

Συστήματα Διαχείρισης Δεδομένων Μεγάλου Όγκου

Εργαστηριακή Άσκηση 2021/22

| Όνομα | Επώνυμο | ΑΜ |
|--------------|------------|---------|
| Κωνσταντίνος | Καμινάρης | 1041756 |
| Σωτήριος | Ταβιανάτος | 1043830 |

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
Καμινάρης Κωνσταντίνος

12/ 09 / 2022

Υπογραφή
Ταβιανάτος Σωτήριος

12 / 09 / 2022

Συνημμένα αρχεία κώδικα

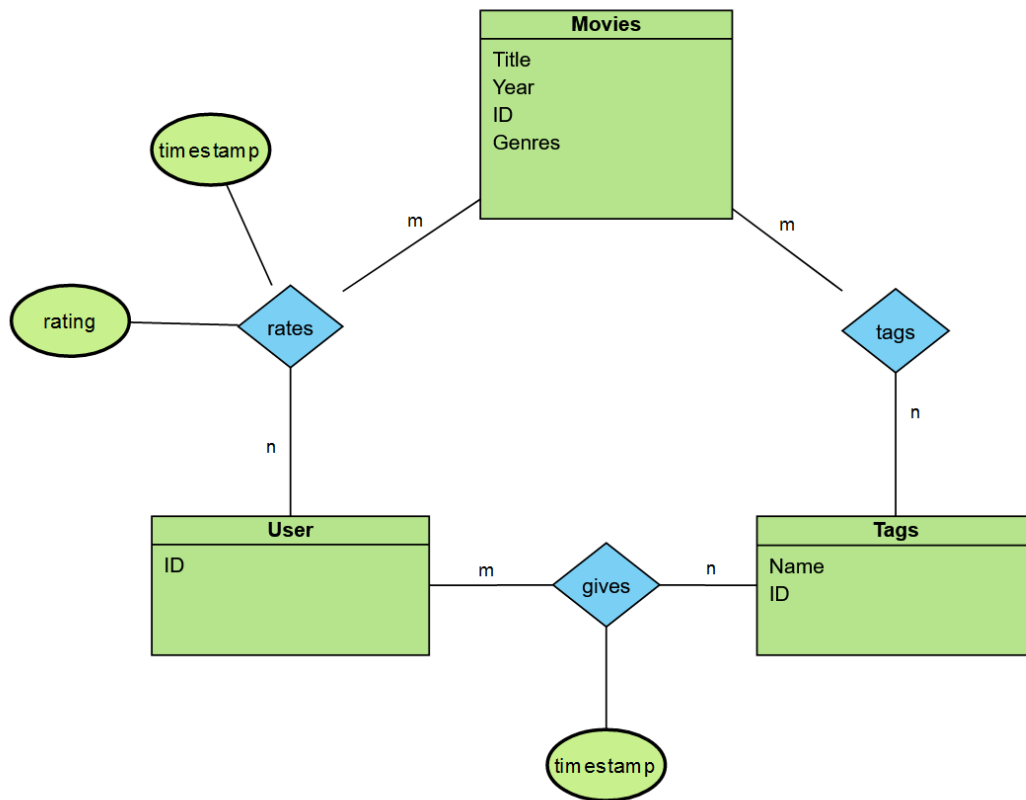
Μαζί με την παρούσα αναφορά υποβάλλουμε τα παρακάτω αρχεία κώδικα

| Αρχείο | Αφορά το ερώτημα | Περιγραφή/Σχόλιο |
|----------------|------------------|---------------------------|
| question1.py | 1 | Εκτελεί το ερ.1 |
| question2.py | 2 | Εκτελεί το ερ.2 |
| question3.py | 3 | Εκτελεί το ερ.3 |
| question4.py | 4 | Εκτελεί το ερ.4 |
| question5.py | 5 | Εκτελεί το ερ.5 |
| import1.py | 1 | Κατασκευάζει τον πίνακα 1 |
| import2.py | 2 | Κατασκευάζει τον πίνακα 2 |
| import3.py | 3 | Κατασκευάζει τον πίνακα 3 |
| import4.py | 4 | Κατασκευάζει τον πίνακα 4 |
| import5.py | 5 | Κατασκευάζει τον πίνακα 5 |
| movie_remap.py | Όλα τα ερωτήματα | Αναδιατάσει το movie.csv |

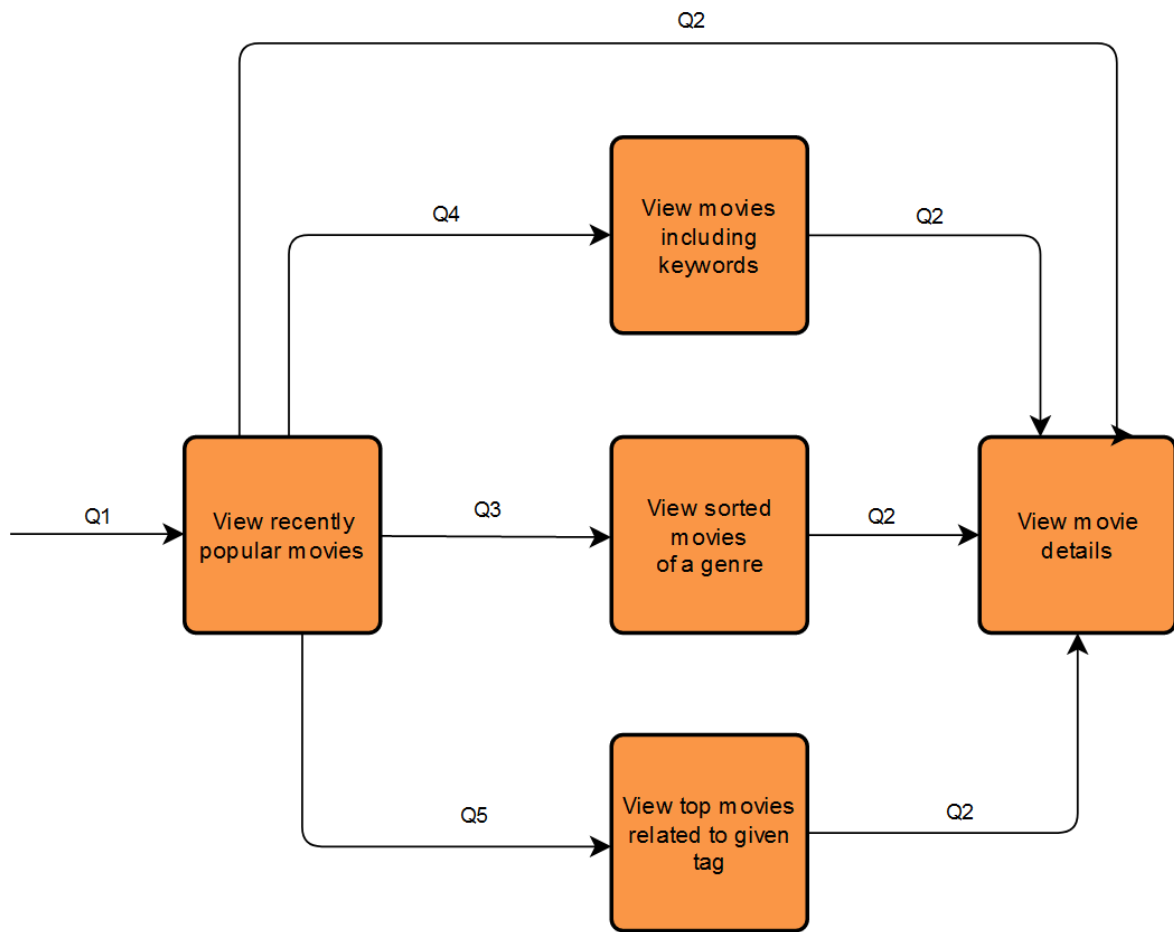
Τεχνικά χαρακτηριστικά περιβάλλοντος λειτουργίας

Χρησιμοποιήθηκε μόνο το σύστημα Astra DB για την εκπόνηση της εργασίας.

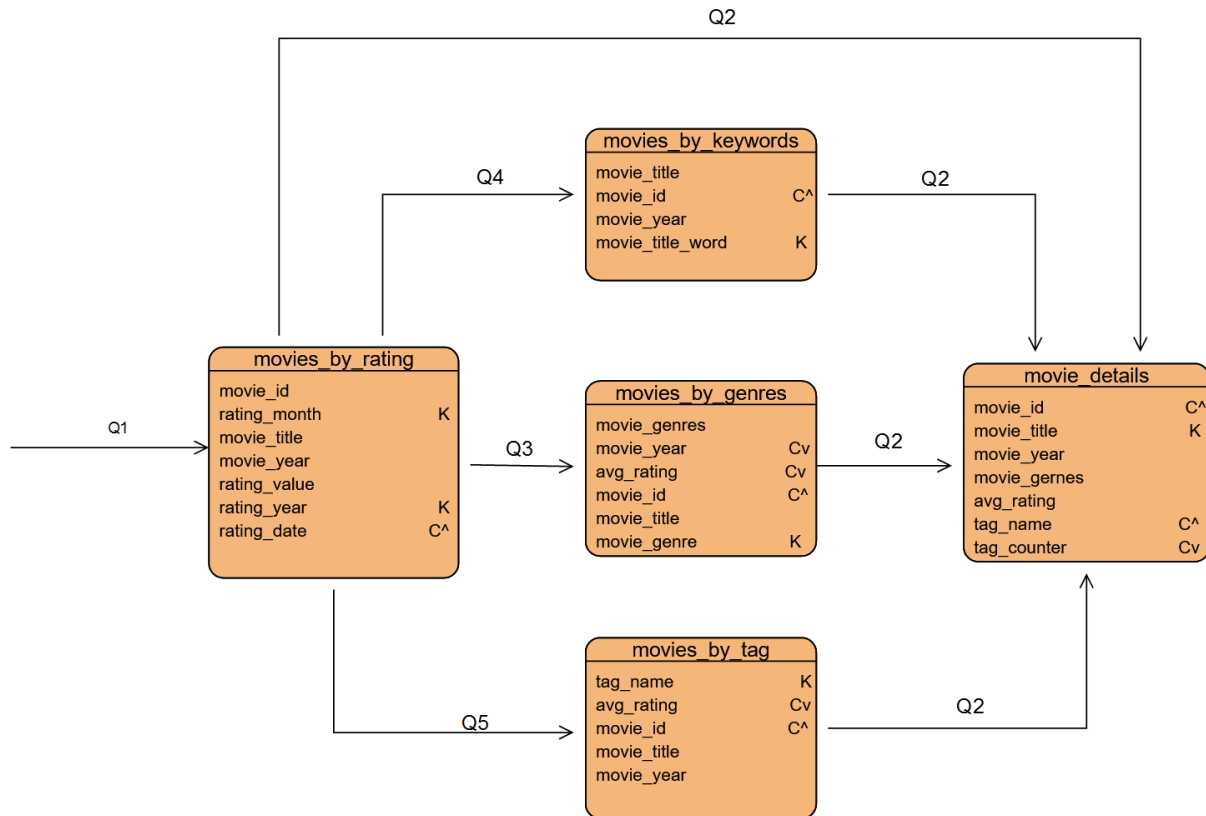
Ερώτημα 1: Σχεδιασμός ΒΔ



Conceptual Data Model



Application Workflow

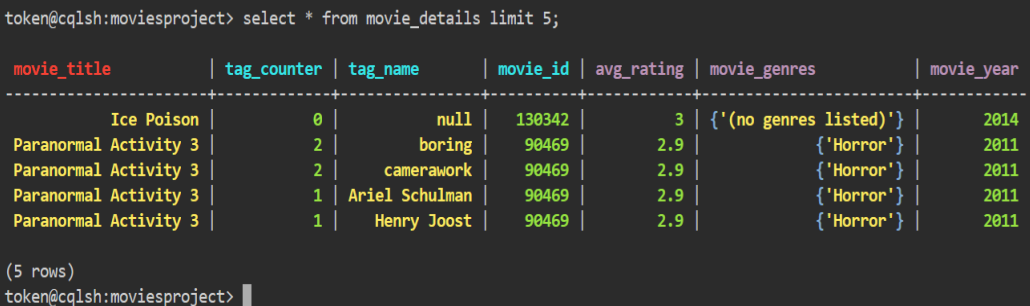


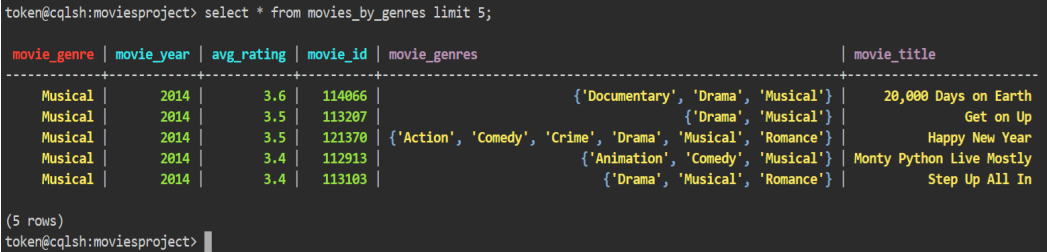
Chebotko Diagram

Όσον αφορά το Chebotko Diagram έχουμε αφαιρέσει το *user_id* σαν στήλη γιατί δεν είναι χρήσιμο για τις ανάγκες της εφαρμογής. Επίσης οι στήλες *rating_month/rating_year*, *movie_title_word* και *movie_genre* στους πίνακες *movies_by_rating*, *movies_by_keywords* και *movies_by_genres* αντίστοιχα αποτελούν στήλες πλεονασμού για να μπορούμε να τρέξουμε ερωτήματα τύπου EQUALITY με το PRIMARY KEY των πινάκων στο περιβάλλον της Cassandra. Τέλος στον πίνακα *movie_details* έχουμε δημιουργήσει τη στήλη *tag_counter* για να μπορούμε να εμφανίζουμε τα top tags για τις ανάγκες του ερωτήματος.

Ερώτημα 2: Ερωτήματα DDL

| Keyspace | moviesproject.movies_by_rating | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---------------|--|---------------------------------|--------------|-------------------------|------------|--------------|------------|--------------|------|---|---------------------------------|------|-------------------------|------|---|------|---|---------------------------------|------|--------------------|------|---|------|---|---------------------------------|------|--------------|------|---|------|---|---------------------------------|------|--------|------|---|------|---|---------------------------------|------|-------------------|------|---|
| DDL statement | CREATE TABLE movies_by_rating (rating_year int, rating_month int, rating_date timestamp, movie_id int, movie_title text, movie_year int, rating_value float, PRIMARY KEY ((rating_year, rating_month), rating_date)) WITH CLUSTERING ORDER BY (rating_date ASC); | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Screenshot | <p>The screenshot shows a terminal window with a CQL query and its results. The query is: <code>token@cqlsh:moviesproject> select * from movies_by_rating limit 5;</code> The results are displayed in a table with 7 columns: <code>rating_year</code>, <code>rating_month</code>, <code>rating_date</code>, <code>movie_id</code>, <code>movie_title</code>, <code>movie_year</code>, and <code>rating_value</code>. The results show 5 rows of data, all with <code>rating_year</code> 2004 and <code>rating_month</code> 8. The <code>movie_title</code> column is truncated in the output.</p> <table><thead><tr><th>rating_year</th><th>rating_month</th><th>rating_date</th><th>movie_id</th><th>movie_title</th><th>movie_year</th><th>rating_value</th></tr></thead><tbody><tr><td>2004</td><td>8</td><td>2004-08-01 00:00:10.000000+0000</td><td>1025</td><td>Sword in the Stone, The</td><td>1963</td><td>4</td></tr><tr><td>2004</td><td>8</td><td>2004-08-01 00:00:25.000000+0000</td><td>2353</td><td>Enemy of the State</td><td>1998</td><td>3</td></tr><tr><td>2004</td><td>8</td><td>2004-08-01 00:00:31.000000+0000</td><td>3421</td><td>Animal House</td><td>1978</td><td>3</td></tr><tr><td>2004</td><td>8</td><td>2004-08-01 00:00:45.000000+0000</td><td>3499</td><td>Misery</td><td>1990</td><td>4</td></tr><tr><td>2004</td><td>8</td><td>2004-08-01 00:01:01.000000+0000</td><td>2194</td><td>Untouchables, The</td><td>1987</td><td>4</td></tr></tbody></table> <p>(5 rows) token@cqlsh:moviesproject></p> | rating_year | rating_month | rating_date | movie_id | movie_title | movie_year | rating_value | 2004 | 8 | 2004-08-01 00:00:10.000000+0000 | 1025 | Sword in the Stone, The | 1963 | 4 | 2004 | 8 | 2004-08-01 00:00:25.000000+0000 | 2353 | Enemy of the State | 1998 | 3 | 2004 | 8 | 2004-08-01 00:00:31.000000+0000 | 3421 | Animal House | 1978 | 3 | 2004 | 8 | 2004-08-01 00:00:45.000000+0000 | 3499 | Misery | 1990 | 4 | 2004 | 8 | 2004-08-01 00:01:01.000000+0000 | 2194 | Untouchables, The | 1987 | 4 |
| rating_year | rating_month | rating_date | movie_id | movie_title | movie_year | rating_value | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2004 | 8 | 2004-08-01 00:00:10.000000+0000 | 1025 | Sword in the Stone, The | 1963 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2004 | 8 | 2004-08-01 00:00:25.000000+0000 | 2353 | Enemy of the State | 1998 | 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2004 | 8 | 2004-08-01 00:00:31.000000+0000 | 3421 | Animal House | 1978 | 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2004 | 8 | 2004-08-01 00:00:45.000000+0000 | 3499 | Misery | 1990 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2004 | 8 | 2004-08-01 00:01:01.000000+0000 | 2194 | Untouchables, The | 1987 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Keyspace | moviesproject.movie_details | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------------------|--|----------------|-------------|------------|-------------------------|------------|--------------|------------|------------|---|------|--------|---|-------------------------|------|-----------------------|---|--------|-------|-----|------------|------|-----------------------|---|------------|-------|-----|------------|------|-----------------------|---|----------------|-------|-----|------------|------|-----------------------|---|-------------|-------|-----|------------|------|
| DDL statement | CREATE TABLE movie_details (movie_title text, tag_counter int, tag_name text, movie_id int, avg_rating float, movie_genres frozen<set<text>>, movie_year int, PRIMARY KEY (movie_title, tag_counter, tag_name, movie_id)) WITH CLUSTERING ORDER BY (tag_counter DESC, tag_name ASC, movie_id ASC); | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Screenshot |  <pre>token@cqlsh:moviesproject> select * from movie_details limit 5;</pre> <table><thead><tr><th>movie_title</th><th>tag_counter</th><th>tag_name</th><th>movie_id</th><th>avg_rating</th><th>movie_genres</th><th>movie_year</th></tr></thead><tbody><tr><td>Ice Poison</td><td>0</td><td>null</td><td>130342</td><td>3</td><td>{'(no genres listed)')}</td><td>2014</td></tr><tr><td>Paranormal Activity 3</td><td>2</td><td>boring</td><td>90469</td><td>2.9</td><td>{'Horror'}</td><td>2011</td></tr><tr><td>Paranormal Activity 3</td><td>2</td><td>camerawork</td><td>90469</td><td>2.9</td><td>{'Horror'}</td><td>2011</td></tr><tr><td>Paranormal Activity 3</td><td>1</td><td>Ariel Schulman</td><td>90469</td><td>2.9</td><td>{'Horror'}</td><td>2011</td></tr><tr><td>Paranormal Activity 3</td><td>1</td><td>Henry Joost</td><td>90469</td><td>2.9</td><td>{'Horror'}</td><td>2011</td></tr></tbody></table> <pre>(5 rows) token@cqlsh:moviesproject></pre> | movie_title | tag_counter | tag_name | movie_id | avg_rating | movie_genres | movie_year | Ice Poison | 0 | null | 130342 | 3 | {'(no genres listed)')} | 2014 | Paranormal Activity 3 | 2 | boring | 90469 | 2.9 | {'Horror'} | 2011 | Paranormal Activity 3 | 2 | camerawork | 90469 | 2.9 | {'Horror'} | 2011 | Paranormal Activity 3 | 1 | Ariel Schulman | 90469 | 2.9 | {'Horror'} | 2011 | Paranormal Activity 3 | 1 | Henry Joost | 90469 | 2.9 | {'Horror'} | 2011 |
| movie_title | tag_counter | tag_name | movie_id | avg_rating | movie_genres | movie_year | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ice Poison | 0 | null | 130342 | 3 | {'(no genres listed)')} | 2014 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Paranormal Activity 3 | 2 | boring | 90469 | 2.9 | {'Horror'} | 2011 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Paranormal Activity 3 | 2 | camerawork | 90469 | 2.9 | {'Horror'} | 2011 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Paranormal Activity 3 | 1 | Ariel Schulman | 90469 | 2.9 | {'Horror'} | 2011 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Paranormal Activity 3 | 1 | Henry Joost | 90469 | 2.9 | {'Horror'} | 2011 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Keyspace | moviesproject.movies_by_genres | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---------------|---|-------------|------------|--|--------------------------|--------------|-------------|---------|------|-----|--------|-------------------------------------|----------------------|---------|------|-----|--------|----------------------|-----------|---------|------|-----|--------|--|----------------|---------|------|-----|--------|------------------------------------|--------------------------|---------|------|-----|--------|---------------------------------|----------------|
| DDL statement | CREATE TABLE movies_by_genres (movie_genre text, movie_year int, avg_rating float, movie_id int, movie_genres frozen<set<text>>, movie_title text, PRIMARY KEY (movie_genre, movie_year, avg_rating, movie_id)) WITH CLUSTERING ORDER BY (movie_year DESC, avg_rating DESC, movie_id ASC); | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Screenshot |  <pre>token@cqlsh:moviesproject> select * from movies_by_genres limit 5;</pre> <table><thead><tr><th>movie_genre</th><th>movie_year</th><th>avg_rating</th><th>movie_id</th><th>movie_genres</th><th>movie_title</th></tr></thead><tbody><tr><td>Musical</td><td>2014</td><td>3.6</td><td>114066</td><td>{'Documentary', 'Drama', 'Musical'}</td><td>20,000 Days on Earth</td></tr><tr><td>Musical</td><td>2014</td><td>3.5</td><td>113207</td><td>{'Drama', 'Musical'}</td><td>Get on Up</td></tr><tr><td>Musical</td><td>2014</td><td>3.5</td><td>121370</td><td>{'Action', 'Comedy', 'Crime', 'Drama', 'Musical', 'Romance'}</td><td>Happy New Year</td></tr><tr><td>Musical</td><td>2014</td><td>3.4</td><td>112913</td><td>{'Animation', 'Comedy', 'Musical'}</td><td>Monty Python Live Mostly</td></tr><tr><td>Musical</td><td>2014</td><td>3.4</td><td>113103</td><td>{'Drama', 'Musical', 'Romance'}</td><td>Step Up All In</td></tr></tbody></table> <p>(5 rows)</p> <pre>token@cqlsh:moviesproject></pre> | movie_genre | movie_year | avg_rating | movie_id | movie_genres | movie_title | Musical | 2014 | 3.6 | 114066 | {'Documentary', 'Drama', 'Musical'} | 20,000 Days on Earth | Musical | 2014 | 3.5 | 113207 | {'Drama', 'Musical'} | Get on Up | Musical | 2014 | 3.5 | 121370 | {'Action', 'Comedy', 'Crime', 'Drama', 'Musical', 'Romance'} | Happy New Year | Musical | 2014 | 3.4 | 112913 | {'Animation', 'Comedy', 'Musical'} | Monty Python Live Mostly | Musical | 2014 | 3.4 | 113103 | {'Drama', 'Musical', 'Romance'} | Step Up All In |
| movie_genre | movie_year | avg_rating | movie_id | movie_genres | movie_title | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Musical | 2014 | 3.6 | 114066 | {'Documentary', 'Drama', 'Musical'} | 20,000 Days on Earth | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Musical | 2014 | 3.5 | 113207 | {'Drama', 'Musical'} | Get on Up | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Musical | 2014 | 3.5 | 121370 | {'Action', 'Comedy', 'Crime', 'Drama', 'Musical', 'Romance'} | Happy New Year | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Musical | 2014 | 3.4 | 112913 | {'Animation', 'Comedy', 'Musical'} | Monty Python Live Mostly | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Musical | 2014 | 3.4 | 113103 | {'Drama', 'Musical', 'Romance'} | Step Up All In | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Keyspace | moviesproject.movies_by_keywords | | | | | | | | | | | | | | | | | | | | | | | | |
|------------------|--|--|------------|-------------|------------|--------|-------|-----------------------------|------|-------|-------|--|------|----------|-----|------------------------------------|------|-------|-------|--------------|------|--------|-------|-----------------------|------|
| DDL statement | CREATE TABLE movies_by_keywords (movie_title_word text, movie_id int, movie_title text, movie_year int, PRIMARY KEY (movie_title_word, movie_id)) WITH CLUSTERING ORDER BY (movie_id ASC); | | | | | | | | | | | | | | | | | | | | | | | | |
| Screenshot | <pre>token@cqlsh:moviesproject> select * from movies_by_keywords limit 5;</pre> <table><thead><tr><th>movie_title_word</th><th>movie_id</th><th>movie_title</th><th>movie_year</th></tr></thead><tbody><tr><td>anseon</td><td>45200</td><td>Coast Guard, The Hae anseon</td><td>2002</td></tr><tr><td>culpa</td><td>90552</td><td>Blame 6 Films to Keep You Awake Películas para no dormir: La culpa</td><td>2006</td></tr><tr><td>libertad</td><td>632</td><td>Land and Freedom Tierra y libertad</td><td>1995</td></tr><tr><td>Bucks</td><td>33988</td><td>Twenty Bucks</td><td>1993</td></tr><tr><td>Script</td><td>81090</td><td>Tales from the Script</td><td>2009</td></tr></tbody></table> <p>(5 rows)</p> <pre>token@cqlsh:moviesproject> </pre> | movie_title_word | movie_id | movie_title | movie_year | anseon | 45200 | Coast Guard, The Hae anseon | 2002 | culpa | 90552 | Blame 6 Films to Keep You Awake Películas para no dormir: La culpa | 2006 | libertad | 632 | Land and Freedom Tierra y libertad | 1995 | Bucks | 33988 | Twenty Bucks | 1993 | Script | 81090 | Tales from the Script | 2009 |
| movie_title_word | movie_id | movie_title | movie_year | | | | | | | | | | | | | | | | | | | | | | |
| anseon | 45200 | Coast Guard, The Hae anseon | 2002 | | | | | | | | | | | | | | | | | | | | | | |
| culpa | 90552 | Blame 6 Films to Keep You Awake Películas para no dormir: La culpa | 2006 | | | | | | | | | | | | | | | | | | | | | | |
| libertad | 632 | Land and Freedom Tierra y libertad | 1995 | | | | | | | | | | | | | | | | | | | | | | |
| Bucks | 33988 | Twenty Bucks | 1993 | | | | | | | | | | | | | | | | | | | | | | |
| Script | 81090 | Tales from the Script | 2009 | | | | | | | | | | | | | | | | | | | | | | |

| Keyspace | moviesproject.movies_by_tag | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----------------------------|--|----------|-----------------------|------------|-------------|------------|----------------------------|-----|-------|------|------|----------------------------|-----|-----|------------------|------|-----------|-----|------|-----------|------|------------|-----|-------|-------------|------|------------|-----|-------|-----------------------|------|
| DDL statement | CREATE TABLE movies_by_tag (tag_name text, avg_rating float, movie_id int, movie_title text, movie_year int, PRIMARY KEY (tag_name, avg_rating, movie_id)) WITH CLUSTERING ORDER BY (avg_rating DESC, movie_id ASC); | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Screenshot | <pre>token@cqlsh:moviesproject> select * from movies_by_tag limit 5;</pre> <table><tr><th>tag_name</th><th>avg_rating</th><th>movie_id</th><th>movie_title</th><th>movie_year</th></tr><tr><td>compareTo:Last Action Hero</td><td>3.4</td><td>63222</td><td>JCVD</td><td>2008</td></tr><tr><td>compareTo:Last Action Hero</td><td>2.8</td><td>485</td><td>Last Action Hero</td><td>1993</td></tr><tr><td>daredevil</td><td>2.6</td><td>6157</td><td>Daredevil</td><td>2003</td></tr><tr><td>loose ends</td><td>3.8</td><td>36529</td><td>Lord of War</td><td>2005</td></tr><tr><td>loose ends</td><td>3.6</td><td>95088</td><td>Safety Not Guaranteed</td><td>2012</td></tr></table> <pre>(5 rows) token@cqlsh:moviesproject> </pre> | tag_name | avg_rating | movie_id | movie_title | movie_year | compareTo:Last Action Hero | 3.4 | 63222 | JCVD | 2008 | compareTo:Last Action Hero | 2.8 | 485 | Last Action Hero | 1993 | daredevil | 2.6 | 6157 | Daredevil | 2003 | loose ends | 3.8 | 36529 | Lord of War | 2005 | loose ends | 3.6 | 95088 | Safety Not Guaranteed | 2012 |
| tag_name | avg_rating | movie_id | movie_title | movie_year | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| compareTo:Last Action Hero | 3.4 | 63222 | JCVD | 2008 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| compareTo:Last Action Hero | 2.8 | 485 | Last Action Hero | 1993 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| daredevil | 2.6 | 6157 | Daredevil | 2003 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| loose ends | 3.8 | 36529 | Lord of War | 2005 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| loose ends | 3.6 | 95088 | Safety Not Guaranteed | 2012 | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Ερώτημα 3: Απαντήσεις ερωτημάτων

| Ερώτημα | Απάντηση |
|--|---|
| Εμφάνιση των 30 ταινιών με την υψηλότερη μέση βαθμολογία μεταξύ 01/01/2015 και 15/01/2015 | <pre> movie_title,movie_year,avgRating "To Wong Foo, Thanks for Everything! Julie Newmar",1995,5.0 Circle of Friends,1995,5.0 "Wonderful, Horrible Life of Leni Riefenstahl, The Macht der Bilder: Leni Riefenstahl, Die",1993,5.0 Judgment Night,1993,5.0 Naked,1993,5.0 </pre> |
| Εμφάνιση όλων των λεπτομερειών για την ταινία Jumanji (κατηγορία, μέση βαθμολογία, top-5 ετικέτες) | <pre> movie_id, movie_title, movie_year, movie_genres, avg_rating, top5_tags 2,Jumanji,1995,"['Adventure','Children','Fantasy']",3.2,"[RobinWilliams,fantasy,time travel,animals,board game]" </pre> |
| Εμφάνιση των ταινιών της κατηγορίας “adventure” ταξινομημένες ως προς το έτος παραγωγής | <pre> movie_id, movie_title, movie_year, avg_rating, movie_genres 119145,Kingsman: The Secret Service,2015,3.6,['Action','Adventure','Comedy','Crime'] 117466,In the Heart of the Sea,2015,3.5,['Action','Adventure','Drama'] 130073,Cinderella,2015,3.4,['Adventure','Children','Drama','Sci-Fi'] 130842,Power/Rangers,2015,3.3,['Action','Adventure','Sci-Fi'] 113345,Jupiter Ascending,2015,2.8,['Action','Adventure','Sci-Fi'] </pre> |
| Εμφάνιση των ταινιών που περιέχουν τη λέξη “star” | <pre> movie_title, movie_id, movie_year "Star Maker, The Uomo delle stelle, L",124,1995 Star Wars: Episode IV - A New Hope,260,1977 Star Trek: Generations,329,1994 </pre> |

| | |
|--|---|
| | <div>Lone Star,800,1996</div> <div>North Star a.k.a. Tashunga,1118,1995</div> |
| Εμφάνιση των 20 ταινιών με την υψηλότερη μέση βαθμολογία για την ετικέτα “comedy”. | <div>movie_id, movie_title, movie_year, avg_rating</div> <div>86055,"Foster Brothers, The Süt kardesler",1976,5.0</div> <div>128991,Johnny Express,2014,4.4</div> <div>108134,"Lizard, The Marmoulak",2004,4.3</div> <div>296,Pulp Fiction,1994,4.2</div> <div>745,Wallace & Gromit: A Close Shave,1995,4.2</div> |

Ερώτημα 4Α: Χρόνοι εισαγωγής δεδομένων

| | Επίπεδο write consistency | | |
|--------------------|---------------------------|--------|-----|
| | ALL | QUORUM | ONE |
| movies_by_rating | 1:42:29 | 40:46 | - |
| movie_details | 1:57 | 1:12 | - |
| movies_by_genres | 0:30 | 0:17 | - |
| movies_by_keywords | 4:45 | 2:46 | - |
| movies_by_tag | 2:35 | 1:38 | - |
| Μέσος όρος | 22:27 | 9:16 | - |

Δεν καταφέραμε να κάνουμε write με consistency level ONE διότι όταν επιχειρούσαμε *dsbulk* με την επιλογή -cl ONE το bash μας απαντούσε με το εξής warning:

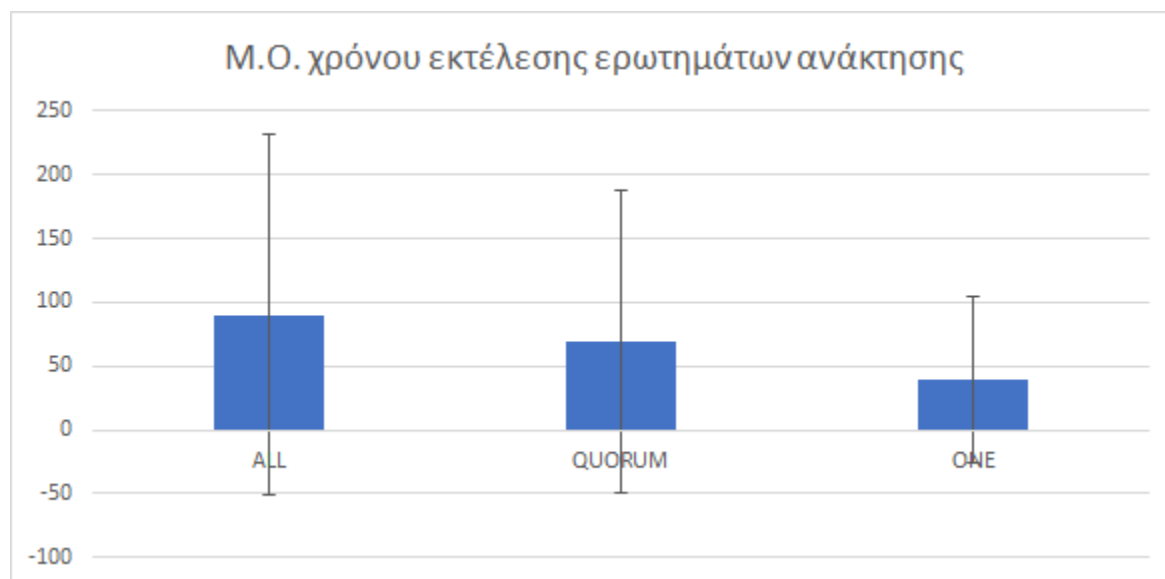
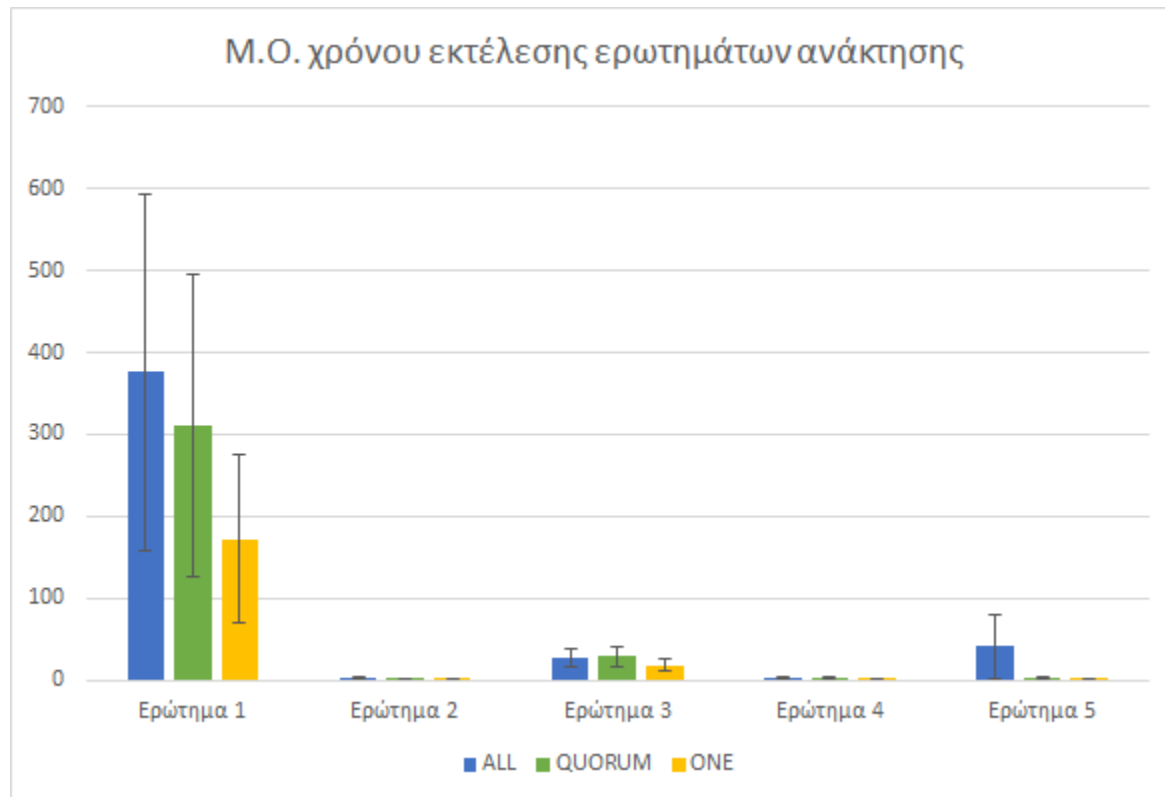
A cloud secure connect bundle was provided together with consistency level ONE, but selected operation performs writes: forcing default consistency level to LOCAL_QUORUM.

στο οποίο error δεν καταφέραμε να βρούμε λύση.

Ερώτημα 4Β: Χρόνοι ανάκτησης δεδομένων

| | Επίπεδο read consistency | | |
|------------|--------------------------|---------|---------|
| | ALL | QUORUM | ONE |
| Ερώτημα 1 | 376.2 | 310.2 | 171.8 |
| Ερώτημα 2 | 2.7 | 1.5 | 1.4 |
| Ερώτημα 3 | 27.9 | 29.2 | 18.7 |
| Ερώτημα 4 | 3.4 | 2.2 | 1.7 |
| Ερώτημα 5 | 41.2 | 2.2 | 1.4 |
| Μέσος όρος | 90.3 msec | 69 msec | 39 msec |

Ερώτημα 4Γ: Σχολιασμός αποτελεσμάτων



Οι γραφικές παραστάσεις δείχνουν πως ο χρόνος απόκρισης των ερωτημάτων μειώνεται ανάλογα με τη χαλάρωση του consistency level, κάτι το οποίο είναι αναμενόμενο καθώς μειώνεται το πλήθος των nodes που πρέπει να προσπελαστούν πριν αποκριθεί η βάση.

Πρακτικά η βάση δεδομένων πρέπει να έχει υψηλό partition tolerance μιας και χρησιμοποιεί τον παγκόσμιο ιστό για την αποθήκευση των δεδομένων. Αυτό σημαίνει πως όσο αυστηροποιείται το consistency, τόσο πέφτει το availability (λόγω του θεωρήματος CAP). Έτσι σε αυστηρότερο consistency level αντιστοιχεί μεγαλύτερος χρόνος απόκρισης.

Βιβλιογραφία

- docs.datastax.com
- [geeksforgeeks.org](https://www.geeksforgeeks.org)
- askubuntu.com/questions
- github.com/datastax/dsbulk
- cassandra.apache.org