



Εθνικό Μετσόβιο Πολυτεχνείο

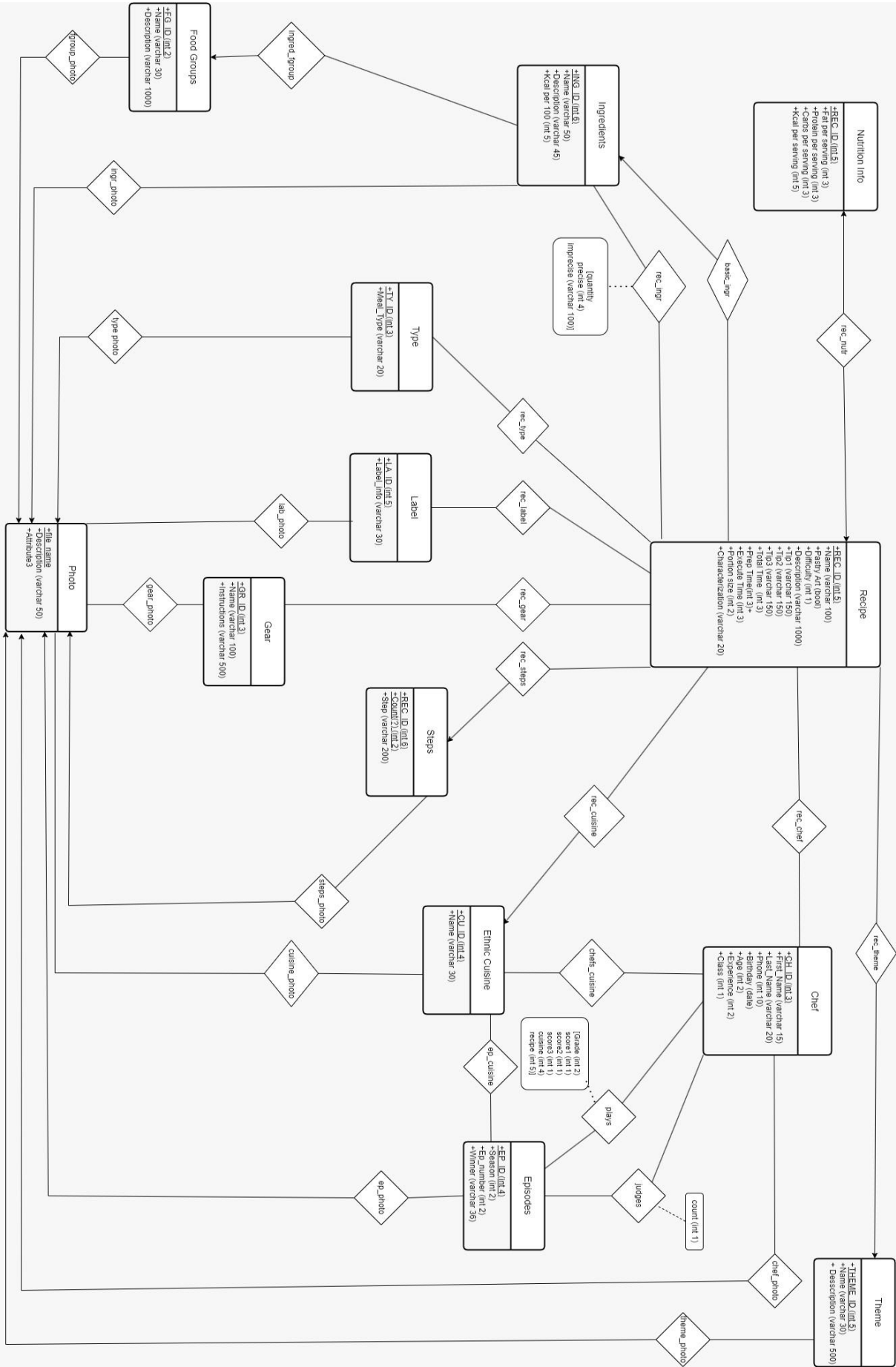
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Εξάμηνο 6^ο – Ακαδημαϊκό Έτος 2023-24

Βάσεις Δεδομένων: Εξαμηνιαία Εργασία

| | |
|-------------|--|
| Συγγραφείς | Καπετανάκη Ειρήνη (Α.Μ.: 03121196) Μάγος Παναγιώτης (Α.Μ.: 03121005) Βασιλείου Νεκτάριος – Μιχαήλ (Α.Μ.: 03121204) |
| Ομάδα | 81 |
| Περιεχόμενα | ER Diagram Οντότητες Σχέσεις Relational Schema Indexes Mock Data Creation Οδηγίες Εγκατάστασης της Εφαρμογής Οδηγίες Χρήσης της Εφαρμογής Οδηγίες Χρήσης για Διαχειριστή Οδηγίες Χρήσης για Μάγειρα Ζητούμενα Queries Βιβλιοθήκες και Frameworks που χρησιμοποιήθηκαν |

ER Diagram



Οντότητες

Στις περισσότερες οντότητες υπάρχει το attribute *file_name*. Αυτό είναι το Foreign Key που θα έχουμε για την αντιστοίχιση του κάθε αντικειμένου με τη φωτογραφία του.

Παρακάτω φαίνονται τα attributes της κάθε οντότητας με την περιγραφή τους:

Recipes

| <i>Attribute</i> | <i>Περιγραφή</i> |
|------------------|---------------------------------------|
| REC_ID | Αναγνωριστικό της συνταγής |
| Name | Όνομα συνταγής |
| Pastry | Αλμυρό ή γλυκό φαγητό |
| Difficulty | Επίπεδο δυσκολίας |
| Description | Περιγραφή συνταγής |
| Tip1 | 1 ^η συμβουλή |
| Tip2 | 2 ^η συμβουλή |
| Tip3 | 3 ^η συμβουλή |
| Total Time | Συνολικός χρόνος εκτέλεσης |
| Prep Time | Χρόνος προετοιμασίας |
| Characterization | Χαρακτηρισμός με βάση το βασικό υλικό |

Το REC_ID διασφαλίζει πως κάθε συνταγή θα έχει ένα μοναδικό αριθμό που θα λειτουργεί ως το αναγνωριστικό της (ID). Τα PRIM_ING_ID, CU_ID, THEME_ID υπάρχουν για τη σωστή λειτουργία των σχέσεων basic_ingr, rec_cuisine, rec_theme αντιστοίχως.

Ingredients

| <i>Attribute</i> | <i>Περιγραφή</i> |
|------------------|----------------------------|
| ING_ID | Αναγνωριστικό του υλικού |
| Name | Όνομα του υλικού |
| Kcal per 100 | Θερμίδες ανά 100 γραμμάρια |

Το ING_ID διασφαλίζει πως κάθε υλικό θα έχει έναν μοναδικό αριθμό που θα λειτουργεί ως το αναγνωριστικό του (ID). Το FG_ID υπάρχει για τη σωστή λειτουργία της σχέσης ingred_fgrouop.

Steps

| <i>Attribute</i> | <i>Περιγραφή</i> |
|------------------|----------------------------|
| REC_ID | Αναγνωριστικό της συνταγής |
| Count | Αναγνωριστικό του βήματος |
| Step | Περιγραφή του βήματος |

Τα REC_ID και Count αποτελούν μαζί το αναγνωριστικό του εκάστοτε βήματος της εκάστοτε συνταγής (Composite Primary Key).

Food Groups

| Attribute | Περιγραφή |
|-------------|-----------------------------------|
| FG_ID | Αναγνωριστικό της ομάδας τροφίμων |
| Name | Όνομα της ομάδας τροφίμων |
| Description | Περιγραφή της ομάδας |

Το FG_ID διασφαλίζει πως κάθε ομάδα τροφίμων θα έχει έναν μοναδικό αριθμό που θα λειτουργεί ως αναγνωριστικό της (ID).

Nutrition Info

| Attribute | Περιγραφή |
|---------------------|-------------------------|
| Fat per serving | Λιπαρά ανά μερίδα |
| Protein per serving | Πρωτεΐνη ανά μερίδα |
| Carbs per serving | Υδατάνθρακες ανά μερίδα |
| Kcal per serving | Θερμίδες ανά μερίδα |

Το REC_ID λειτουργεί ως Foreign Key και αντιστοιχίζει κάθε συνταγή με τη διατροφική της αξία (το REC_ID είναι μοναδικό για κάθε συνταγή).

Ethnic Cuisine

| Attribute | Περιγραφή |
|-----------|------------------------------------|
| CU_ID | Αναγνωριστικό της εθνικής κουζίνας |
| Name | Όνομα της εθνικής κουζίνας |

Το CU_ID διασφαλίζει πως κάθε εθνική κουζίνα θα έχει έναν μοναδικό αριθμό που θα λειτουργεί ως αναγνωριστικό της (ID).

Type

| Attribute | Περιγραφή |
|-----------|----------------------------------|
| TY_ID | Αναγνωριστικό του τύπου γεύματος |
| Meal_type | Όνομα του τύπου γεύματος |

Το TY_ID διασφαλίζει πως κάθε τύπος γεύματος θα έχει έναν μοναδικό αριθμό που θα λειτουργεί ως αναγνωριστικό του (ID).

Label

| Attribute | Περιγραφή |
|------------|----------------------------|
| LA_ID | Αναγνωριστικό της ετικέτας |
| Label Name | Όνομα της ετικέτας |
| Label_Info | Περιγραφή για την ετικέτα |

Το LA_ID διασφαλίζει πως κάθε ετικέτα θα έχει έναν μοναδικό αριθμό που θα λειτουργεί ως αναγνωριστικό της (ID).

Gear

| <i>Attribute</i> | <i>Περιγραφή</i> |
|------------------|-------------------------------|
| GR_ID | Αναγνωριστικό του εξοπλισμού |
| Name | Όνομα του εξοπλισμού |
| Instructions | Οδηγίες χρήσης του εξοπλισμού |

Το GR_ID διασφαλίζει πως κάθε εξοπλισμός θα έχει έναν μοναδικό αριθμό που θα λειτουργεί ως αναγνωριστικό του (ID).

Chefs

| <i>Attribute</i> | <i>Περιγραφή</i> |
|------------------|--|
| CH_ID | Αναγνωριστικό του μάγειρα |
| First_Name | Όνομα του μάγειρα |
| Last_Name | Επώνυμο του μάγειρα |
| Phone | Τηλέφωνο του μάγειρα |
| Birthday | Γενέθλια του μάγειρα |
| Age | Ηλικία του μάγειρα |
| Experience | Χρόνια εμπειρίας του μάγειρα |
| Class | Επίπεδο κατάρτισης του μάγειρα (1 έως 5) |

Το CH_ID διασφαλίζει πως κάθε μάγειρας θα έχει έναν μοναδικό αριθμό που θα λειτουργεί ως αναγνωριστικό του (ID).

Episodes

| <i>Attribute</i> | <i>Περιγραφή</i> |
|------------------|--|
| EP_ID | Αναγνωριστικό του επεισοδίου |
| Season | Η σεζόν στην οποία ανήκει το επεισόδιο |
| EP_Number | Ο αριθμός του επεισοδίου |
| Winner | Ο νικητής του επεισοδίου |

Το EP_ID διασφαλίζει πως κάθε επεισόδιο θα έχει έναν μοναδικό αριθμό που θα λειτουργεί ως αναγνωριστικό του (ID).

Theme

| <i>Attribute</i> | <i>Περιγραφή</i> |
|------------------|---------------------------|
| THEME_ID | Αναγνωριστικό του θέματος |
| Name | Όνομα του θέματος |

| | |
|-------------|-----------------------|
| Description | Περιγραφή του θέματος |
|-------------|-----------------------|

Το THEME_ID διασφαλίζει πως κάθε θέμα θα έχει έναν μοναδικό αριθμό που θα λειτουργεί ως αναγνωριστικό του (ID).

Photo

| Attribute | Περιγραφή |
|-------------|-------------------------------|
| file_name | Αναγνωριστικό της φωτογραφίας |
| Description | Περιγραφή της φωτογραφίας |

Σχέσεις

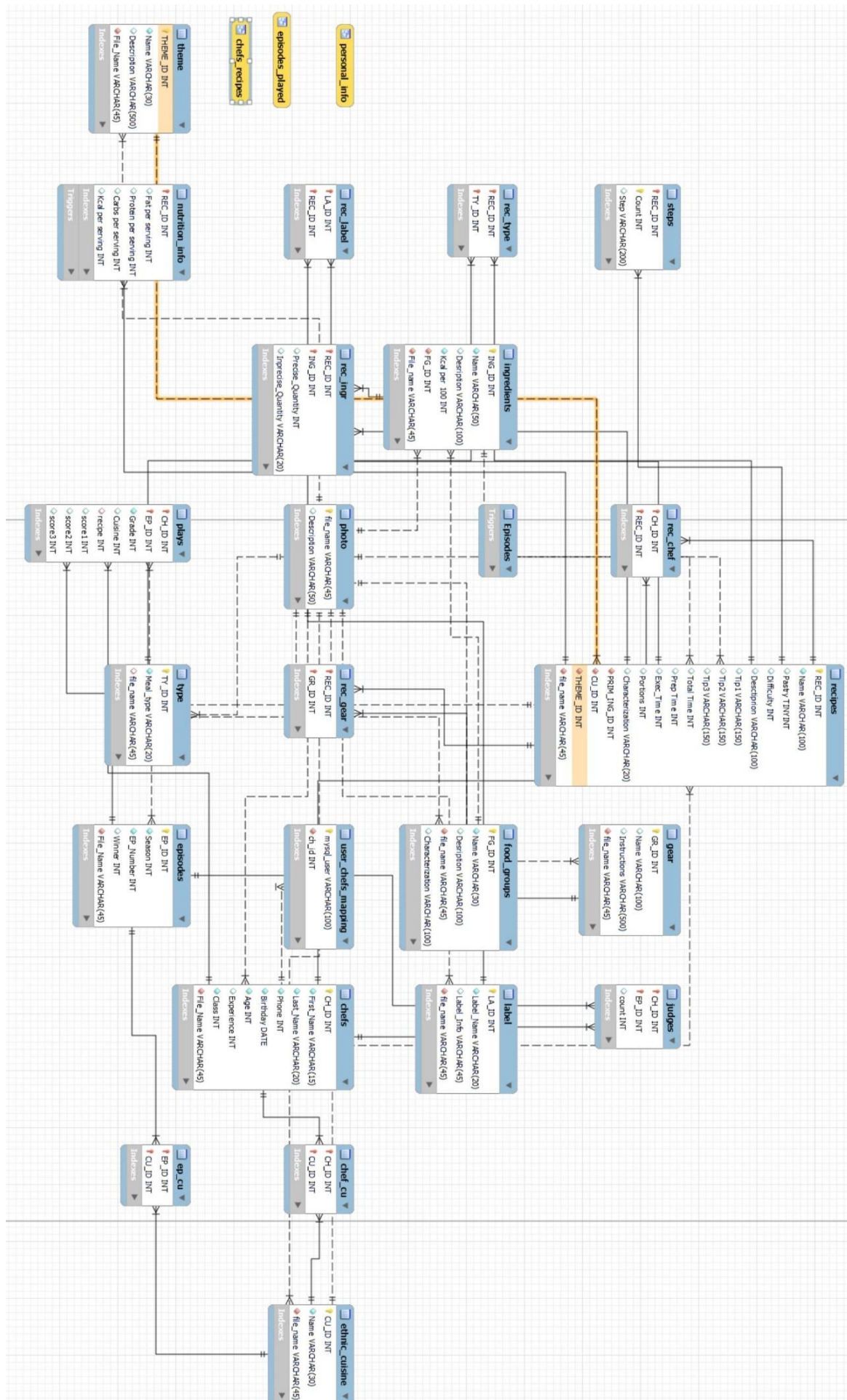
Έχουμε τις παρακάτω σχέσεις:

| Σχέση | Τύπος | Περιγραφή |
|---------------|-------|---|
| rec_nutr | 1:1 | Ποια είναι η διατροφική αξία μιας συνταγής / Σε ποια συνταγή αντιστοιχεί μια διατροφική αξία |
| rec_ingr | N:M | Ποια είναι τα υλικά μιας συνταγής / Σε ποιες συνταγές χρησιμοποιείται ένα υλικό |
| basic_ingr | 1:N | Ποιο είναι το βασικό υλικό μιας συνταγής / Σε ποιες συνταγές είναι ένα υλικό βασικό |
| rec_type | 1:N | Ποιος είναι ο τύπος μιας συνταγής / Ποιες συνταγές ανήκουν σε αυτόν τον τύπο |
| rec_label | 1:1 | Ποια είναι η ετικέτα μιας συνταγής / Σε ποια συνταγή αντιστοιχεί μια ετικέτα |
| rec_gear | N:M | Ποιος είναι ο εξοπλισμός μια συνταγής / Σε ποιες συνταγές ανήκει αυτός ο εξοπλισμός |
| rec_steps | N:1 | Ποια είναι τα βήματα μιας συνταγής / Σε ποια συνταγή ανήκει αυτό το βήμα |
| rec_cuisine | 1:N | Σε ποια κουζίνα ανήκει μια συνταγή / Ποιες συνταγές ανήκουν σε αυτήν την κουζίνα |
| rec_chef | N:M | Ποιοι μάγειρες ξέρουν μια συνταγή / Ποιες συνταγές ξέρει ένας μάγειρας |
| rec_theme | 1:N | Ποιο είναι το θέμα μιας συνταγής / Ποιες συνταγές έχουν αυτό το θέμα |
| ingred_fgroup | 1:N | Σε ποια ομάδα τροφίμων ανήκει ένα υλικό / Ποια υλικά ανήκουν σε αυτήν την ομάδα τροφίμων |
| chefs_cuisine | N:M | Σε ποιες κουζίνες ειδικεύεται αυτός ο μάγειρας / Ποιοι μάγειρες ειδικεύονται σε αυτήν την κουζίνα |
| ep_cuisine | N:M | Ποιες κουζίνες παίζουν σε ένα επεισόδιο / Σε ποια επεισόδια έχει παίξει μια κουζίνα |
| plays | N:M | Ποιοι μάγειρες παίζουν σε ένα επεισόδιο / Σε ποια επεισόδια έχει παίξει ένας μάγειρας |
| judges | N:M | Ποιοι μάγειρες κρίνουν σε ένα επεισόδιο / Σε ποια επεισόδια έχει κρίνει ένας μάγειρας |
| rec_photo | 1:N | Φωτογραφία συνταγής / Φωτογραφίες όλων των συνταγών |

| | | |
|---------------|-----|---|
| nut_photo | 1:N | Φωτογραφία διατροφικής αξίας / Φωτογραφίες όλων των διατροφικών πληροφοριών |
| fgroup_photo | 1:N | Φωτογραφία ομάδας τροφίμων / Φωτογραφίες όλων των ομάδων τροφίμων |
| ingr_photo | 1:N | Φωτογραφία υλικού / Φωτογραφίες όλων των υλικών |
| type_photo | 1:N | Φωτογραφία τύπου γεύματος / Φωτογραφίες όλων των τύπων γεύματος |
| lab_photo | 1:N | Φωτογραφία ετικέτας / Φωτογραφίες όλων των ετικετών |
| gear_photo | 1:N | Φωτογραφία εξοπλισμού / Φωτογραφίες όλων των εξοπλισμών |
| cuisine_photo | 1:N | Φωτογραφία εθνικής κουζίνας / Φωτογραφίες όλων των εθνικών κουζινών |
| ep_photo | 1:N | Φωτογραφία επεισοδίου / Φωτογραφίες όλων των επεισοδίων |
| chef_photo | 1:N | Φωτογραφία μάγειρα / Φωτογραφίες όλων των μάγειρων |
| theme_photo | 1:N | Φωτογραφία θέματος / Φωτογραφίες όλων των θεμάτων |

Το file_name διασφαλίζει πως κάθε φωτογραφία θα έχει έναν μοναδικό αριθμό που θα λειτουργεί ως αναγνωριστικό της (ID).

Relational Schema



Στους περισσότερους πίνακες υπάρχει το attribute *file_name* ως Foreign Key (NOT NULL), το οποίο διασφαλίζει πως κάθε αντικείμενο της βάσης μας έχει δική του φωτογραφία.

Παρακάτω ακολουθούν οι πίνακες με τα attributes και τα type – constraints των attributes:

Recipes

Πίνακας που χρησιμοποιήσαμε για τις συνταγές, έχει τα ακόλουθα attributes:

| <i>Attribute</i> | <i>Type</i> | <i>Constraints</i> |
|------------------|-------------|---------------------------------------|
| REC_ID | INT | Primary Key, NOT NULL, AUTO INCREMENT |
| Name | VARCHAR 100 | NOT NULL |
| Pastry | TINYINT | NOT NULL DEFAULT 0 |
| Difficulty | INT | NOT NULL, Difficulty < 6 |
| Description | VARCHAR 100 | - |
| Tip1 | VARCHAR 150 | - |
| Tip2 | VARCHAR 150 | - |
| Tip3 | VARCHAR 150 | - |
| Total Time | INT | Total time = Prep Time + Exec Time |
| Prep Time | INT | - |
| Exec Time | INT | - |
| Characterization | VARCHAR20 | - |
| PRIM_ING_ID | INT | Foreign Key, NOT NULL |
| CU_ID | INT | Foreign Key, NOT NULL |
| THEME_ID | INT | Foreign Key, NOT NULL |
| file_name | VARCHAR 45 | Foreign Key, NOT NULL |

Το Primary Key μας, έχει το attribute AUTO INCREMENT, ούτως ώστε κάθε συνταγή να έχει μοναδικό ID.

Το Pastry που χρησιμοποιείται για να κρίνουμε αν η συνταγή αφορά αλμυρό ή γλυκό γεύμα είναι τύπου TINYINT, που σημαίνει ότι μπορεί να πάρει τιμές από -128 έως 127. Χρησιμοποιείται το 0 ως FALSE και το 1 ως TRUE.

Το Difficulty παίρνει τιμές από 1 έως και 5.

Ingredients

Πίνακας που χρησιμοποιήσαμε για τα υλικά, έχει τα ακόλουθα attributes:

| <i>Attribute</i> | <i>Type</i> | <i>Constraints</i> |
|------------------|-------------|---------------------------------------|
| ING_ID | INT | Primary Key, NOT NULL, AUTO INCREMENT |
| Name | VARCHAR 50 | NOT NULL |
| Kcal per 100 | INT | NOT NULL |
| FG_ID | INT | NOT NULL |
| file_name | VARCHAR 45 | Foreign Key, NOT NULL |

Το Primary Key μας, έχει το attribute AUTO INCREMENT, ούτως ώστε κάθε υλικό να έχει μοναδικό ID.

Steps

Πίνακας που χρησιμοποιήσαμε για τα βήματα της εκάστοτε συνταγής, έχει τα ακόλουθα attributes:

| <i>Attribute</i> | <i>Type</i> | <i>Constraints</i> |
|------------------|-------------|------------------------------------|
| REC_ID | INT | Primary Key, Foreign Key, NOT NULL |
| Count | INT | Primary Key, NOT NULL |
| Step | VARCHAR 200 | - |

Τα REC_ID και Count αποτελούν το Composite Primary Key που λειτουργεί ως αναγνωριστικό για κάθε βήμα που υπάρχει σε κάθε συνταγή και για αυτό είναι και τα δύο NOT NULL. Για παράδειγμα, αν σε δύο συνταγές υπάρχει ακριβώς το ίδιο βήμα, τότε αυτό θα έχει σε κάθε συνταγή διαφορετικό REC_ID οπότε διαφορετικό ID (ακόμα και αν τύχει να έχουν το ίδιο Count, δηλαδή να είναι και τα δύο το ν-οστό βήμα της συνταγής τους).

Food Groups

Πίνακας που χρησιμοποιήσαμε για τις ομάδες τροφίμων, έχει τα ακόλουθα attributes:

| <i>Attribute</i> | <i>Type</i> | <i>Constraints</i> |
|------------------|-------------|---------------------------------------|
| FG_ID | INT | Primary Key, NOT NULL, AUTO INCREMENT |
| Name | VARCHAR 30 | NOT NULL |
| Description | VARCHAR 50 | - |
| file_name | VARCHAR 45 | Foreign Key, NOT NULL |

Το Primary Key μας, έχει το attribute AUTO INCREMENT, ούτως ώστε κάθε ομάδα τροφίμων να έχει μοναδικό ID.

Nutrition Info

Πίνακας που χρησιμοποιήσαμε για τις διατροφικές πληροφορίες, έχει τα ακόλουθα attributes:

| <i>Attribute</i> | <i>Type</i> | <i>Περιγραφή</i> |
|---------------------|-------------|------------------------------------|
| REC_ID | INT | Primary Key, Foreign Key, NOT NULL |
| Fat per serving | INT | DEFAULT 0 |
| Protein per serving | INT | DEFAULT 0 |
| Carbs per serving | INT | DEFAULT 0 |
| Kcal per serving | INT | DEFAULT 0 |

Το Primary Key μας είναι παράλληλα και Foreign Key (προερχόμενο από τον πίνακα Recipes). Είναι μοναδικό για τις διατροφικές πληροφορίες της κάθε συνταγής δεδομένου ότι το REC_ID είναι μοναδικό για κάθε συνταγή.

Ethnic Cuisine

Πίνακας που χρησιμοποιήσαμε για τις εθνικές κουζίνες, έχει τα ακόλουθα attributes:

| <i>Attribute</i> | <i>Type</i> | <i>Constraints</i> |
|------------------|-------------|---------------------------------------|
| CU ID | INT | Primary Key, NOT NULL, AUTO INCREMENT |
| Name | VARCHAR 30 | NOT NULL, UNIQUE |
| file_name | VARCHAR 45 | Foreign Key, NOT NULL |

Το Primary Key μας, έχει το attribute AUTO INCREMENT, ούτως ώστε κάθε εθνική κουζίνα να έχει μοναδικό ID.

Το Name είναι Unique για να έχει κάθε εθνική κουζίνα διαφορετικό όνομα.

Type

Πίνακας που χρησιμοποιήσαμε για τους τύπους γευμάτων, έχει τα ακόλουθα attributes:

| <i>Attribute</i> | <i>Type</i> | <i>Constraints</i> |
|------------------|-------------|---------------------------------------|
| TY ID | INT | Primary Key, NOT NULL, AUTO INCREMENT |
| Meal type | VARCHAR 20 | NOT NULL |
| file_name | VARCHAR 45 | Foreign Key, NOT NULL |

Το Primary Key μας, έχει το attribute AUTO INCREMENT, ούτως ώστε κάθε τύπος γεύματος να έχει μοναδικό ID.

Label

Πίνακας που χρησιμοποιήσαμε για τις ετικέτες, έχει τα ακόλουθα attributes:

| <i>Attribute</i> | <i>Type</i> | <i>Constraints</i> |
|------------------|-------------|---------------------------------------|
| LA ID | INT | Primary Key, NOT NULL, AUTO INCREMENT |
| Label Name | VARCHAR 20 | NOT NULL |
| Label Info | VARCHAR 45 | DEFAULT NULL |
| file_name | VARCHAR 45 | Foreign Key, NOT NULL |

Το Primary Key μας, έχει το attribute AUTO INCREMENT, ούτως ώστε κάθε ετικέτα να έχει μοναδικό ID.

Gear

Πίνακας που χρησιμοποιήσαμε για τον εξοπλισμό, έχει τα ακόλουθα attributes:

| <i>Attribute</i> | <i>Type</i> | <i>Constraints</i> |
|------------------|-------------|---------------------------------------|
| GR ID | INT | Primary Key, NOT NULL, AUTO INCREMENT |
| Name | VARCHAR 100 | NOT NULL |
| Instructions | VARCHAR 500 | DEFAULT NULL |
| file_name | VARCHAR 45 | Foreign Key, NOT NULL |

Το Primary Key μας, έχει το attribute AUTO INCREMENT, ούτως ώστε κάθε εξοπλισμός να έχει μοναδικό ID.

Chefs

Πίνακας που χρησιμοποιήσαμε για τους μάγειρες, έχει τα ακόλουθα attributes:

| <i>Attribute</i> | <i>Type</i> | <i>Constraints</i> |
|------------------|-------------|---------------------------------------|
| CH_ID | INT | Primary Key, NOT NULL, AUTO INCREMENT |
| First Name | VARCHAR 15 | NOT NULL |
| Last Name | VARCHAR 20 | NOT NULL |
| Phone | INT | NOT NULL |
| Birthday | DATE | NOT NULL |
| Age | INT | Age >17 |
| Experience | INT | Experience <Age - 18 |
| Class | INT | Class <6 |
| file_name | VARCHAR 45 | Foreign Key, NOT NULL |

Το Primary Key μας, έχει το attribute AUTO INCREMENT, ούτως ώστε κάθε μάγειρας να έχει μοναδικό ID.

Episodes

Πίνακας που χρησιμοποιήσαμε για τα επεισόδια, έχει τα ακόλουθα attributes:

| <i>Attribute</i> | <i>Type</i> | <i>Constraints</i> |
|------------------|-------------|---------------------------------------|
| EP_ID | INT | Primary Key, NOT NULL, AUTO INCREMENT |
| Season | INT | NOT NULL |
| EP Number | INT | NOT NULL |
| Winner | VARCHAR 36 | - |
| file_name | VARCHAR 45 | Foreign Key, NOT NULL |

Το Primary Key μας, έχει το attribute AUTO INCREMENT, ούτως ώστε κάθε επεισόδιο να έχει μοναδικό ID.

Theme

Πίνακας που χρησιμοποιήσαμε για τις ομάδες τροφίμων, έχει τα ακόλουθα attributes:

| <i>Attribute</i> | <i>Type</i> | <i>Constraints</i> |
|------------------|-------------|---------------------------------------|
| THEME_ID | INT | Primary Key, NOT NULL, AUTO INCREMENT |
| Name | VARCHAR 30 | NOT NULL, UNIQUE |
| Description | VARCHAR 500 | - |
| file_name | VARCHAR 45 | Foreign Key, NOT NULL |

Το Primary Key μας, έχει το attribute AUTO INCREMENT, ούτως ώστε κάθε θέμα να έχει μοναδικό ID.

Το Name είναι Unique για να έχει θέμα διαφορετικό όνομα.

Photo

Πίνακας που χρησιμοποιήσαμε για τις ετικέτες, έχει τα ακόλουθα attributes:

| <i>Attribute</i> | <i>Type</i> | <i>Constraints</i> |
|------------------|-------------|-----------------------|
| file_name | VARCHAR 45 | Primary Key, NOT NULL |
| Description | VARCHAR 50 | - |

Εν προκειμένω, το file_name είναι Primary Key.

Τώρα, προχωράμε στους πίνακες του relational που αποτελούσαν σχέσεις N:M μεταξύ δύο οντοτήτων στο ER και γι'αυτο δεν έχουν ήδη καλυφθεί παραπάνω:

Οι πίνακες Rec_Label, Rec_Type, EP_CU, REC_GEAR, Rec_Chef, Chef_CU, JUDGES, περιέχουν μονάχα ως Foreign και ταυτόχρονα Primary Keys τα IDs των πινάκων που συσχετίζουν. Είναι όλα τύπου INT και έχουν το constraint NOT NULL. Πιο συγκεκριμένα:

| <i>Πίνακας</i> | <i>Attributes</i> | <i>Περιγραφή</i> |
|----------------|-------------------|-----------------------------------|
| Rec_Label | LA_ID, REC_ID | Σχέση συνταγής – ετικέτας |
| Rec_Type | REC_ID, TY_ID | Σχέση συνταγής – τύπου γεύματος |
| EP_CU | EP_ID, CU_ID | Σχέση συνταγής – εθνικής κουζίνας |
| REC_GEAR | REC_ID, GR_ID | Σχέση συνταγής – εξοπλισμού |
| Rec_Chef | CH_ID, REC_ID | Σχέση συνταγής – μάγειρα |
| Chef_CU | CH_ID, CU_ID | Σχέση μάγειρα – εθνικής κουζίνας |

Επιπλέον έχουμε τους εξής δύο πίνακες:

PLAYS

Πίνακας που χρησιμοποιήσαμε για τη σχέση μάγειρα (παίχτη) – επεισοδίου, έχει τα ακόλουθα attributes:

| <i>Attribute</i> | <i>Type</i> | <i>Constraints</i> |
|------------------|-------------|------------------------------------|
| CH_ID | INT | Primary Key, Foreign Key, NOT NULL |
| EP_ID | INT | Primary Key, Foreign Key, NOT NULL |
| Grade | INT | Grade = Score1 + Score2 + Score3 |
| Cuisine | INT | Foreign Key, NOT NULL |
| recipe | INT | Foreign Key, NOT NULL |
| Score1 | INT | Score 1<6 |
| Score2 | INT | Score2 <6 |
| Score3 | INT | Score3<6 |

Το Grade υπάρχει για να αποθηκεύεται η βαθμολογία του κάθε μάγειρα που διαγωνίστηκε στο αντίστοιχο επεισόδιο. Παίρνει τιμή από το 0 έως και το 15 (άθροισμα βαθμολογίας τριών κριτών).

Τα Score1, Score2, Score3 παίρνουν τιμές από 0 έως 5.

JUDGES

Πίνακας που χρησιμοποιήσαμε για τη σχέση μάγειρα (κριτή) – επεισοδίου, έχει τα ακόλουθα attributes:

| <i>Attribute</i> | <i>Type</i> | <i>Constraints</i> |
|------------------|-------------|------------------------------------|
| CH_ID | INT | Primary Key, Foreign Key, NOT NULL |
| EP_ID | INT | Primary Key, Foreign Key, NOT NULL |
| Count | INT | Count < 4 |

REC_INGR

Πίνακας που χρησιμοποιήσαμε για τη σχέση συνταγής – υλικών, έχει τα ακόλουθα attributes:

| <i>Attribute</i> | <i>Type</i> | <i>Constraints</i> |
|--------------------|-------------|------------------------------------|
| REC_ID | INT | Primary Key, Foreign Key, NOT NULL |
| ING_ID | INT | Primary Key, Foreign Key, NOT NULL |
| Precise Quantity | INT | DEFAULT 0 |
| Imprecise Quantity | VARCHAR 20 | DEFAULT NULL |

Τα Precise_Quantity και Imprecise_Quantity υπάρχουν για να αποθηκεύονται η ακριβής και η μη ακριβής δοσολογία ενός υλικού σε μια συνταγή αντιστοίχως.

Indexes

Όλα τα indexes με το όνομα *PRIMARY* δημιουργούνται αυτόματα και αντιστοιχούν στο Primary Key του εκάστοτε πίνακα. Κάνουν ευκολότερη την αναζήτηση που χρειάζεται το ID του εκάστοτε πίνακα. Στους παρακάτω πίνακες, ο *Λόγος Ύπαρξης* τους θα σημειώνεται με «Εξηγήθηκε».

Σχεδόν σε όλους τους πίνακες υπάρχουν indexes για τα Foreign Keys. Από το όνομα του index φανερώνεται ποιους πίνακες συσχετίζει το αντίστοιχο Foreign Key. Συντελούν στην ευκολότερη αναζήτηση που χρειάζονται τα αντίστοιχα Foreign Keys. Στους παρακάτω πίνακες, ο *Λόγος Ύπαρξης* τους θα σημειώνεται με «Εξηγήθηκε».

Παρακάτω φαίνονται τα indexes όλων των πινάκων, καθώς και αν είναι Unique και Visible:

Recipes

| <i>Index</i> | <i>Unique</i> | <i>Visible</i> | <i>Λόγος Ύπαρξης</i> |
|------------------|---------------|----------------|--|
| PRIMARY | ✓ | ✓ | Εξηγήθηκε |
| test_idx | ✗ | ✓ | Εξηγήθηκε |
| Primary ingr_idx | ✗ | ✓ | Εξηγήθηκε |
| Rec Theme_idx | ✗ | ✓ | Εξηγήθηκε |
| photo_idx | ✗ | ✓ | Εξηγήθηκε |
| idx_recipe_name | ✗ | ✓ | Ευκολότερη εύρεση του ονόματος μιας συνταγής |

* Το test_idx χρησιμοποιείται για τις εθνικές κουζίνες

Ingredients

| <i>Index</i> | <i>Unique</i> | <i>Visible</i> | <i>Λόγος Ύπαρξης</i> |
|---------------|---------------|----------------|--|
| PRIMARY | ✓ | ✓ | Εξηγήθηκε |
| ing_photo_idx | ✗ | ✓ | Εξηγήθηκε |
| ing_fg | ✗ | ✓ | Εξηγήθηκε |
| idx_ingr_name | ✗ | ✓ | Ευκολότερη εύρεση του ονόματος ενός υλικού |

Steps

| <i>Index</i> | <i>Unique</i> | <i>Visible</i> | <i>Λόγος Ύπαρξης</i> |
|--------------|---------------|----------------|----------------------|
| PRIMARY | ✓ | ✓ | Εξηγήθηκε |

Food Groups

| <i>Index</i> | <i>Unique</i> | <i>Visible</i> | <i>Λόγος Ύπαρξης</i> |
|--------------|---------------|----------------|---|
| PRIMARY | ✓ | ✓ | Εξηγήθηκε |
| fg_photo_idx | ✓ | ✓ | Εξηγήθηκε |
| idx_fd_name | ✗ | ✓ | Ευκολότερη εύρεση του ονόματος μιας ομάδας τροφίμων |

Nutrition Info

| <i>Index</i> | <i>Unique</i> | <i>Visible</i> | <i>Λόγος Ύπαρξης</i> |
|--------------|---------------|----------------|----------------------|
| PRIMARY | ✓ | ✓ | Εξηγήθηκε |

Ethnic Cuisine

| <i>Index</i> | <i>Unique</i> | <i>Visible</i> | <i>Λόγος Ύπαρξης</i> |
|--------------|---------------|----------------|--|
| PRIMARY | ✓ | ✓ | Εξηγήθηκε |
| Name UNIQUE | ✓ | ✓ | Ευκολότερη εύρεση του ονόματος μιας εθνικής κουζίνας |
| cu_photo_idx | ✗ | ✓ | Εξηγήθηκε |

Type

| <i>Index</i> | <i>Unique</i> | <i>Visible</i> | <i>Λόγος Ύπαρξης</i> |
|----------------|---------------|----------------|--|
| PRIMARY | ✓ | ✓ | Εξηγήθηκε |
| type_photo_idx | ✗ | ✓ | Εξηγήθηκε |
| idx_type | ✗ | ✓ | Ευκολότερη εύρεση του ονόματος ενός τύπου γεύματος |

Label

| <i>Index</i> | <i>Unique</i> | <i>Visible</i> | <i>Λόγος Ύπαρξης</i> |
|---------------|---------------|----------------|--|
| PRIMARY | ✓ | ✓ | Εξηγήθηκε |
| file_name_idx | ✗ | ✓ | Εξηγήθηκε |
| idx_label | ✗ | ✓ | Ευκολότερη εύρεση του ονόματος μιας ετικέτας |

Gear

| <i>Index</i> | <i>Unique</i> | <i>Visible</i> | <i>Λόγος Ύπαρξης</i> |
|----------------|---------------|----------------|--|
| PRIMARY | ✓ | ✓ | Εξηγήθηκε |
| gear_photo_idx | ✗ | ✓ | Εξηγήθηκε |
| idx_gear | ✗ | ✓ | Ευκολότερη εύρεση του ονόματος ενός εξοπλισμού |

Chefs

| <i>Index</i> | <i>Unique</i> | <i>Visible</i> | <i>Λόγος Ύπαρξης</i> |
|----------------|---------------|----------------|---|
| PRIMARY | ✓ | ✓ | Εξηγήθηκε |
| chef_photo_idx | ✗ | ✓ | Εξηγήθηκε |
| idx_fullname | ✗ | ✓ | Ευκολότερη εύρεση του ονόματος ενός μάγειρα |

Episodes

| <i>Index</i> | <i>Unique</i> | <i>Visible</i> | <i>Λόγος Ύπαρξης</i> |
|---------------|---------------|----------------|--|
| PRIMARY | ✓ | ✓ | Εξηγήθηκε |
| file_name_idx | ✗ | ✓ | Εξηγήθηκε |
| idx_winner | ✗ | ✓ | Ευκολότερη εύρεση του νικητή ενός επεισοδίου |

Theme

| <i>Index</i> | <i>Unique</i> | <i>Visible</i> | <i>Λόγος Ύπαρξης</i> |
|---------------|---------------|----------------|---|
| PRIMARY | ✓ | ✓ | Εξηγήθηκε |
| Name UNIQUE | ✓ | ✓ | Ευκολότερη εύρεση του ονόματος ενός θέματος |
| file_name_idx | ✗ | ✓ | Εξηγήθηκε |

Photo

| <i>Index</i> | <i>Unique</i> | <i>Visible</i> | <i>Λόγος Ύπαρξης</i> |
|--------------|---------------|----------------|----------------------|
| PRIMARY | ✓ | ✓ | Εξηγήθηκε |

chef_cu

| <i>Index</i> | <i>Unique</i> | <i>Visible</i> | <i>Λόγος Ύπαρξης</i> |
|--------------|---------------|----------------|----------------------|
| PRIMARY | ✓ | ✓ | Εξηγήθηκε |
| CU_idx | ✗ | ✓ | Εξηγήθηκε |

ep_cu

| <i>Index</i> | <i>Unique</i> | <i>Visible</i> | <i>Λόγος Ύπαρξης</i> |
|--------------|---------------|----------------|----------------------|
| PRIMARY | ✓ | ✓ | Εξηγήθηκε |
| CU_idx | ✗ | ✓ | Εξηγήθηκε |

judges

| <i>Index</i> | <i>Unique</i> | <i>Visible</i> | <i>Λόγος Ύπαρξης</i> |
|--------------|---------------|----------------|----------------------|
| PRIMARY | ✓ | ✓ | Εξηγήθηκε |
| EP_idx | ✗ | ✓ | Εξηγήθηκε |

plays

| <i>Index</i> | <i>Unique</i> | <i>Visible</i> | <i>Λόγος Ύπαρξης</i> |
|--------------|---------------|----------------|---|
| PRIMARY | ✓ | ✓ | Εξηγήθηκε |
| EP_idx | ✗ | ✓ | Εξηγήθηκε |
| idx_ep_rec | ✗ | ✓ | Ευκολότερη εύρεση των συνταγών που έπαιξαν σε ένα επεισόδιο |

rec_chef

| <i>Index</i> | <i>Unique</i> | <i>Visible</i> | <i>Λόγος Ύπαρξης</i> |
|--------------|---------------|----------------|----------------------|
| PRIMARY | ✓ | ✓ | Εξηγήθηκε |
| REC_idx | ✗ | ✓ | Εξηγήθηκε |

rec_gear

| <i>Index</i> | <i>Unique</i> | <i>Visible</i> | <i>Λόγος Ύπαρξης</i> |
|--------------|---------------|----------------|----------------------|
| PRIMARY | ✓ | ✓ | Εξηγήθηκε |
| GR_idx | ✗ | ✓ | Εξηγήθηκε |

rec_ingr

| <i>Index</i> | <i>Unique</i> | <i>Visible</i> | <i>Λόγος Ύπαρξης</i> |
|--------------|---------------|----------------|----------------------|
| PRIMARY | ✓ | ✓ | Εξηγήθηκε |
| ING_idx | ✗ | ✓ | Εξηγήθηκε |

rec_label

| <i>Index</i> | <i>Unique</i> | <i>Visible</i> | <i>Λόγος Ύπαρξης</i> |
|--------------|---------------|----------------|----------------------|
| PRIMARY | ✓ | ✓ | Εξηγήθηκε |
| REC_idx | ✗ | ✓ | Εξηγήθηκε |

rec_type

| <i>Index</i> | <i>Unique</i> | <i>Visible</i> | <i>Λόγος Ύπαρξης</i> |
|--------------|---------------|----------------|----------------------|
| PRIMARY | ✓ | ✓ | Εξηγήθηκε |
| TY_idx | ✗ | ✓ | Εξηγήθηκε |

Mock Data Creation

Τα δεδομένα τα οποία φορτώσαμε στην βάση δεδομένων μας δημιουργήθηκαν μεσω python notebook, οπου έγινε χρήση της βιβλιοθήκης Faker, και της βιβλιοθήκης Random

Με την Faker δημιουργήσαμε όλα τα δεδομένα για τις προσωπικές πληροφορίες των chef, όλα τα description, tips συνταγών και όλα τα filenames των φωτογραφιών, καθώς δεν βρήκαμε πραγματικές φωτογραφίες για να φορτώσουμε στην βάση μας.

Επιπλέον, χρησιμοποιήσαμε το ChatGPT για να δημιουργήσουμε λίστες με ονόματα συνταγών, food groups, national cuisines, εξοπλισμου, μορφές, τύπους, ετικέτες και θεματικές ενότητες συνταγών για να ονομάσουμε τις αντίστοιχες οντότητες

Τέλος με την βιβλιοθήκη random, αναθέσαμε τυχαία chef_ids σε recipe_ids και όλους τους άλλους πίνακες της βάσης μας που απαιτείται να έχουν δεδομένα πριν την δημιουργία των επεισοδίων.

Για τον κάθε πίνακα δημιουργήθηκε ένα csv αρχείο το οποίο έπειτα φορτώσαμε στην βάση

DDL, DML, role and user, query Scripts

Τα αρχεία κώδικα sql που χρησιμοποιούνται για την κατασκευή και ανάπτυξη του database βρίσκονται στο παρακάτω github repository:

[Panagiotis-Magos/Cooking_Show \(github.com\)](#)

Πιο συγκεκριμένα υπάρχουν:

- Το αρχείο DDL για κατασκευή των πινάκων, indexes, precedures και triggers ονομάζεται ‘full structure.sql’
- Το αρχείο DML για να συμπληρωθεί με mock data η βάση δεδομένων ονομάζεται ‘full insertions.sql’
- Το αρχείο για δημιουργία χρηστών ονομάζεται ‘roles and users.sql’
- Το αρχείο με τα ζητούμενα queries ονομάζεται ‘queries_for_database.sql’

* Στο παραπάνω repository στο αρχείο README βρίσκονται και τα ER Diagram, Relational Schema σε καλύτερη ανάλυση, καθώς και η παρούσα αναφορά

Οδηγίες Εγκατάστασης της Εφαρμογής

Αρχικά ο χρήστης πρέπει να κατεβάσει κάποιο εργαλείο DBMS και ενδείκνυται και κάποιον graphical client. Η εγκατάσταση αυτών είναι εκτός του πεδίου αυτής της εργασίας και ο χρήστης παραπέμπεται στις σχετικές ιστοσελίδες των συγκεκριμένων εργαλείων. Προτείνεται για αυτόν τον οδηγό η εγκατάσταση MySQL Workbench.

Μετά την εγκατάσταση του εργαλείου, για την δημιουργία της βάσης πρέπει να τρέξουν τα παρακάτω αρχεία αυστηρά με αυτή την σειρά:

1. Το DDL αρχείο ‘full insertions.sql’, που ο χρήστης θα βρει παραπάνω. Το αρχείο αυτό θα δημιουργήσει μία βάση δεδομένων με όνομα “cooking_contestdb”, διαγράφοντας αν υπάρχει μια παλιά με το ίδιο όνομα.
 - 1.1. Όλα τα υπόλοιπα παραδοτέα αρχεία εκτός από το ‘queries_for_database.sql’ χρησιμοποιούν αυτόματα αυτή την βάση για να διευκολύνουν την εγκατάσταση. Τα

- queries όμως πρέπει να τα τρέξει ο χρήστης εντός της βάσης (πχ να την επιλέξει με διπλό κλικ αν δουλεύει στο MySQL Workbench και να φορτώσει το αρχείο ως query).
2. Το DML αρχείο ‘full insertions.sql’ ή οποιοδήποτε άλλο αρχείο με δεδομένα που ο χρήστης θέλει να φορτώσει στην βάση.
 - 2.1. Σημείωση: Προκειμένου να τηρηθούν όλα τα constraints σχετικά με τα foreign keys της βάσης, προτείνεται έντονα η εισαγωγή δεδομένων με την ίδια σειρά με αυτή στο ‘full insertions.sql’.
 - 2.2. Επιπρόσθετα μετά από εισαγωγή νέου επεισοδίου, απαιτείται να τρέξει ο χρήστης μία φορά το query “CALL set_all_winners()”, προκειμένου να φορτωθούν σωστά οι νικητές στα νέα επεισόδια (το query αυτό μπορεί απλά να προστεθεί κάτω από το “INSERT INTO episodes ...” στο νέο DML αρχείο). Σε περίπτωση που προσθέσει μόνο ένα επεισόδιο, μπορεί απλά να τρέξει το query “CALL get_winner(EP_ID)”, για το νέο EP_ID που πρόσθεσε.
 3. Το αρχείο ‘roles and users.sql’ για την δημιουργία ρόλων χρήστη. Το αρχείο παράγει ενδεικτικά έναν χρήστη ‘admin_user’ με password ‘bored admin’ με εξουσιοδότηση διαχειριστή, και δύο χρήστες ‘chef3’ και ‘chef2’ με password ‘password1’ και ‘password2’ αντίστοιχα, με εξουσιοδότηση μάγαιρα.
 - 3.1. Σε περίπτωση που ο διαχειριστής το επιθυμεί, αφού έχει εισαγάγει όλους τους μάγειρες στην βάση δεδομένων, μπορεί να δημιουργήσει έναν λογαριασμό για κάθε μάγαιρα εύκολα τρέχοντας το query “CALL set_all_chefs()”. Ο κάθε chef έχει μετά την εκτέλεση της παραπάνω εντολής έναν λογαριασμό με όνομα “ChefX” και κωδικό “passwordX” όπου X ο αριθμός μητρώου του μάγαιρα (CH_ID). Σε περίπτωση που προσθέσει μόνο έναν μάγαιρα, μπορεί απλά να τρέξει το query “CALL add_chef(CH_ID)”, για το νέο CH_ID που πρόσθεσε.
 - 3.2. Όλοι οι παραγόμενοι χρήστες στα πλαίσια της άσκησης είναι στο “@localhost”
 - 3.3. Αν ο διαχειριστής επιλέξει να δημιουργήσει νέο χρήστη μάγαιρα χωρίς να χρησιμοποιήσει τις built in διεργασίες που του προσφέρουμε, πρέπει να προσθέσει τον νέο μάγαιρα στο τραπέζι “user_chefs_mapping” (καλώντας ένα query “INSERT INTO user_chefs_mapping (mysql_user, ch_id) VALUES (username, CH_ID);”), αφού όλες οι άδειες κάθε μάγαιρα συνδέονται με τον αριθμό μητρώου του και συνεπώς ένας χρήστης μάγαιρας χωρίς αριθμό μητρώου δεν μπορεί να κάνει τίποτα.
 4. Το αρχείο ‘queries_for_database.sql’ που περιέχει τα queries για τα ερωτήματα της εργασίας, ή οποιοδήποτε άλλο αρχείο ή query για να εξερευνήσει/τροποποιήσει το database.

Οδηγίες Χρήσης της Εφαρμογής

Οδηγίες Χρήσης για Διαχειριστή

Ο διαχειριστής έχει πλήρη έλεγχο της βάσης δεδομένων (ALL PRIVILEGES), και μπορεί να την χειριστεί ανάλογα και να την τροποποιήσει όπως επιθυμεί με queries SQL, γνώση της οποίας υποθέτουμε ότι ήδη κατέχει.

Οδηγίες Χρήσης για Μάγαιρα

Οι μάγειρες έχουν περιορισμένο έλεγχο στη βάση και μπορούν να δουν και να αλλάξουν αποκλειστικά στοιχεία που σχετίζονται με αυτούς. Προκειμένου να χρησιμοποιήσουν την βάση, καλούν procedures στο format “CALL procedure(procedure arguments)”. Πιο συγκεκριμένα, έχει στην διαθεσή του τις εντολές:

- call get_personal_info(); Η οποία επιστρέφει τα προσωπικά του στοιχεία

- call update_personal_info(first name, last name, phone, birthday, age, experience, class, filename); Η οποία ενημερώνει τα προσωπικά του στοιχεία (σημείωση: πρέπει όλα τα πεδία να είναι συμπληρωμένα, ακόμα και αυτά που δεν αλλάζουν)
- call get_episodes_played(); Η οποία επιστρέφει τα επεισόδια στα οποία έπαιξε ο μάγειρας
- call get_chefs_recipes(); Η οποία επιστρέφει τις συνταγές που έχει ο μάγειρας στο σύστημα
- call update_recipe(rec_id, difficulty, description, Tip1, Tip2, Tip3, total_time, prep_time, cook_time, portions, file_name, protein, carbs, fat); Η οποία του επιτρέπει να αλλάξει κάποια συνταγή που του έχει ανατατεθεί (το rec_id χρησιμοποιείται για λόγους ταυτοποίησης της συνταγής και δεν μεταβάλλεται, και ο μάγειρας μπορεί να το βρει από την λίστα συνταγών του καλώντας το get_chefs_recipes())
- call add_recipe(name, pastry, difficulty, description, Tip1, Tip2, Tip3, total_time, prep_time, cook_time, portions, primary_ing, cuisine, theme, file_name, protein, carbs, fat); Η οποία του επιτρέπει να προσθέσει μία καινούργια συνταγή, (αυτόματα ενημερώνεται και το τραπέζι rec_chef ότι γνωρίζει αυτή την συνταγή)
- call get_recipe_steps(rec_id); Η οποία επιστρέφει τα βήματα της ζητούμενης συνταγής (μόνο αν αυτή σχετίζεται με τον μάγειρα)
- call add_step(rec_id, step_count, step_desc); Η οποία επιτρέπει στον μάγειρα να προσθέσει ένα βήμα σε κάποια συνταγή που σχετίζεται με αυτόν
- call delete_recipe_step(rec_id, step_count); Η οποία του επιτρέπει να διαγράψει κάποιο βήμα σε κάποια συνταγή που σχετίζεται με αυτόν
- call get_recipe_ingredients(rec_id); Η οποία του επιστρέφει τα συστατικά της συνταγής που επιλέξει αν αυτή σχετίζεται με αυτόν
- call add_ingredient(rec_id,ing_id, precise_quantity, inprecise_quantity); Η οποία του επιτρέπει να προσθέσει ένα συστατικό σε συνταγή που σχετίζεται με αυτόν. Το πεδίο precise_quantity δέχεται αριθμό σε γραμμάρια της ποσότητας του υλικού που θα χρησιμοποιηθεί, ενώ το το πεδίο inprecise_quantity δέχεται ένα string που περιγράφει την ποσότητα του υλικού
- call delete_recipe_ingredient(rec_id,ing_id); Η οποία του επιτρέπει να αφαιρέσει ένα συστατικό από συνταγή που σχετίζεται με αυτόν.

Ο κώδικας όλων των procedures στα οποία έχει πρόσβαση ο μάγειρας βρίσκεται στο ‘roles and users.sql’.

Ζητούμενα Queries

Σημείωση: Ορισμένα δεδομένα της βάσης, όπως οι μάγειρες που παίζουν στα επεισόδια, οι βαθμολογίες που δίνουν οι κριτές, κλπ, παράγονται δυναμικά με τυχαίο τρόπο (τα συγκεκριμένα παραδείγματα όποτε εισάγεται επεισόδιο). Συνεπώς υπάρχει περίπτωση να παραχθούν διαφορετικά αποτελέσματα για ορισμένα queries όταν τα τρέξει κανείς τοπικά αφού έχει ξαναδημιουργήσει την βάση, σε σχέση με αυτά που παρουσιάζονται εδώ. Αυτό δεν οφείλεται σε σφάλμα των queries αλλά σε αποκλίσεις των δεδομένων που έχουν παραχθεί τυχαία.

1) Μέσος Όρος Αξιολογήσεων (σκορ) ανά μάγειρα και Εθνική κουζίνα.

```
-- question 3.1
-- average grade of each chef
SELECT
    CH_ID,
    AVG(Grade) AS mean_grade
FROM
    plays
GROUP BY
    CH_ID;

-- average grade of each cuisine
SELECT
    Cuisine,
    AVG(Grade) AS mean_grade
FROM
    plays
GROUP BY
    Cuisine;
```

| | CH_ID | mean_grade | | Cuisine | mean_grade |
|---|-------|------------|---|---------|------------|
| ▶ | 2 | 6.1250 | ▶ | 1 | 7.6296 |
| | 3 | 8.1538 | | 2 | 6.5926 |
| | 4 | 6.0000 | | 3 | 7.4815 |
| | 5 | 7.7727 | | 4 | 7.7200 |
| | 6 | 7.1250 | | 5 | 7.1714 |
| | 7 | 5.7692 | | 6 | 6.8696 |
| | 8 | 8.6667 | | 7 | 7.4000 |
| | 9 | 8.1500 | | 8 | 7.8276 |

2) Για δεδομένη Εθνική κουζίνα και έτος, ψάχνουμε ποιοι μάγειρες που έχουν την συγκεκριμένη εθνική κουζίνα συμμετείχαν σε επεισόδιο με συνταγή που ανήκει σε αυτή.

```
SELECT DISTINCT ep.Season,p.Cuisine,cu.Name,p.CH_ID,c.First_Name,c.Last_Name
FROM plays p
JOIN episodes ep ON p.EP_ID=ep.EP_ID
JOIN Chefs c ON c.CH_id=p.CH_id
JOIN ethnic_cuisine cu ON p.Cuisine=cu.CU_ID
ORDER BY ep.Season,p.Cuisine;
```

-- Second query is just which Chef belongs to each cuisine

```
SELECT cu.Name,c.First_Name, c.Last_Name
FROM chef_cu cc
JOIN chefs c ON cc.CH_ID=c.CH_ID
JOIN ethnic_cuisine cu ON cu.CU_ID=cc.CU_ID;
```

| | Season | Cuisine | Name | CH_ID | First_Name | Last_Name |
|---|--------|---------|---------|-------|------------|-----------|
| ▶ | 1 | 1 | Italian | 30 | Brandon | Lamb |
| | 1 | 1 | Italian | 2 | Kevin | Ware |
| | 1 | 1 | Italian | 11 | Kimberly | Schwartz |
| | 1 | 2 | Chinese | 15 | Andrew | Perez |
| | 1 | 2 | Chinese | 3 | Carolyn | Cox |
| | 1 | 2 | Chinese | 50 | Deborah | Patel |
| | 1 | 2 | Chinese | 16 | Travis | Myers |
| | 1 | 2 | Chinese | 36 | Willie | Hunter |
| | 1 | 3 | Mexican | 15 | Andrew | Perez |
| | 1 | 3 | Mexican | 41 | Kelly | Newton |

| | Name | First_Name | Last_Name |
|---|----------|------------|-----------|
| ▶ | American | Tyler | Murray |
| | American | Melissa | Garcia |
| | American | Kimberly | Schwartz |
| | American | Andrew | Perez |
| | American | Travis | Myers |
| | American | David | Mccann |
| | American | Justin | Holland |
| | American | Jacqueline | Hendrix |
| | American | Anthony | Sherman |
| | American | Daisy | Salazar |
| | American | Joseph | Hernandez |
| | American | Mary | Quinn |

3) Βρίσκουμε τους 5 μάγειρες που έχουν τις περισσότερες συνταγές και είναι μικρότεροι από 30 χρονών.

```
-- question 3.3
-- it doesnt specify how many chefs should i return so i said top 5
SELECT
    c.CH_ID,
    c.First_Name,
    c.Last_Name,
    c.Age,
    COUNT(r.REC_ID) AS Recipe_Count
FROM
    chefs c
JOIN
    rec_chef r ON c.CH_ID = r.CH_ID
WHERE
    c.Age < 30
GROUP BY
    c.CH_ID, c.First_Name, c.Last_Name, c.Age
ORDER BY
    Recipe_Count DESC
LIMIT 5;
```

| | CH_ID | First_Name | Last_Name | Age | Recipe_Count |
|---|-------|------------|-----------|-----|--------------|
| ▶ | 29 | Joseph | Hernandez | 21 | 63 |
| | 48 | Tricia | Floyd | 25 | 63 |
| | 31 | Mary | Quinn | 20 | 63 |
| | 20 | Justin | Holland | 26 | 63 |
| | 36 | Willie | Hunter | 28 | 63 |

4) Βρίσκουμε τους μάγειρες που δεν έχουν γίνει κριτές σε κανένα επεισόδιο.

```
-- question 3.4
SELECT
    c.CH_ID,
    c.First_Name,
    c.Last_Name
FROM
    chefs c
LEFT JOIN
    judges j ON c.CH_ID = j.CH_ID
WHERE
    j.CH_ID IS NULL;
```

| | CH_ID | First_Name | Last_Name |
|---|-------|------------|-----------|
| ▶ | 33 | Marvin | Davies |

5) Βρίσκουμε τους κριτές έχουν συμμετάσχει στον ίδιο αριθμό επεισοδίων σε διάστημα μίας σεζόν με περισσότερες από 2 εμφανίσεις. Το ζητούμενο ήταν περισσότερες από 3 εμφανίσεις αλλά καθώς δεν υπήρχαν κριτές που πληρούσαν τις προϋποθέσεις, μειώσαμε τον αριθμό των εμφανίσεων για να μην επιστρέψει κενό πίνακα το query.

```
-- question 3.5
-- i reduced the appearances to more than 2 instead of three to not get empty result

WITH judge_appearances AS (
    SELECT
        j.CH_ID,
        e.Season,
        COUNT(j.EP_ID) AS EpisodeCount
    FROM
        judges j
    JOIN
        episodes e ON j.EP_ID = e.EP_ID
    GROUP BY
        j.CH_ID, e.Season
)
SELECT
    j1.CH_ID AS Judge1_ID,
    j2.CH_ID AS Judge2_ID,
    j1.Season,
    j1.EpisodeCount
FROM
    judge_appearances j1
JOIN
    judge_appearances j2
    ON j1.Season = j2.Season
    AND j1.EpisodeCount = j2.EpisodeCount
    AND j1.CH_ID < j2.CH_ID
WHERE
    j1.EpisodeCount > 2
ORDER BY
    j1.Season, j1.EpisodeCount;
```

| | Judge1_ID | Judge2_ID | Season | EpisodeCount |
|---|-----------|-----------|--------|--------------|
| ▶ | 2 | 15 | 3 | 3 |
| | 18 | 26 | 3 | 3 |
| | 2 | 26 | 3 | 3 |
| | 15 | 26 | 3 | 3 |
| | 2 | 18 | 3 | 3 |
| | 15 | 18 | 3 | 3 |
| | 3 | 4 | 4 | 3 |
| | 22 | 27 | 4 | 3 |
| | 4 | 27 | 4 | 3 |
| | 3 | 27 | 4 | 3 |
| | 4 | 22 | 4 | 3 |
| | 3 | 22 | 4 | 3 |

6) Ανάμεσα σε ζεύγη labels που είναι κοινά στις συνταγές, βρίσκουμε τα top-3 ζεύγη που εμφανίστηκαν τις περισσότερες φορές σε επεισόδια.

Παρακάτω φαίνεται η εκδοχή του query χωρίς force index:

```
-- question 3.6
-- basic/optimised version
EXPLAIN
WITH label_pairs AS (
  SELECT
    r1.REC_ID,
    r1.LA_ID AS label1,
    r2.LA_ID AS label2
  FROM
    rec_label r1
  JOIN
    rec_label r2 ON r1.REC_ID = r2.REC_ID AND r1.LA_ID < r2.LA_ID
),

Pair_ep_count AS (
  SELECT
    lp.label1,
    lp.label2,
    COUNT(DISTINCT p.EP_ID) AS EpCount
  FROM
    label_pairs lp
  JOIN
    plays p ON lp.REC_ID = p.recipe
  GROUP BY
    lp.label1, lp.label2
)

SELECT
  pc.label1,
  pc.label2,
  pc.EpCount
FROM
  Pair_ep_count pc
ORDER BY
  pc.EpCount DESC
LIMIT 3;
```

Και εδώ φαίνεται η εκδοχή του query με force index:

```
-- version with force index

EXPLAIN
WITH label_pairs AS (
  SELECT
    r1.REC_ID,
    r1.LA_ID AS label1,
    r2.LA_ID AS label2
  FROM
    rec_label r1 FORCE INDEX (PRIMARY,REC_idx)
  JOIN
    rec_label r2 FORCE INDEX (PRIMARY,REC_idx)
    ON r1.REC_ID = r2.REC_ID AND r1.LA_ID < r2.LA_ID
),

Pair_ep_count AS (
  SELECT
    lp.label1,
    lp.label2,
    COUNT(DISTINCT p.EP_ID) AS EpCount
  FROM
    label_pairs lp

  JOIN
    plays p FORCE INDEX (idx_ep_rec)
    ON lp.REC_ID = p.recipe
  GROUP BY
    lp.label1, lp.label2
)

SELECT
  pc.label1,
  pc.label2,
  pc.EpCount
FROM
  Pair_ep_count pc
ORDER BY
  pc.EpCount DESC
LIMIT 3;
```

Και οι δύο εκδοχές έχουν το ίδιο αποτέλεσμα:

| | label1 | label2 | EpCount |
|---|--------|--------|---------|
| ► | 5 | 8 | 45 |
| | 7 | 9 | 38 |
| | 7 | 11 | 36 |

Με την εντολή EXPLAIN παίρνουμε τα traces των δύο εκδοχών.

Without force index:

| | id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
|---|----|-------------|------------|------------|-------|-----------------|------------|---------|-----------------------------|------|----------|--|
| ▶ | 1 | PRIMARY | <derived2> | NULL | ALL | NULL | NULL | NULL | NULL | 2514 | 100.00 | Using filesort |
| | 2 | DERIVED | r1 | NULL | index | PRIMARY_REC_idx | REC_idx | 4 | NULL | 328 | 100.00 | Using index; Using temporary; Using filesort |
| | 2 | DERIVED | r2 | NULL | ref | PRIMARY_REC_idx | REC_idx | 4 | cooking_contestdb.r1.REC_ID | 3 | 33.33 | Using where; Using index |
| | 2 | DERIVED | p | NULL | ref | idx_ep_rec | idx_ep_rec | 5 | cooking_contestdb.r1.REC_ID | 6 | 100.00 | Using index |

With force index:

| | id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
|---|----|-------------|------------|------------|-------|-----------------|------------|---------|-----------------------------|------|----------|--|
| ▶ | 1 | PRIMARY | <derived2> | NULL | ALL | NULL | NULL | NULL | NULL | 2514 | 100.00 | Using filesort |
| | 2 | DERIVED | r1 | NULL | index | PRIMARY_REC_idx | REC_idx | 4 | NULL | 328 | 100.00 | Using index; Using temporary; Using filesort |
| | 2 | DERIVED | r2 | NULL | ref | PRIMARY_REC_idx | REC_idx | 4 | cooking_contestdb.r1.REC_ID | 3 | 33.33 | Using where; Using index |
| | 2 | DERIVED | p | NULL | ref | idx_ep_rec | idx_ep_rec | 5 | cooking_contestdb.r1.REC_ID | 6 | 100.00 | Using index |

Στο εναλλακτικό query plan με force index, προσπαθούμε να καθοδηγήσουμε την αναζήτηση αναγκάζοντας το σύστημα να χρησιμοποιήσει τα indexes που του ορίζουμε εμείς. Στη συγκεκριμένη περίπτωση το αναγκάζουμε να χρησιμοποιήσει το PRIMARY (δηλαδή βασισμένο στο LABEL_ID και REC_ID) και REC_idx για τους πίνακες rec_label, και το index idx_ep_rec για το plays, το οποίο είναι το index που έχει δημιουργηθεί για τις συνταγές που ανατίθενται σε chef και μαγειρεύονται σε επεισόδια.

Παρατηρούμε ότι το σύστημα αυτόματα χρησιμοποιεί τα ίδια indexes, με αποτέλεσμα τα παραγόμενα traces να είναι ίδια, καθώς αυτός είναι ο βέλτιστος τρόπος να αναζητηθεί η βάση στο συγκεκριμένο query.

Δοκιμάζοντας να κάνουμε force index με διαφορετικά indexes (λόγου χάρη το EP_idx στο plays, το οποίο είναι βασισμένο στο EP_ID), παρατηρούμε ότι το παραγόμενο trace είναι διαφορετικό. Στο συγκεκριμένο παράδειγμα ας πούμε, το type της τελευταίας σειράς (table p) θα γινόταν ALL (αφού το index που του δίνουμε είναι άσχετο από την αναζήτηση που κάνει και αναγκάζεται να ψάξει όλον το πίνακα), ενώ τα rows που εξέταζε θα γίνονταν 600, επειδή θα εξέταζε όλον τον πίνακα plays. Η δική μας υλοποίηση με force index όμως είναι βέλτιστη.

7) Βρίσκουμε όλους τους μάγειρες που συμμετείχαν τουλάχιστον 5 λιγότερες φορές από τον μάγειρα με τις περισσότερες συμμετοχές σε επεισόδια.

```
-- question 3.7
WITH max1 AS (
  SELECT COUNT(p.EP_ID) AS EpisodeCount
  FROM chefs c
  JOIN plays p ON c.CH_ID=p.CH_ID
  GROUP BY
    p.CH_ID
  ORDER BY
    COUNT(p.EP_ID) DESC
  LIMIT 1
)

SELECT c.CH_ID, c.First_Name, c.Last_Name, COUNT(p.EP_ID) AS EpisodeCount
FROM chefs c
JOIN plays p ON c.CH_ID=p.CH_ID
GROUP BY c.CH_ID, c.First_Name, c.Last_Name
HAVING COUNT(p.EP_ID) <= (SELECT EpisodeCount FROM max1) - 5
ORDER BY EpisodeCount DESC;
```

| | CH_ID | First_Name | Last_Name | EpisodeCount |
|---|-------|------------|-----------|--------------|
| ▶ | 37 | Valerie | Ayers | 21 |
| | 26 | Michael | Gross | 19 |
| | 29 | Joseph | Hernandez | 18 |
| | 16 | Travis | Myers | 18 |
| | 49 | James | Keller | 18 |
| | 50 | Deborah | Patel | 17 |
| | 47 | Katie | Brown | 16 |
| | 14 | Joshua | Collins | 16 |
| | 4 | Deborah | Hernandez | 15 |
| | 19 | David | Mccann | 15 |
| | 15 | Andrew | Perez | 14 |
| | 9 | Leslie | Singleton | 14 |

8) Βρίσκουμε το επεισόδιο στο οποίο χρησιμοποιήθηκαν τα περισσότερα εξαρτήματα (εξοπλισμός):

```
-- question 3.8
-- without force index
EXPLAIN
WITH EpGearCount AS (
  SELECT
    p.EP_ID,
    COUNT(rg.GR_ID) AS GearCount
  FROM
    rec_gear rg
  JOIN
    plays p ON rg.REC_ID = p.recipe
  GROUP BY
    p.EP_ID
)
SELECT
  ec.EP_ID,
  ec.GearCount
FROM
  EpGearCount ec
ORDER BY
  ec.GearCount DESC
LIMIT 1;
```

```
-- with force index
EXPLAIN
WITH EpGearCount AS (
  SELECT
    p.EP_ID,
    COUNT(rg.GR_ID) AS GearCount
  FROM
    rec_gear rg FORCE INDEX (PRIMARY)
  JOIN
    plays p FORCE INDEX (idx_ep_rec) ON rg.REC_ID = p.recipe
  GROUP BY
    p.EP_ID
)
SELECT
  ec.EP_ID,
  ec.GearCount
FROM
  EpGearCount ec
ORDER BY
  ec.GearCount DESC
LIMIT 1;
```

Αποτέλεσμα:

| | EP_ID | GearCount |
|---|-------|-----------|
| ▶ | 32 | 59 |

Και traces:

Without force index:

| | id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
|---|----|-------------|------------|------------|-------|---------------------------|------------|---------|-----------------------------|------|----------|------------------------------|
| ▶ | 1 | PRIMARY | <derived2> | HULL | ALL | HULL | HULL | HULL | HULL | 2677 | 100.00 | Using filesort |
| | 2 | DERIVED | rg | HULL | index | PRIMARY | GR_idx | 4 | HULL | 415 | 100.00 | Using index; Using temporary |
| | 2 | DERIVED | p | HULL | ref | PRIMARY,EP_idx,idx_ep_rec | idx_ep_rec | 5 | cooking_contestdb.rg.REC_ID | 6 | 100.00 | Using index |

With force index:

| | id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
|---|----|-------------|------------|------------|-------|---------------|------------|---------|-----------------------------|------|----------|------------------------------|
| ▶ | 1 | PRIMARY | <derived2> | HULL | ALL | HULL | HULL | HULL | HULL | 2677 | 100.00 | Using filesort |
| | 2 | DERIVED | rg | HULL | index | PRIMARY | PRIMARY | 8 | HULL | 415 | 100.00 | Using index; Using temporary |
| | 2 | DERIVED | p | HULL | ref | idx_ep_rec | idx_ep_rec | 5 | cooking_contestdb.rg.REC_ID | 6 | 100.00 | Using index |

Αντίστοιχα με το ερώτημα 6, χρησιμοποιούμε force index για να αναγκάσουμε το σύστημα να χρησιμοποιήσει το PRIMARY index του rec_gear (το οποίο σχετίζεται με το GR_ID και το rec_id) και

το `idx_ep_rec` του `plays` (το οποίο σχετίζεται με τις συνταγές γιατί αναζητώντας μέσω αυτών θα βρούμε πόσος εξοπλισμός χρησιμοποιείται σε κάθε επεισόδιο).

Πάλι παρατηρούμε ότι οι αναζητήσεις `rows` του συστήματος είναι ίδιες καθώς και αυτό χρησιμοποιεί τα ίδια `indexes`, ωστόσο καθώς καθοδηγούμε το σύστημα στο να χρησιμοποιήσει απευθείας το `idx_ep_rec` του `plays`, το αντίστοιχο πεδίο `possible keys` έχει μόνο ένα (το `idx_ep_rec`) αντί να έχει όλα τα `indexes` του `plays`.

9) Φτιάχνουμε λίστα με μέσο όρο αριθμού γραμμάρων υδατανθράκων στο διαγωνισμό ανά έτος:

-- question 3.9

```
WITH carbs AS(
  SELECT DISTINCT e.Season,n.REC_ID, n.`Carbs per serving`
  FROM nutrition_info n
  JOIN plays p ON n.REC_ID = p.recipe
  JOIN episodes e ON e.EP_ID =p.EP_ID
  GROUP BY e.EP_ID, n.REC_ID
  ORDER BY e.Season
)

SELECT DISTINCT n.Season,AVG(n.`Carbs per serving`)
FROM carbs n
GROUP BY n.Season;
```

| | Season | AVG(n. `Carbs per serving`) |
|---|--------|-----------------------------|
| ▶ | 1 | 47.3065 |
| | 2 | 50.6102 |
| | 3 | 49.8571 |
| | 4 | 55.1000 |
| | 5 | 49.0500 |
| | 6 | 47.4576 |

10) Βρίσκουμε τις Εθνικές κουζίνες που έχουν τον ίδιο αριθμό συμμετοχών σε διαγωνισμούς, σε διάστημα δύο συνεχόμενων ετών, με τουλάχιστον τρεις συμμετοχές ετησίως:

```

266 WITH ParticipantCount AS (
267     SELECT
268         p.Cuisine AS CU_ID,
269         e.Season,
270         COUNT(DISTINCT p.CH_ID) AS ParticipantCount
271     FROM
272         plays p
273     JOIN
274         episodes e ON p.EP_ID = e.EP_ID
275     GROUP BY
276         p.Cuisine, e.Season
277     HAVING
278         COUNT(DISTINCT p.CH_ID) >= 3
279 ),
280 ConsecutiveSeasons AS (
281     SELECT
282         p1.CU_ID,
283         p1.Season AS S1,
284         p2.Season AS S2,
285         p1.ParticipantCount + p2.ParticipantCount AS TotalParticipants
286     FROM
287         ParticipantCount p1
288     JOIN
289         ParticipantCount p2 ON p1.CU_ID = p2.CU_ID AND p2.Season = p1.Season + 1
290 ),
291 MatchingCuisines AS (
292     SELECT
293         cs1.CU_ID AS CU_ID1,
294         cs2.CU_ID AS CU_ID2,
295         cs1.S1,
296         cs1.S2,
297         cs1.TotalParticipants
298     FROM
299         ConsecutiveSeasons cs1
300     JOIN
301         ConsecutiveSeasons cs2 ON cs1.TotalParticipants = cs2.TotalParticipants
302         AND cs1.CU_ID < cs2.CU_ID
303         AND cs1.S1 = cs2.S1
304         AND cs1.S2 = cs2.S2
305 )
306 SELECT
307     mc.CU_ID1,
308     mc.CU_ID2,
309     mc.S1,
310     mc.S2,
311     mc.TotalParticipants
312 FROM
313     MatchingCuisines mc
314 ORDER BY
315     mc.TotalParticipants DESC;

```

| | CU_ID1 | CU_ID2 | S1 | S2 | TotalParticipants |
|---|--------|--------|----|----|-------------------|
| ▶ | 7 | 10 | 3 | 4 | 12 |
| | 4 | 9 | 5 | 6 | 11 |
| | 4 | 20 | 5 | 6 | 11 |
| | 9 | 20 | 5 | 6 | 11 |
| | 2 | 15 | 2 | 3 | 11 |
| | 2 | 14 | 5 | 6 | 10 |
| | 6 | 14 | 2 | 3 | 10 |
| | 14 | 15 | 1 | 2 | 10 |
| | 2 | 10 | 1 | 2 | 10 |
| | 2 | 14 | 1 | 2 | 10 |
| | 2 | 15 | 1 | 2 | 10 |
| | 5 | 6 | 3 | 4 | 10 |

11) Βρίσκουμε τους 5 κριτές που έχουν δώσει συνολικά την υψηλότερη βαθμολογία σε ένα μάγειρα. (όνομα κριτή, όνομα μάγειρα και συνολικό σκορ βαθμολόγησης)

```
-- question3.11
WITH JudgeScores AS (
  SELECT
    j.CH_ID AS Judge_ID,
    j.EP_ID,
    p.CH_ID AS Chef_ID,
    p.Score1 AS Score,
    1 AS count
  FROM
    judges j
  JOIN
    plays p ON j.EP_ID = p.EP_ID AND j.count = 1
  UNION ALL
  SELECT
    j.CH_ID AS Judge_ID,
    j.EP_ID,
    p.CH_ID AS Chef_ID,
    p.Score2 AS Score,
    2 AS count
  FROM
    judges j
  JOIN
    plays p ON j.EP_ID = p.EP_ID AND j.count = 2
  UNION ALL
  SELECT
    j.CH_ID AS Judge_ID,
    j.EP_ID,
    p.CH_ID AS Chef_ID,
    p.Score3 AS Score,
    3 AS count
  FROM
    judges j
  JOIN
    plays p ON j.EP_ID = p.EP_ID AND j.count = 3
)
SELECT
  j.CH_ID AS Judge_ID,
  c.CH_ID AS Chef_ID,
  SUM(js.Score) AS TotalScore
FROM
  JudgeScores js
JOIN
  judges j ON js.Judge_ID = j.CH_ID
JOIN
  chefs c ON js.Chef_ID = c.CH_ID
GROUP BY
  j.CH_ID, c.CH_ID
ORDER BY
  TotalScore DESC
LIMIT 5;
```

| | Judge_ID | Chef_ID | TotalScore |
|---|----------|---------|------------|
| ▶ | 18 | 27 | 128 |
| | 15 | 29 | 126 |
| | 7 | 29 | 120 |
| | 4 | 14 | 119 |
| | 18 | 5 | 112 |

12) Βρίσκουμε το πιο τεχνικά δύσκολο, από πλευράς συνταγών, επεισόδιο του διαγωνισμού ανά έτος:

-- question 3.12

```
WITH ep_stats1 AS(
SELECT DISTINCT e.Season, e.EP_Number, r.REC_ID, r.Difficulty
FROM episodes e
JOIN plays p ON p.EP_ID=e.EP_ID
JOIN recipes r ON p.recipe=r.REC_ID
GROUP BY
p.EP_ID, r.REC_ID),

ep_stats2 AS(
SELECT DISTINCT e.Season, e.EP_Number, SUM(e.Difficulty) AS "Total Difficulty"
FROM ep_stats1 e
GROUP BY e.Season, e.EP_Number
HAVING SUM(e.Difficulty)= SUM(e.Difficulty)
ORDER BY e.Season, `Total Difficulty` DESC),

max_difficulty AS(
SELECT DISTINCT e.Season, MAX(e.`Total Difficulty`) AS "Highest Difficulty"
FROM ep_stats2 e
GROUP BY e.Season
)

SELECT m.Season, e.EP_Number, m.`Highest Difficulty`
FROM max_difficulty m JOIN ep_stats2 e ON e.Season=m.Season AND e.`Total Difficulty`= m.`Highest Difficulty`;
```

| | Season | EP_Number | Highest Difficulty |
|---|--------|-----------|--------------------|
| ▶ | 1 | 7 | 40 |
| | 2 | 3 | 38 |
| | 3 | 5 | 37 |
| | 4 | 7 | 41 |
| | 5 | 7 | 39 |
| | 6 | 1 | 39 |

13) Βρίσκουμε το επεισόδιο συγκέντρωσε τον χαμηλότερο συνολικά βαθμό επαγγελματικής κατάρτισης (κριτές και μάγειρες). Όμως καθώς εμείς έχουμε υποθέσει πως η υψηλότερη επαγγελματική κατάρτιση είναι αυτή με βαθμό 1 και η χαμηλότερη αυτή με βαθμό 5, αντί να βρούμε το min total count βρίσκουμε το max:

```
-- question 3.13
-- im doing max class count as 5 if the lowst and 1 the highest score

WITH PlayerClasses AS (
    SELECT
        e.EP_ID,
        SUM(c.Class) AS PlayerClassCount
    FROM
        episodes e
    LEFT JOIN
        plays p ON e.EP_ID = p.EP_ID
    LEFT JOIN
        chefs c ON p.CH_ID = c.CH_ID
    GROUP BY
        e.EP_ID
),
JudgeClasses AS (
    SELECT
        e.EP_ID,
        SUM(c.Class) AS JudgeClassCount
    FROM
        episodes e
    LEFT JOIN
        judges j ON e.EP_ID = j.EP_ID
    LEFT JOIN
        chefs c ON j.CH_ID = c.CH_ID
    GROUP BY
        e.EP_ID
),
TotalClasses AS (
    SELECT
        p.EP_ID,
        p.PlayerClassCount,
        j.JudgeClassCount,
        (p.PlayerClassCount + j.JudgeClassCount) AS TotalClassCount
    FROM
        PlayerClasses p
    JOIN
        JudgeClasses j ON p.EP_ID = j.EP_ID
)
SELECT
    t.EP_ID,
    t.TotalClassCount
FROM
    TotalClasses t
ORDER BY
    t.TotalClassCount DESC
LIMIT 1;
```

| | EP_ID | TotalClassCount |
|---|-------|-----------------|
| ▶ | 18 | 45 |

14) Βρίσκουμε την θεματική ενότητα με τις περισσότερες εμφανίσεις στο διαγωνισμό:

```
-- question 3.14
SELECT
    r.THEME_ID,
    t.Name,
    COUNT(*) AS ParticipationCount
FROM
    plays p
JOIN
    recipes r ON p.recipe = r.REC_ID
JOIN
    theme t ON r.THEME_ID = t.THEME_ID
GROUP BY
    r.THEME_ID, t.Name
ORDER BY
    COUNT(*) DESC
LIMIT 1;
```

| | THEME_ID | Name | ParticipationCount |
|---|----------|--------------|--------------------|
| ▶ | 40 | Clean Eating | 35 |

15) Βρίσκουμε τις ομάδες τροφίμων που δεν έχουν εμφανιστεί ποτέ στον διαγωνισμό:

```
-- question 3.15
WITH AllFoodGroups AS (
    SELECT FG_ID, Name
    FROM food_groups
),
ParticipatedFoodGroups AS (
    SELECT DISTINCT fg.FG_ID
    FROM recipes r
    JOIN ingredients i ON r.PRIM_ING_ID = i.ING_ID
    JOIN food_groups fg ON i.FG_ID = fg.FG_ID
    JOIN plays p ON r.REC_ID = p.recipe
)

SELECT afg.FG_ID, afg.Name
FROM AllFoodGroups afg
LEFT JOIN ParticipatedFoodGroups pfg ON afg.FG_ID = pfg.FG_ID
WHERE pfg.FG_ID IS NULL;
```

| | FG_ID | Name |
|---|-------|---------|
| ▶ | 11 | Seafood |

Βιβλιοθήκες και Frameworks που χρησιμοποιήθηκαν

- MySQL Workbench για την κατασκευή της Βάσης
- Jupyter Notebook για την κατασκευή των δεδομένων για την Βάση
- Συγκεκριμένα, οι βιβλιοθήκες Python: faker, pandas, random