for row in cursor1.fetchall()}

cursor2.execute("SELECT cook\_id FROM cook")

results2 = cursor2.fetchall()

available\_cook\_ids = [result[0] for result in results2 if result[0] not in existing\_cook\_ids]

judge\_ids = random.sample(available\_cook\_ids, 3)

cursor1.close()

cursor2.close()

return judge\_ids

conn = mysql.connector.connect(

host="localhost",

user="root",

password="",

database="project84\_DB\_2024"

)

max\_retries = 15 # Maximum number of retries

insert\_counts = {} # Dictionary to store the number of successful inserts for each episode\_id

for episode\_id in range(1, 31):

insert\_counts[episode\_id] = 0 # Initialize insert count for each episode\_id

retries = 0

while retries < max\_retries and insert\_counts[episode\_id] < 3:

try:

judge\_ids = pick\_judge(conn , episode\_id)

cursor = conn.cursor()

success = False # Flag to indicate a successful insert

for judge\_id in judge\_ids :

insert\_query = "INSERT INTO judge (cook\_id, episode\_id) VALUES (%s, %s)"

try:

cursor.execute(insert\_query, (judge\_id, episode\_id))

conn.commit()

print(f"Inserted: Episode ID: {episode\_id}, Cook ID: {judge\_id}")

insert\_counts[episode\_id] += 1 # Update insert count for the episode\_id

success = True # Mark the insert as successful

if insert\_counts[episode\_id] >= 3:

break # Exit the inner for-loop if reached 10 inserts

except mysql.connector.IntegrityError as e:

if e.sqlstate == '45000': # Custom error code for SIGNAL SQLSTATE

print(f"Insert failed for Episode ID {episode\_id}: {e.msg}. Retrying...")

# Retry by calling pick\_episode\_data again

judge\_ids = pick\_judge(conn ,episode\_id)

break # Exit the inner for-loop to retry with new data

else:

print(f"Insert failed for Episode ID {episode\_id}: {e.msg}. Retrying...")

# If there's a duplicate entry, pick new random national cuisine, cook, and recipe

judge\_ids = pick\_judge(conn ,episode\_id)

break # Exit the inner for-loop to retry with new data

else:

print(f"No judge found for episode id : {episode\_id}")

cursor.close()

if success:

break # Exit the while-loop if an insert was successful

except mysql.connector.Error as err:

print(f"Error: {err}")

conn.rollback()

finally:

retries += 1

conn.close()

**evaluation inserts:**

Με ένα Join βρίσκουμε όλους τους συνδιασμούς κριτών - μαγείρων που υπάρχουν στα επισόδια , επιλέγουμε τυχαία μια βαθμολογία από το 1 στο 5 και γεμίζουμε τον πίνακα evaluation.

**ΚΏΔΙΚΑΣ :**

# evaluation inserts

import mysql.connector # type: ignore

import random

conn = mysql.connector.connect(

host="localhost",

user="root",

password="",

database="project84\_DB\_2024"

)

cursor1 = conn.cursor() # for cook and judge ids

cursor2 = conn.cursor() # for inserts in evaluation

cursor1.execute("SELECT j.judge\_id, ecr.cook\_id FROM judge j JOIN episode\_cook\_recipe ecr ON j.episode\_id = ecr.episode\_id")

results1 = cursor1.fetchall()

# Insert into evaluation table

for judge\_id, cook\_id in results1:

grade = random.randint(1, 5)

cursor2.execute("INSERT INTO evaluation (cook\_id, judge\_id, grade) VALUES (%s, %s, %s)", (cook\_id, judge\_id, grade))

# Commit the transaction

conn.commit()

cursor1.close()

cursor2.close()

conn.close()

Ο κώδικας στο αρχείο **Indexes.sql** περιέχει εντολές που χρησιμοποιούνται βελτιστοποιούν τις επιδόσεις των ερωτημάτων, επιτρέποντας γρηγορότερη ανάκτηση δεδομένων μειώνοντας τον χρόνο αναζήτησης και τον αριθμό των απαιτούμενων δίσκων για πρόσβαση στα δεδομένα.

Ας αναλύσουμε τη χρησιμότητα κάθε ευρετηρίου :

1. **Πίνακας recipe**:
   * **idx\_recipe\_category**: Αυτό το ευρετήριο δημιουργείται πάνω στο πεδίο recipe\_category. Είναι χρήσιμο για ερωτήματα που φιλτράρουν συνταγές ανά κατηγορία.
   * **idx\_difficulty\_level**: Δημιουργείται πάνω στο πεδίο **difficulty\_level**. Χρησιμοποιείται για ερωτήματα που φιλτράρουν ή υπολογίζουν τις μέσες δυσκολίες των συνταγών.
2. **Πίνακας cook**:
   * **idx\_cook\_last\_name**: Αυτό το ευρετήριο δημιουργείται πάνω στο πεδίο **last\_name**. Χρησιμοποιείται για την επιτάχυνση ερωτημάτων που περιέχουν φίλτρα ή ομαδοποιήσεις βάσει του επωνύμου του μάγειρα.
   * **idx\_cook\_first\_name**: Δημιουργείται πάνω στο πεδίο first\_name. Παρέχει επιπλέον επιτάχυνση σε ερωτήματα που χρησιμοποιούν το πρώτο όνομα του μάγειρα.
   * **idx\_cook\_age**: Ευρετήριο πάνω στο πεδίο age. Χρησιμοποιείται για ερωτήματα που εστιάζουν σε ηλικιακά κριτήρια.
3. **Πίνακας national\_cuisine**:
   * **idx\_natcuis\_name**: Ευρετήριο πάνω στο πεδίο natcuis\_name. Χρησιμοποιείται για ερωτήματα που φιλτράρουν ή ομαδοποιούν κατά την εθνική κουζίνα.
4. **Πίνακας tags**:
   * **idx\_tag\_name**: Ευρετήριο πάνω στο πεδίο tag\_name. Χρησιμοποιείται για ερωτήματα που φιλτράρουν ή συνδέουν συνταγές με ετικέτες.
5. **Πίνακας episode**:
   * **idx\_episode\_season**: Ευρετήριο πάνω στο πεδίο season. Χρησιμοποιείται για ερωτήματα που φιλτράρουν συνταγές ανά εποχή.
   * **idx\_episode\_episode\_name**: Χρησιμοποιείται για ερωτήματα που επιστρέφουν τα ονόματα των επεισοδίων.
6. **Πίνακας foodgroups**:
   * **idx\_judge\_foodgroups\_name**: Ευρετήριο πάνω στο πεδίο foodgroups\_name. Χρησιμοποιείται για ερωτήματα που φιλτράρουν ή συνδέουν συνταγές με ομάδες τροφίμων.

**Φαινεται αναλυτικα στα σχόλια του κλωδικα σε ποια Queries είναι το καθένα Index .**

**ΚΏΔΙΚΑΣ : Indexes.sql**

--Αν το ερώτημα εκτελείται αργά, μπορείτε να εξετάσετε την ανάλυση του σχεδίου εκτέλεσης του query

--(query execution plan) με την εντολή EXPLAIN, ώστε να δείτε αν τα ευρετήρια χρησιμοποιούνται όπως

--αναμένεται:

-----------------------------------------------------------------------------------------------------------

-- 1. ΠΙΝΑΚΑΣ recipe

-- QUERY 12 :difficulty\_level

CREATE INDEX idx\_recipe\_category ON recipe(recipe\_category);

CREATE INDEX idx\_difficulty\_level ON recipe(difficulty\_level);

-----------------------------------------------------------------------------------------------------------

--2. ΠΙΝΑΚΑΣ cook

-- QUERY 1 : SELECT c.first\_name, c.last\_name

-- QUERY 2 : SELECT c.first\_name, c.last\_name

-- QUERY 3 : SELECT c.first\_name, c.last\_name, WHERE c.age < 30 ,

-- QUERY 4 : SELECT c.first\_name, c.last\_name

-- QUERY 5 : SELECT c.first\_name, c.last\_name

-- QUERY 7 : SELECT c.first\_name, c.last\_name,

--Query 11 : SELECT c.last\_name

CREATE INDEX idx\_cook\_last\_name ON cook(first\_name);

CREATE INDEX idx\_cook\_first\_name ON cook(last\_name);

CREATE INDEX idx\_cook\_age ON cook(age);

-----------------------------------------------------------------------------------------------------------

--3. Πίνακας national\_cuisine

--QUERY 1 :SELECT natcuis\_name,

--QUERY 2 :SELECT natcuis\_name , where nc.natcuis\_name

--QUERY 10:SELECT natcuis\_name

CREATE INDEX idx\_natcuis\_name ON national\_cuisine(natcuis\_name);

-----------------------------------------------------------------------------------------------------------

--4. Πίνακας tags

--QUERY 6:SELECT t1.tag\_name

CREATE INDEX idx\_tag\_name ON tags(tag\_name);

-----------------------------------------------------------------------------------------------------------

--5. Πίνακας episode

--QUERY 2 :season, where e.season = 2

--QUERY 5 :season,

--QUERY 8 :SELECT episode\_name

--QUERY 9: season

--QUERY 10 : SELECT e2.season

--QUERY 12 : season ,SELECT episode\_name

--QUERY 13 :SELECT episode\_name

CREATE INDEX idx\_episode\_season ON episode(season);

CREATE INDEX idx\_episode\_episode\_name ON episode(episode\_name);

-----------------------------------------------------------------------------------------------------------

--6. Πίνακας foodgroups

--QUERΥ 15 : SELECT fg.foodgroups\_name ,WHERE fg.foodgroups\_id IS NULL

CREATE INDEX idx\_judge\_foodgroups\_name ON foodgroups(foodgroups\_name);

Ο κώδικας στο αρχείο **Queries.sql:**

**Παρατήρησή για το Query 5:**

Σε τυχαία δεδομένα είναι δύσκολο να έχουμε έναν κριτή να εμφανίζεται περισσότερες από τρεις φορές σε μία σεζόν, γι αυτό τον λόγο το query υπάρχει πιθανότητα να μην εμφανίζει αποτέλεσμα. Για να φανεί η αποτελεσματικότητα του προτείνουμε να γίνει αλλαγή από μεγαλύτερο του 3 σε μεγαλύτερο του 2 ή 1**.**

**ΚΏΔΙΚΑΣ : Queries.sql**

-------------------------------------------------------------------------Query 1---------------------------------------------------------------------

--Eugene

--per cook and nat cius

SELECT c.cook\_id,c.first\_name, c.last\_name,r.natcuis\_id,n.natcuis\_name,avg(grade)

FROM cook c

JOIN evaluation e ON c.cook\_id = e.cook\_id

JOIN judge j on e.judge\_id = j.judge\_id

JOIN episode\_cook\_recipe ecr ON e.cook\_id = ecr.cook\_id AND j.episode\_id = ecr.episode\_id

JOIN recipe r on r.recipe\_id = ecr.recipe\_id

JOIN national\_cuisine n on r.natcuis\_id = n.natcuis\_id

GROUP BY cook\_id,natcuis\_id;

-------------------------------------------------------------------------Query 2---------------------------------------------------------------------

SELECT c.first\_name,c.last\_name, nc.natcuis\_name

from cook c

JOIN cook\_nat\_cuis cn on c.cook\_id = cn.cook\_id

JOIN national\_cuisine nc on cn.natcuis\_id = nc.natcuis\_id

where nc.natcuis\_name = 'Afghan cuisine';

SELECT c.first\_name,c.last\_name,e.season

from cook c

JOIN episode\_cook\_recipe cer on c.cook\_id = cer.cook\_id

JOIN episode e on e.episode\_id = cer.episode\_id

where e.season = 2 ; --needs more data in other seasons

-------------------------------------------------------------------------Query 3---------------------------------------------------------------------

SELECT

c.first\_name, c.last\_name, c.age, COUNT(ecr.recipe\_id) AS total\_recipes

FROM

cook c

JOIN

episode\_cook\_recipe ecr ON c.cook\_id = ecr.cook\_id

WHERE

c.age < 30

GROUP BY

c.cook\_id

ORDER BY

total\_recipes DESC;

-------------------------------------------------------------------------Query 4---------------------------------------------------------------------

SELECT

c.first\_name, c.last\_name

FROM

cook c

LEFT JOIN

judge j ON c.cook\_id = j.cook\_id

WHERE

j.judge\_id IS NULL;

-------------------------------------------------------------------------Query 5---------------------------------------------------------------------

SELECT c.first\_name, c.last\_name, e.season, count(e.episode\_id) as part\_count

FROM cook c

JOIN judge j on j.cook\_id = c.cook\_id

JOIN episode e on e.episode\_id = j.episode\_id

GROUP BY j.cook\_id,e.season

HAVING part\_count>3

ORDER BY part\_count;

-------------------------------------------------------------------------Query 6---------------------------------------------------------------------

SELECT t1.tag\_name AS tag\_name1, t2.tag\_name AS tag\_name2, COUNT(\*) AS pair\_count

FROM recipe\_tag rt1

JOIN episode\_cook\_recipe ecr ON rt1.recipe\_id = ecr.recipe\_id

JOIN recipe\_tag rt2 ON rt1.recipe\_id = rt2.recipe\_id AND rt1.tag\_id < rt2.tag\_id

JOIN tags t1 ON rt1.tag\_id = t1.tag\_id

JOIN tags t2 ON rt2.tag\_id = t2.tag\_id

GROUP BY rt1.tag\_id, rt2.tag\_id

ORDER BY pair\_count DESC

LIMIT 3;

-------------------------------------------------------------------------Query 7---------------------------------------------------------------------

SELECT c.cook\_id, c.first\_name, c.last\_name, COUNT(ecr.episode\_id) AS participation

FROM cook c

JOIN episode\_cook\_recipe ecr ON c.cook\_id = ecr.cook\_id

GROUP BY c.cook\_id, c.first\_name, c.last\_name

HAVING COUNT(ecr.episode\_id) <= (

SELECT MAX(participation\_count) - 5

FROM (

SELECT c.cook\_id, COUNT(ecr.episode\_id) AS participation\_count

FROM cook c

JOIN episode\_cook\_recipe ecr ON c.cook\_id = ecr.cook\_id

GROUP BY c.cook\_id

) AS table1

);

-------------------------------------------------------------------------Query 8---------------------------------------------------------------------

SELECT episode\_name, COUNT(DISTINCT re.equipment\_id) AS equip\_count

FROM episode\_cook\_recipe ecr

JOIN recipe\_equipment re ON re.recipe\_id = ecr.recipe\_id

JOIN episode ep ON ep.episode\_id = ecr.episode\_id

GROUP BY ecr.episode\_id

ORDER BY equip\_count DESC

LIMIT 10;

-------------------------------------------------------------------------Query 9---------------------------------------------------------------------

SELECT

YEAR(e.episode\_date) AS year, e.season,

AVG(r.carbohydrate\_per\_portion \* r.quantity\_of\_servings) AS total\_carbohydrates

FROM

episode e

JOIN

episode\_cook\_recipe ecr ON e.episode\_id = ecr.episode\_id

JOIN

recipe r ON ecr.recipe\_id = r.recipe\_id

GROUP BY

e.season

ORDER BY

year;

-------------------------------------------------------------------------Query 10---------------------------------------------------------------------

SELECT table1.natcuis\_name as nat\_cuis, table1.season as name\_of\_season\_1, table2.season as name\_of\_season\_2,

table1.cuis\_count1, table2.cuis\_count2,(table1.cuis\_count1 + table2.cuis\_count2) AS total\_count

FROM

(SELECT nc.natcuis\_name, e.season, r.natcuis\_id, count(ecr.episode\_id) as cuis\_count1

FROM national\_cuisine nc

JOIN recipe r ON r.natcuis\_id = nc.natcuis\_id

JOIN episode\_cook\_recipe ecr ON r.recipe\_id = ecr.recipe\_id

JOIN episode e ON e.episode\_id = ecr.episode\_id

GROUP BY nc.natcuis\_id,e.season

HAVING cuis\_count1 >= 3) AS table1

JOIN

(SELECT nc2.natcuis\_name, e2.season, r2.natcuis\_id, count(ecr2.episode\_id) as cuis\_count2

FROM national\_cuisine nc2

JOIN recipe r2 ON r2.natcuis\_id = nc2.natcuis\_id

JOIN episode\_cook\_recipe ecr2 ON r2.recipe\_id = ecr2.recipe\_id

JOIN episode e2 ON e2.episode\_id = ecr2.episode\_id

GROUP BY nc2.natcuis\_id,e2.season

HAVING cuis\_count2 >= 3) AS table2

ON table1.natcuis\_id = table2.natcuis\_id

WHERE ABS(table1.season - table2.season) =1 and table1.season< table2.season

ORDER BY total\_count DESC ;

-------------------------------------------------------------------------Query 11---------------------------------------------------------------------

SELECT

c2.last\_name as judge\_name,

c.last\_name as cook\_name,

e.grade,

ecr.episode\_id

FROM

judge j

JOIN

evaluation e on j.judge\_id = e.judge\_id

JOIN

episode\_cook\_recipe ecr on ecr.cook\_id = e.cook\_id AND ecr.episode\_id = j.episode\_id

JOIN

cook c on c.cook\_id = ecr.cook\_id

JOIN

cook c2 on c2.cook\_id = j.cook\_id

WHERE

c.last\_name = 'D Acampo'

ORDER BY

e.grade DESC

LIMIT 5;

--or for more general :

WITH RankedJudges AS (

SELECT

c2.last\_name AS judge\_name,

c.last\_name AS cook\_name,

e.grade,

ecr.episode\_id,

ROW\_NUMBER() OVER (PARTITION BY c.last\_name ORDER BY e.grade DESC) AS rank

FROM

judge j

JOIN

evaluation e ON j.judge\_id = e.judge\_id

JOIN

episode\_cook\_recipe ecr ON ecr.cook\_id = e.cook\_id

JOIN

cook c ON c.cook\_id = ecr.cook\_id

JOIN

cook c2 ON c2.cook\_id = j.cook\_id

)

SELECT

judge\_name,

cook\_name,

grade,

episode\_id

FROM

RankedJudges

WHERE

rank <= 5;

-------------------------------------------------------------------------Query 12---------------------------------------------------------------------

WITH avg\_diff\_per\_episode AS (

SELECT

ep.season,

ep.episode\_name,

ep.episode\_id,

AVG(re.difficulty\_level) AS avg\_dif

FROM

episode\_cook\_recipe ecr

JOIN

episode ep ON ep.episode\_id = ecr.episode\_id

JOIN

recipe re ON re.recipe\_id = ecr.recipe\_id

GROUP BY

ep.season, ep.episode\_name, ep.episode\_id

),

max\_avg\_diff\_per\_season AS (

SELECT

season,

MAX(avg\_dif) AS highest\_dif

FROM

avg\_diff\_per\_episode

GROUP BY

season

)

SELECT

adp.season,

adp.episode\_name,

mad.highest\_dif

FROM

max\_avg\_diff\_per\_season mad

JOIN

avg\_diff\_per\_episode adp

ON mad.season = adp.season

AND mad.highest\_dif = adp.avg\_dif;

-------------------------------------------------------------------------Query 13---------------------------------------------------------------------

SELECT

combined.episode\_id,

combined.episode\_name,

combined.episode\_date,

SUM(

CASE combined.position\_level

WHEN 'cook A' THEN 1

WHEN 'cook B' THEN 2

WHEN 'cook C' THEN 3

WHEN 'chef assistant' THEN 4

WHEN 'chef' THEN 5

END

) AS total\_position\_level

FROM (

SELECT

e.episode\_id,

e.episode\_name,

e.episode\_date,

c.position\_level

FROM

episode e

JOIN

episode\_cook\_recipe ecr ON e.episode\_id = ecr.episode\_id

JOIN

cook c ON ecr.cook\_id = c.cook\_id

UNION ALL

SELECT

e.episode\_id,

e.episode\_name,

e.episode\_date,

c.position\_level

FROM judge j

JOIN cook c ON j.cook\_id = c.cook\_id

JOIN episode e ON j.episode\_id = e.episode\_id

) AS combined

GROUP BY combined.episode\_id

ORDER BY

total\_position\_level ASC

;

--LIMIT 1;

-- needs more inserts

-------------------------------------------------------------------------Query 14---------------------------------------------------------------------

select ts.name as thematic\_section\_name ,COUNT(ts.them\_sec\_id) as appearance\_count

from episode\_cook\_recipe ecr

join recipe r on ecr.recipe\_id = r.recipe\_id

join recipe\_thematic\_section rts on r.recipe\_id = rts.recipe\_id

join thematic\_section ts on rts.them\_sec\_id = ts.them\_sec\_id

group by ts.them\_sec\_id

order by appearance\_count DESC

LIMIT 1;

-------------------------------------------------------------------------Query 15---------------------------------------------------------------------

SELECT fg.foodgroups\_name

FROM ingredient\_VS\_recipe ir

JOIN ingredient i ON i.ingredient\_id = ir.ingredient\_id

JOIN episode\_cook\_recipe ecr ON ecr.recipe\_id= ir.recipe\_id

RIGHT JOIN foodgroups fg ON fg.foodgroups\_id = i.foodgroups\_id

WHERE i.ingredient\_id IS NULL;

**Έχουμε δημιουργήσει δύο ρόλους**

• **Διαχειριστής** -> administrator ο οποίος έχει δυνατότητα να επεξεργάζεται όλους τους πίνακες, views, triggers κτλ

• **Μάγειρας** -> cook\_role , ο οποίος έχει την δυνατότητα να δει και να προσθέσει πράγματα που αφορούν τις συνταγές αλλά όχι να επεξεργαστεί κάτι που υπάρχει ήδη καθώς θα μπορούσε να ανήκει σε κάποιον άλλο μάγειρα. Δεν έχει επίσης πρόσβαση σε πίνακες όπως το cook, καθώς δεν θέλουμε να ξέρει τις προσωπικές πληροφορίες των άλλων μαγείρων.

**ΚΏΔΙΚΑΣ: Roles\_Users\_Grants.sql**

---------------Role Creation -> Cook -----------------------

CREATE ROLE cook\_role;

------------- Give a grant to a role--------------

GRANT SELECT, INSERT ON project84\_DB\_2024.recipe TO cook\_role;

---Για να προσθέσει κάποιος μία συνταγή πρέπει να προσθέσει τον εξοπλισμό της και επισης να μπορεί να προσθέσει κάποιον νέο εξοπλισμό αν χρειάζεται

GRANT SELECT, INSERT ON project84\_DB\_2024.equipment TO cook\_role;

GRANT SELECT, INSERT ON project84\_DB\_2024.recipe\_equipment TO cook\_role;

--- Αντίστοιχα πρέπει να προσθέσει την εθνική κουζίνα της συνταγής,τα υλικά που απαιτούνται και τα διάφορα tags ή meal\_types

GRANT SELECT, INSERT ON project84\_DB\_2024.national\_cuisine TO cook\_role;

GRANT SELECT, INSERT ON project84\_DB\_2024.ingredient TO cook\_role;

GRANT SELECT, INSERT ON project84\_DB\_2024.ingredient\_VS\_recipe TO cook\_role;

GRANT SELECT, INSERT ON project84\_DB\_2024.tags TO cook\_role;

GRANT SELECT, INSERT ON project84\_DB\_2024.recipe\_tag TO cook\_role;

GRANT SELECT, INSERT ON project84\_DB\_2024.recipe\_meal\_type TO cook\_role;

GRANT SELECT, INSERT ON project84\_DB\_2024.meal\_type TO cook\_role;

GRANT SELECT, INSERT ON project84\_DB\_2024.recipe\_step TO cook\_role;

GRANT SELECT, INSERT ON project84\_DB\_2024.thematic\_section TO cook\_role;

GRANT SELECT, INSERT ON project84\_DB\_2024.recipe\_thematic\_section TO cook\_role;

--Θα μπορούν επισης να δουν αλλα όχι να προσθέσουμε κατι που αφορά τα επεισόδια

GRANT SELECT ON project84\_DB\_2024.episode TO cook\_role;

-------------Users----------------------

CREATE USER 'gordon\_ramsey'@'localhost' IDENTIFIED by 'password1';

GRANT cook\_role to 'gordon\_ramsey'@'localhost';

---How to give grants to individual user ( meaning not though roles)

GRANT SELECT, INSERT ON project84\_DB\_2024.recipe TO 'gordon\_ramsey'@'localhost';

---mysql -u gordon\_ramsey -p

-------- Role Creation -> Administrator -----------------

CREATE ROLE administrator;

GRANT ALL PRIVILEGES ON \*.\* TO administrator WITH GRANT OPTION;

-- Creating a BACKUP containing all the sql code for the recreation of the database------------

mysqldump -u username -p masterchef > masterchef\_backup.sql

--restoration

source /path/to/backup\_file.sql;