

Συστήματα Διαχείρισης Δεδομένων Μεγάλου Όγκου

Εργαστηριακή Άσκηση 2021/22

Όνομα	Επώνυμο	ΑΜ
Αγάθη	Σκύρλα	1064888

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

Αγάθη Σκύρλα



19/09/2022

Υπογραφή

___ / ___ / 2022

Συνημμένα αρχεία κώδικα

Μαζί με την παρούσα αναφορά υποβάλλουμε τα παρακάτω αρχεία κώδικα

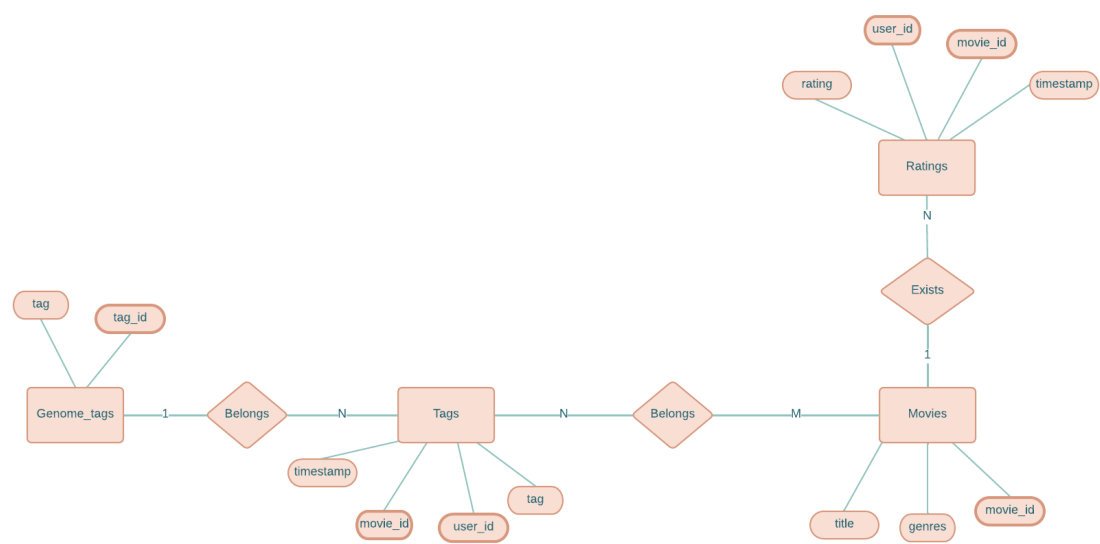
Αρχείο	Αφορά το ερώτημα	Περιγραφή/Σχόλιο
<i>query2_insert.py , query2_select.py</i>	2	<i>Αφορούν την εισαγωγή των δεδομένων στην βάση δεδομένων και την μέτρηση του χρόνου για την εκτέλεση των read consistency level</i>
<i>query3_insert.py , query3_select.py</i>	3	<i>Αφορούν την εισαγωγή των δεδομένων στην βάση δεδομένων και την μέτρηση του χρόνου για την εκτέλεση των read consistency level</i>
<i>query4_insert.py , query4_select.py</i>	4	<i>Αφορούν την εισαγωγή των δεδομένων στην βάση δεδομένων και την μέτρηση του χρόνου για την εκτέλεση των read consistency level</i>
<i>query5_insert.py , query5_select.py</i>	5	<i>Αφορούν την εισαγωγή των δεδομένων στην βάση δεδομένων και την μέτρηση του χρόνου για την εκτέλεση των read consistency level</i>

Τεχνικά χαρακτηριστικά περιβάλλοντος λειτουργίας

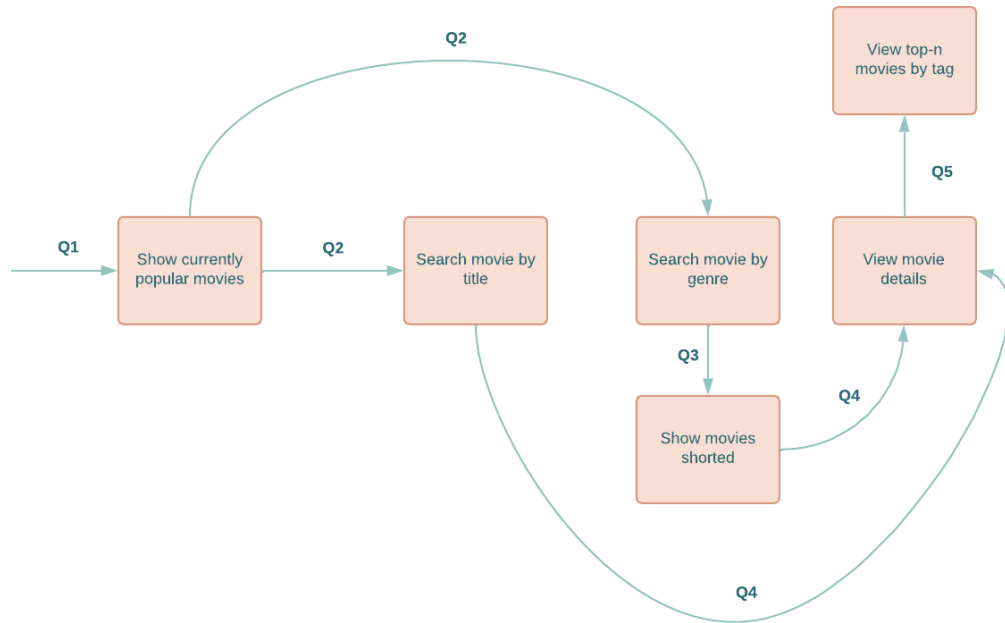
Χαρακτηριστικό	Τιμή
CPU model	Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 1.99 GHz
CPU clock speed	1.80GHz 1.99 GHz
Physical CPU cores	4
Logical CPU cores	8
RAM	8
Secondary Storage Type	

Ερώτημα 1: Σχεδιασμός ΒΔ

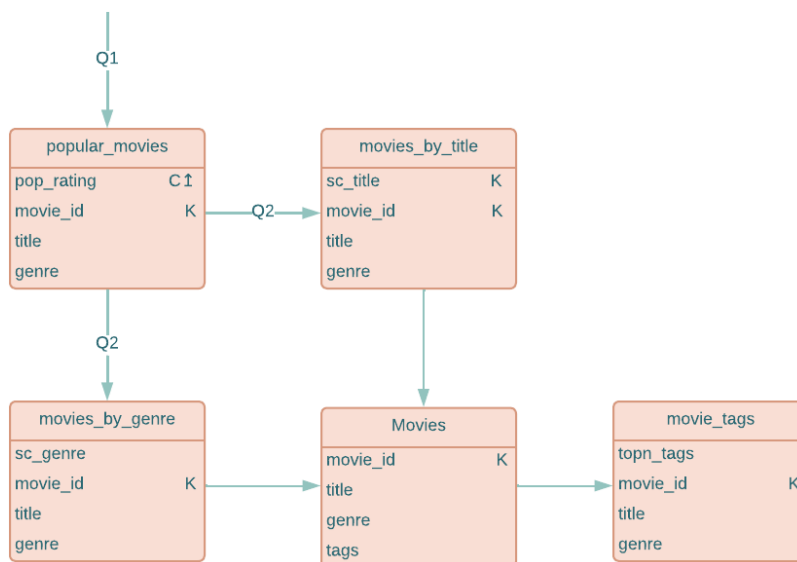
Εννοιολογικό μοντέλο:



Application Workflow:

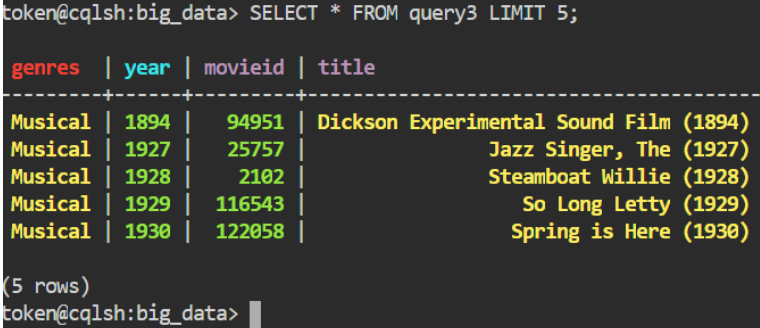


Chebotko diagram:



Ερώτημα 2: Ερωτήματα DDL

Keyspace	query2
DDL statement	<pre>CREATE TABLE big_data.query2 (movieid int PRIMARY KEY, genres text, rating float, tag text, title list<text>);</pre>
Screenshot	

Keyspace	query3
DDL statement	<pre>CREATE TABLE big_data.query3 (genres text, year int, movieid int, title text, PRIMARY KEY (genres, year)) WITH CLUSTERING ORDER BY (year ASC);</pre>
Screenshot	

Keyspace	query4
DDL statement	<pre>CREATE TABLE big_data.query4 (movieid int, rating float, genres list<text>, title list<text>,</pre>

	<i>PRIMARY KEY (movieid, rating)</i>);																								
Screenshot	<pre>token@cqlsh:big_data> SELECT * FROM query4 LIMIT 5;</pre> <table><tr><th>movieid</th><th>rating</th><th>genres</th><th>title</th></tr><tr><td>4317</td><td>2.70171</td><td>['Comedy', 'Romance']</td><td>['Love', 'Potion', '#9', '(1992)']</td></tr><tr><td>51678</td><td>3.55357</td><td>['Drama']</td><td>['Julius', 'Caesar', '(1953)']</td></tr><tr><td>77328</td><td>3.48113</td><td>['Crime', 'Drama', 'Mystery', 'Thriller']</td><td>['Red', 'Riding:', '1974', '(2009)']</td></tr><tr><td>3372</td><td>3.43692</td><td>['Action', 'War']</td><td>['Bridge', 'at', 'Remagen,', 'The', '(1969)']</td></tr><tr><td>96748</td><td>2.7</td><td>['Horror', 'Thriller']</td><td>['247°F', '(2011)']</td></tr></table> <p>(5 rows)</p> <pre>token@cqlsh:big_data></pre>	movieid	rating	genres	title	4317	2.70171	['Comedy', 'Romance']	['Love', 'Potion', '#9', '(1992)']	51678	3.55357	['Drama']	['Julius', 'Caesar', '(1953)']	77328	3.48113	['Crime', 'Drama', 'Mystery', 'Thriller']	['Red', 'Riding:', '1974', '(2009)']	3372	3.43692	['Action', 'War']	['Bridge', 'at', 'Remagen,', 'The', '(1969)']	96748	2.7	['Horror', 'Thriller']	['247°F', '(2011)']
movieid	rating	genres	title																						
4317	2.70171	['Comedy', 'Romance']	['Love', 'Potion', '#9', '(1992)']																						
51678	3.55357	['Drama']	['Julius', 'Caesar', '(1953)']																						
77328	3.48113	['Crime', 'Drama', 'Mystery', 'Thriller']	['Red', 'Riding:', '1974', '(2009)']																						
3372	3.43692	['Action', 'War']	['Bridge', 'at', 'Remagen,', 'The', '(1969)']																						
96748	2.7	['Horror', 'Thriller']	['247°F', '(2011)']																						

Keyspace	query5																														
DDL statement	<pre>CREATE TABLE big_data.query5 (tag text, rating float, genres list<text>, movieid int, title text, PRIMARY KEY (tag, rating)) WITH CLUSTERING ORDER BY (rating ASC);</pre>																														
Screenshot	<p>The screenshot shows a CQLSH terminal window with the command <code>token@cqlsh:big_data> SELECT * FROM query5 LIMIT 5;</code> and its output. The output is a table with 5 columns: tag, rating, genres, movieid, and title. The data is as follows:</p> <table><thead><tr><th>tag</th><th>rating</th><th>genres</th><th>movieid</th><th>title</th></tr></thead><tbody><tr><td>compareTo:Last Action Hero</td><td>2.77874</td><td>['Action', 'Adventure', 'Comedy', 'Fantasy']</td><td>485</td><td>Last Action Hero (1993)</td></tr><tr><td>dance</td><td>2.93889</td><td>['Drama', 'Romance']</td><td>2942</td><td>Flashdance (1983)</td></tr><tr><td>dance</td><td>3.09104</td><td>['Drama']</td><td>3791</td><td>Footloose (1984)</td></tr><tr><td>dance</td><td>3.17668</td><td>['Drama', 'Romance']</td><td>4054</td><td>Save the Last Dance (2001)</td></tr><tr><td>dance</td><td>3.20921</td><td>['Drama', 'Musical', 'Romance']</td><td>1088</td><td>Dirty Dancing (1987)</td></tr></tbody></table> <p>(5 rows) token@cqlsh:big_data></p>	tag	rating	genres	movieid	title	compareTo:Last Action Hero	2.77874	['Action', 'Adventure', 'Comedy', 'Fantasy']	485	Last Action Hero (1993)	dance	2.93889	['Drama', 'Romance']	2942	Flashdance (1983)	dance	3.09104	['Drama']	3791	Footloose (1984)	dance	3.17668	['Drama', 'Romance']	4054	Save the Last Dance (2001)	dance	3.20921	['Drama', 'Musical', 'Romance']	1088	Dirty Dancing (1987)
tag	rating	genres	movieid	title																											
compareTo:Last Action Hero	2.77874	['Action', 'Adventure', 'Comedy', 'Fantasy']	485	Last Action Hero (1993)																											
dance	2.93889	['Drama', 'Romance']	2942	Flashdance (1983)																											
dance	3.09104	['Drama']	3791	Footloose (1984)																											
dance	3.17668	['Drama', 'Romance']	4054	Save the Last Dance (2001)																											
dance	3.20921	['Drama', 'Musical', 'Romance']	1088	Dirty Dancing (1987)																											

Keyspace	<code>movies</code>																		
DDL statement	<code>CREATE TABLE big_data.movies (movieid int PRIMARY KEY, genres text, title text);</code>																		
Screenshot	<pre>token@cqlsh:big_data> SELECT * FROM movies LIMIT 5;</pre> <table><thead><tr><th>movieid</th><th>genres</th><th>title</th></tr></thead><tbody><tr><td>4317</td><td>Comedy Romance</td><td>Love Potion #9 (1992)</td></tr><tr><td>51678</td><td>Drama</td><td>Julius Caesar (1953)</td></tr><tr><td>77328</td><td>Crime Drama Mystery Thriller</td><td>Red Riding: 1974 (2009)</td></tr><tr><td>3372</td><td>Action War</td><td>Bridge at Remagen, The (1969)</td></tr><tr><td>96748</td><td>Horror Thriller</td><td>247°F (2011)</td></tr></tbody></table> <pre>(5 rows) token@cqlsh:big_data></pre>	movieid	genres	title	4317	Comedy Romance	Love Potion #9 (1992)	51678	Drama	Julius Caesar (1953)	77328	Crime Drama Mystery Thriller	Red Riding: 1974 (2009)	3372	Action War	Bridge at Remagen, The (1969)	96748	Horror Thriller	247°F (2011)
movieid	genres	title																	
4317	Comedy Romance	Love Potion #9 (1992)																	
51678	Drama	Julius Caesar (1953)																	
77328	Crime Drama Mystery Thriller	Red Riding: 1974 (2009)																	
3372	Action War	Bridge at Remagen, The (1969)																	
96748	Horror Thriller	247°F (2011)																	

Keyspace	tag
DDL statement	<pre>CREATE TABLE big_data.tag (userid int, movieid int,</pre>

	<pre>tag text, timestamp timestamp, PRIMARY KEY (userid, movieid));</pre>																								
Screenshot	<pre>token@cqlsh:big_data> SELECT * FROM tag LIMIT 5;</pre> <table><thead><tr><th>userid</th><th>movieid</th><th>tag</th><th>timestamp</th></tr></thead><tbody><tr><td>1584</td><td>45186</td><td>Good</td><td>2006-05-22 20:00:55.000000+0000</td></tr><tr><td>65053</td><td>4239</td><td>based on a true story</td><td>2012-10-05 23:20:36.000000+0000</td></tr><tr><td>65053</td><td>4239</td><td>cocaine</td><td>2012-10-05 23:20:18.000000+0000</td></tr><tr><td>65053</td><td>4239</td><td>dysfunctional family</td><td>2012-10-05 23:20:20.000000+0000</td></tr><tr><td>65053</td><td>4239</td><td>true story</td><td>2012-10-05 23:20:28.000000+0000</td></tr></tbody></table> <pre>(5 rows) token@cqlsh:big_data></pre>	userid	movieid	tag	timestamp	1584	45186	Good	2006-05-22 20:00:55.000000+0000	65053	4239	based on a true story	2012-10-05 23:20:36.000000+0000	65053	4239	cocaine	2012-10-05 23:20:18.000000+0000	65053	4239	dysfunctional family	2012-10-05 23:20:20.000000+0000	65053	4239	true story	2012-10-05 23:20:28.000000+0000
userid	movieid	tag	timestamp																						
1584	45186	Good	2006-05-22 20:00:55.000000+0000																						
65053	4239	based on a true story	2012-10-05 23:20:36.000000+0000																						
65053	4239	cocaine	2012-10-05 23:20:18.000000+0000																						
65053	4239	dysfunctional family	2012-10-05 23:20:20.000000+0000																						
65053	4239	true story	2012-10-05 23:20:28.000000+0000																						

Keyspace	genome_tags												
DDL statement	CREATE TABLE big_data.genome_tags (tagid int PRIMARY KEY, tag text);												
Screenshot	<pre>token@cqlsh:big_data> SELECT * FROM genome_tags LIMIT 5;</pre> <table> <thead> <tr> <th>tagid</th><th>tag</th></tr> </thead> <tbody> <tr> <td>769</td><td>paris</td></tr> <tr> <td>23</td><td>adapted from:comic</td></tr> <tr> <td>114</td><td>based on comic</td></tr> <tr> <td>660</td><td>modern fantasy</td></tr> <tr> <td>893</td><td>screwball</td></tr> </tbody> </table> <pre>(5 rows) token@cqlsh:big_data></pre>	tagid	tag	769	paris	23	adapted from:comic	114	based on comic	660	modern fantasy	893	screwball
tagid	tag												
769	paris												
23	adapted from:comic												
114	based on comic												
660	modern fantasy												
893	screwball												

Keyspace	rating																								
DDL statement	<pre>CREATE TABLE big_data.rating (userid int, movieid int, rating float, timestamp timestamp, PRIMARY KEY (userid, movieid)) WITH CLUSTERING ORDER BY (movieid ASC);</pre>																								
Screenshot	<pre>token@cqlsh:big_data> SELECT * FROM rating LIMIT 5;</pre> <table><thead><tr><th>userid</th><th>movieid</th><th>rating</th><th>timestamp</th></tr></thead><tbody><tr><td>769</td><td>1</td><td>5</td><td>1996-06-17 11:00:32.000000+0000</td></tr><tr><td>769</td><td>10</td><td>5</td><td>1996-06-17 11:00:11.000000+0000</td></tr><tr><td>769</td><td>11</td><td>3</td><td>1996-06-17 11:13:58.000000+0000</td></tr><tr><td>769</td><td>12</td><td>3</td><td>1996-07-23 15:59:32.000000+0000</td></tr><tr><td>769</td><td>17</td><td>5</td><td>1996-06-17 11:14:43.000000+0000</td></tr></tbody></table> <pre>(5 rows) token@cqlsh:big_data></pre>	userid	movieid	rating	timestamp	769	1	5	1996-06-17 11:00:32.000000+0000	769	10	5	1996-06-17 11:00:11.000000+0000	769	11	3	1996-06-17 11:13:58.000000+0000	769	12	3	1996-07-23 15:59:32.000000+0000	769	17	5	1996-06-17 11:14:43.000000+0000
userid	movieid	rating	timestamp																						
769	1	5	1996-06-17 11:00:32.000000+0000																						
769	10	5	1996-06-17 11:00:11.000000+0000																						
769	11	3	1996-06-17 11:13:58.000000+0000																						
769	12	3	1996-07-23 15:59:32.000000+0000																						
769	17	5	1996-06-17 11:14:43.000000+0000																						

Ερώτημα 3: Απαντήσεις ερωτημάτων

Ερώτημα	Απάντηση
Εμφάνιση των 30 ταινιών με την υψηλότερη μέση βαθμολογία μεταξύ 01/01/2015 και 15/01/2015	-
Εμφάνιση όλων των λεπτομερειών για την ταινία Jumanji (κατηγορία, μέση βαθμολογία, top-5 ετικέτες)	<i>movieid genres rating tag title</i> -----+-----+-----+-----+----- ----- 2 Adventure Children Fantasy 3.21198 scary ['Jumanji', '(1995)']
Εμφάνιση των ταινιών της κατηγορίας “adventure” ταξινομημένες ως προς το έτος παραγωγής	<i>genres year movieid title</i> -----+-----+-----+----- ----- Adventure 1902 32898 Trip to the Moon, A (Voyage dans la lune, Le) (1902) Adventure 1913 90339 Last Days of Pompeii, The (Gli ultimi giorni di Pompeii) (1913) Adventure 1914 84852 Judith of Bethulia (1914) Adventure 1915 69509 Vampires, Les (1915) Adventure 1916 91562 Judex (1916)
Εμφάνιση των ταινιών που περιέχουν τη λέξη “star”	<i>movieid rating genres title</i> -----+-----+-----+----- -----+----- ----- 60981 3 ['Action', 'Animation'] ['Fist', 'of', 'the', 'North', 'Star', '(1986)'] 25996 3.68343 ['Drama', 'Musical'] ['Star', 'Is', 'Born,', 'A', '(1954)'] 26285 3.47 ['Comedy', 'Sci-Fi', 'Thriller'] ['Dark', 'Star', '(1974)'] 115558 3.5 ['Animation', 'Children', 'Musical', 'Romance'] ['Amazon', 'Jack', '2:', 'The', 'Movie', 'Star', '(a.k.a.', 'Hugo', 'the', 'Movie', 'Star)', '(Jungledyret', '2', '-', 'den', 'store', 'filmhelt)', '(1996)'] 121211 0.5 ['Adventure', 'Comedy', 'Drama', 'War']

Εμφάνιση των 20 ταινιών με την υψηλότερη μέση βαθμολογία για την ετικέτα "comedy".	<i>tag</i> <i>rating</i> <i>genres</i> <i>movieid</i> <i>title</i>
	-----+-----+-----
	-----+-----+-----
	--
	Comedy 2.29272 ['Action', 'Comedy', 'Sci-Fi', 'Western'] 2701 Wild Wild West (1999)
	Comedy 2.66215 ['Comedy'] 585 Brady Bunch Movie, The (1995)
	Comedy 2.70276 ['Comedy', 'Thriller'] 784 Cable Guy, The (1996)
	Comedy 2.74796 ['Comedy', 'Horror', 'Mystery', 'Thriller'] 1717 Scream 2 (1997)
	Comedy 2.76049 ['Comedy'] 813 Larger Than Life (1996)

Παραθέτω και screenshot των αποτελεσμάτων :

```
token@cqlsh:big_data> SELECT * FROM query2 WHERE title CONTAINS 'Jumanji' ALLOW FILTERING;

movieid | genres | rating | tag | title
-----+-----+-----+-----+-----
2 | Adventure|Children|Fantasy | 3.21198 | scary | ['Jumanji', '(1995)']

(1 rows)
token@cqlsh:big_data>
```

!Γνωρίζω ήδη ότι δεν βγάζει 5 tags

```
token@cqlsh:big_data> SELECT * FROM query3 WHERE genres='Adventure' ORDER BY year ASC ALLOW FILTERING;

genres | year | movieid | title
-----+-----+-----+-----
Adventure | 1902 | 32898 | Trip to the Moon, A (Voyage dans la lune, Le) (1902)
Adventure | 1913 | 90339 | Last Days of Pompeii, The (Gli ultimi giorni di Pompeii) (1913)
Adventure | 1914 | 84852 | Judith of Bethulia (1914)
Adventure | 1915 | 69509 | Vampires, Les (1915)
Adventure | 1916 | 91562 | Judex (1916)

token@cqlsh> use big_data;
token@cqlsh:big_data> SELECT * FROM query4 WHERE title CONTAINS 'Star' ALLOW FILTERING;

movieid | rating | genres | title
-----+-----+-----+-----
60981 | 3 | ['Action', 'Animation'] | ['Fist', 'of', 'the', 'North', 'Star', '(1986)']
25996 | 3.68343 | ['Drama', 'Musical'] | ['Star', 'Is', 'Born', 'A', '(1954)']
26285 | 3.47 | ['Comedy', 'Sci-Fi', 'Thriller'] | ['Bark', 'Star', '(1974)']
115558 | 3.5 | ['Animation', 'Children', 'Musical', 'Romance'] | ['Amazon', 'Jack', '2:', 'The', 'Movie', 'Star', '(a.k.a.', 'Hugo', 'the', 'Movie', 'Star)', '(Jungl
edyret', '2', '-', 'den', 'store', 'filmhelt)', '(1996)']
121211 | 0.5 | ['Adventure', 'Comedy', 'Drama', 'War'] | ['Major', 'Movie', 'Star', '(2008)']

token@cqlsh:big_data> SELECT * FROM query5 WHERE tag='Comedy' LIMIT 20 ALLOW FILTERING;

tag | rating | genres | movieid | title
-----+-----+-----+-----+-----
Comedy | 2.29272 | ['Action', 'Comedy', 'Sci-Fi', 'Western'] | 2701 | Wild Wild West (1999)
Comedy | 2.66215 | ['Comedy'] | 585 | Brady Bunch Movie, The (1995)
Comedy | 2.70276 | ['Comedy', 'Thriller'] | 784 | Cable Guy, The (1996)
Comedy | 2.74796 | ['Comedy', 'Horror', 'Mystery', 'Thriller'] | 1717 | Scream 2 (1997)
Comedy | 2.76049 | ['Comedy'] | 813 | Larger Than Life (1996)
```

Ερώτημα 4Α: Χρόνοι εισαγωγής δεδομένων

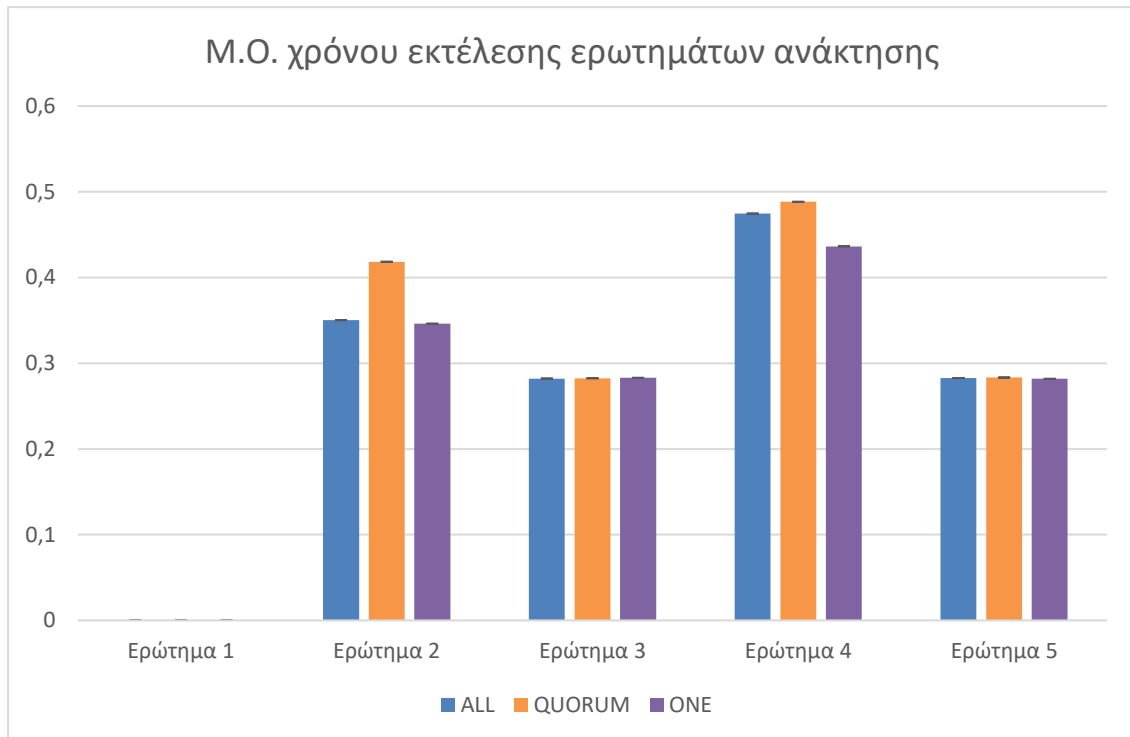
Οι μετρήσεις αυτές δεν έγιναν !

	Επίπεδο write consistency		
	ALL	QUORUM	ONE
query2	[χρόνος εκτέλεσης]	[χρόνος εκτέλεσης]	[χρόνος εκτέλεσης]
query3	[χρόνος εκτέλεσης]	[χρόνος εκτέλεσης]	[χρόνος εκτέλεσης]
query4
query5	[χρόνος εκτέλεσης]	[χρόνος εκτέλεσης]	[χρόνος εκτέλεσης]
Μέσος όρος			

Ερώτημα 4Β: Χρόνοι ανάκτησης δεδομένων

	Επίπεδο read consistency		
	ALL	QUORUM	ONE
Ερώτημα 1	-	-	-
Ερώτημα 2	0.35023216	0.41836447	0.34608349
Ερώτημα 3	0.2820806187	0.2825076528999 9	0.2816981550999 9
Ερώτημα 4	0.4745280857000 4	0.4883208457999 8	0.4745280857000 4
Ερώτημα 5	0.2827195974	0.2832948269999 9	0.2818070862999 8
Μέσος όρος	0.34739011545001	0.36812194892499	0.346029204275

Ερώτημα 4Γ: Σχολιασμός αποτελεσμάτων



Οι μετρήσεις για το ερώτημα 1 δεν έγιναν, αφού δεν υλοποιήθηκε το ερώτημα 1. Παρατηρούμε ότι οι μετρήσεις για τα ερωτήματα, ειδικά για το ερώτημα 3 και 5, αντικειμενικά δεν ανταπεξέρχονται σε αυτά που πρεσβεύουν κανονικά τα consistency levels. Πιστεύω για αυτό ευθύνεται το ότι στα αυτά τα δύο queries τα insert δεν έγιναν ολοκληρωμένα, λόγω του τεράστιου όγκου δεδομένων, το σύστημα μου δεν ανταποκρίθηκε. Κανονικά το επίπεδο Quorum αντιστοιχεί στο 51% των αντίγραφων nodes σε όλα τα κέντρα δεδομένων, και μπορεί να οδηγήσει σε αργό read. Ενώ το ALL αντιστοιχεί σε πολύ αργό read. Το level ONE είναι το πιο γρήγορο, αφού μία αντιγραφή node επιστρέφει τα δεδομένα.

Βιβλιογραφία

https://docs.datastax.com/en/developer/python-driver/3.23/getting_started/

<https://www.baeldung.com/cassandra-consistency-levels>