Συστήματα Διαχείρισης Δεδομένων Μεγάλου Όγκου

Εργαστηριακή Άσκηση 2022/23

|  |  |  |
| --- | --- | --- |
| **Όνομα** | **Επώνυμο** | **ΑΜ** |
| Πάρης | Σεργιάννης | 1067467 |
| Στυλιανός | Στυλιανάκης | 1059713 |

Βεβαιώνω ότι είμαι συγγραφέας της παρούσας εργασίας και ότι έχω αναφέρει ή παραπέμψει σε αυτήν, ρητά και συγκεκριμένα, όλες τις πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, προτάσεων ή λέξεων, είτε αυτές μεταφέρονται επακριβώς (στο πρωτότυπο ή μεταφρασμένες) είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για το συγκεκριμένο μάθημα/σεμινάριο/πρόγραμμα σπουδών.

Έχω ενημερωθεί ότι σύμφωνα με τον εσωτερικό κανονισμό λειτουργίας του Πανεπιστημίου Πατρών άρθρο 50§6, τυχόν προσπάθεια αντιγραφής ή εν γένει φαλκίδευσης της εξεταστικής και εκπαιδευτικής διαδικασίας από οιονδήποτε εξεταζόμενο, πέραν του μηδενισμού, συνιστά βαρύ πειθαρχικό παράπτωμα.

|  |  |
| --- | --- |
| Υπογραφή    16 / 2 / 2024 | Υπογραφή    16 / 2 / 2024 |

**Συνημμένα αρχεία κώδικα**

Μαζί με την παρούσα αναφορά υποβάλλουμε τα παρακάτω αρχεία κώδικα

|  |  |  |
| --- | --- | --- |
| Αρχείο | Αφορά το ερώτημα | Περιγραφή/Σχόλιο |
| main.py | όλα | Σε αυτο το αρχείο καλούμε τις συναρτήσεις από το db\_functions.py, για να εκτελέσουμε τα inserts και τα selects στην βάση. Επιλέγουμε mode αλλάζοντας την μεταβλητή  operation\_mode. |
| db\_functions.py | όλα | Περιέχει όλες τις συναρτήσεις που αφορούν τις βασικές λειτουργίες της βασης, οι οποίες καλούνται από την main.py. |
| query\_generation.py | 2, 4 | Περιέχει τα table schemas και χρησιμοποιείται για την κατασκευή ερωτημάτων. |
| data\_shortening.py | όλα | Script που χρησιμοποιείται για την κατασκευή csv με ένα υποσύνολο δεδομένων, με σκοπό την ταχύτερη υλοποίηση των ερωτημάτων. |

# Τεχνικά χαρακτηριστικά περιβάλλοντος λειτουργίας

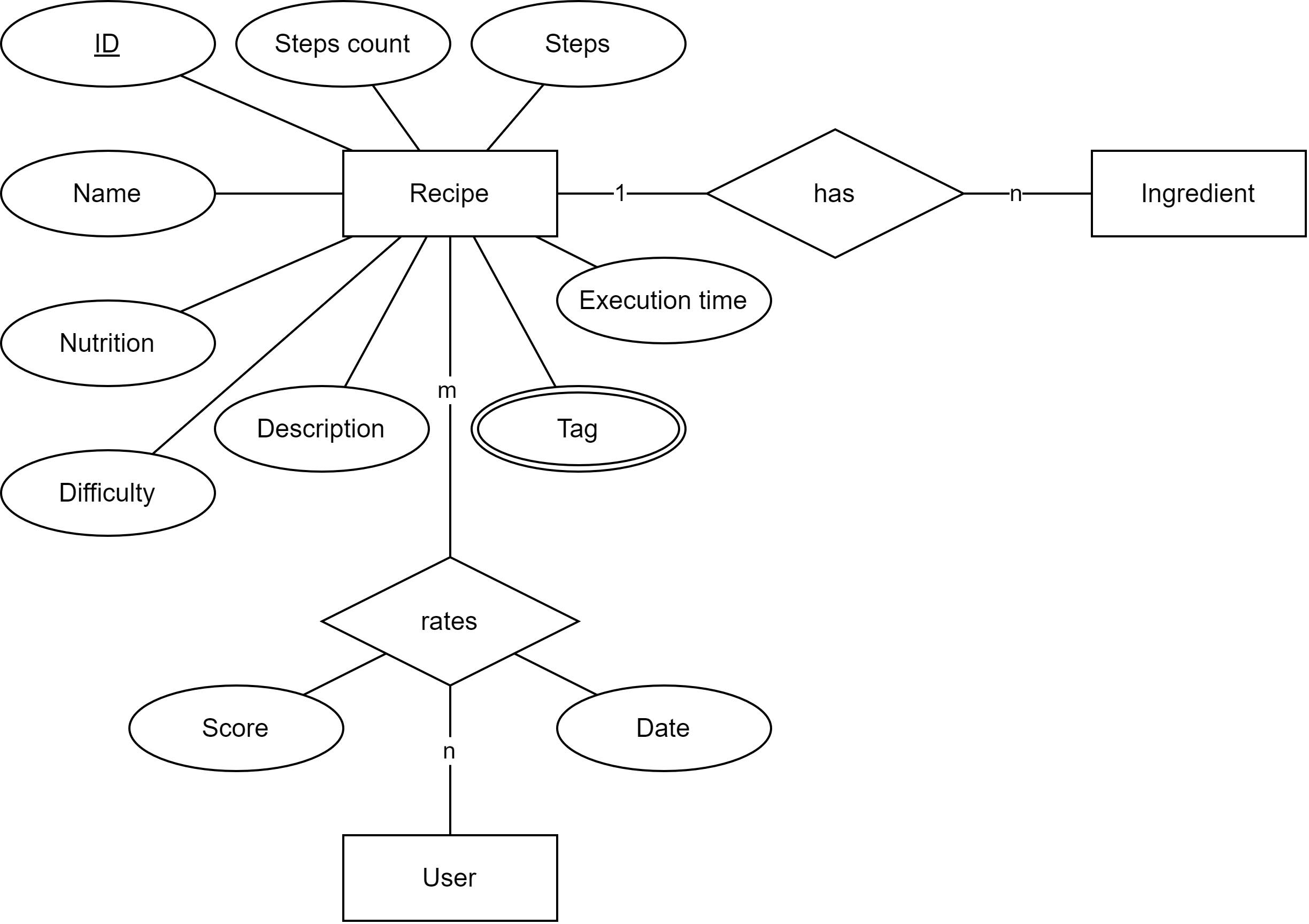
Για την υλοποίηση της εργασίας χρησιμοποιήθηκε αποκλειστικά το Astra DB.

# Ερώτημα 1: Σχεδιασμός ΒΔ

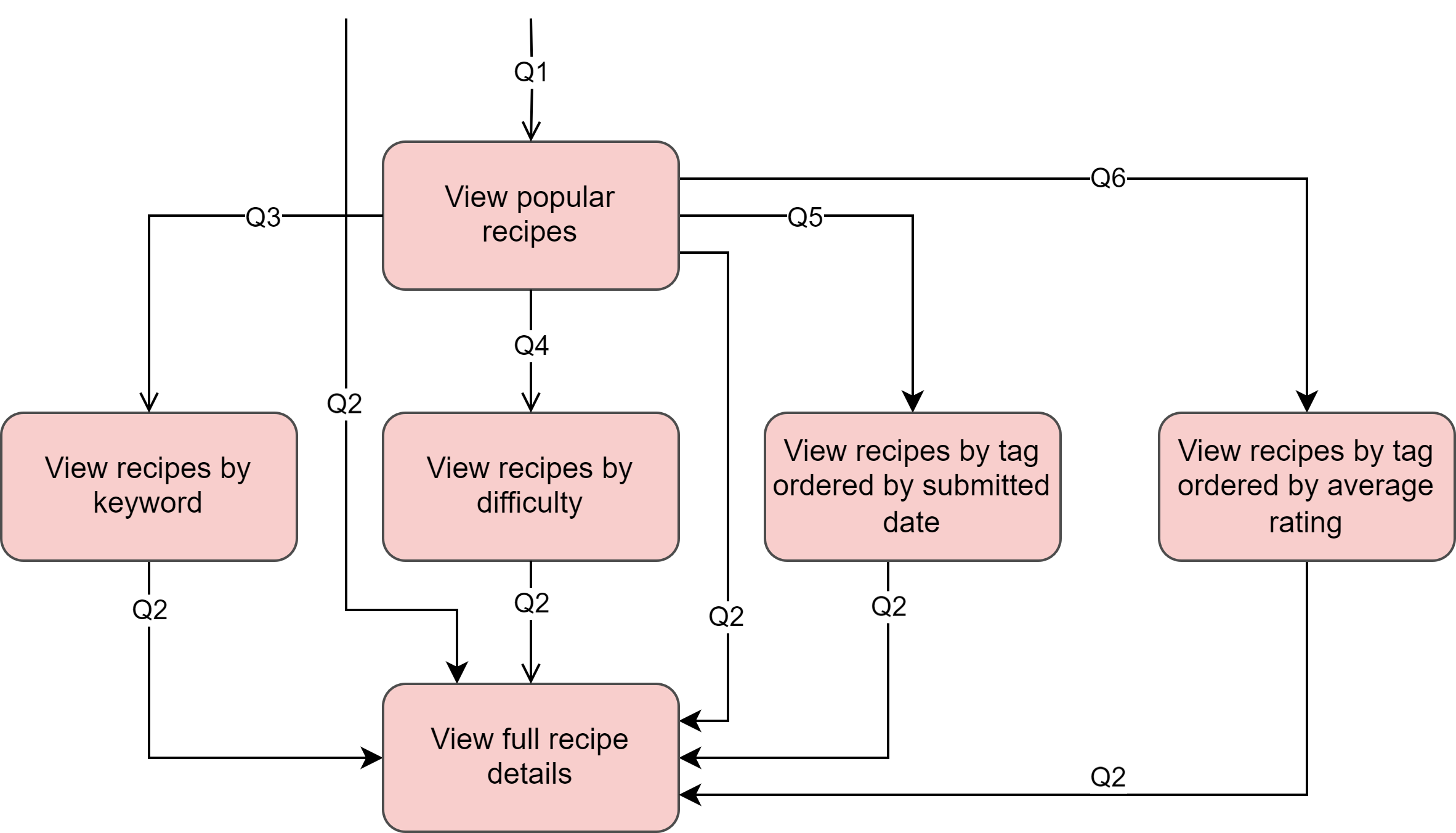
### Απαιτήσεις χρήστη:

1. Εμφάνιση των συνταγών που είναι δημοφιλείς (έχουν καλή βαθμολογία) εντός ενός χρονικού διαστήματος (π.χ. τους τελευταίους 3 μήνες) – αυτό μπορεί να αποτελεί κάλλιστα την αρχική οθόνη «προτάσεων» προς το χρήστη.
2. Να αναζητήσει την/τις συνταγές που περιέχουν κάποιες λέξεις – κλειδιά στον τίτλο.
3. Να αναζητήσει συνταγές με βάση την κατηγορία δυσκολίας (π.χ. χρόνος παρασκευής ή πλήθος βημάτων) και να τις λάβει με βάση κάποια ταξινόμηση (π.χ. μέση βαθμολογία.
4. Να δει τις λεπτομερείς πληροφορίες για κάποια συνταγή (κατηγορία δυσκολίας, μέση βαθμολογία, ετικέτες, υλικά, διαδικασία παρασκευής).
5. Να δει τις top-n συνταγές που σχετίζονται με κάποια ετικέτα.

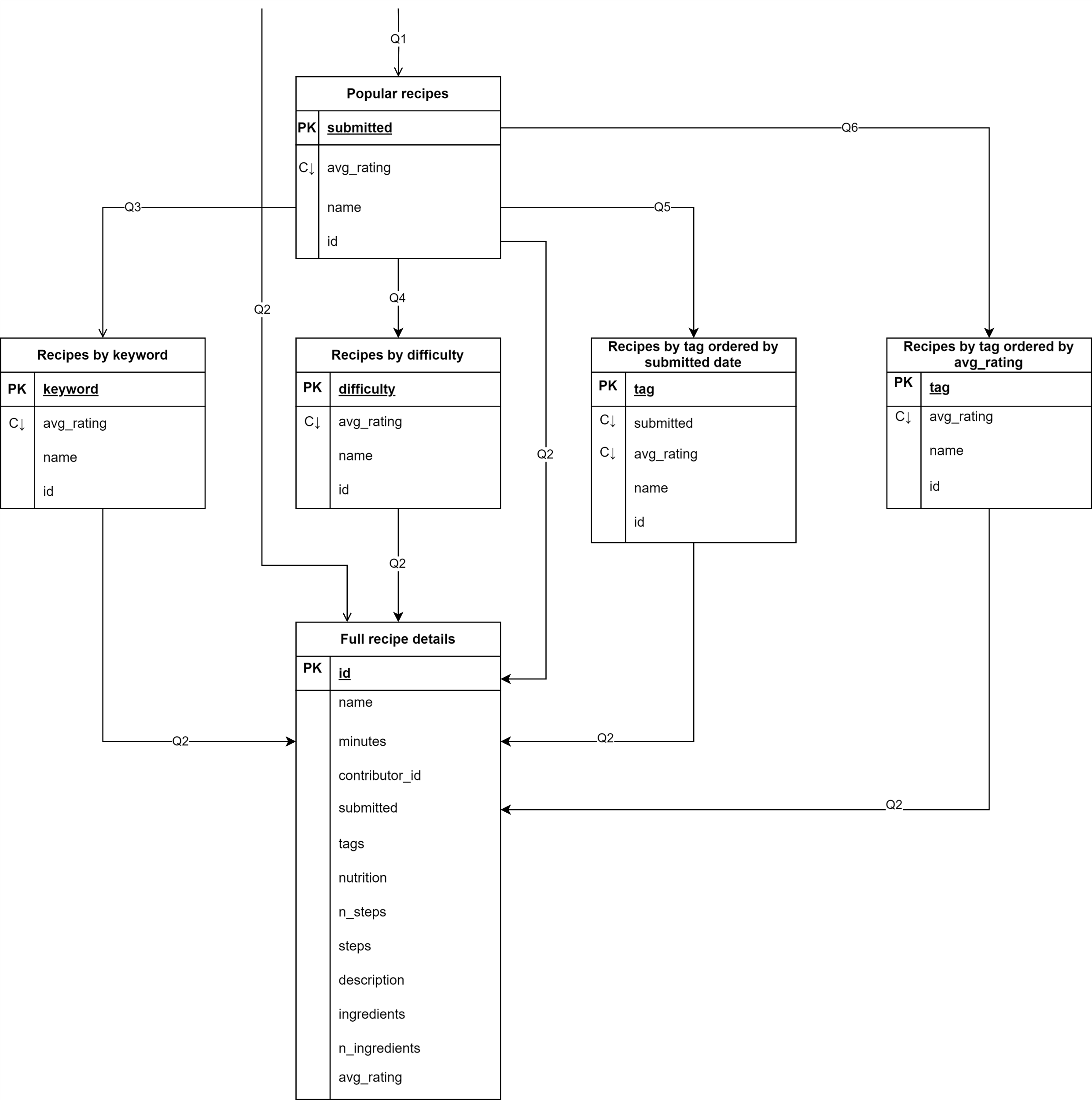
### Εννοιολογικό μοντέλο



### Application workflow



### Chebotko diagram



# Ερώτημα 2: Ερωτήματα DDL

[επαναλαμβάνετε τον παρακάτω πίνακα για κάθε keyspace στη ΒΔ σας]

|  |  |
| --- | --- |
| **Keyspace** | recipes.popular\_recipes |
| **DDL statement** | CREATE TABLE recipes.popular\_recipes (  id int,  submitted date,  avg\_rating float,  name text,  PRIMARY KEY (submitted, avg\_rating, id)  ) WITH CLUSTERING ORDER BY (avg\_rating DESC, id ASC); |
| **Screenshot** |  |

|  |  |
| --- | --- |
| **Keyspace** | recipes.recipes\_difficulty |
| **DDL statement** | CREATE TABLE recipes.recipes\_difficulty (  difficulty text,  id int,  avg\_rating float,  name text,  PRIMARY KEY (difficulty, avg\_rating, name)  ) WITH CLUSTERING ORDER BY (avg\_rating DESC, name ASC); |
| **Screenshot** |  |

|  |  |
| --- | --- |
| **Keyspace** | recipes.recipes\_tag\_rating |
| **DDL statement** | CREATE TABLE recipes.recipes\_tag\_rating (  tag text,  id int,  avg\_rating float,  name text,  PRIMARY KEY (tag, avg\_rating, id)  ) WITH CLUSTERING ORDER BY (avg\_rating DESC, id ASC); |
| **Screenshot** |  |

|  |  |
| --- | --- |
| **Keyspace** | recipes.recipes\_tag\_submitted |
| **DDL statement** | CREATE TABLE recipes.recipes\_tag\_submitted (  tag text,  submitted date,  avg\_rating float,  id int,  name text,  PRIMARY KEY (tag, submitted, avg\_rating, id)  ) WITH CLUSTERING ORDER BY (submitted DESC, avg\_rating DESC, id ASC); |
| **Screenshot** |  |

|  |  |
| --- | --- |
| **Keyspace** | recipes.recipes\_keywords |
| **DDL statement** | CREATE TABLE recipes.recipes\_keywords (  id int,  keywords set<text>,  avg\_rating float,  name text,  PRIMARY KEY (id, avg\_rating, name)  ) WITH CLUSTERING ORDER BY (avg\_rating DESC, name ASC); |
| **Screenshot** |  |

|  |  |
| --- | --- |
| **Keyspace** | recipes.recipes\_details |
| **DDL statement** | CREATE TABLE recipes.recipes\_details (  id int,  name text,  minutes int,  contributor\_id int,  submitted date,  tags set<text>,  nutrition list<float>,  n\_steps smallint,  steps list<text>,  description text,  ingredients set<text>,  n\_ingredients smallint,  avg\_rating float,  difficulty text,  keywords set<text>,  PRIMARY KEY (name)  ); |
| **Screenshot** |  |

# Ερώτημα 3: Απαντήσεις ερωτημάτων

|  |  |
| --- | --- |
| **Ερώτημα** | **Απάντηση** |
| Εμφάνιση των 30 συνταγών με την υψηλότερη μέση βαθμολογία μεταξύ 01/01/2012 και 31/05/2012 |  |
| Εμφάνιση όλων των λεπτομερειών για την ταινία «curried bean salad» (κατηγορία δυσκολίας, διατροφικές αξίες, βήματα, περιγραφή, μέση βαθμολογία) |  |
| Εμφάνιση των ταινιών της κατηγορίας «εύκολη» ταξινομημένες ως προς τη μέση βαθμολογία τους |  |
| Εμφάνιση των συνταγών που περιέχουν την ετικέτα “course” με ταξινόμηση ανά ημερομηνία προσθήκης (πιο πρόσφατες πρώτα) |  |
| Εμφάνιση των 20 συνταγών με την υψηλότερη μέση βαθμολογία για την ετικέτα “30-minutes-or-less”. |  |

# Ερώτημα 4Α: Χρόνοι εισαγωγής δεδομένων

ΣΗΜΕΙΩΣΗ: Η AstraDB απαγορεύει inserts με consistency level ONE επειδή τα θεωρεί κακή πρακτική. Γι αυτό τον λόγο εμείς τα εκτελέσαμε με consistency level TWO και παραθέτουμε τα αποτελέσματά μας.  
  
Εδώ κάναμε απευθείας τον μέσο όρο των insert operations στον κώδικα και τα παραθέτουμε απευθείας.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Επίπεδο write consistency** | | |
| **ALL** | **QUORUM** | **TWO** |
| **recipes** | *61.46* | *63.22* | *67.35* |
| **Μέσος όρος** | *61.46* | *63.22* | *67.35* |

A graph of a bar chart

Description automatically generated with medium confidence

# Ερώτημα 4Β: Χρόνοι ανάκτησης δεδομένων

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Επίπεδο write consistency** | | |
| **ALL** | **QUORUM** | **TWO** |
| **Ερώτημα 1** | *0.0948* | *0.0930* | *0.0929* |
| **Ερώτημα 2** | *0.0921* | *0.0928* | *0.0979* |
| **Ερώτημα 3** | *0.0927* | *0.0939* | *0.0945* |
| **Ερώτημα 4** | *0.0935* | *0.1002* | *0.0922* |
| **Ερώτημα 5** | *0.0939* | *0.0924* | *0.0919* |
| **Μέσος όρος** | 0.0934 | 0.0945 | 0.0939 |

# A graph of multiple colored bars Description automatically generated with medium confidence Ερώτημα 4Γ: Σχολιασμός αποτελεσμάτων

# Στα ερωτήματα εγγραφής βλέπουμε ξεκάθαρη ανοδική τάση όσο το consistency level γίνεται αυστηρότερο. Αυτό είναι εντελώς αναμενόμενο, αφού αυξάνονται τα μηνύματα acknowledge που ανταλάσσουν μεταξύ τους οι server της βάσης πρωτού βεβαιώσουν ένα write. Αυτό στην ουσία είναι το tradeoff του θεωρήματος CAP, σύμφωνα με το οποίο όσο αυξάνουμε το Consistency θα πρέπει να αναμένουμε μειωμένο Availability (κρατώντας σταθερό το Partition Tolerance). Στα ερωτήματα ανάγνωσης βλέπουμε παρόμοιο trend, με λίγη αστάθεια, ίσως λόγω αυξημένου traffic στους server της AstraDB.

# Βιβλιογραφία

* DataStax Cassandra Query Language Documentation: https://docs.datastax.com/en/cql-oss/3.x/cql/cqlIntro.html
* Pandas python library documentation: https://pandas.pydata.org/docs/index.html
* Drawio diagram tool: https://app.diagrams.net/