

Συστήματα Διαχείρισης Δεδομένων Μεγάλου Όγκου

Εργαστηριακή Άσκηση 2021/22

Όνομα	Επώνυμο	ΑΜ
ΛΕΩΝΙΔΑΣ	ΤΣΑΝΤΑΡΛΙΩΤΗΣ	1059642

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.

Συνημμένα αρχεία κώδικα

Μαζί με την παρούσα αναφορά υποβάλλω τα παρακάτω αρχεία κώδικα

Αρχείο	Αφορά το ερώτημα	Περιγραφή/Σχόλιο
select_query_1.py	Ερώτημα 3 - Ερωτήματα προς υλοποίηση - Ερώτημα 1	Εμφανίζει τις 30 ταινίες με την υψηλότερη μέση βαθμολογία μεταξύ 01/01/2015 και 15/01/2015
select_query_2.py	Ερώτημα 3 - Ερωτήματα προς υλοποίηση - Ερώτημα 2	Εμφανίζει όλες τις λεπτομέρειες για την ταινία Jumanji (κατηγορία, μέση βαθμολογία, top-5 ετικέτες)
select_query_3.py	Ερώτημα 3 - Ερωτήματα προς υλοποίηση - Ερώτημα 3	Εμφανίζει τις ταινίες της κατηγορίας “adventure” ταξινομημένες ως προς το έτος παραγωγής
select_query_4.py	Ερώτημα 3 - Ερωτήματα προς υλοποίηση - Ερώτημα 4	Εμφανίζει τις ταινίες που περιέχουν τη λέξη “star”
select_query_5.py	Ερώτημα 3 - Ερωτήματα προς υλοποίηση - Ερώτημα 5	Εμφανίζει τις 20 ταινίες με την υψηλότερη μέση βαθμολογία για την ετικέτα “comedy”.
create_table.txt	Ερώτημα 2	Περιέχει όλες τις CREATE TABLE της βάσης δεδομένων

cqlsh_commands.py	Ερώτημα 2	Περιέχει όλες τις εντολές COPY που χρησιμοποίησα για να εισάγω τα δεδομένα στη βάση
-------------------	-----------	---

Σημείωση: Την εισαγωγή των δεδομένων στη βάση την έκανα με τα κατάλληλα COPY commands (που βρίσκονται στο αρχείο `cqlsh_commands.py`), απευθείας από το `cqlsh shell` του Cassandra και όχι με Python scripts που ζητάει η εκφώνηση διότι ο DataStax Python Driver υποστηρίζει μόνο CQL commands και η COPY είναι cqlsh command. Θεωρητικά μπορούμε να εισάγουμε τα δεδομένα με τον DataStax Python Driver με την INSERT αλλά η ταχύτητα εισαγωγής μειώνεται δραματικά (με τον driver είχα ταχύτητα εισαγωγής 3000-5000 rows/sec ενώ με την COPY είχα 70000-90000 rows/sec). Οπότε για να εισάγουμε αρχεία όπως το `rating.csv` που περιέχουν 20M rows, ο απαραίτητος χρόνος είναι τεράστιος.

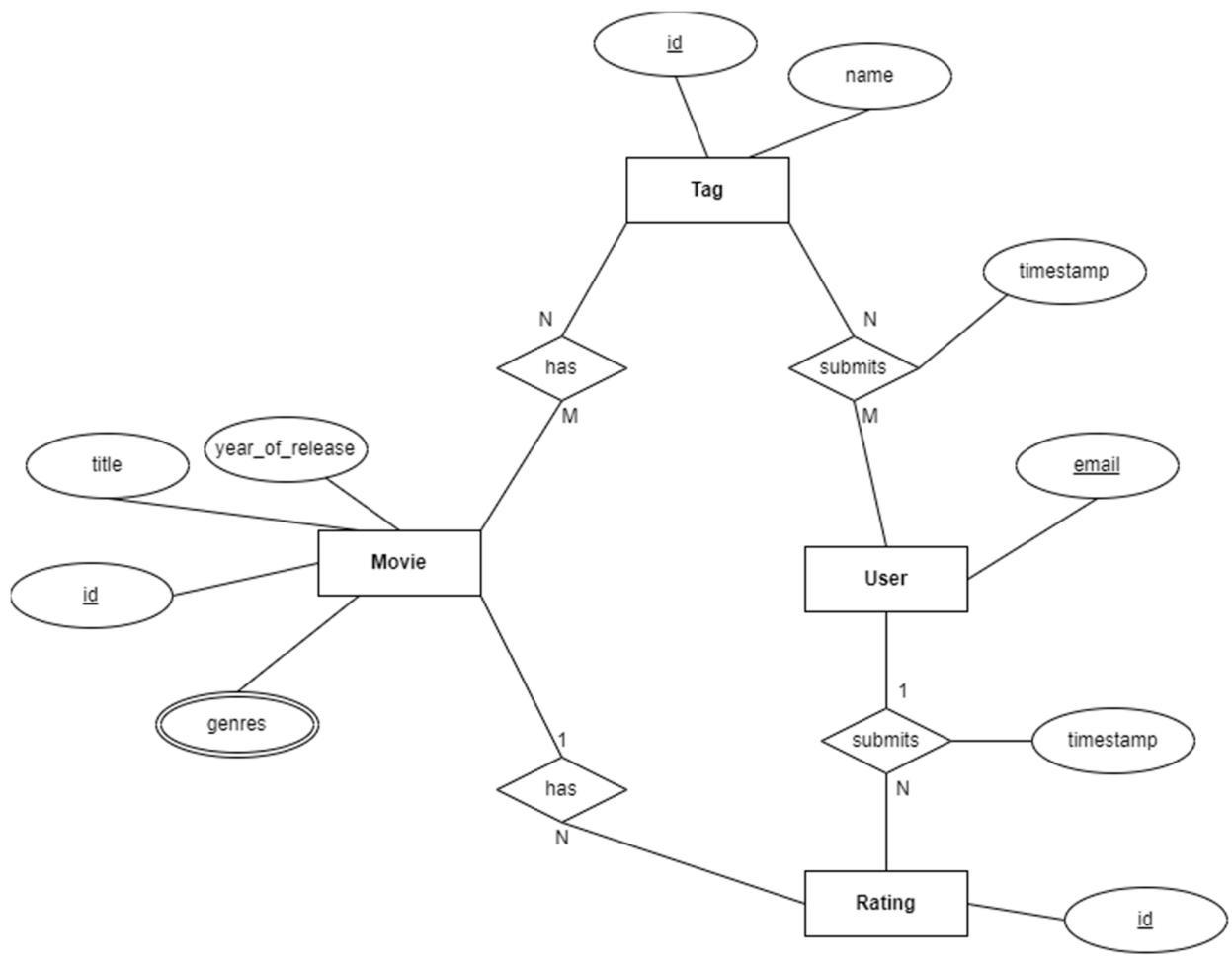
https://www.tutorialspoint.com/cassandra/cassandra_cqlsh.htm

Τεχνικά χαρακτηριστικά περιβάλλοντος λειτουργίας

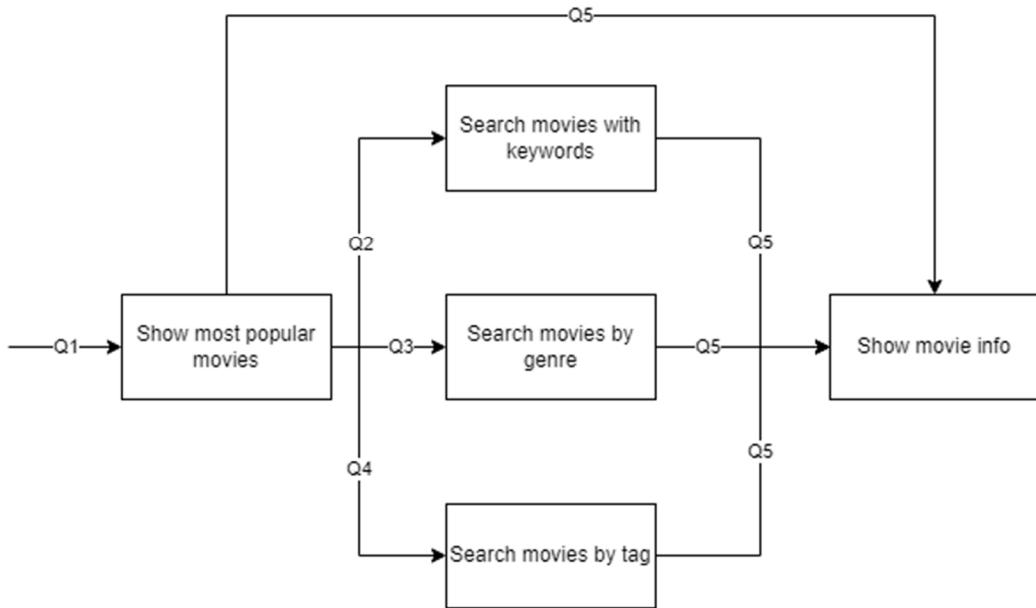
Χαρακτηριστικό	Τιμή
CPU model	Ryzen 5 1600
CPU clock speed	3.2 GHz
Physical CPU cores	6
Logical CPU cores	12
RAM	16 GB
Secondary Storage Type	SSD

Ερώτημα 1: Σχεδιασμός ΒΔ

Εννοιολογικό Μοντέλο

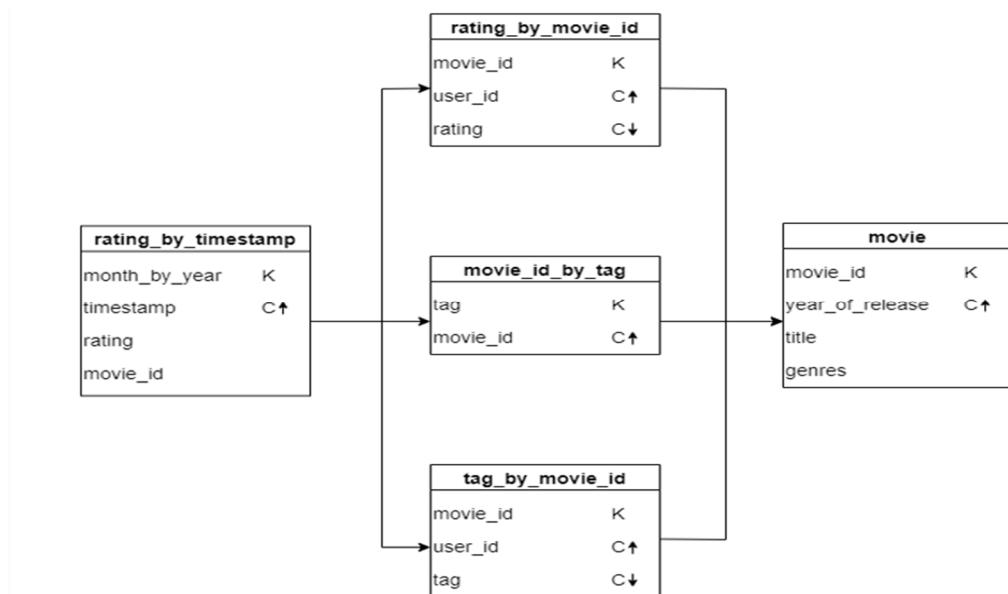


Application Workflow



- Q1: Find the most popular movies (of a prefixed time period)
- Q2: Find all the movies that contain the keyword in their title
- Q3: Find movies that belong to a genre (show most popular first)
- Q4: Find the n most popular movies by a tag
- Q5: Show movie information (genre, rating, top tags)

Chebotko diagram



Ερώτημα 2: Ερωτήματα DDL

TABLE	rating_by_timestamp																																												
DDL statement	<pre>CREATE TABLE IF NOT EXISTS rating_by_timestamp (movieid int, rating float, timestamp timestamp, month_by_year text, PRIMARY KEY ((month_by_year), timestamp))WITH CLUSTERING ORDER BY (timestamp ASC);</pre>																																												
Screenshot	<pre>cqlsh:big_data_project> select * from rating_by_timestamp limit 10;</pre> <table border="1"> <thead> <tr> <th>month_by_year</th> <th>timestamp</th> <th>movieid</th> <th>rating</th> </tr> </thead> <tbody> <tr><td>2010-07</td><td>2010-07-01 02:23:42.000000+0000</td><td>60110</td><td>3</td></tr> <tr><td>2010-07</td><td>2010-07-01 02:25:31.000000+0000</td><td>54736</td><td>4</td></tr> <tr><td>2010-07</td><td>2010-07-01 02:25:32.000000+0000</td><td>65882</td><td>2.5</td></tr> <tr><td>2010-07</td><td>2010-07-01 02:26:23.000000+0000</td><td>5540</td><td>3.5</td></tr> <tr><td>2010-07</td><td>2010-07-01 02:26:36.000000+0000</td><td>77837</td><td>4</td></tr> <tr><td>2010-07</td><td>2010-07-01 02:27:51.000000+0000</td><td>74156</td><td>2.5</td></tr> <tr><td>2010-07</td><td>2010-07-01 02:28:11.000000+0000</td><td>32</td><td>4</td></tr> <tr><td>2010-07</td><td>2010-07-01 02:28:16.000000+0000</td><td>59315</td><td>4</td></tr> <tr><td>2010-07</td><td>2010-07-01 02:28:18.000000+0000</td><td>4963</td><td>4.5</td></tr> <tr><td>2010-07</td><td>2010-07-01 02:28:22.000000+0000</td><td>5445</td><td>3</td></tr> </tbody> </table> <p>(10 rows)</p>	month_by_year	timestamp	movieid	rating	2010-07	2010-07-01 02:23:42.000000+0000	60110	3	2010-07	2010-07-01 02:25:31.000000+0000	54736	4	2010-07	2010-07-01 02:25:32.000000+0000	65882	2.5	2010-07	2010-07-01 02:26:23.000000+0000	5540	3.5	2010-07	2010-07-01 02:26:36.000000+0000	77837	4	2010-07	2010-07-01 02:27:51.000000+0000	74156	2.5	2010-07	2010-07-01 02:28:11.000000+0000	32	4	2010-07	2010-07-01 02:28:16.000000+0000	59315	4	2010-07	2010-07-01 02:28:18.000000+0000	4963	4.5	2010-07	2010-07-01 02:28:22.000000+0000	5445	3
month_by_year	timestamp	movieid	rating																																										
2010-07	2010-07-01 02:23:42.000000+0000	60110	3																																										
2010-07	2010-07-01 02:25:31.000000+0000	54736	4																																										
2010-07	2010-07-01 02:25:32.000000+0000	65882	2.5																																										
2010-07	2010-07-01 02:26:23.000000+0000	5540	3.5																																										
2010-07	2010-07-01 02:26:36.000000+0000	77837	4																																										
2010-07	2010-07-01 02:27:51.000000+0000	74156	2.5																																										
2010-07	2010-07-01 02:28:11.000000+0000	32	4																																										
2010-07	2010-07-01 02:28:16.000000+0000	59315	4																																										
2010-07	2010-07-01 02:28:18.000000+0000	4963	4.5																																										
2010-07	2010-07-01 02:28:22.000000+0000	5445	3																																										

TABLE	rating_by_movie_id
DDL statement	<pre>CREATE TABLE IF NOT EXISTS rating_by_movie_id (userid int, movieid int,</pre>

	<pre> rating float, PRIMARY KEY ((movieid), userid, rating))WITH CLUSTERING ORDER BY (userid ASC, rating DESC); </pre>																																	
Screenshot	<pre>cqlsh:big_data_project> select * from rating_by_movie_id limit 10;</pre> <table border="1"> <thead> <tr> <th>movieid</th><th>userid</th><th>rating</th></tr> </thead> <tbody> <tr><td>4317</td><td>7</td><td>3</td></tr> <tr><td>4317</td><td>116</td><td>1</td></tr> <tr><td>4317</td><td>156</td><td>3</td></tr> <tr><td>4317</td><td>279</td><td>3.5</td></tr> <tr><td>4317</td><td>413</td><td>1</td></tr> <tr><td>4317</td><td>424</td><td>4.5</td></tr> <tr><td>4317</td><td>609</td><td>0.5</td></tr> <tr><td>4317</td><td>632</td><td>1</td></tr> <tr><td>4317</td><td>635</td><td>5</td></tr> <tr><td>4317</td><td>903</td><td>3</td></tr> </tbody> </table>	movieid	userid	rating	4317	7	3	4317	116	1	4317	156	3	4317	279	3.5	4317	413	1	4317	424	4.5	4317	609	0.5	4317	632	1	4317	635	5	4317	903	3
movieid	userid	rating																																
4317	7	3																																
4317	116	1																																
4317	156	3																																
4317	279	3.5																																
4317	413	1																																
4317	424	4.5																																
4317	609	0.5																																
4317	632	1																																
4317	635	5																																
4317	903	3																																

TABLE	tag_by_movie_id
DDL statement	<pre> CREATE TABLE IF NOT EXISTS tag_by_movie_id (userid int, movieid int, tag text, PRIMARY KEY ((movieid), userid, tag))WITH CLUSTERING ORDER BY (userid ASC, tag DESC); </pre>

Screenshot	cqlsh:big_data_project> select * from tag_by_movie_id limit 10;
	<pre> movieid userid tag -----+-----+ 4317 52437 Sandra Bullock 4317 123297 CLV 51678 1741 playwright:Shakespeare 51678 1741 adapted from:play 51678 10573 shakespeare 51678 45928 Marlon Brando 51678 70201 Joseph L. Mankiewicz 51678 78356 Marlon Brando 51678 88738 based on a play 51678 123297 BD-R (10 rows) </pre>

TABLE	movie
DDL statement	CREATE TABLE IF NOT EXISTS movie(movieid int, title text, genres text, year_of_release int, PRIMARY KEY ((movieid), year_of_release))WITH CLUSTERING ORDER BY (year_of_release ASC);

Screenshot	cqlsh:big_data_project> select * from movie limit 10;																																												
	<table border="1"> <thead> <tr> <th>movieid</th><th>year_of_release</th><th>genres</th><th>title</th></tr> </thead> <tbody> <tr><td>4317</td><td>1992</td><td>Comedy Romance</td><td>Love Potion #9 (1992)</td></tr> <tr><td>51678</td><td>1953</td><td>Drama</td><td>Julius Caesar (1953)</td></tr> <tr><td>77328</td><td>2009</td><td>Crime Drama Mystery Thriller</td><td>Red Riding: 1974 (2009)</td></tr> <tr><td>3372</td><td>1969</td><td>Action War</td><td>Bridge at Remagen, The (1969)</td></tr> <tr><td>96748</td><td>2011</td><td>Horror Thriller</td><td>247°F (2011)</td></tr> <tr><td>1584</td><td>1997</td><td>Drama Sci-Fi</td><td>Contact (1997)</td></tr> <tr><td>7034</td><td>1998</td><td>Drama Romance</td><td>Show Me Love (Fucking Åmål) (1998)</td></tr> <tr><td>47076</td><td>2003</td><td>Drama</td><td>Good Morning, Night (Buongiorno, notte) (2003)</td></tr> <tr><td>118193</td><td>1961</td><td>Children Drama</td><td>Hand in Hand (1961)</td></tr> <tr><td>71650</td><td>1980</td><td>Documentary</td><td>Werner Herzog Eats His Shoe (1980)</td></tr> </tbody> </table> <p>(10 rows)</p>	movieid	year_of_release	genres	title	4317	1992	Comedy Romance	Love Potion #9 (1992)	51678	1953	Drama	Julius Caesar (1953)	77328	2009	Crime Drama Mystery Thriller	Red Riding: 1974 (2009)	3372	1969	Action War	Bridge at Remagen, The (1969)	96748	2011	Horror Thriller	247°F (2011)	1584	1997	Drama Sci-Fi	Contact (1997)	7034	1998	Drama Romance	Show Me Love (Fucking Åmål) (1998)	47076	2003	Drama	Good Morning, Night (Buongiorno, notte) (2003)	118193	1961	Children Drama	Hand in Hand (1961)	71650	1980	Documentary	Werner Herzog Eats His Shoe (1980)
movieid	year_of_release	genres	title																																										
4317	1992	Comedy Romance	Love Potion #9 (1992)																																										
51678	1953	Drama	Julius Caesar (1953)																																										
77328	2009	Crime Drama Mystery Thriller	Red Riding: 1974 (2009)																																										
3372	1969	Action War	Bridge at Remagen, The (1969)																																										
96748	2011	Horror Thriller	247°F (2011)																																										
1584	1997	Drama Sci-Fi	Contact (1997)																																										
7034	1998	Drama Romance	Show Me Love (Fucking Åmål) (1998)																																										
47076	2003	Drama	Good Morning, Night (Buongiorno, notte) (2003)																																										
118193	1961	Children Drama	Hand in Hand (1961)																																										
71650	1980	Documentary	Werner Herzog Eats His Shoe (1980)																																										

TABLE	movie_id_by_tag																						
DDL statement	<pre>CREATE TABLE IF NOT EXISTS movie_id_by_tag(movieid int, tag text, PRIMARY KEY ((tag), movieid))WITH CLUSTERING ORDER BY (movieid ASC);</pre>																						
Screenshot	cqlsh:big_data_project> select * from movie_id_by_tag limit 10; <table border="1"> <thead> <tr> <th>tag</th><th>movieid</th></tr> </thead> <tbody> <tr><td>compareTo:Last Action Hero</td><td>485</td></tr> <tr><td>compareTo:Last Action Hero</td><td>63222</td></tr> <tr><td>daredevil</td><td>6157</td></tr> <tr><td>loose ends</td><td>36529</td></tr> <tr><td>loose ends</td><td>53953</td></tr> <tr><td>loose ends</td><td>64997</td></tr> <tr><td>loose ends</td><td>77780</td></tr> <tr><td>loose ends</td><td>95088</td></tr> <tr><td>Script</td><td>26403</td></tr> <tr><td>Script</td><td>62394</td></tr> </tbody> </table> <p>(10 rows)</p>	tag	movieid	compareTo:Last Action Hero	485	compareTo:Last Action Hero	63222	daredevil	6157	loose ends	36529	loose ends	53953	loose ends	64997	loose ends	77780	loose ends	95088	Script	26403	Script	62394
tag	movieid																						
compareTo:Last Action Hero	485																						
compareTo:Last Action Hero	63222																						
daredevil	6157																						
loose ends	36529																						
loose ends	53953																						
loose ends	64997																						
loose ends	77780																						
loose ends	95088																						
Script	26403																						
Script	62394																						

Ερώτημα 3: Απαντήσεις ερωτημάτων

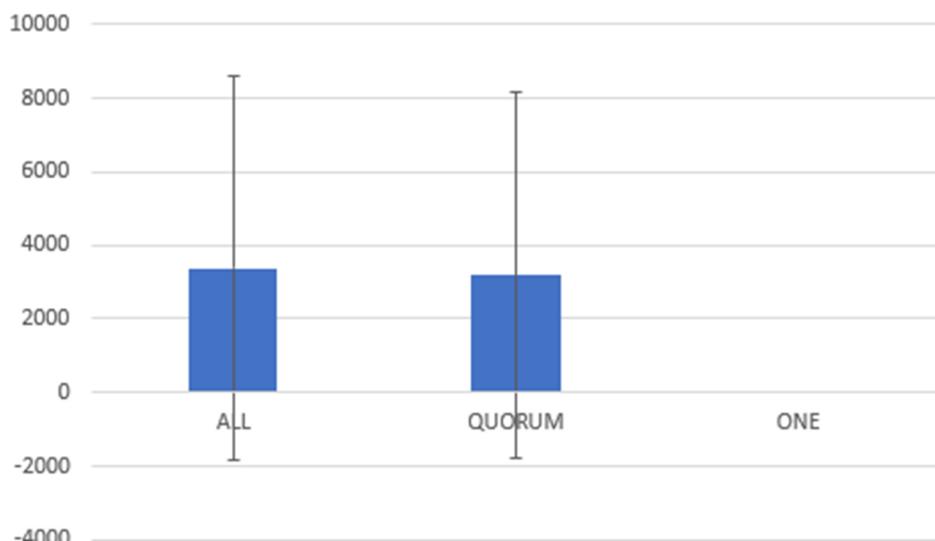
Ερώτημα	Απάντηση
Εμφάνιση των 30 ταινιών με την υψηλότερη μέση βαθμολογία μεταξύ 01/01/2015 και 15/01/2015	<pre>leonidastsant@pop-os:~/Documents/Big_Data_Project/Python_Scripts\$ python3 select_query_1.py movieid title rating 0 123107 The Phantom of the Opera (1990) 5.0 1 80337 Still Bill (2009) 5.0 2 80939 Conviction (2010) 5.0 3 81083 Kaboom (2010) 5.0 4 82306 Student Prince in Old Heidelberg, The (1927) 5.0</pre>
Εμφάνιση όλων των λεπτομερειών για την ταινία Jumanji (κατηγορία, μέση βαθμολογία, top-5 ετικέτες)	<pre>leonidastsant@pop-os:~/Documents/Big_Data_Project/Python_Scripts\$ python3 select_query_2.py Movie ID: 2 Title: Jumanji (1995) Genres: Adventure Children Fantasy Rating: 3.2 Top Tags:['Robin Williams', 'time travel', 'fantasy', 'animals', 'board game']</pre>
Εμφάνιση των ταινιών της κατηγορίας “adventure” ταξινομημένες ως προς το έτος παραγωγής	<pre>leonidastsant@pop-os:~/Documents/Big_Data_Project/Python_Scripts\$ python3 select_query_3.py movieid title genres year_of_release 0 32898 Trip to the Moon, A (Voyage dans la lune, Le) ... Action Adventure Fantasy Sci-Fi 1902 1 90339 Last Days of Pompeii, The (Gli ultimi giorni d... Adventure Drama 1913 2 84852 Judith of Bethulia (1914) Adventure Drama Romance 1914 3 74097 Cabiria (1914) Adventure Drama War 1914 4 69509 Vampires, Les (1915) Action Adventure Drama Horror Thriller 1915</pre>
Εμφάνιση των ταινιών που περιέχουν τη λέξη “star”	<pre>leonidastsant@pop-os:~/Documents/Big_Data_Project/Python_Scripts\$ python3 select_query_4.py movieid title genres year_of_release 0 98241 Starry starry night (Xing kong) (2011) Children Drama Fantasy 2011 1 91914 Mad Bastards (2010) Drama 2010 2 86758 Bastards, The (Los bastardos) (2008) Crime Drama Thriller 2008 3 68574 Django the Bastard (Strangers Gundown, The) (... Action Thriller Western 1969 4 100099 Pictures of the Old World (Obrazy starého svet...) Documentary 1972</pre>
Εμφάνιση των 20 ταινιών με την υψηλότερη μέση βαθμολογία για την ετικέτα “comedy”	<pre>leonidastsant@pop-os:~/Documents/Big_Data_Project/Python_Scripts\$ python3 select_query_5.py movieId title average_rating 0 86055 Foster Brothers, The (Süt kardesler) (1976) 5.0 1 128991 Johnny Express (2014) 4.4 2 108134 Lizard, The (Marmoulak) (2004) 4.3 3 4973 Amelie (Fabuleux destin d'Amélie Poulain, Le) ... 4 745 Wallace & Gromit: A Close Shave (1995) 4.2</pre>

Ερώτημα 4A: Χρόνοι εισαγωγής δεδομένων

Χρόνοι εκτέλεσης (sec)

Keyspace	ALL	QUORUM	ONE
Table 1 (rating_by_timestamp)	2,634	2,367	
Table 2 (rating_by_movie_id)	13,866	13,160	
Table 3 (tag_by_movie_id)	123	112	
Table 4 (movie)	19	18	
Table 5 (movie_id_by_tag)	240	217	
μέσος όρος	3376.4	3174.8	#DIV/0!
τυπική απόκλιση	5963.98812	5666.821658	#DIV/0!
διάστημα εμπιστοσύνης	5227.570019	4967.096916	#DIV/0!

M.O. χρόνου εκτέλεσης ερωτημάτων εισαγωγής



Για consistency level = ONE, το terminal μου εμφανίζει το παρακάτω το μήνυμα

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

Ερώτημα 4B: Χρόνοι ανάκτησης δεδομένων

Ωμά Δεδομένα				
Εκτέλεση	Ερώτημα	ALL	QUORUM	ONE
1	Ερώτημα 1	5.05	4.81	4.56
1	Ερώτημα 2	1.14	1.1	1.06
1	Ερώτημα 3			
1	Ερώτημα 4			
1	Ερώτημα 5	3.56	3.43	3.04
2	Ερώτημα 1	4.07	4.03	3.9
2	Ερώτημα 2	0.52	0.71	0.71
2	Ερώτημα 3			
2	Ερώτημα 4			
2	Ερώτημα 5	1.69	1.66	1.59
3	Ερώτημα 1	3.2	3.08	2.85
3	Ερώτημα 2	0.5	0.51	0.5
3	Ερώτημα 3			
3	Ερώτημα 4			
3	Ερώτημα 5	1.73	1.65	1.61
4	Ερώτημα 1	3.25	3.24	3.17
4	Ερώτημα 2	0.53	0.5	0.52
4	Ερώτημα 3			
4	Ερώτημα 4			
4	Ερώτημα 5	1.97	1.84	1.65
5	Ερώτημα 1	3.46	3.4	3.5
5	Ερώτημα 2	0.55	0.53	0.55
5	Ερώτημα 3			
5	Ερώτημα 4			
5	Ερώτημα 5	1.69	1.7	1.69
6	Ερώτημα 1	3.69	3.56	3.55
6	Ερώτημα 2	0.5	0.53	0.5
6	Ερώτημα 3			
6	Ερώτημα 4			
6	Ερώτημα 5	3.51	3.68	3.15

7	Ερώτημα 1	3.54	3.54	3.51
7	Ερώτημα 2	1.13	1	0.99
7	Ερώτημα 3			
7	Ερώτημα 4			
7	Ερώτημα 5	2	2.02	1.92
8	Ερώτημα 1	3.9	3.83	3.71
8	Ερώτημα 2	0.52	0.55	0.54
8	Ερώτημα 3			
8	Ερώτημα 4			
8	Ερώτημα 5	2.36	2.29	2.22
9	Ερώτημα 1	4.16	4.1	3.96
9	Ερώτημα 2	0.52	0.53	0.48
9	Ερώτημα 3			
9	Ερώτημα 4			
9	Ερώτημα 5	2.13	2.14	1.98
10	Ερώτημα 1	4.55	4.55	4.25
10	Ερώτημα 2	0.48	0.52	0.53
10	Ερώτημα 3			
10	Ερώτημα 4			
10	Ερώτημα 5	2.69	2.46	2.27

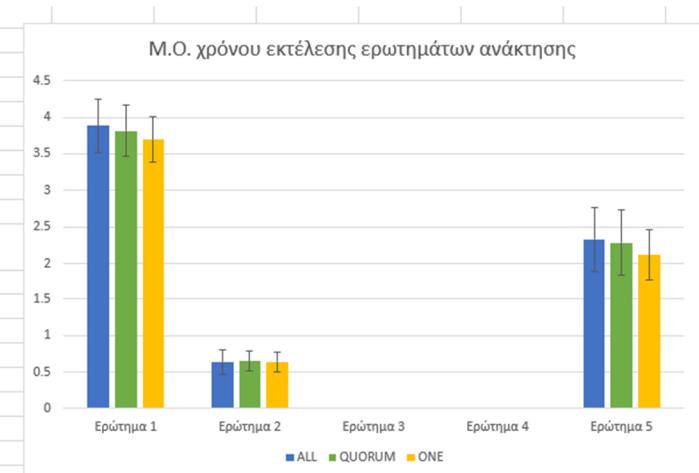
Μέσοι όροι			
Ερώτημα	ALL	QUORUM	ONE
Ερώτημα 1	3.887	3.814	3.696
Ερώτημα 2	0.639	0.648	0.638
Ερώτημα 3	#DIV/0!	#DIV/0!	#DIV/0!
Ερώτημα 4	#DIV/0!	#DIV/0!	#DIV/0!
Ερώτημα 5	2.333	2.287	2.112
μέσος όρος	#DIV/0!	#DIV/0!	#DIV/0!
τυπική απόκλιση	#DIV/0!	#DIV/0!	#DIV/0!
διάστημα εμπιστοσύνης	#DIV/0!	#DIV/0!	#DIV/0!

Τυπικές αποκλίσεις

Ερώτημα	ALL	QUORUM	ONE
Ερώτημα 1	0.588747258	0.561905488	0.500182189
Ερώτημα 2	0.262104729	0.221299194	0.214258411
Ερώτημα 3	#DIV/0!	#DIV/0!	#DIV/0!
Ερώτημα 4	#DIV/0!	#DIV/0!	#DIV/0!
Ερώτημα 5	0.706588832	0.723648933	0.572087794

Διαστήματα Εμπιστοσύνης

Ερώτημα	ALL	QUORUM	ONE
Ερώτημα 1	0.364902626	0.34826623	0.310010436
Ερώτημα 2	0.162451209	0.137160141	0.132796299
Ερώτημα 3	#DIV/0!	#DIV/0!	#DIV/0!
Ερώτημα 4	#DIV/0!	#DIV/0!	#DIV/0!
Ερώτημα 5	0.437940248	0.448514014	0.354577173



Για την υλοποίηση του 3ου και 4ου query, έχω χρησιμοποιήσει LIKE condition (π.χ. στο 4o: " SELECT * FROM movie WHERE title LIKE '%star%' "), το οποίο στο Cassandra είναι εφικτό μόνο με τη χρήση SSTable Attached Secondary Index (SASI). Το πρόβλημα είναι ότι το AstraDB δεν υποστηρίζει ακόμα αυτού του είδους indexes οπότε και παίρνω το παρακάτω error όταν πάω να δημιουργήσω έναν:

InvalidRequest: Error from server: code=2200 [Invalid query] message="Tables cannot have more than 0 SASI indexes, failed to create SASI index title_index on table movie"

Συνεπώς δεν μπορώ να πάρω μετρήσεις για τα ερωτήματα 3 και 4.

Ερώτημα 4Γ: Σχολιασμός αποτελεσμάτων

Παρατηρούμε ότι τους πιο γρήγορους χρόνους τους παίρνουμε για consistency level ONE, μετά για QUORUM και τέλος για ALL. Αυτό είναι αναμενόμενο διότι κατά την ανάγνωση δεδομένων:

- Το επίπεδο συνέπειας ONE σημαίνει ότι μόνο ένας κόμβος αντιγράφων επιστρέφει τα δεδομένα. Η ανάκτηση δεδομένων είναι ταχύτερη σε αυτή την περίπτωση.
- Το επίπεδο συνέπειας QUORUM σημαίνει ότι το 51% ή η πλειοψηφία των κόμβων αντιγράφων σε όλα τα κέντρα δεδομένων ανταποκρίνεται. Στη συνέχεια, ο συντονιστής επιστρέφει τα δεδομένα στον πελάτη. Στην περίπτωση πολλαπλών κέντρων δεδομένων, η καθυστέρηση της επικοινωνίας μεταξύ των κέντρων δεδομένων οδηγεί σε αργή ανάγνωση.
- Το επίπεδο συνέπειας ALL σημαίνει ότι όλοι οι κόμβοι αντιγράφων ανταποκρίνονται, και στη συνέχεια ο συντονιστής επιστρέφει τα δεδομένα στον πελάτη. Δεδομένου ότι όλοι οι κόμβοι αντιγράφων πρέπει να επιβεβαιώσουν, η λειτουργία ανάγνωσης είναι η πιο αργή σε αυτή την περίπτωση. Επιπλέον, εάν ένας από τους κόμβους αντιγράφων είναι εκτός λειτουργίας κατά τη διάρκεια της λειτουργίας ανάγνωσης, αποτυγχάνει και η διαθεσιμότητα υποφέρει.

Κατά την εγγραφή δεδομένων:

- Το επίπεδο συνέπειας ONE σημαίνει ότι χρειάζεται επιβεβαίωση από έναν μόνο κόμβο αντιγράφων. Δεδομένου ότι μόνο ένα αντίγραφο χρειάζεται επιβεβαίωση, η λειτουργία εγγραφής είναι ταχύτερη σε αυτή την περίπτωση.
- Το επίπεδο συνέπειας QUORUM σημαίνει ότι χρειάζεται επιβεβαίωση από το 51% ή την πλειοψηφία των κόμβων αντιγράφων σε όλα τα κέντρα δεδομένων.
- Το επίπεδο συνέπειας ALL σημαίνει ότι χρειάζεται επιβεβαίωση από όλους τους κόμβους αντιγράφων. Δεδομένου ότι όλοι οι κόμβοι αντιγράφων χρειάζονται επιβεβαίωση, η λειτουργία εγγραφής είναι η πιο αργή σε αυτή την περίπτωση. Επιπλέον, εάν ένας από τους κόμβους αντιγράφων είναι εκτός λειτουργίας κατά τη διάρκεια της λειτουργίας εγγραφής, αποτυγχάνει και η διαθεσιμότητα υποφέρει.

Το θεώρημα CAP δηλώνει ότι μια βάση δεδομένων δεν μπορεί να εγγυηθεί ταυτόχρονα τη συνέπεια, τη διαθεσιμότητα και την ανοχή διαμερισμού. Το Cassandra περιγράφεται συνήθως ως ένα σύστημα "AP", που σημαίνει ότι επιλέγει την πλευρά της διασφάλισης της διαθεσιμότητας των δεδομένων, ακόμη και αν αυτό σημαίνει ότι θυσιάζεται η συνέπεια. Συνεπώς αν αυξήσουμε τη συνέπεια, μειώνεται η διαθεσιμότητα αφού όπως ειπώθηκε από το θεώρημα CAP μια βάση δεδομένων δεν μπορεί να εγγυηθεί ταυτόχρονα τη συνέπεια, τη διαθεσιμότητα και την ανοχή διαμερισμού.