ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

Εξαμηνιαία Εργασία (Ακ. Έτος 2023-2024)

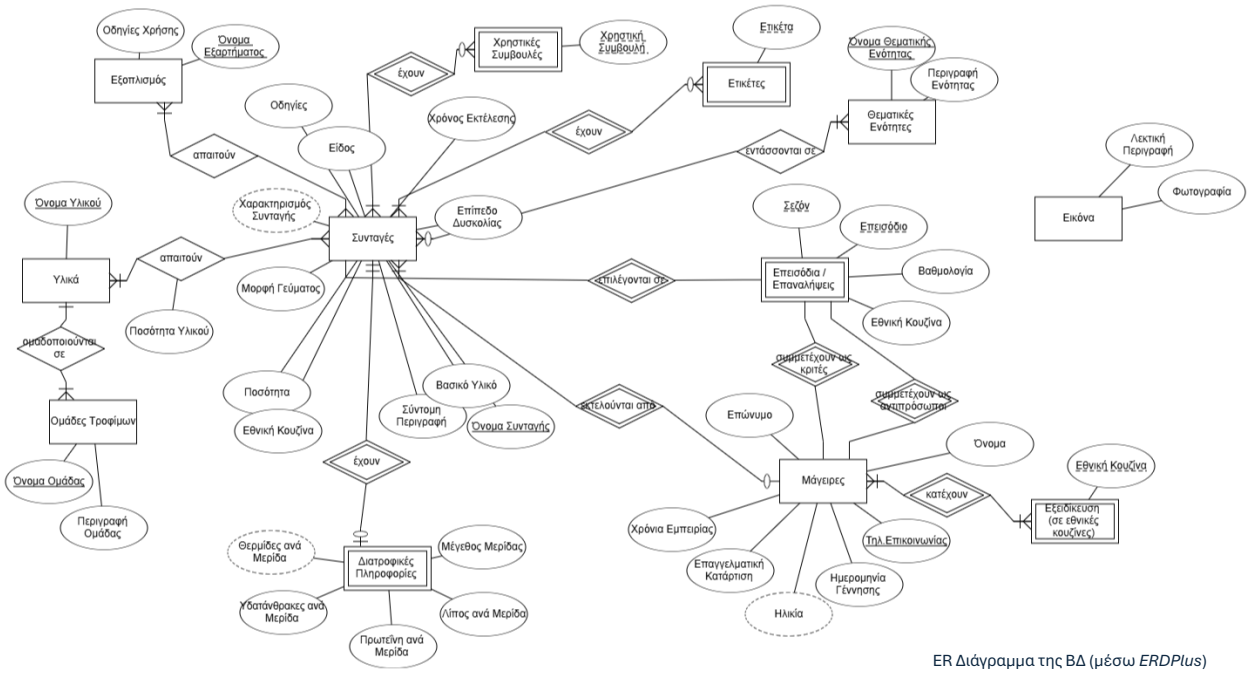
Κωνσταντίνος Γκαβογιάννης

(ΑΜ: 03121820)

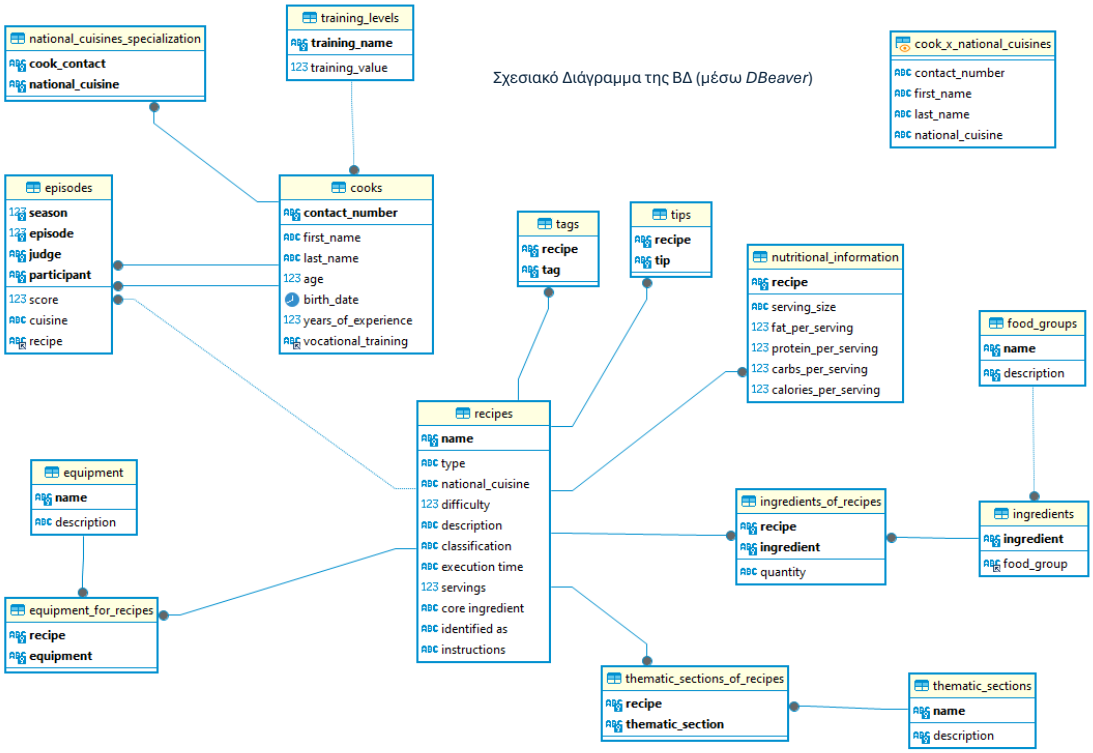
Γεώργιος Μεγαλιός

(ΑΜ: 03121118)

1. ER & Σχεσιακό Διάγραμμα της ΒΔ με αιτιολόγηση και ευρετήρια:



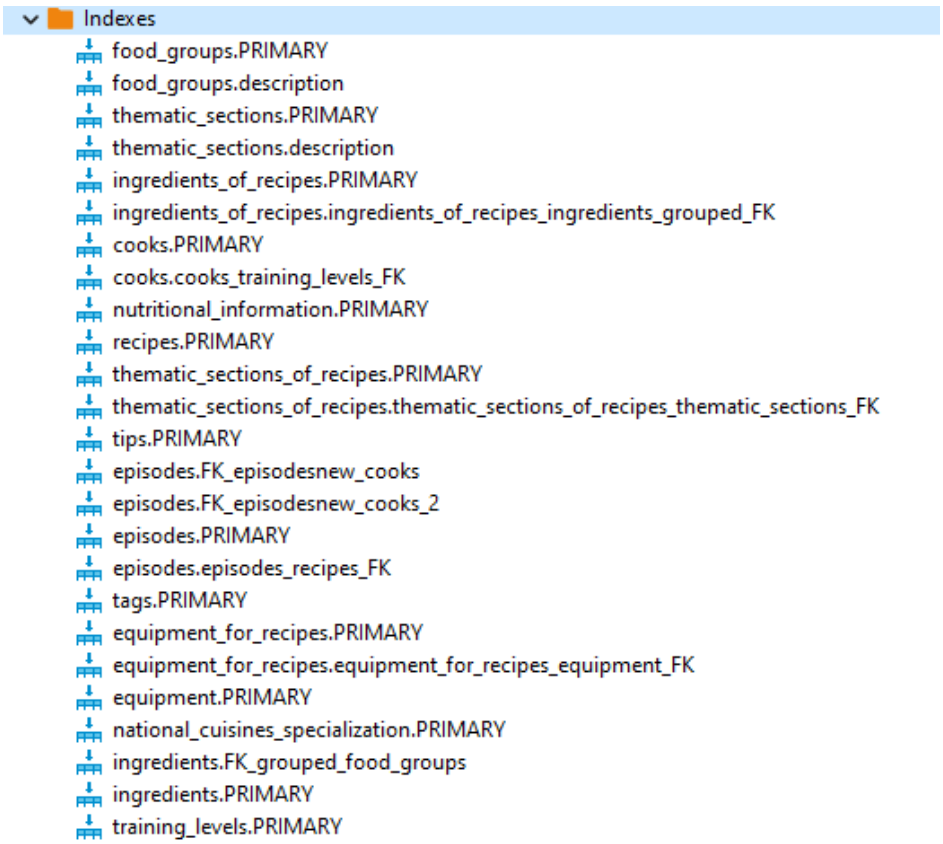
ER Διάγραμμα της ΒΔ (μέσω *ERDPlus*)



Σχεσιακό Διάγραμμα της ΒΔ (μέσω *DBeaver*)

Στο παραπάνω σχεσιακό διάγραμμα το primary key του καθενός table αντιπροσωπεύεται από έντονα γράμματα και διαχωρίζεται ρητά από τις υπόλοιπες ιδιότητες. Οφείλουμε να επισημάνουμε κάποιες επιπρόσθετες παρατηρήσεις, προς πληρότητα της βάσης:

- Για λόγους αισθητικής, έχει δημιουργηθεί ένα view με το όνομα *"cook_x_national_cuisines"*. Η υλοποίηση του αποσκοπεί αποκλειστικά στην ομορφότερη απεικόνιση του table *"national_cuisines_specialization"*, προσφέροντας μια πιο ξεκάθαρη εικόνα για την εξειδίκευση κάθε μάγειρα.
- Η έλλειψη χρόνου από πλευράς μας, επέφερε επιπτώσεις στην ολοκλήρωση της βάσης. Ως εκ τούτου, **από κάθε σχέση της βάσης απουσιάζει δυστυχώς μια τελευταία ιδιότητα (*"image"*), στην οποία αποβλέπαμε να συμπεριλάβουμε τις εικόνες του κάθε αντικειμένου (εξοπλισμός, συνταγές κτλ).**
- Προς υλοποίηση της σχέσης των μαγείρων, **έγινε η υπόθεση ότι ο καθένας χαρακτηρίζεται μοναδικά από το τηλέφωνο επικοινωνίας του, το οποίο και υποχρεούται να είναι πάντα ένα (κινητό, όχι σταθερό).**
- Multivalued attributes όπως τα tags και τα tips (τα οποία ανήκαν αρχικά στην σχέση των συνταγών) εξελίχθηκαν καταληκτικά σε δικές τους οντότητες, με τους κατάλληλους περιορισμούς.
- Περί των βημάτων που απαιτούνται για την υλοποίηση μιας συνταγής, αυτά απεικονίστηκαν εν τέλει ως ένα ενιαίο κείμενο, το οποίο περιλαμβάνει το σύνολο τους με την σωστή σειρά (*"instructions"*). Ερχομένη σε αντίθεση με το αρχικό μας πλάνο να υλοποιηθούν ως ξεχωριστή οντότητα, η απόφαση μας στηρίχτηκε στο τι θεωρήσαμε περισσότερο αποδοτικό, βάσει των περιστάσεων. Λόγω απελπισίας της τελευταίας στιγμής, η ιδιότητα αφέθηκε εξολοκλήρου στην default τιμή της, σε καθένα από τα πεδία της σχέσης *recipes*.
- Κατά την υλοποίηση της βάσης, εισάχθηκαν συνολικά 53 μάγειρες, 1800 επεισόδια (έξι σεζόν), 59 είδη εξοπλισμού, 179 υλικά, 9 θεματικές ενότητες και 102 συνταγές.
- Παρατίθενται τα indexes που χρησιμοποιήθηκαν στην βάση, έτσι ακριβώς όπως καταγράφηκαν στο περιβάλλον του DBeaver:



Τα παραπάνω indexes αφορούν αποκλειστικά τους περιορισμούς ακεραιότητας των primary και foreign keys, και δεν υλοποιούν επιπρόσθετες λειτουργίες.

- Η δομή της υπόλοιπης αναφοράς ακολουθεί με πιστό τρόπο τις οδηγίες της εκφώνησης. Πληροφορίες οι οποίες ενδέχεται να έχουν παραλειφθεί απορρέουν από την δυσκολία που αντιμετωπίσαμε ως προς την **πλήρη** υλοποίηση της βάσης, εξαιτίας διαφόρων παραγόντων. Το παρών αποτελεί την βέλτιστη δυνατή προσπάθεια από μέρους μας, και ευελπιστούμε σε ένα ικανοποιητικό, αν και ανεπαρκές, αποτέλεσμα.

2. DDL και DML script:

Παρακάτω παρατίθεται το σύνολο του DDL/DML κώδικα, τόσο για την δημιουργία της βάσης, όσων και των επιμέρους συστατικών της (tables):

Δημιουργία Βάσης (cooking_competition):

```
CREATE DATABASE `cooking_competition` /*!40100 DEFAULT CHARACTER SET latin1 COLLATE latin1_swedish_ci */;
```

Δημιουργία table 'recipes':

```
-- cooking_competition.recipes definition

CREATE TABLE `recipes` (
  `name` varchar(50) NOT NULL DEFAULT '',
  `type` varchar(20) NOT NULL DEFAULT '',
  `national_cuisine` varchar(20) NOT NULL DEFAULT '',
  `difficulty` int(11) NOT NULL DEFAULT 0,
  `description` varchar(200) NOT NULL DEFAULT '',
  `classification` varchar(20) NOT NULL DEFAULT '',
  `execution time` tinytext NOT NULL,
  `servings` int(11) NOT NULL DEFAULT 0,
  `core ingredient` varchar(15) NOT NULL DEFAULT '0',
  `identified as` varchar(20) NOT NULL DEFAULT '0',
  `instructions` varchar(100) DEFAULT 'Instructions need to be added.',
  PRIMARY KEY (`name`) USING BTREE
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;
```

Δημιουργία table 'tags':

```
-- cooking_competition.tags definition

CREATE TABLE `tags` (
  `recipe` varchar(50) NOT NULL,
  `tag` varchar(50) NOT NULL,
  PRIMARY KEY (`recipe`,`tag`) USING BTREE,
  CONSTRAINT `tags_recipes_FK` FOREIGN KEY (`recipe`) REFERENCES `recipes` (`name`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;
```

Δημιουργία table 'thematic_sections':

```
-- cooking_competition.thematic_sections definition

CREATE TABLE `thematic_sections` (
  `name` varchar(25) NOT NULL,
  `description` varchar(100) NOT NULL,
  PRIMARY KEY (`name`),
  UNIQUE KEY `description` (`description`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;
```

Δημιουργία table 'cooks':

```
-- cooking_competition.cooks definition

CREATE TABLE `cooks` (
  `contact_number` varchar(20) NOT NULL,
  `first_name` varchar(20) NOT NULL,
  `last_name` varchar(20) NOT NULL,
  `age` int(11) DEFAULT NULL,
  `birth_date` date NOT NULL,
  `years_of_experience` int(11) NOT NULL,
  `vocational_training` varchar(25) NOT NULL,
  PRIMARY KEY (`contact_number`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;
```

***Προκειμένου να επιλυθεί το ερώτημα 3.13, προέκυψε η ανάγκη για την δημιουργία ενός καινούργιου table, υπό την ονομασία *'training_levels'* (αντιστοίχιση των πιθανών επαγγελματικών καταρτίσεων με έναν σχετικό αριθμό). Ως εκ τούτου, προέκυψε ο επιπρόσθετος περιορισμός foreign key της σχέσης *'cooks'*, ως προς την προαναφερθούσα:**

```
CONSTRAINT `cooks training levels FK` FOREIGN KEY (`vocational training`) REFERENCES `training levels` (`training name`)
```

Δημιουργία table 'equipment':

```
-- cooking_competition.equipment definition

CREATE TABLE `equipment` (
  `name` varchar(30) NOT NULL,
  `description` varchar(200) NOT NULL DEFAULT '',
  PRIMARY KEY (`name`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;
```

Δημιουργία table 'food_groups':

```
-- cooking_competition.food_groups definition

CREATE TABLE `food_groups` (
  `name` varchar(50) NOT NULL,
  `description` varchar(100) NOT NULL,
  PRIMARY KEY (`name`),
  UNIQUE KEY `description` (`description`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;
```

Δημιουργία trigger 'tips_before_insert':

```
CREATE DEFINER=`root`@`localhost` TRIGGER `tips_before_insert` BEFORE INSERT ON `tips` FOR EACH ROW BEGIN
    DECLARE tip_count INT;
    SELECT COUNT(*) INTO tip_count FROM tips  WHERE `recipe` = NEW.`recipe`;
    IF tip_count >= 3 THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Cannot insert more than 3 tips per recipe';
    END IF;
END;
```

***Το παραπάνω trigger αποβλέπει στην εφαρμογή ενός άνω ορίου, ως προς το πλήθος των χρηστικών συμβουλών που μπορεί να έχει μία συνταγή. Καθιστώντας αδύνατο να προστεθούν παραπάνω από τρεις συμβουλές πριν την εισαγωγή μιας καινούργιας, το *'tips_before_insert'* εφαρμόζεται με ευκολονόητο τρόπο στην σχέση 'tips' της βάσης και συμμορφώνεται απόλυτα με τα ζητούμενα της άσκησης.**

Δημιουργία table 'training_levels':

```
-- cooking_competition.training_levels definition

CREATE TABLE `training_levels` (
  `training_name` varchar(25) NOT NULL,
  `training_value` int(11) DEFAULT NULL,
  PRIMARY KEY (`training_name`) USING BTREE
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;
```

***Όπως αναφέρθηκε και προηγουμένως, η παραπάνω σχέση δημιουργήθηκε αποκλειστικά προς επίλυση του ερωτήματος 3.13, με βάσει την εκφώνηση. Το κάθε πιθανό ενδεχόμενο επαγγελματικής κατάρτισης αντιστοιχίζεται σε έναν αντιπροσωπευτικό αριθμό, ή αλλιώς βαθμό.**

Δημιουργία table 'ingredients':

```
-- cooking_competition.ingredients definition

CREATE TABLE `ingredients` (
  `ingredient` varchar(30) NOT NULL,
  `food_group` varchar(50) NOT NULL,
  PRIMARY KEY (`ingredient`),
  KEY `FK_grouped_food_groups` (`food_group`) USING BTREE,
  CONSTRAINT `ingredients_grouped_food_groups_FK` FOREIGN KEY (`food_group`) REFERENCES `food_groups` (`name`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;
```

Δημιουργία table 'ingredients_of_recipes':

```
-- cooking_competition.ingredients_of_recipes definition

CREATE TABLE `ingredients_of_recipes` (
  `recipe` varchar(50) NOT NULL,
  `ingredient` varchar(30) NOT NULL,
  `quantity` varchar(50) DEFAULT '0',
  PRIMARY KEY (`recipe`,`ingredient`),
  KEY `ingredients_of_recipes_ingredients_grouped_FK` (`ingredient`),
  CONSTRAINT `ingredients_of_recipes_ingredients_grouped_FK` FOREIGN KEY (`ingredient`) REFERENCES `ingredients` (`ingredient`),
  CONSTRAINT `ingredients_of_recipes_recipes_FK` FOREIGN KEY (`recipe`) REFERENCES `recipes` (`name`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;
```

Δημιουργία table 'national_cuisines_specialization':

```
-- cooking_competition.national_cuisines_specialization definition

CREATE TABLE `national_cuisines_specialization` (
  `cook_contact` varchar(20) NOT NULL,
  `national_cuisine` varchar(20) NOT NULL,
  PRIMARY KEY (`cook_contact`,`national_cuisine`),
  CONSTRAINT `national_cuisines_specialization_cooks_FK` FOREIGN KEY (`cook_contact`) REFERENCES `cooks` (`contact_number`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;
```

Δημιουργία table 'tips':

```
-- cooking_competition.tips definition

CREATE TABLE `tips` (
  `recipe` varchar(50) NOT NULL,
  `tip` varchar(200) NOT NULL,
  PRIMARY KEY (`recipe`,`tip`) USING BTREE,
  CONSTRAINT `FK__recipe` FOREIGN KEY (`recipe`) REFERENCES `recipes` (`name`) ON DELETE CASCADE ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;
```

Δημιουργία table 'thematic_sections_of_recipes':

```
-- cooking_competition.thematic_sections_of_recipes definition

CREATE TABLE `thematic_sections_of_recipes` (
  `recipe` varchar(50) NOT NULL,
  `thematic_section` varchar(25) NOT NULL,
  PRIMARY KEY (`recipe`,`thematic_section`),
  KEY `thematic_sections_of_recipes_thematic_sections_FK` (`thematic_section`),
  CONSTRAINT `thematic_sections_of_recipes_recipes_FK` FOREIGN KEY (`recipe`) REFERENCES `recipes` (`name`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `thematic_sections_of_recipes_thematic_sections_FK` FOREIGN KEY (`thematic_section`) REFERENCES `thematic_sections` (`name`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;
```

Δημιουργία table 'episodes':

```
-- cooking_competition.episodes definition

CREATE TABLE `episodes` (
  `season` int(11) NOT NULL,
  `episode` int(11) NOT NULL,
  `judge` varchar(20) NOT NULL,
  `score` int(11) NOT NULL,
  `participant` varchar(20) NOT NULL,
  `cuisine` varchar(50) DEFAULT NULL,
  `recipe` varchar(50) NOT NULL,
  PRIMARY KEY (`season`,`episode`,`judge`,`participant`),
  KEY `FK_episodesnew_cooks` (`judge`) USING BTREE,
  KEY `FK_episodesnew_cooks_2` (`participant`) USING BTREE,
  KEY `episodes_recipes_FK` (`recipe`),
  CONSTRAINT `FK_episodes_cooks` FOREIGN KEY (`judge`) REFERENCES `cooks` (`contact_number`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `FK_episodes_cooks_2` FOREIGN KEY (`participant`) REFERENCES `cooks` (`contact_number`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `episodes_recipes_FK` FOREIGN KEY (`recipe`) REFERENCES `recipes` (`name`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;
```

Δημιουργία table 'equipment_for_recipes':

```
-- cooking_competition.equipment_for_recipes definition

CREATE TABLE `equipment_for_recipes` (
  `recipe` varchar(50) NOT NULL,
  `equipment` varchar(30) NOT NULL,
  PRIMARY KEY (`recipe`,`equipment`),
  KEY `equipment_for_recipes_equipment_FK` (`equipment`),
  CONSTRAINT `equipment_for_recipes_equipment_FK` FOREIGN KEY (`equipment`) REFERENCES `equipment` (`name`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `equipment_for_recipes_recipes_FK` FOREIGN KEY (`recipe`) REFERENCES `recipes` (`name`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;
```

Δημιουργία table 'nutritional_information':

```
-- cooking_competition.nutritional_information definition

CREATE TABLE `nutritional_information` (
  `recipe` varchar(50) NOT NULL,
  `serving_size` varchar(50) DEFAULT NULL,
  `fat_per_serving` decimal(5,2) NOT NULL,
  `protein_per_serving` decimal(5,2) NOT NULL,
  `carbs_per_serving` decimal(5,2) NOT NULL,
  `calories_per_serving` decimal(6,2) GENERATED ALWAYS AS (`fat_per_serving` * 9 + `protein_per_serving` * 4 + `carbs_per_serving` * 4) STORED,
  PRIMARY KEY (`recipe`),
  CONSTRAINT `nutritional_information_ibfk_1` FOREIGN KEY (`recipe`) REFERENCES `recipes` (`name`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;
```

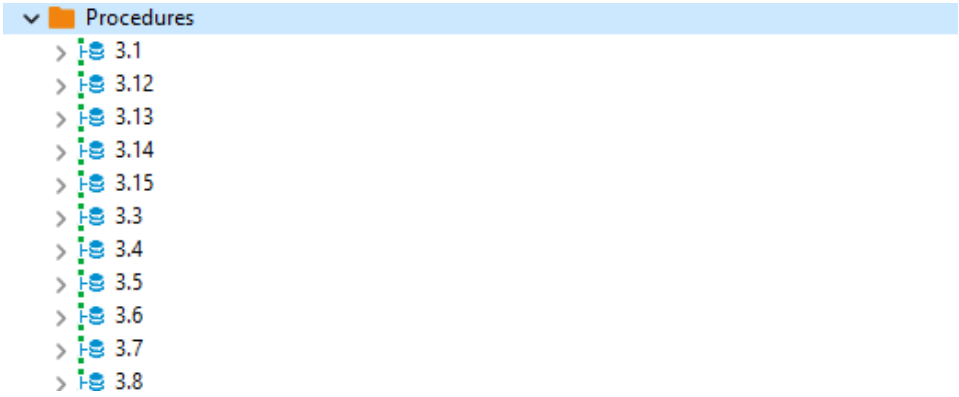
***Καθώς η συγκεκριμένη σχέση παράχθηκε τελευταία ως προς τις υπόλοιπες, κατέστη αδύνατο να της δώσουμε την απαραίτητη σημασία. Δραστικά επηρεασμένες από την έλλειψη χρόνου, πληροφορίες όπως οι θερμίδες ανά μερίδα είναι λάθος ορισμένες.**

Δημιουργία view 'cook_x_national_cuisines':

```
-- cooking_competition.cook_x_national_cuisines source

create or replace
algorithm = UNDEFINED view `cooking_competition`.`cook_x_national_cuisines` as
select
    `cooking_competition`.`cooks`.`contact_number` as `contact_number`,
    `cooking_competition`.`cooks`.`first_name` as `first_name`,
    `cooking_competition`.`cooks`.`last_name` as `last_name`,
    `cooking_competition`.`national_cuisines_specialization`.`national_cuisine` as `national_cuisine`
from
    (`cooking_competition`.`national_cuisines_specialization`
join `cooking_competition`.`cooks` on
    (`cooking_competition`.`cooks`.`contact_number` = `cooking_competition`.`national_cuisines_specialization`.`cook_contact`));
```

Εμπιστευόμενοι την προσωπική μας κρίση, **αποθηκεύσαμε τις απαντήσεις των ερωτημάτων της εκφώνησης υπό την μορφή procedures**. Αξιοποιώντας την δυνατότητα που αυτά προσφέρουν ως προς την αποθήκευση τον αποτελεσμάτων με γρήγορο και σαφή τρόπο (μέσου της εντολής call()), θεωρήσαμε ότι αποτελούν αποδεκτό μέσο για την οργάνωση των σκέψεων μας.



Όπως γίνεται φανερό βάσει της παραπάνω εικόνας, πραγματοποιήθηκε η υλοποίηση έντεκα ερωτημάτων από τα δεκαπέντε, στηριζόμενοι εν μέρει και σε κάποιες δικές μας υποθέσεις, λόγω ασάφειας των εκφωνήσεων. Δεδομένου πως ο κώδικας των procedures συγκαταλέγεται στον DDL, παρατίθενται αυτούσια οι προσωπικές μας λύσεις ως προς τα ερωτήματα της άσκησης, συμπληρωμένες με τυχόν διευκρινίσεις και υποθέσεις που κρίναμε απαραίτητες.

Απάντηση του ερωτήματος 3.1:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `cooking_competition`.`3.1`()
begin
    SELECT `participant`,ROUND(AVG(score),2) AS average_score
    FROM `episodes`
    GROUP BY `participant`;

    SELECT `cuisine`,ROUND(AVG(score),2) AS average_score
    FROM `episodes`
    GROUP BY `cuisine`;
END;
```

Απάντηση του ερωτήματος 3.3:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `cooking_competition`.`3.3`()
begin
    SELECT c.last_name, c.first_name, c.age, COUNT(e.recipe) AS recipe_count
    FROM cooks AS c
    JOIN episodes AS e
    ON c.contact_number=e.participant
    WHERE c.age < 30
    GROUP BY c.last_name, c.first_name
    ORDER BY recipe_count desc;
END;
```

*Ο παραπάνω κώδικας εμφανίζει στην έξοδο του το σύνολο των μαγείρων (ηλικία < 30 ετών) που έχουν συμμετάσχει στο διαγωνισμό (ως αντιπρόσωποι) τουλάχιστον μία φορά. Η τελευταία εντολή (ORDER BY…desc;) διασφαλίζει ότι η απεικόνιση θα πραγματοποιηθεί με φθίνουσα σειρά, βάσει του συνόλου των συνταγών που τους έχει ανατεθεί. Ως εκ τούτου, ο ζητούμενος μάγειρας του ερωτήματος προσδιορίζεται στην κορυφή των αποτελεσμάτων, θεωρώντας πως αυτός ο τρόπος συνιστά μια πληρέστερη εικόνα.

Απάντηση του ερωτήματος 3.4:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `cooking_competition`.`3.4`()
begin
    SELECT `contact_number`,first_name,last_name
    FROM `cooks`
    WHERE `contact_number` NOT IN (SELECT `judge` FROM `episodes`);
END;
```

Απάντηση του ερωτήματος 3.5:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `cooking_competition`.`3.5`()
begin
    WITH JudgeEpisodeCounts AS (
   SELECT
      e.judge,
      e.season,
      COUNT(e.episode)/10  AS episode_count
   FROM
      episodes e
   GROUP BY
      e.judge, e.season
   HAVING
       COUNT(e.episode)/10>3
)
SELECT
    j1.judge,
    j2.judge,
    j1.season,
    j1.episode_count
FROM
    JudgeEpisodeCounts j1
JOIN
    JudgeEpisodeCounts j2 ON j1.episode_count = j2.episode_count
                         AND j1.season = j2.season
                         AND j1.judge <> j2.judge
ORDER BY
    j1.season, j1.episode_count;
END;
```

---

*Προς απάντηση του ερωτήματος, προβήκαμε στην υπόθεση ότι η φράση «σε διάστημα ενός έτους» αναφέρεται στην ίδια, υπ' αριθμόν, σεζόν. Ως εκ τούτου, ο παραπάνω κώδικας προσδιορίζει τους κριτές που έχουν συμμετάσχει στον ίδιο αριθμό επεισοδίων σε διάστημα της ίδιας σεζόν. Επιπρόσθετα, στην έξοδο παράγονται διπλότυπα, τα οποία πρέπει να αγνοηθούν για προφανείς λόγους (π.χ. {κριτής 1, κριτής 2} και {κριτής 2, κριτής 1}, με την αντίστροφη σειρά).

Απάντηση του ερωτήματος 3.6:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `cooking_competition`.`3.6`()
begin
WITH ConcatTags AS (
SELECT
    GROUP_CONCAT(t.tag ORDER BY t.tag) AS tag_combination,
    t.recipe
FROM
    `tags` t
WHERE t.recipe IN (
select recipe FROM episodes
)
GROUP BY
    t.recipe
)
SELECT
    tag_combination,
    COUNT(*) AS tag_count
FROM
    ConcatTags
GROUP BY
    tag_combination
ORDER BY
    tag_count DESC
LIMIT 3;
END;
```

Απάντηση του ερωτήματος 3.7:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `cooking_competition`.`3.7`()
begin
WITH ChefEpisodeCounts AS (
    SELECT
        e.participant,
        COUNT(e.episode) AS episode_count
    FROM
        episodes e
    GROUP BY
        e.participant
),
MaxEpisodes AS (
    SELECT
        MAX(episode_count) AS max_episode_count
    FROM
        ChefEpisodeCounts
)
SELECT
    c.participant,
    c.episode_count
FROM
    ChefEpisodeCounts c
JOIN
    MaxEpisodes m ON c.episode_count <= m.max_episode_count - 5
ORDER BY
    c.episode_count DESC;
END;
```

***Ως «συμμετοχές» ενός μάγειρα μετρήθηκαν αποκλειστικά αυτές που αναφέρονται στην δράση του ως αντιπρόσωπος, και όχι ως κριτής. Ως εκ τούτου, ο μάγειρας με τις περισσότερες συμμετοχές σε επεισόδια (και ο οποιοσδήποτε άλλος ζητείται) καθορίστηκε με βάση τις εμφανίσεις του ως διαγωνιζόμενος, στην οποία περίπτωση η συμμετοχή του είναι πιο ενεργή.**

Απάντηση του ερωτήματος 3.8:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `cooking_competition`.`3.8`()
begin
    WITH NumberOfEquipment AS (
SELECT
    r.name,
    COUNT(er.equipment) AS equipment_count
FROM
    recipes r
JOIN
    equipment_for_recipes er ON r.name = er.recipe
GROUP BY
    r.name
ORDER BY
    equipment_count DESC
)
,
MaxEquipment AS (
    SELECT
        MAX(equipment_count) AS max_equipment_count
    FROM
        NumberOfEquipment ne
)

SELECT UNIQUE (e.recipe),e.season, e.episode, ne.equipment_count
FROM episodes as e
JOIN NumberOfEquipment ne ON
e.recipe =ne.name
JOIN MaxEquipment ON
ne.equipment_count=MaxEquipment.max_equipment_count;
END;
```

Απάντηση του ερωτήματος 3.12:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `cooking_competition`.`3.12`()
begin
    WITH aux AS(
SELECT
    UNIQUE (r.name),
    season,
    episode,
    r.difficulty
FROM
    episodes e
JOIN
    recipes r ON e.recipe = r.name
),

aux2 AS(
SELECT season, episode,SUM(difficulty)/10 AS difficult
FROM aux
GROUP BY season,episode
ORDER BY difficult DESC
)
SELECT season,episode, MAX(difficult) AS AvarageDifficulty
from aux2
GROUP BY season;
END;
```

Απάντηση του ερωτήματος 3.13:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `cooking_competition`.`3.13`()
begin
    WITH avg_training AS (
SELECT
    e.season , e.episode,
    AVG(tl.training_value) AS average_training_level
FROM
    episodes e
JOIN
    cooks c ON e.participant = c.contact_number
JOIN
    training_levels tl ON c.vocational_training = tl.training_name
GROUP BY
    e.episode , e.season
ORDER BY
    average_training_level ASC
)
,
minval AS
(SELECT
      MIN(average_training_level) AS min_avg_training_level
FROM
      avg_training)

SELECT
    at.season,
    at.episode,
    at.average_training_level
FROM
    avg_training at
JOIN
   minval
ON
   at.average_training_level = minval.min_avg_training_level;
END;
```

Απάντηση του ερωτήματος 3.14:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `cooking_competition`.`3.14`()
begin
SELECT
    ts.name AS thematic_section,
    COUNT(e.episode) AS appearance_count
FROM
    thematic_sections ts
JOIN
    thematic_sections_of_recipes tsr ON ts.name = tsr.thematic_section
JOIN
    episodes e ON tsr.recipe = e.recipe
GROUP BY
    ts.name
ORDER BY
    appearance_count DESC
LIMIT 1;
END;
```

Απάντηση του έρωτηματος 3.15:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `cooking_competition`.`3.15`()
begin
    SELECT name
FROM food_groups
WHERE name NOT in

(SELECT
    i.food_group
FROM
    ingredients i
JOIN
    ingredients_of_recipes ir ON i.ingredient = ir.ingredient
JOIN
    episodes e ON ir.recipe = e.recipe);
END;
```

Στηριζόμενοι στις γνώσεις μας και σε αρκετές προσωπικές διερμηνείες περί των όσων ζητούνται, θεωρούμε πως οι παρατιθέμενες απαντήσεις βρίσκονται επί τον πλείστον κοντά στην τελική τους μορφή.

Όσον αφορά τον ζητούμενο DML κώδικα, για προφανείς λόγους χωρητικότητας αυτός παρατίθεται ως ξεχωριστό τμήμα του repository (GitHub): [cooking-competition-DB24/cooking_competition_dml.sql at master ·

geo10meg/cooking-competition-DB24 (github.com) . Περισσότερες πληροφορίες μπορεί να βρεθούν στο τμήμα '4' της αναφοράς.

3. Αναλυτικά βήματα εγκατάστασης της εφαρμογής σας καθώς και τυχόν βιβλιοθηκών που απαιτούνται:

Προς υλοποίηση της βάσης δεδομένων μας, εργαστήκαμε σε διαφορετικούς υπολογιστές και περιβάλλοντα, διαμοιραζόμενοι όμως παράλληλα τις ίδιες πληροφορίες και κώδικα.

-Ως σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων χρησιμοποιήθηκε από κοινού η εφαρμογή MariaDB (MySQL), την οποία και εγκαταστήσαμε ως το πρώτο στάδιο της διαδικασίας (v.11.3).

Αξιοποιώντας την εντολή `mysql -u your_username -p` και θέτοντας το σύνηθες 'root' ως username, συνδεθήκαμε στο σύστημα χωρίς ατυχείς δυσκολίες. Αξίζει να σημειωθεί πως στην προκείμενη περίπτωση ο MariaDB server έτρεξε στον προεπιλεγμένο *localhost*, απαιτώντας μη περαιτέρω διευκρινίσεις.

```
MariaDB [(none)]> use cooking_competition;
Database changed
MariaDB [cooking_competition]> show tables;
+---------------------------------+
| Tables_in_cooking_competition   |
+---------------------------------+
| cook_x_national_cuisines        |
| cooks                           |
| episodes                        |
| equipment                       |
| equipment_for_recipes           |
| food_groups                     |
| ingredients                     |
| ingredients_of_recipes          |
| national_cuisines_specialization |
| nutritional_information         |
| recipes                         |
| tags                            |
| thematic_sections               |
| thematic_sections_of_recipes    |
| tips                            |
+---------------------------------+
15 rows in set (0.001 sec)
```

-Διαφοροποιώντας τις επιλογές μας ως προς το εργαλείο διαχείρισης και ανάπτυξης της βάσης, οι εφαρμογές **DBeaver** και **HeidiSQL** έπαιξαν καθοριστικό ρόλο στην διευκόλυνση της εναπομείνουσας διαδικασίας. Εκμετάλλευομενοι τόσο το **όμορφο, καθαρό περιβάλλον** της πρώτης, όσο και της ευχρηστία που προσέφερε η δεύτερη (υποστηρίζοντας, επιπρόσθετα, και

την ελληνική γλώσσα), συνδυάσαμε επιτυχώς τα πλεονεκτήματα της καθεμίας σε ένα εννιαίο και τελικό έργο, εν πλήρη συνεργασία.

Εγκαθιδρύοντας σύνδεση μεταξύ των παραπάνω εφαρμογών και του συστήματος MariaDB, προχωρήσαμε στην αποκλειστική υλοποίηση της βάσης διαμέσου αυτών.

-Έχοντας ολοκληρώσει το τελικό όραμα της βάσης δεδομένων μας (ακόμη και ελλιπές), προχωρήσαμε στην εξαγωγή ενός link για το git repository της εφαρμογής μας. Έπειτα από την σχετική εγκατάσταση του συστήματος ελέγχου εκδόσεων Git και την εφαρμογή κάποιων εντολών στο cmd του υπολογιστή, τα δεδομένα της βάσης μεταφέρθηκαν εξολοκλήρου σε αντίστοιχο repository του GitHub.

4. Σύνδεσμος για το git repo της εφαρμογής:

Δίχως περιττολογίες, παρατίθεται αυτούσια ο ζητούμενος σύνδεσμος για το repository: geo10meg/cooking-competition-DB24 (github.com) .

Κατά την συγγραφή της παρούσας αναφοράς, το repository αποτελείται από τα εξής τρία (3) περιεχόμενα:

- cooking_competition.sql: Αρχείο SQL του συνόλου της βάσης, το οποίο περιέχει όλον τον απαραίτητο κώδικα για την δημιουργία της.
- cooking_competition_ddl.sql: Αρχείο SQL του DDL κώδικα της βάσης, εξαιρουμένων όλων των δεδομένων που εισάγαμε. Αποτελεί το ήμισυ του συνολικού αρχείου.

- cooking_competition_dml.sql: Αρχείο SQL του DML κώδικα της βάσης, εξαιρουμένων των CREATE δηλώσεων. Αποτελεί το δεύτερο ήμισυ του συνολικού αρχείου.

Ας σημειωθεί πως το παρών repository μπορεί να υποβληθεί σε αλλαγές, προκειμένου να προστεθούν πληροφορίες και να βελτιωθεί η αισθητική του περιβάλλοντος. **Ο κώδικας θα διατηρηθεί άθικτος.**

5. Καταληκτικά:

Έχοντας πλέον παραθέσει και το σχετικό repository για τον κώδικα της βάσης, η παρούσα αναφορά φτάνει στο τέλος της. Στηριζόμενοι στα όσα καταφέραμε να υλοποιήσουμε και βρισκόμενοι στην θέση να αναγνωρίσουμε τις προφανείς της ελλείψεις, η βάση που αναπτύχθηκε αποτελεί την βέλτιστη δυνατή μας προσπάθεια, δεδομένων των συνθηκών.

Παράγοντες όπως τα χρονικά περιθώρια, οι λοιπές υποχρεώσεις και οι περιορισμένες, προσωπικές μας γνώσεις, μας οδήγησαν καταληκτικά σε έναν μονόδρομο ατέλειας. Αποδεχόμενοι την δύσκολη πορεία και αποφασίζοντας να εργαστούμε μόνο υπό την σφαίρα δυνατοτήτων μας, ευελπιστούμε το τελικό «προϊόν» να ανταποκρίνεται σε αρκετά από τα ζητούμενα, προσφέροντας μια επαρκή, έως και υψηλή, λειτουργία.

Σας ευχαριστούμε για την κατανόηση και σας ευχόμαστε ένα απολαυστικό καλοκαίρι!

ΟΜΑΔΑ PROJECT 5

ΓΕΩΡΓΙΟΣ ΜΕΤΑΛΛΙΟΣ

ΚΩΝΣΤΑΝΤΙΝΟΣ ΓΚΑΒΟΓΙΑΝΝΗΣ