Συστήματα Διαχείρισης Δεδομένων Μεγάλου Όγκου

Εργαστηριακή Άσκηση 2021/22

|  |  |  |
| --- | --- | --- |
| **Όνομα** | **Επώνυμο** | **ΑΜ** |
| Αλέξανδρος-Παναγιώτης | Σοϊλεμεζίδης | 1067522 |
| Κυριακή | Λιούμη | 1067410 |

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

Αλέξανδρος-Παναγιώτης Σοϊλεμεζίδης

4 / 7 / 2022

Υπογραφή

Κυριακή Λιούμη

4 / 7 / 2022

**Συνημμένα αρχεία κώδικα**

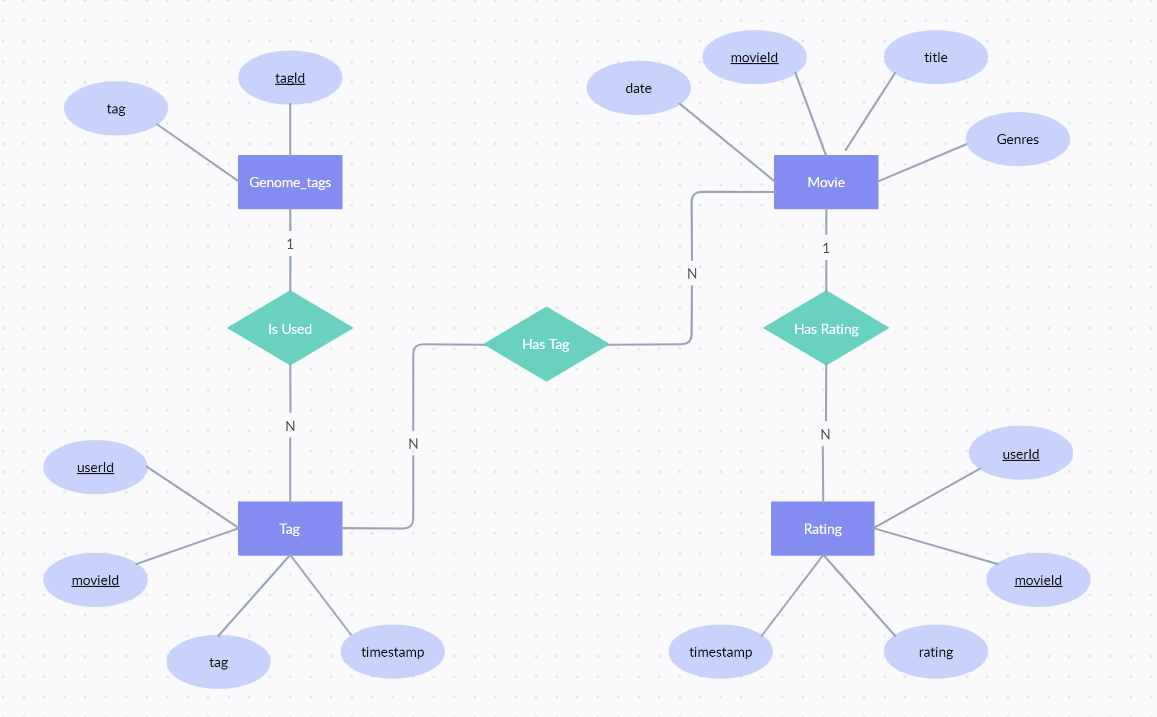
Μαζί με την παρούσα αναφορά υποβάλλουμε τα παρακάτω αρχεία κώδικα

|  |  |  |
| --- | --- | --- |
| **Αρχείο** | **Αφορά το ερώτημα** | **Περιγραφή/Σχόλιο** |
| connect\_db.py | 2,3 | Σύνδεση με cassandra |
| load.py | 2,3 | Φόρτωση των csv, επεξεργασία και δημιουργία νέων αρχείων, τα οποία περιλαμβάνουν ό,τι χρειάζεται για κάθε query |
| main.py | 2,3 | Δημιουργία table για κάθε query και insert |

Τεχνικά χαρακτηριστικά περιβάλλοντος λειτουργίας

Τα τρέξαμε σε Astra DB

Ερώτημα 1: Σχεδιασμός ΒΔ

Εννοιολογικό Μοντέλο

Εξηγήσεις: Φτιάξαμε 4 οντότητες, όσα και τα csv που μας δόθηκαν, με τα γνωρίσματα που περιέχονται σε κάθε αρχείο csv. Τα κλειδιά φαίνονται με υπογράμμιση. Για τις συσχετίσεις έχουμε ότι 1 ταινία θα έχει πολλά (Ν) ratings, 1 genome tag έχει πολλά (Ν) tags και πολλές (Ν) ταινίες έχουν πολλά (Ν) tags.

Application Workflow

Εικόνα που περιέχει κείμενο, ουρανός, στιγμιότυπο οθόνης

Περιγραφή που δημιουργήθηκε αυτόματα

Εξηγήσεις: Χωρίσαμε το query 3 σε δύο μέρη. Στο Q3.1 έχουμε τις ταινίες ανά κατηγορία, ταξινομημένες με βάση την μέση βαθμολογία και στο Q3.2 έχουμε τις ταινίες ανά κατηγορία, ταξινομημένες με βάση το έτος που βγήκαν. Καθώς το Q4 μας φάνηκε το πιο ειδικό από όλα, κάναμε τις συνδέσεις όπως φαίνεται παραπάνω.

Chebotko Diagram

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, εσωτερικό

Περιγραφή που δημιουργήθηκε αυτόματα

Εξηγήσεις: Για κάθε query έχουμε φτιάξει έναν πίνακα. Όπου ‘Κ ’ είναι το κλειδί και όπου ‘C’ είναι η ταξινόμηση που χρησιμοποιούμε. DESC για φθίνουσα και ASC για αύξουσα. Οι συνδέσεις εξηγούνται παραπάνω.

Ερώτημα 2: Ερωτήματα DDL

|  |  |
| --- | --- |
| **Keyspace.table** | Movies.movies\_by\_rating |
| **DDL statement** | CREATE TABLE movies\_by\_rating (movieId int, title text, rating float, stamp timestamp, PRIMARY KEY ((movieId), stamp) ) WITH comment = 'Q1. Find best rated movies on specific timeframe' AND CLUSTERING ORDER BY (stamp DESC); |

|  |  |
| --- | --- |
| **Keyspace.table** | Movies.movies\_by\_keyword |
| **DDL statement** | CREATE TABLE movies\_by\_keyword (movieId int, title text, rating float, PRIMARY KEY ((movieId), title) ) WITH comment = 'Q2. Find movies by using keywords'; |

|  |  |
| --- | --- |
| **Keyspace.table** | Movies.movies\_by\_genre\_rating |
| **DDL statement** | CREATE TABLE movies\_by\_genre\_rating (movieId int, title text, genres set, rating float, PRIMARY KEY ((movieId), genres, rating) ) WITH comment = 'Q3\_1. Find movies by genre, sorted by rating' AND CLUSTERING ORDER BY (genres ASC, rating DESC); |

|  |  |
| --- | --- |
| **Keyspace.table** | Movies.movies\_by\_genre\_date |
| **DDL statement** | CREATE TABLE movies\_by\_genres(release\_date) (movieId int, title text, genres text, date int, PRIMARY KEY ((movieId), genres, date) ) WITH comment = 'Q3\_2. Find movies by genre, sorted by release date' AND CLUSTERING ORDER BY (genres ASC, date DESC); |

|  |  |
| --- | --- |
| **Keyspace.table** | Movies.movie\_info |
| **DDL statement** | CREATE TABLE movie\_info (movieId int, title text, genres set, rating float, tag text, PRIMARY KEY ((movieId), tag) ) WITH comment = 'Q4. View more info about a movie'; |

|  |  |
| --- | --- |
| **Keyspace.table** | Movies.movies\_by\_tag |
| **DDL statement** | CREATE TABLE movies\_by\_tag (movieId int, title text, rating float, tag text, PRIMARY KEY ((movieId), tag) ) WITH comment = 'Q5. View movies relevant to a tag' AND CLUSTERING ORDER BY (tag ASC); |

Εξηγήσεις: Αρχικά είχαμε σκοπό, όπως αναφέρετε στην εκφώνηση, να φορτώσουμε τα δεδομένα μας στη βάση δεδομένων AstraDB μέσω python. Ωστόσο, παρά τις διαφορετικές μεθόδους που δοκιμάσαμε (σειριακά και μέσω batch query preperation), τα δεδομένα των πινάκων μας ήταν απαγορευτικά μεγάλα (20million+ rows) το οποίο είχε ως αποτέλεσμα πάντα είτε η σύνδεση να κάνει time out, είτε ο χρόνος φόρτωσης των δεδομένων να είναι της τάξης των εκατοντάδων ωρών. Για το λόγο αυτό δυστυχώς δεν καταφέραμε να χρησιμοποιήσουμε τη βάση δεδομέων AstraDB, και συνεπώς δε μπορούμε να παραθέσουμε screenshots, αποτελέσματα, και δεν είναι δυνατό να ολοκληρώσουμε το 4ο ερώτημα.

Ερώτημα 3: Απαντήσεις ερωτημάτων

|  |  |
| --- | --- |
| **Ερώτημα** | **Απάντηση** |
| Εμφάνιση των 30 ταινιών με την υψηλότερη μέση βαθμολογία μεταξύ 01/01/2015 και 15/01/2015 | SELECT movieId, title, AVG(rating) FROM Movies.movies\_by\_rating WHERE date > ‘01-01-2015 AND date < ‘15-01-2015’ ORDER BY rating DESC LIMIT 30 ALLOW FILTERING; |
| Εμφάνιση όλων των λεπτομερειών για την ταινία Jumanji (κατηγορία, μέση βαθμολογία, top-5 ετικέτες | SELECT title, genres, rating, COUNT(tag) FROM Movies.movie\_info WHERE movieId = 2 ORDER BY tag DESC LIMIT 5; |
| Εμφάνιση των ταινιών της κατηγορίας “adventure” ταξινομημένες ως προς το έτος παραγωγής | SELECT movieId, title, date FROM Movies.movies\_by\_genre\_date WHERE genres CONTAINS ‘adventure’ ORDER BY date ASC; |
| Εμφάνιση των ταινιών που περιέχουν τη λέξη “star” | SELECT movieId, title FROM Movies.movies\_by\_keyword WHERE title = “\*star\*”; |
| Εμφάνιση των 20 ταινιών με την υψηλότερη μέση βαθμολογία για την ετικέτα “comedy” | SELECT movieId, title, AVG(rating) FROM Movies.movies\_by\_tag WHERE tag = ‘comedy’ ORDER BY rating DESC LIMIT 5 ALLOW FILTERING; |

Βιβλιογραφία

* <https://cassandra.apache.org/doc/latest/cassandra/data_modeling/data_modeling_queries.html>
* <https://cassandra.apache.org/doc/latest/cassandra/data_modeling/data_modeling_logical.html>
* <https://cassandra.apache.org/_/quickstart.html>