

ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΤΜΗΜΑΤΟΣ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

AgensGraph PostgreSQL

Διπλωματική Εργασία

του

Κοσμά Θεόδωρου Καίσαρη

Θεσσαλονίκη, Ιανουάριος 2023

AgensGraph PostgreSQL

Κοσμάς Θεόδωρος Καίσαρης

Πτυχίο Μηχανικών πληροφορικής, Τ.Ε.Ι Δυτικής Μακεδονίας

Διπλωματική Εργασία

υποβαλλόμενη για τη μερική εκπλήρωση των απαιτήσεων του

ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΤΙΤΛΟΥ ΣΠΟΥΔΩΝ ΣΤΗΝ ΕΦΑΡΜΟΣΜΕΝΗ
ΠΛΗΡΟΦΟΡΙΚΗ

Επιβλέπων Καθηγητής

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την ηη/μμ/εεεε

Όνοματεπώνυμο 1

Όνοματεπώνυμο 2

Όνοματεπώνυμο 3

.....

.....

.....

Κοσμάς Θεόδωρος Καίσαρης

Περίληψη

Εντός του πεδίου διαχείρισης δεδομένων η ευρεία χρήση της (SQL) Structured Query Language την έχει εδραιώσει ως τον βασικό πυλώνα για την αλληλεπίδραση του χρήστη με τις σχεσιακές βάσεις δεδομένων. Μεταξύ αυτών η PostgreSQL, ένας αξιόπιστος Διαχειριστής Συστήματος Σχεσιακής Βάσης Δεδομένων (RDBMS), έχει επιδοθεί στη διαχείριση δομημένων δεδομένων με τον πλούσιο σε δυνατότητες σύνολο χαρακτηριστικών του. Ωστόσο, καθώς οι πολυπλοκότητες των δεδομένων εξελίσσονται, η ανάγκη για ευέλικτα μοντέλα δεδομένων έχει οδηγήσει στην εμφάνιση των βάσεων δεδομένων γράφων, ως μια εναλλακτική πρακτική.

Αυτή η διατριβή αναλαμβάνει μια εξερεύνηση του τοπίου δεδομένων, αντιθέτοντας τις βάσεις της SQL με τις καινοτόμες όψεις των βάσεων δεδομένων γράφων. Αυτό το ταξίδι ξεκινά με μια εστιασμένη ματιά στην Neo4j, μια διακεκριμένη λύση βάσης δεδομένων γράφων, όπου τα πλεονεκτήματα και οι περιορισμοί της αναδεικνύονται στη διαχείριση περίπλοκων συνδεδεμένων σεναρίων δεδομένων.

Μια ομαλή μετάβαση κατευθύνει την προσοχή μας στην AgensGraph, μια πολυμορφική βάση δεδομένων γράφων βαθιά ριζωμένη στην αρχιτεκτονική του PostgreSQL. Μια καθοριστική χαρακτηριστική της AgensGraph κρύβεται στη δυνατότητά της να εναρμονίζει αβίαστα τα σχεσιακά και μοντέλα δεδομένων γράφων εντός ενός μοναδικού περιβάλλοντος βάσης δεδομένων. Αποκαλύπτοντας τη δομή της AgensGraph, αυτή η μελέτη αναδεικνύει τη συμβατότητά της όχι μόνο με τη SQL αλλά και με την εξειδικευμένη γλώσσα ερωτημάτων Cypher. Σχετική για αυτή την έρευνα είναι η αποκάλυψη του μοντέλου γραφικών ιδιοτήτων, αποκαλύπτοντας τους κόμβους της, τις σχέσεις τους και τη δυναμική της διαδικασίας του γραφικού περιηγητή στη χάραξη αποδοτικών συνδέσεων σε πολύπλοκες διασυνδέσεις δεδομένων.

Επιπλέον, η μελέτη εμβαθύνει στον πρωταγωνιστικό ρόλο της AgensGraph στην επίλυση των διλημάτων που εμφανίζονται σε πρακτικές εφαρμογές. Εξετάζοντας τα χαρακτηριστικά απόδοσης, την κλιμακοποίηση και την ευελιξία στην πλοήγηση σε περίπλοκες συνδεδεμένες δομές δεδομένων, η συνεργία των σχεσιακών και μοντέλων γράφων δεδομένων αναδύεται ως ένα πολυπόθητο πλεονέκτημα. Αυτή η ένωση επιτρέπει στους προγραμματιστές να αξιοποιούν τα πλεονεκτήματα και των δύο παραδειγμάτων, δημιουργώντας ένα εύφορο πεδίο για τη δημιουργία ανθεκτικών, δεδομένων-κεντρικών λύσεων.

Συνοψίζοντας, αυτή η διατριβή συμβάλλει ουσιαστικά στη συζήτηση που αφορά τα σύγχρονα παραδείγματα διαχείρισης δεδομένων. Μέσα από τον ιστορικό διάλογο μεταξύ της SQL και των βάσεων δεδομένων γράφων, αυτή η εργασία αναδεικνύει την ανάγκη για ευέλικτες, αποδοτικές και ασφαλείς προσεγγίσεις της διαχείρισης δεδομένων. Σε αυτό το πλαίσιο, η AgensGraph αναδεικνύεται ως μια καινοτόμος επιλογή που ανταποκρίνεται στις σύγχρονες απαιτήσεις για αποδοτική διαχείριση δεδομένων, καταστρατηγώντας τα όρια μεταξύ σχεσιακών και μοντέλων γράφων και προσφέροντας έναν ολοκληρωμένο, ευέλικτο και ασφαλή τρόπο αντιμετώπισης των πολύπλοκων απαιτήσεων των δεδομένων της σύγχρονης εποχής.

Λέξεις Κλειδιά: AgensGraph, Sql, PostgreSql, Cypher, Neo4j

Abstract

Within the realm of data management, the ubiquity of Structured Query Language (SQL) has cemented its role as a cornerstone for interacting with relational databases. Among these, PostgreSQL, a stalwart Relational Database Management System (RDBMS), has adeptly navigated structured data with its robust feature set. Yet, as data complexities evolve, the call for adaptable data models has given rise to graph databases as an alternative paradigm.

This thesis embarks on an exploration of the data landscape, contrasting SQL's foundations with the innovative facets of graph databases. This journey begins with a focused gaze upon Neo4j, a notable graph database solution, where its merits and limitations take center stage in handling intricate, interconnected data scenarios.

A seamless transition directs attention to AgensGraph, a multi-model graph database deeply rooted in PostgreSQL's architecture. A defining feature of AgensGraph lies in its ability to seamlessly harmonize the relational and graph data models within a singular database environment. Unveiling AgensGraph's structural framework, this study accentuates its compatibility not only with SQL but also with the specialized Cypher query language. Pivotal to this exploration is the exposition of the property graph model, unveiling its nodes, relationships, and the potency of graph traversal in efficiently mapping complex data interdependencies.

Moreover, the study delves into AgensGraph's pivotal role in resolving the quandaries that manifest in practical applications. Scrutinizing its performance attributes, scalability, and the flexibility to navigate intricate connected data formations, the synergy of the relational and graph data models emerges as a potent asset. This union empowers developers to harness the strengths of both paradigms, fostering a fertile ground for crafting robust, data-centric solutions.

In summation, this thesis contributes substantively to the discourse surrounding modern data management paradigms. Through a dialectic juxtaposition of SQL and graph databases and a meticulous dissection of AgensGraph's proficiencies, it sheds light on the transformative potential of graph databases in reshaping data governance and analysis within our increasingly interwoven digital landscapes.

Keywords: AgensGraph, Sql, PostgreSql, Cypher, Neo4j

Πρόλογος – Ευχαριστίες

Στο σημείο αυτό, θα ήθελα να ευχαριστήσω όλους όσους συνέλαβαν στην εκπόνηση της διπλωματικής μου εργασίας. Οφείλω να εκφράσω τις θερμές μου ευχαριστίες προς την επιβλέπουσα την εργασία Καθηγήτρια κ. Κολωνιάρη Γεωργία. Επίσης ευχαριστώ θερμά την οικογένεια και τους φίλους μου, για την συμπαράσταση και κατανόηση που έδειξαν σε όλη την περίοδο εκπόνησης της εργασίας μου.

Περιεχόμενα

1. Εισαγωγή	1
1.1 Πρόβλημα – Σημαντικότητα του θέματος	1
1.2 Σκοπός – Στόχοι	1
1.3 Συνεισφορά	2
1.4 Διάθρωση της μελέτης	3
2. Σχεσιακές βάσεις δεδομένων	4
2.1 Τα θεμέλια των συστημάτων βάσεων δεδομένων και η αρχιτεκτονική τους	4
2.2 Το μοντέλο οντοτήτων-συσχέτισης	5
2.3 Ευρητηρίαση και κατακερματισμός	5
2.4 Αποθήκευση δεδομένων και επιχειρηματική νοημοσύνη	6
2.5 SQL	7
2.6 PostgreSQL	7
3. Βάσεις Δεδομένων Γράφων και η γλώσσα Cypher	10
3.1 Neo4j	10
3.2 Η γλώσσα Cypher	12
3.3 Τυπικές προδιαγραφές	14
3.4 Αντιστοίχιση μοτίβων και πρότυπων	15
3.5 Ιστορικές παρατηρήσεις	18
3.6 Τρέχουσες εξελίξεις	18
4. Γράφοι και AgensGraph	20
4.1 Γράφος	21
4.2 AgensGraph	27
4.3 Deployment	31
4.4 Αρχιτεκτονική	32
4.5 Ασφάλεια	34
4.6 Πιστοποίηση με Κωδικό Πρόσβασης	35
5. Σύγκριση	37
5.1 Εισαγωγή Δεδομένων	38
5.2 Σύγκριση AgensGraph με TigerGraph	46
5.3 Σύγκριση AgensGraph με Neo4j	47
5.4 Χαρακτηριστικά και εγκατάσταση	51

5.5 Πείραμα	55
6. Μελλοντικές επεκτάσεις	58

Κατάλογος Εικόνων

Εικόνα 1: Neo4j.....	11
Εικόνα 2: Labeled Property Graph Model.....	28
Εικόνα 3: AgensGraph Simple Data Model.....	29
Εικόνα 4: Edge Label inheritance.....	31
Εικόνα 5: AgensGraph Architecture.....	33
Εικόνα 6: Σύνολο δεδομένων.....	37
Εικόνα 7: Neo4j εγγραφή	45
Εικόνα 8: Model	46
Εικόνα 9: Σελίδα κατεβάσματος	47
Εικόνα 10: Σχήμα πειράματος	48
Εικόνα 11: AgensGraph command line	48
Εικόνα 12: Απαιτήσεις AgensGraph	49
Εικόνα 13: Model AgensGraph.....	49
Εικόνα 14: Query 13.....	49

1 Εισαγωγή

1.1 Πρόβλημα – Σημαντικότητα του θέματος

Η AgensGraph είναι ένα ανοιχτού κώδικα πολυμορφικό σύστημα διαχείρισης βάσεων δεδομένων που αναπτύχθηκε από την BitnineGlobal. Είναι κατασκευασμένη πάνω στη βάση δεδομένων PostgreSQL και υποστηρίζει και τις γλώσσες ερωτημάτων SQL και Cypher, καθιστώντας την ιδανική επιλογή για αποθήκευση και αναζήτηση δεδομένων γράφων. Ένα σημαντικό χαρακτηριστικό της AgensGraph είναι η ικανότητα της να ανταποκρίνεται σε πολύπλοκα και με πολλές σχέσεις δεδομένα. Αυτό την καθιστά ιδανικό για εφαρμογές που απαιτούν υψηλά συνδεδεμένα δεδομένα, όπως κοινωνικά δίκτυα, συστάσεις συστημάτων και ανίχνευση απάτης. Ένα άλλο σημαντικό στοιχείο της AgensGraph είναι η ευελιξία της. Είναι σχεδιασμένη να αντεπεξέλθει σε μεγάλους όγκους δεδομένων και μπορεί εύκολα να κλιμακωθεί, προσθέτοντας περισσότερες μηχανές στο cluster. Επιπλέον, η AgensGraph έχει καλή απόδοση, μπορεί να αντεπεξέλθει σε μεγάλα σύνολα δεδομένων και πολύπλοκες ερωτήσεις με υψηλή ταχύτητα και χαμηλή καθυστέρηση. Αυτό την καθιστά κατάλληλη για αναλυτικές εφαρμογές και λήψεις αποφάσεων σε πραγματικό χρόνο. Η AgensGraph είναι μια καινούρια βάση δεδομένων και κατέχει μεγάλο κοινό το οποίο συνεισφέρει στην ανάπτυξη και στην υποστήριξη της. Στη διπλωματική θα μελετηθούν οι σχεσιακές και μη σχεσιακές βάσεις δεδομένων και θα συγκριθούν με την AgensGraph.

1.2 Σκοπός – Στόχοι

Στην εργασία μου, εμβάθυνα σε μια εκτεταμένη εξερεύνηση της AgensGraph, ενός αναδυόμενου και καινοτόμου συστήματος βάσεων δεδομένων, που συνδυάζει τις δυνατότητες των βάσεων δεδομένων γράφων με τα χαρακτηριστικά που συνήθως συσχετίζονται με τις σχετικές βάσεις δεδομένων. Ο βασικός στόχος της έρευνάς μου ήταν να αξιολογήσω εκτενώς τα πλεονεκτήματα, τους περιορισμούς και τα πρακτικά σενάρια χρήσης που περιβάλλουν την AgensGraph μέσα στο σύγχρονο τοπίο της διαχείρισης δεδομένων. Εστιάζοντας σε πρακτικά παραδείγματα επιδίωξα να αποκαλύψω τη δυνατότητά της να διαχειρίζεται επιδέξια πολύπλοκες σχέσεις δεδομένων, διατηρώντας παράλληλα τα αναμενόμενα επίπεδα ακεραιότητας και συνοχής που συνδέονται παραδοσιακά με τις σχετικές βάσεις δεδομένων. Μέσω εκτενών

αξιολογήσεων απόδοσης και προσεκτικών συγκριτικών αναλύσεων, σκόπευα να προσφέρω μια ολιστική κατανόηση των ξεχωριστών πλεονεκτημάτων της AgensGraph σε σύγκριση με τις συμβατικές βάσεις δεδομένων και εναλλακτικές λύσεις βάσεων δεδομένων γράφων. Αυτή η προσπάθεια φωτίζει όχι μόνο τις τεχνικές λεπτομέρειες που είναι ουσιώδεις για τη AgensGraph, αλλά συνεισφέρει επίσης στον ευρύτερο διάλογο περί προηγμένων τεχνολογιών βάσεων δεδομένων και τον κρίσιμο ρόλο τους στο σχεδιασμό της πορείας των εφαρμογών που τρέφονται από τα πολλά δεδομένα.

1.3 Συνεισφορά

Μία από τις σημαντικές συνεισφορές αυτής της έρευνας βρίσκεται στην εκτεταμένη αξιολόγηση και ανάλυση των δυνατοτήτων της AgensGraph ως πρωτοποριακού υβριδικού συστήματος βάσης δεδομένων. Αυτή η μελέτη εξετάζει τα χαρακτηριστικά της που συνδυάζουν τα πλεονεκτήματα που συνδέονται με τις βάσεις δεδομένων γράφων, με τις λειτουργίες που προσφέρουν οι σχετικές βάσεις δεδομένων, αναδεικνύοντας έτσι την ικανότητά της να διαχειρίζεται αποτελεσματικά περίπλοκες σχέσεις δεδομένων και ιεραρχίες. Μέσω πρακτικών αξιολογήσεων και προσεκτικών αναλύσεων απόδοσης αυτή η έρευνα όχι μόνο υπογραμμίζει την επάρκεια της AgensGraph στην μοντελοποίηση περίπλοκων αλληλεπιδράσεων, αλλά δείχνει επίσης το επιδέξιο χειρισμό της παροχής ομαλών μηχανισμών ερωτήσεων και διάβασης. Επιπλέον, αυτό το έργο εμβαθύνει στην ικανότητα της AgensGraph να υποστηρίζει προηγμένες δομές δεδομένων, όπως τα JSON και οι δείκτες GIST, ενισχύοντας έτσι την προσαρμοστικότητά της για σύγχρονες εφαρμογές, προσδιορίζοντας τα πλεονεκτήματά της τόσο στον τομέα της μοντελοποίησης δεδομένων όσο και στην απόδοση των ερωτημάτων. Αυτή η μελέτη παρέχει εμπειρία και γνώση σε προγραμματιστές, μηχανικούς δεδομένων και ερευνητές που διερευνούν προηγμένες λύσεις βάσεων δεδομένων. Τα αποτελέσματα εμπλουτίζουν τον αυξανόμενο αποθεματικό γνώσης σχετικά με τις υβριδικές βάσεις δεδομένων, διευκολύνοντας την ενημερωμένη λήψη αποφάσεων σχετικά με τις επιλογές αρχιτεκτονικής βάσεων δεδομένων και προωθώντας την υιοθέτηση καινοτόμων λύσεων, όπως της AgensGraph.

1.4 Διάρθρωση της μελέτης

Η συγκεκριμένη έρευνα αποτελείται από πέντε κεφάλαια. Ειδικότερα, στο πρώτο κεφάλαιο αναλύονται οι στόχοι της έρευνας καθώς και η συνεισφορά της στον ερευνητικό κόσμο. Τέλος, περιγράφεται η δομή της εργασίας. Στο δεύτερο κεφάλαιο, αναφερόμαστε στα θεμέλια των συστημάτων βάσεων δεδομένων και για την αρχιτεκτονική τους. Στο τρίτο κεφάλαιο αναφέρουμε μια από της πιο γνωστές βάσεις δεδομένων γράφων, την neo4j. Στο τέταρτο κεφάλαιο αναφερόμαστε στην κύρια έρευνα της διπλωματικής μας που είναι η βάση δεδομένων AgensGraph. Και στο πέμπτο και τελευταίο κεφάλαιο κάνουμε κάποιες συγκρίσεις και αναφέρουμε τα συμπεράσματα.

2 Σχεσιακές βάσεις δεδομένων

2.1 Τα θεμέλια των συστημάτων βάσεων δεδομένων και η αρχιτεκτονική τους

Οι σχεσιακές βάσεις δεδομένων αποθηκεύουν τα δεδομένα σε μια δομημένη μορφή, χρησιμοποιώντας πίνακες που αποτελούνται από γραμμές και στήλες. Κάθε πίνακας αντιπροσωπεύει μια οντότητα, και οι στήλες αντιπροσωπεύουν τα χαρακτηριστικά αυτής της οντότητας. Το σχεσιακό μοντέλο βασίζεται στην ιδέα ότι όλα τα δεδομένα μπορούν να αναπαριστούν ως συλλογή σχέσεων, όπου κάθε σχέση είναι ένας πίνακας και κάθε γραμμή στον πίνακα αντιπροσωπεύει μια πλειάδα. Η αρχιτεκτονική μιας σχεσιακής βάσης δεδομένων περιλαμβάνει ένα σύστημα διαχείρισης βάσεων δεδομένων (DBMS) που είναι υπεύθυνο για τη διαχείριση των δεδομένων που αποθηκεύονται στη βάση δεδομένων, συμπεριλαμβανομένης της δημιουργίας και τροποποίησης πινάκων, της αναζήτησης και τροποποίησης δεδομένων και της εφαρμογής περιορισμών ακεραιότητας δεδομένων. Το DBMS περιλαμβάνει επίσης έναν βελτιστοποιητή αναζήτησης που είναι υπεύθυνος για τον προσδιορισμό του πιο αποτελεσματικού τρόπου εκτέλεσης μιας αναζήτησης και έναν διαχειριστή αποθήκευσης, που είναι υπεύθυνος για τη διαχείριση της φυσικής αποθήκευσης των δεδομένων, συμπεριλαμβανομένης της χρήσης τεχνικών ευρετηρίου και cache για τη βελτιστοποίηση της απόδοσης. Ένα από τα κύρια πλεονεκτήματα των σχεσιακών βάσεων δεδομένων είναι η δεδομένη ανεξαρτησία, που σημαίνει ότι η λογική οργάνωση των δεδομένων είναι διαφορετική από τον τρόπο που φυσικά αποθηκεύεται, επιτρέποντας στις εφαρμογές να έχουν πρόσβαση στα δεδομένα χωρίς να είναι σε θέση να γνωρίζουν τον τρόπο αποθήκευσης. Ένα άλλο σημαντικό χαρακτηριστικό των σχεσιακών βάσεων δεδομένων είναι η ακεραιότητα των δεδομένων, η οποία προστατεύει την ακρίβεια, την ακαταλληλότητα και την αξιόπιστη προέλευση των δεδομένων, εφαρμόζοντας περιορισμούς και κανόνες που ελέγχουν τις σχέσεις και τις δομές των δεδομένων. Το σχεσιακό μοντέλο χρησιμοποιείται ευρεία στην πρακτική, είναι το πιο δημοφιλές μοντέλο που χρησιμοποιείται σε εμπορικά συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS) όπως το MySQL, το Oracle, το SQL Server και το PostgreSQL.

2.2 Το μοντέλο οντοτήτων-συσχέτισης

Το εννοιολογικό μοντέλο δεδομένων γνωστό ως το Entity-Relationship (ER) χρησιμοποιείται για να αναπαριστά τη δομή βάσεων δεδομένων. Είναι μια γραφική αναπαράσταση των συνδέσεων μεταξύ των στοιχείων, που είναι πράγματα ή ιδέες, που διατηρούνται στη βάση δεδομένων. Επειδή επιτρέπει στον χρήστη να δημιουργήσει μια γραφική αναπαράσταση των σχέσεων μεταξύ πινάκων που μπορούν να μεταφραστούν σε ένα φυσικό σχεδιασμό βάσης δεδομένων, το ER μοντέλο χρησιμοποιείται για τον σχεδιασμό σχεσιακών βάσεων δεδομένων. Τα στοιχεία, τα χαρακτηριστικά και οι σχέσεις αποτελούν τα τρία βασικά μέρη του μοντέλου ER. Τα ορθογώνια στο διάγραμμα ER αντιπροσωπεύουν τα στοιχεία, που είναι έννοιες ή πράγματα που καταγράφονται στη βάση δεδομένων. Οι ελλείψεις στο διάγραμμα ER αντιπροσωπεύουν τα χαρακτηριστικά, που είναι οι ιδιότητες ή οι ιδιότητες ενός στοιχείου. Τα διαμάντια στο διάγραμμα ER αντιπροσωπεύουν τις σχέσεις, που είναι οι συνδέσεις μεταξύ των στοιχείων. Το ER μοντέλο είναι χρήσιμο για τον σχεδιασμό βάσης δεδομένων επειδή επιτρέπει στον χρήστη να αναγνωρίσει τα στοιχεία και τις σχέσεις που είναι σημαντικές για το σύστημα και βοηθά στην εξασφάλιση ότι η βάση δεδομένων είναι δομημένη με έναν τρόπο που είναι εύκολος να κατανοηθεί και να διαχειριστεί. Επίσης, επιτρέπει την αναγνώριση των περιορισμών και των κανόνων που πρέπει να διατηρηθούν για να προστατευθεί η ακεραιότητα των δεδομένων. Χρησιμοποιώντας το για να κατασκευάσει ένα λογικό μοντέλο της βάσης δεδομένων, το οποίο μπορεί να χρησιμοποιηθεί για να δημιουργηθεί ένας φυσικός σχεδιασμός βάσης δεδομένων, χρησιμοποιώντας ένα συγκεκριμένο σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS).

2.3 Ευρετηρίαση και κατακερματισμός

Η ευρετηρίαση και η αποθήκευση είναι τεχνικές που χρησιμοποιούνται για να βελτιώσουν την απόδοση των συστημάτων βάσεων δεδομένων σε σχεσιακές βάσεις δεδομένων. Η ευρετηρίαση είναι η διαδικασία δημιουργίας ενός ευρετηρίου, μιας δομής δεδομένων, που βελτιώνει την ταχύτητα των ενεργειών ανάκτησης δεδομένων σε έναν πίνακα βάσης δεδομένων, με κόστος πρόσθετων εγγραφών και χώρου αποθήκευσης για να διατηρηθεί. Υπάρχουν διάφοροι τύποι ευρετηρίων όπως B-tree, Bitmap και Hash ευρετήρια. Τα δεδομένα οποιουδήποτε μεγέθους μπορούν να αντιστοιχιστούν σε δεδομένα σταθερού μεγέθους χρησιμοποιώντας μια διαδικασία που ονομάζεται αποθήκευση. Αυτό χρησιμοποιείται συνήθως για αναζήτηση δεδομένων σε μια βάση δεδομένων και μπορεί να βελτιώσει την απόδοση ενεργειών όπως η αναζήτηση και η ανάκτηση δεδομένων. Αυτές οι δύο μέθοδοι μπορούν να συνδυαστούν για τα καλύτερα αποτελέσματα επειδή και οι δύο λειτουργούν για να επιταχύνουν τις διαδικασίες ανάκτησης δεδομένων μειώνοντας το ποσό των δεδομένων που πρέπει να αναζητηθούν.

2.4 Αποθήκευση δεδομένων και επιχειρηματική ευφυΐα

Τα συστήματα αποθήκευσης δεδομένων και επιχειρησιακής πληροφόρησης σχεδιάστηκαν για να αποθηκεύουν, να διαχειρίζονται και να αναλύουν μεγάλες ποσότητες δεδομένων από διάφορες πηγές προκειμένου να υποστηρίξουν την λήψη αποφάσεων και το στρατηγικό σχεδιασμό για τους οργανισμούς. Η διαδικασία σχεδίασης και υλοποίησης αυτών των συστημάτων περιλαμβάνει διάφορα βήματα, όπως το σχεδιασμό δεδομένων, την ενσωμάτωση δεδομένων, τη διαχείριση ποιότητας δεδομένων και την πρόσβαση και την ανάλυση δεδομένων. Τα συστήματα αποθήκευσης δεδομένων δημιουργούνται ώστε να κρατά τα δεδομένα από πολλές πηγές σε ένα μέρος και να διαμορφωθούν σε μια μορφή που είναι καλύτερη για αναφορές και ανάλυση. Για να οργανώνονται τα δεδομένα σε πράγματα και διαστάσεις που μπορούν να προσπελαστούν και να αξιολογηθούν γρήγορα, αυτό συνήθως περιλαμβάνει τη χρήση εργαλείων διαμόρφωσης διαστάσεων. Τα συστήματα επιχειρησιακής πληροφόρησης σχεδιάστηκαν για να παρέχουν στους χρήστες πρόσβαση στα δεδομένα που αποθηκεύονται στο data warehouse και να τους παρέχουν τα εργαλεία και τις οπτικοποιήσεις που χρειάζονται για

να αναλύσουν και να λάβουν αποφάσεις με βάση αυτά τα δεδομένα. Αυτό συνήθως περιλαμβάνει τη χρήση αναφορών, ταμπλό και εργαλεία OLAP (Online Analytical Processing) [1].

2.5 SQL

Σε βάσεις δεδομένων σχεσιακού τύπου, η SQL (Structured Query Language) είναι μια πρότυπη γλώσσα για την αποθήκευση, ανάκτηση και επεξεργασία δεδομένων. Επιτρέπει στους τελικούς χρήστες να αλληλεπιδρούν με βάσεις δεδομένων και να εκτελούν εργασίες όπως προσθήκη, αφαίρεση και ενημέρωση βάσεων δεδομένων. Η SQL χρησιμοποιείται από σχεδόν όλες τις επιχειρήσεις μεσαίας έως μεγάλης κλίμακας, συμπεριλαμβανομένων του Facebook, Microsoft, LinkedIn και Accenture. Η SQL κατέλαβε την πρώτη θέση ως το πιο χρησιμοποιούμενο περιβάλλον βάσης δεδομένων και ήταν η τρίτη πιο δημοφιλέστερη γλώσσα προγραμματισμού από το StackOverflow το 2021. Η SQL έχει πολλές χρήσεις και χρησιμοποιείται στο μάρκετινγκ, στην υγεία και στα χρηματοοικονομικά για ανάλυση δεδομένων και επιχειρησιακή ανάλυση, ανάπτυξη και επιστήμη δεδομένων. Για την δημιουργία εφαρμογών για κινητά και ιστού, οι προγραμματιστές χρησιμοποιούν την SQL. Ένα όνομα χρήστη και ο κωδικός πρόσβασης που δημιουργείται κατά την εγκατάσταση μιας εφαρμογής κινητού μετατρέπονται αμέσως σε έναν κώδικα που είναι αδύνατο για τους χάκερ να σπάσουν, και αποθηκεύονται με ασφάλεια σε μια βάση δεδομένων με την χρήση της SQL. Επίσης, όταν αλληλεπιδράμε με μια εφαρμογή, εγγραφόμαστε σε ένα ενημερωτικό δελτίο ή κάνουμε μια αγορά ηλεκτρονικά, τα δεδομένα μας συλλέγονται και αποθηκεύονται σε μια βάση δεδομένων.

Τέλος οι αναλυτές δεδομένων είναι ειδικοί που διαθέτουν γνώση και βλέπουν πρότυπα στα δεδομένα για να υποστηρίξουν τη διαδικασία προσανατολισμού της επιχείρησης. Αναλύουν και εξάγουν δεδομένα χρησιμοποιώντας SQL για να παράγουν γνώσεις που μπορούν να εφαρμοστούν στην πράξη. Οι μάνατζερ και οι ενδιαφερόμενοι χρησιμοποιούν τα πρότυπα που βρίσκουν οι αναλυτές δεδομένων για να καθοδηγήσουν τις ενέργειές τους και να αυξήσουν την αξία της επιχείρησης του οργανισμού [5].

2.6 PostgreSQL

Η PostgreSQL είναι ένα ισχυρό και ανοιχτού κώδικα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS) που είναι γνωστό για τις ισχυρές δυνατότητές της, συμπεριλαμβανομένων των προηγμένων δυνατοτήτων σχεδίασης δεδομένων και ρυθμίσεων βελτιστοποίησης απόδοσης. Η PostgreSQL υποστηρίζει προηγμένες τεχνικές για σχεδίαση δεδομένων, συμπεριλαμβανομένου της κληρονομικότητας, πίνακα προηγμένων τύπων δεδομένων, υποστήριξης για αποθηκευμένες διαδικασίες, συναγερμούς και χρήστης καθορισμένων συναρτήσεων. Μπορεί να διαχειρίζεται τεράστια ποσότητα δεδομένων και να προσφέρει γρήγορη ταχύτητα αναζήτησης. Περιέχει επίσης υποστήριξη για αποθήκευση δεδομένων και business intelligence.

Η PostgreSQL έχει μια ποικιλία επιλογών για προσαρμογή απόδοσης, συμπεριλαμβανομένων caching, indexing και τη δυνατότητα αλλαγής ρυθμίσεων χρήσης μνήμης και δίσκου. Σε πλατφόρμες πολλαπλών πυρήνων υποστηρίζει επίσης την εκτέλεση παράλληλων ερωτημάτων, το οποίο βελτιώνει την απόδοση ερωτημάτων. Επιπλέον, παρέχει σύνθετες λειτουργίες, συμπεριλαμβανομένων replication, full-text search και υποστήριξη για χωρο-γεωγραφικά δεδομένα. Η PostgreSQL παρέχει επίσης ενσωματωμένη υποστήριξη για SQL, την πρότυπη γλώσσα για σχεσιακές βάσεις δεδομένων και μπορεί να χρησιμοποιηθεί με μια ποικιλία από προγραμματιστικές γλώσσες, όπως Java, Python και C++. Επιπλέον, υποστηρίζει μια πληθώρα από λειτουργικά συστήματα, συμπεριλαμβανομένων Windows, Linux και macOS.

Η PostgreSQL χρησιμοποιείται σε:

- Apple App Store: Για να διαχειρίζεται εκατομμύρια συναλλαγές καθημερινά και ήταν σε θέση να αυξήσει εύκολα την απαίτηση.
- New York Times: Για αποθήκευση και ανάκτηση μεγάλων ποσοτήτων δεδομένων και είναι σε θέση να ανταποκρίνεται σε υψηλή φόρτο δεδομένων και κυκλοφορίας στον ιστότοπό της.
- Οργανισμός Ηνωμένων Εθνών: Για την αποθήκευση δεδομένων σχετικά με την επικαλυπτόμενη ασφάλεια τροφής και την αγροτική στατιστική και είναι σε θέση να διαχειρίζεται μεγάλα ποσά δεδομένων από διάφορες πηγές.

- The Internet Archive: Για αποθήκευση και ανάκτηση δισεκατομμυρίων αντικειμένων, όπως σελίδες ιστού, βιβλία και βίντεο και είναι σε θέση να ανταποκρίνεται σε υψηλό φόρτο δεδομένων και κυκλοφορίας στον ιστότοπό του.

Και σε πολλά άλλα. Η PostgreSQL είναι γνωστή για τις προχωρημένες δυνατότητές της όπως την αναζήτηση κειμένου, την υποστήριξη δεδομένων χωρικής διάστασης και την αντιγραφή.

- Η αναζήτηση πλήρους κειμένου επιτρέπει την γρήγορη και αποδοτική αναζήτηση δεδομένων κειμένου με την ευρετηριοποίηση και την αναζήτηση πολλών ποσών δεδομένων κειμένου. Υποστηρίζει διάφορες γλώσσες και λειτουργίες, όπως το stemming, τα συνώνυμα και την βαθμολόγηση.
- Η υποστήριξη δεδομένων χωρικού επιπέδου επιτρέπει την αποθήκευση και την επεξεργασία δεδομένων με ένα χωρικό συστατικό, όπως σημεία, γραμμές και πολύγωνα. Παρέχει υποστήριξη για διάφορους τύπους χωρικών δεδομένων, χωρικούς παράγοντες και χωρικά ευρετήρια.
- Η αντιγραφή είναι μια τεχνική που χρησιμοποιείται για την αντιγραφή δεδομένων από μια βάση δεδομένων σε μια άλλη, επιτρέποντας την αύξηση της διαθεσιμότητας δεδομένων και την ανάκτηση από καταστροφές. Το PostgreSQL υποστηρίζει διάφορες τεχνικές αντιγραφής, όπως την αντιγραφή ροής, την αντιγραφή λογικής και την αντιγραφή πολλαπλών κεντρικών [2].

Όλες αυτές οι λειτουργίες καθιστούν την PostgreSQL σε ένα πολύ δυνατό και ευέλικτο εργαλείο διαχείρισης δεδομένων. Η PostgreSQL, όπως αναφέραμε, είναι ένα ανοιχτού κώδικα σύστημα διαχείρισης βάσης δεδομένων σχεσιακού τύπου, που αρχικά αναπτύχθηκε στο πανεπιστήμιο της Καλιφόρνιας, Berkeley το 1980. Το έργο ηγήθηκε από τον Michael Stonebraker, ο οποίος επίσης βοήθησε στην ανάπτυξη των βάσεων δεδομένων Ingres και Illustra. Η PostgreSQL κυκλοφόρησε το 1996 ως "Postgres95". Από τότε, έχει υποστεί πολλές βελτιώσεις και ενημερώσεις και έχει γίνει μια δημοφιλής επιλογή για χρήση σε επιχειρήσεις και ιστοσελίδες εφαρμογών [3].

3 Βάσεις Δεδομένων Γράφων και η γλώσσα Cypher

Τα συστήματα διαχείρισης βάσεων δεδομένων ιεραρχούν και αποθηκεύουν τα δεδομένα σε συλλογές δεδομένων και μέσω των πινάκων σύνδεσης και με τη χρήση κλειδιών πραγματοποιείται η σύνδεση των δεδομένων. Αποτέλεσμα αυτής της διαδικασίας είναι να μειώνεται η αποδοτικότητα του συστήματος στην απάντηση περίπλοκων ερωτημάτων, ενώ η περιγραφή πολύπλοκων σχέσεων του πραγματικού κόσμου στις σχεσιακές βάσεις δεδομένων είναι δύσκολη. Σήμερα, παρατηρείται έντονο ενδιαφέρον για τις βάσεις δεδομένων γράφων, οι οποίες έχουν την δυνατότητα να αποθηκεύουν τα δεδομένα με μορφή γράφου. Έτσι, υπάρχει η δυνατότητα να απεικονίζονται με φυσικό τρόπο στη βάση δεδομένων όλες οι συνδέσεις που υφίστανται στον πραγματικό κόσμο.

3.1 Neo4j

Από το 2015 το project Cypher ανοιχτού κώδικα επιδιώκει να ενεργοποιήσει τη χρήση της Cypher ως τυποποιημένης γλώσσας ικανής για πολλαπλές ανεξάρτητες υλοποιήσεις και για την παροχή ενός πλαισίου για τη συλλογική εξέλιξη νέων γλωσσικών χαρακτηριστικών μεταξύ υλοποιητών. Ο στόχος είναι η cypher να ωριμάσει σε μία τυπική γλώσσα του κλάδου για την αναζήτηση γράφων ιδιοτήτων, παίζοντας έναν συμπληρωματικό ρολό σε αυτό του προτύπου SQL για σχεσιακή αναζήτηση δεδομένων. Η cypher 9 είναι η πρώτη έκδοση της γλώσσας που διέπεται από τον ανοιχτό κώδικα cypher. Η cypher 10 είναι μια γλώσσα σύνθεσης και υποστηρίζει προβολές γραφημάτων και πολλαπλά επώνυμα γραφήματα. Το μοντέλο δεδομένων neo4j που χρησιμοποιείται από την cypher είναι αυτό της ιδιοκτησίας γραφικών παραστάσεων και είναι το πιο δημοφιλές μοντέλο δεδομένων γράφων στη βιομηχανία και τείνει να γίνει το ίδιο και στον ακαδημαϊκό χώρο.

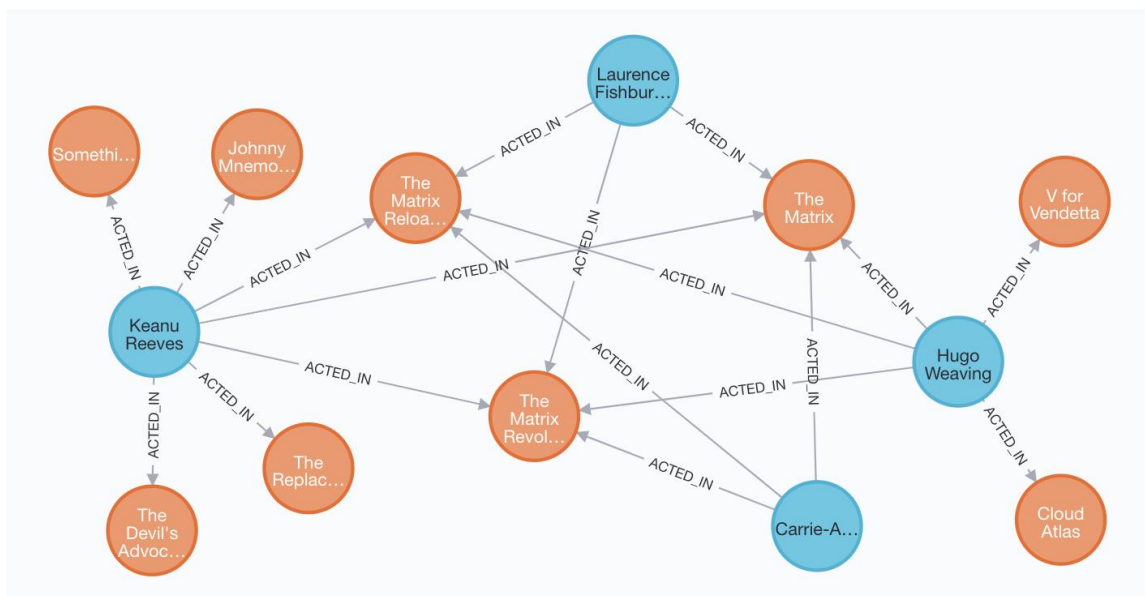
Το μοντέλο περιλαμβάνει κόμβους που αντιπροσωπεύουν οντότητες π.χ. άτομα και σχέσεις π.χ. συνώνυμα με τα άκρα και αντιπροσωπεύει τις συνδέσεις ή τις σχέσεις μεταξύ των οντοτήτων. Στο μοντέλο γράφων οι σχέσεις είναι το ίδιο σημαντικές με τις

ίδιες τις οντότητες. Επιπλέον οποιοσδήποτε αριθμός ιδιοτήτων με τη μορφή ζευγών κλειδιού- τιμής μπορεί να σχετίζεται με τους κόμβους και τις σχέσεις.

Αυτό επιτρέπει τη μοντελοποίηση και αναζήτηση συνθετών δεδομένων.

Η γλώσσα συνοδεύεται από μία πλήρως επισημοποιημένη σημασιολογία του πυρήνα κατασκευής της. Η ανάγκη για αυτό πηγάζει από το γεγονός ότι η cypher εκτός από την υλοποίηση σε βιομηχανικό προϊόν με σημαντική πελατειακή βάση , έχει συλλεχτεί από άλλους και υπάρχουν αρκετές υλοποιήσεις της. Δεδομένης της έλλειψης πρότυπου για την γλώσσα, μπορεί να χρειαστούν χρόνια να ολοκληρωθεί, όπως έγινε για την SQL έχει γίνει πειστικό να γίνει αποδεκτό το επίσημο μοντέλο δεδομένων και το νόημα των κυρίων κατασκευών. Μια τυπική σημασιολογία έχει άλλα πλεονεκτήματα για παράδειγμα, επιτρέπει σε κάποιον να συλλογιστεί σχετικά με την ισοδυναμία των ερωτημάτων και να αποδείξει την ορθότητα των υφιστάμενων ή να ανακαλύψει νέες βελτιστοποιήσεις.

Η ανάγκη της τυπικής σημασιολογίας έχει από πολύ καιρό αποδεχτεί στον τομέα των γλωσσών προγραμματισμού και για πολλές κοινές γλώσσες έχει ολοκληρωθεί η σημασιολογία τους. Πρόσφατα έγιναν παρόμοιες προσπάθειες για τον πυρήνα της SQL με στόχο την απόδειξη της ορθότητας SQL βελτιστοποιήσεων και κατανοώντας την εκφραστικότητα των χαρακτηριστικών. Η ύπαρξη της τυπικής σημασιολογίας της cypher την κάνει να είναι δυνατή για διαφορετικές υλοποιήσεις για να συμφωνήσουν σχετικά με τα βασικά χαρακτηριστικά της και δίνει την δυνατότητα για μια υλοποίηση αναφοράς με την οποία θα συγκριθούν άλλα συστήματα. Η παροχή σημασιολογίας για μια υπάρχουσα πραγματική γλώσσα πχ cypher που καλύπτει όλες της ιδιοσυγκρασίες της είναι πολύ πιο δύσκολο από ότι για τους θεωρητικούς λογισμούς υποκείμενα κύρια χαρακτηριστικά των γλωσσών προγραμματισμού.



Εικόνα 1: Neo4j source (<https://neo4j.com/developer/example-project/>)

3.2 Η γλώσσα Cypher

Η cypher είναι μια δηλωτική γλώσσα γραφημάτων ιδιοτήτων, επίσης παρέχει δυνατότητες τόσο για αναζήτηση όσο και για τροποποίηση δεδομένων επειδή καθορίζει τους ορισμούς των σχημάτων.

Γραμμικά ερωτήματα : Ένα ερώτημα cypher λαμβάνει ως είσοδο ένα γράφημα ιδιοτήτων και βγάζει έναν πίνακα. Αυτοί οι πίνακες μπορούν να θεωρηθούν ότι παρέχουν δεσμεύσεις για παραμέτρους, που μαρτυρούν ορισμένα μοτίβα σε ένα γράφημα , να έχει γίνει κάποια πρόσθετη επεξεργασία σε αυτά. Η δομή της cypher είναι γραμμικά ερωτημάτων . Αυτό επιτρέπει στους χρήστες να σκεφτούν την επεξεργασία ερωτήματος , όπως ξεκινά από την αρχή του ερωτήματος κειμένου και μετά προχωρώντας γραμμικά μέχρι το τέλος. Κάθε ρήτρα σε ένα ερώτημα είναι μια συνάρτηση που παίρνει έναν πίνακα και βγάζει έναν πίνακα και που μπορεί και στα δυο να επεκταθεί ο αριθμός των

πεδίων και να προστεθούν νέες πλειάδες. Όλο το ερώτημα είναι τότε η σύνθεση αυτών των συναρτήσεων. Αυτή είναι η γραμμική σειρά του. Οι ρήτρες νοούνται καθαρά δηλωτικά και χρειάζεται να γίνει διάταξη εκ νέου της εκτέλεσης των ρητρών , εάν αυτό δεν αλλάξει τη σημασιολογία του ερωτήματος . Έτσι, αντί να δηλώνεται η προβολή στην αρχή του ερωτήματος όπως κάνει η SQL με το ‘select’, στην cypher, η προβολή πηγαίνει στο τέλος του ερωτήματος ως ‘return’. Η γραμμική ροή ερωτημάτων στην cypher επεκτείνεται στην σύνθεση ερωτημάτων και χρησιμοποιώντας το ‘with’ το ερώτημα συνεχίζει με τον προβαλλόμενο πίνακα από το τμήμα του ερωτήματος πριν από το ‘with’ ως πίνακα οδήγησης για το ερώτημα μέρους μετά του ‘with’. Η ρήτρα ‘with’ επιτρέπει τις ίδιες προβολές με το ‘return’, συμπεριλαμβανομένων των συναθροίσεων. Υποστηρίζει επίσης φιλτράρισμα με βάση στα προβλεπόμενα πεδία. Εκτός από αυτόν τον γραμμικό τρόπο σύνταξης ερωτημάτων η cypher υποστηρίζει και ένθετα υπο-ερωτήματα πχ ‘Union’.

Ταίριασμα μοτίβων: Η κεντρική ιδέα στα ερωτήματα cypher είναι η αντιστοίχιση προτύπων. Τα μοτίβα της cypher εκφράζονται σε οπτική μορφή ως ‘ASCII art’ όπως (a) – [r] -> (b). Η ρήτρα ‘match’ στην cypher χρησιμοποιεί ένα τέτοιο μοτίβο και εισάγει νέες σειρές (συνώνυμες με έγγραφες) με δεσμεύσεις των ταιριασμένων περιπτώσεων του μοτίβου στο ερωτηθέν γράφημα. Η λειτουργικότητα της cypher επηρεάστηκε από το Xpath και το SPARQL και τα μοτίβα του εκφράζουν μια περιορισμένη μορφή κανονικής διαδρομής ερωτημάτων: η συνένωση και ο διαχωρισμός μιας έννοιας σχέσης τύπου καθώς και διαδρομές μεταβλητού μήκους (ουσιαστικά , μεταβατικό κλείσιμο). Η cypher υποστηρίζει ακόμα και μονοπάτια αντιστοίχισης και επιστροφής ως τιμές.

Τροποποίηση δεδομένων: Η cypher υποστηρίζει μια πλούσια γλώσσα ενημέρωσης τροποποιώντας το γράφημα ,παρέχει ενημέρωση ρητρών ,επαναχρησιμοποίηση της γλώσσας προτύπων οπτικού γράφων και την ίδια απλή από πάνω έως κάτω σημασιολογία, μοντέλο όπως το υπόλοιπο cypher . Οι βασικές ρήτρες για ενημερώσεις περιλαμβάνουν ‘create’ για την δημιουργία νέων κόμβων και σχέσεων, ‘delete’ για κατάργηση οντοτήτων και ‘set’ για ενημέρωση ιδιοτήτων. Ακόμα η cypher παρέχει μια ρήτρα που ονομάζεται ‘merge’ που προσπαθεί να ταιριάζει με το δεδομένο μοτίβο και δημιουργεί μοτίβο εάν δεν βρέθηκε αντιστοιχία . Η εφαρμογή της cypher μπορεί να χρησιμοποιήσει πρωταρχικά στοιχεία συγχρονισμού βάσης δεδομένων π.χ. κλείδωμα για

να διασφαλίσει ότι τα μοτίβα ταιριάζουν με το ‘merge’ και είναι μοναδικά μέσα στην βάση δεδομένων.

Πραγματιστική: Η cypher είναι σκόπιμα παρόμοια με την SQL για να βοηθήσει τη μετάβαση των χρηστών μεταξύ των δυο γλωσσών. Ακολουθεί την ίδια σύνταξη ρήτρας και υλοποιεί την καθιερωμένη σημασιολογία για πολλές λειτουργίες. Η cypher έχει ενσωματωμένη υποστήριξη για παραμέτρους ερωτήματος , καθιστώντας εύκολη την εξάλειψη των προβλημάτων της ένεσης ερωτήματος. Η επιλογή των προεπιλογών ευθυγραμμίζεται με την κοινή χρήση. Η σύνταξη για ομαδοποίηση και η συγκέντρωση είναι απλή , η γλώσσα έκφρασης περιλαμβάνει ισχυρά χαρακτηριστικά , όπως τεμαχισμό και κατανόηση λιστών, υπαρξιακά υπό-ερωτήματα και εργασία με μονοπάτια.

Εφαρμογή Neo4j: Η εκτέλεση ερωτημάτων στο neo4j ακολουθεί ένα συμβατικό μοντέλο που περιγράφεται από τον ‘Volcano Optimizer Generator’ . Η τελική συλλογή ερωτήματος χρησιμοποιεί είτε ένα απλό μοντέλο εκτελέσεως που βασίζεται σε επαναληπτικές πλειάδες ή μεταγλωττίζει το ερώτημα σε java bytecode με μια εκτέλεση που βασίζεται σε μοντέλο push-based. Ένα σχέδιο εκτέλεσης για ένα ερώτημα της cypher στο neo4j περιέχει σε μεγάλο βαθμό τους ίδιους τελεστές ,όπως στις μηχανές σχεσιακής βάσης δεδομένων και ένα πρόσθετο τελεστή που ονομάζεται ‘Expand’ . Το ‘Expand’ σημασιολογικά είναι αρκετά παρόμοιο σε μια σχεσιακή ένωση . Βρίσκει ζεύγη κόμβων που συνδέονται μέσω μιας άκρης. Ως προς την υλοποίηση αξιοποιεί το γεγονός ότι στα δεδομένα η αναπαράσταση του neo4j περιέχει σαφής αναφορές από κάθε κόμβο μέσω των ακρών του στους σχετικούς κόμβους. Αυτό σημαίνει ότι το ‘Expand’ μπορεί να διαβάσει τυχόν περιττά δεδομένα ή να προχωρήσει μέσω μιας έμμεσης κατεύθυνσης , όπως ένα ευρετήριο για την εύρεση σχετικών κόμβων.

3.3 Τυπικές προδιαγραφές

Οι επίσημες προδιαγραφές των γλωσσών έχουν πολλαπλά πλεονεκτήματα σε σχέση με την τεκμηρίωση γραμμένης σε φυσική γλώσσα . Μπορούν να χρησιμοποιηθούν στην λογική σχετικά με την γλώσσα και να αποδείξουν ότι οι βελτιστοποιήσεις είναι σωστές, μπορούν να οδηγήσουν σε μια υλοποίηση αναφοράς, που θα μπορούσε να

χρησιμοποιηθεί για να επαληθευτεί εάν μια συγκεκριμένη εφαρμογή της γλώσσας τηρεί τις προδιαγραφές της και αφήνουν λιγότερο περιθώριο ασάφειας σε σύγκριση με περιγραφές φυσικής γλώσσας. Τα βασικά στοιχεία της cypher είναι:

- Μοντέλο δεδομένων, που περιλαμβάνει τιμές, γραφήματα και πινάκες.
- Γλώσσα ερωτήματος, που περιλαμβάνει εκφράσεις, μοτίβα, προτάσεις και ερωτήματα.

Οι τιμές θα μπορούσαν να είναι απλές ή σύνθετες, όπως λίστες και χάρτες. Οι εκφράσεις δηλώνουν τιμές. Μοτίβα εμφανίζονται στο 'match'. Οι ρήτρες και τα ερωτήματα είναι ακολουθίες ρητρών. Κάθε ρήτρα υποδηλώνει μια συνάρτηση από πίνακες σε πίνακες. Μια σχέση και δυο συναρτήσεις ορίζουν μια επίσημη σημασιολογία της cypher. Η σχέση αντιστοίχισης προτύπου ελέγχει αν μια διαδρομή X σε ένα γράφημα Γ ικανοποιεί ένα μοτίβο Π κάτω από μια ανάθεση τιμών Y στις ελεύθερες μεταβλητές του μοτίβου. Αυτό γράφεται ως $(\Pi, \Gamma, Y) = X$.

3.4 Αντιστοίχιση μοτίβων και πρότυπων

Σύνταξη μοτίβων: Η γραμματική της cypher ορίζεται από την αμοιβαία αναδρομή εκφράσεων, προτύπων, από ρήτρες και ερωτήματα. Η περιγραφή των μοτίβων θα κάνει αναφορά σε εκφράσεις, οι οποίες θα καλύψουν στην συνέχεια και θα υποδηλώνουν τιμές. Αντί για την πραγματική σύνταξη cypher, χρησιμοποιείται μια αφηρημένη μαθηματική σημειογραφία που προσφέρεται πιο φυσικά σε μια επίσημη επεξεργασία.

Ένα πρότυπο κόμβου περιγράφεται από ένα τρίγωνο (a, L, P), όπου:

- Το 'a' είναι ένα προαιρετικό όνομα για τον κόμβο.
- Το 'L' είναι ένα πιθανώς κενό σύνολο ετικετών κόμβου.
- Το 'P' είναι ένα πιθανώς κενό χάρτης από κλειδιά ('K') προς εκφράσεις. Αναπαριστά τα χαρακτηριστικά του κόμβου.

Για παράδειγμα, το πρότυπο $(x:Person:Male \{name: expr1, age: expr2\})$ στη σύνταξη Cypher αναπαριστάται ως $(x, \{Person, Male\}, \{name \mapsto e1, age \mapsto e2\})$, όπου τα 'e1' και 'e2' είναι αναπαραστάσεις εκφράσεων.

Ένα πρότυπο σχέσης καθορίζεται από μια πλειάδα (d, a, T, P, I), όπου:

- Το 'd' καθορίζει την κατεύθυνση του προτύπου: '→' (αριστερά προς τα δεξιά), '←' (δεξιά προς τα αριστερά) ή '↔' (αμφίδρομη).
- Το 'a' είναι ένα προαιρετικό όνομα για τη σχέση.
- Το 'T' είναι ένα πιθανώς κενό σύνολο τύπων σχέσης.
- Το 'P' είναι ένα πιθανώς κενό χάρτης από κλειδιά ('K') προς εκφράσεις, αναπαριστώντας χαρακτηριστικά της σχέσης.
- Το 'I' καθορίζει την περιοχή του προτύπου, είτε ως 'nil' είτε ως μια πλειάδα '(m,n)' όπου τα 'm' και 'n' είναι ακέραιοι ή 'nil'.

Ένα πρότυπο διαδρομής είναι μια εναλλασσόμενη ακολουθία προτύπων κόμβου και προτύπων σχέσης: $\chi_1 \rho_1 \chi_2 \dots \rho_{n-1} \chi_n$. Ορίζει μια διαδρομή στο γράφο.

Ένα πρότυπο διαδρομής μπορεί να ονοματοποιηθεί και να αναφερθεί ως π πάνω από a . Είναι σταθερό αν όλα τα πρότυπα σχέσης είναι σταθερά, διαφορετικά είναι μεταβλητού μήκους. Το κείμενο εισάγει την έννοια της ικανοποίησης για πρότυπα διαδρομών σε έναν γράφο ιδιοτήτων. Δεδομένου ενός γράφου Ιδιοτήτων G , μιας διαδρομής p με αναγνωριστικά κόμβων και σχέσεων, και μιας ανάθεσης u , η σχέση ικανοποίησης καθορίζει εάν το πρότυπο διαδρομής ικανοποιείται από τον δεδομένο γράφο και την ανάθεση. Τα σταθερά πρότυπα καθορίζουν μοναδικά τόσο το μήκος όσο και τις δυνατές αναθέσεις μεταβλητών των διαδρομών που τα ικανοποιούν. Τα πρότυπα μεταβλητού μήκους καθορίζονται στη συνέχεια με βάση σύνολα σταθερών προτύπων. Στην cypher γίνεται προσπάθεια να επιστραφούν τα «ταιριάσματα», για ένα μοτίβο σε (α) γράφημα, και όχι απλώς να ελεγχθεί αν το μοτίβο ικανοποιείται (υπάρχει αντιστοιχία).

Ταίριασμα μοτίβων: Το σύνολο των ελεύθερων μεταβλητών ενός μοτίβου κόμβου $\chi=(a,l,p)$ που συμβολίζεται με $free(x)$, είναι (α) οπότε το a δεν είναι μηδέν και κενό. Σε διαφορετική περίπτωση για ένα μοτίβο π ορίζεται το σύνολο ελεύθερο (π) αναλογικά. Στη συνέχεια για ένα μοτίβο διαδρομής π ορίζεται $free(\pi)$ να είναι ένωση όλων των ελεύθερων μεταβλητών κάθε κόμβου και μοτίβα σχέσης συμβαίνουν σε αυτά. Η Cypher επιτρέπει σε κάποιον να ταιριάζει μια πλειάδα $\pi=(\pi_1, \dots, \pi_n)$ μοτίβων διαδρομής, το καθένα προαιρετικά με

Ο ισομορφισμός γράφων είναι η έννοια που χρησιμοποιείται συνήθως σε ερωτήματα γράφων. Αρχικά, ο μηχανισμός που ταιριάζει με το μοτίβο της Cypher μπορεί να φαίνεται μια προέκταση του, που οφείλεται σε πρόσθετα χαρακτηριστικά, όπως τιμές που σχετίζονται με κόμβους και σχέσεις. Όμως η Cypher απομακρύνεται από τον ισομορφισμό γράφων. Γιατί στον καθορισμό του πόσο άκαμπτα μοτίβα ικανοποιούνται, εξετάστηκαν μόνο μονοπάτια στα οποία τα αναγνωριστικά σχέσης εμφανίζονται το πολύ μια φορά. Δηλαδή ένα μονοπάτι δεν μπορεί να διασχίσει την ίδια σχέση (άκρη) περισσότερες από μία φορές, γιατί υπάρχει ένας τέτοιος περιορισμός και γιατί το επηρεάζει η πολυπλοκότητα της αντιστοίχισης προτύπων αντιστοίχιση μοτίβου γράφων είναι ένα τυπικό πρόβλημα NP-complete για ένα γράφημα G και ένα μοτίβο π , ελέγχοντας αν υπάρχει αντιστοιχία για το π in G είναι NP- πλήρες και είναι συνεπής με την εύρεση προτύπων στη σχεσιακή βάση δεδομένων D , ελέγχοντας αν το $Q(D)$ επιστρέφει μια πλειάδα και είναι επίσης NP-complete.

Η Cypher υπερβαίνει την απλή αντιστοίχιση, επιστρέφοντας μονοπάτια και τότε εμφανίζεται το πρόβλημα. Έστω ότι υπάρχει ένα γράφημα με ένα μόνο κόμβο η , με μεμονωμένη σχέση r , της οποίας η πηγή και ο στόχος είναι η και π το μοτίβο $(\chi)-\{O1\dots\}-\chi$. Κάτω από τη σημασιολογία ισομορφισμού θα έκανε επιστροφή γιατί υπάρχουν άπειρες διαδρομές από το η στο η και φαίνεται ότι ένα τέτοιο μοτίβο θα ταιριάζει άπειρες φορές δηλαδή για κάθε m μεγαλύτερο ή ίσο με το O , θα παράγει ένα ταίριασμα που διασχίζει τη μοναδική ακμή στο γράφημα m φορές. Η Cypher για αποφυγή αυτής της κατάστασης επέλεξε να απαγορεύσει την επανάληψη ακμών σχέσης ενώ ταιριάζουν μοτίβα.

Χρησιμοποιώντας το πρόβλημα της ύπαρξης δύο ασύνδετων μονοπατιών στο G γράφημα, μπορεί να αποδειχθεί ότι ελέγχοντας αν υπάρχει αντιστοιχία για το π στο G σύμφωνα με τη σημασιολογία της Cypher παραμένει NP-complete. Σε γεγονός, διαφορετικά από την περίπτωση της αντιστοίχισης που οφείλεται στον ισομορφισμό, υπάρχει ακόμα και μια σταθερή διαδρομή για την οποία ισχύει αυτό. Τα σταθερά μοτίβα έχουν μάλλον περίεργο σχήμα και δεν βασίζονται σε πολυετή πείρα, που είναι συνήθως φαινόμενο στην πράξη. Τα βασικά συστατικά της cypher δηλαδή οι εκφράσεις, οι ρήτρες, τα ερωτήματα μαζί με την αντιστοίχιση μοτίβων θα αποτελέσουν την επισημοποίηση του πυρήνα της cypher. Ένα ερώτημα είναι είτε μια ακολουθία ρητρών που τελειώνει με την δήλωση 'return' ή μια ένωση δυο ερωτημάτων. Η δήλωση 'return' περιέχει την λίστα επιστροφής (*) ή είναι ακολουθία εκφράσεων, προαιρετικά

ακολουθούμενη από AS α, που παρέχει τα όνομα τα τους. Υποθέτοντας ότι υπάρχει μια ενετική συνάρτηση α που αντιστοιχίζει εκφράσεις σε ονόματα και περιέχεται η σημασιολογία των ερωτημάτων. Η σημασιολογία τόσο των ερωτημάτων όσο και των προτάσεων, σε σχέση με ένα γράφημα ιδιοτήτων G, είναι μια συνάρτηση από πίνακες σε πίνακες, οπότε θα περιγραφεί η τιμή του σε πίνακα T, i.e. $[[query]]G(T)$.

Σημασιολογία προτάσεων: Το νόημα των ρητρών cypher είναι και πάλι συναρτήσεις που μεταφέρουν τους πίνακες σε πίνακες και χωρίζονται σε δυο κατηγορίες. Τις ρήτρες αντιστοίχισης που είναι ουσιαστικά δηλώσεις αντιστοίχισης προτύπων και είναι της μορφής OPTIONAL MATCH μοτίβο πλειάδας WHERE expr. Το OPTIONAL και το WHERE θα μπορούσαν να παραληφθούν.

3.5 Ιστορικές παρατηρήσεις

Η γλώσσα ερωτημάτων cypher προέκυψε από την εξέλιξη της βάσης δεδομένων γραφημάτων neo4j, η οποία με την σειρά της προήρθε από ένα μοντέλο δεδομένων που επινοήθηκε για πρώτη φορά το 2000 από τους ιδρυτές του neo4j στην πορεία δημιουργίας ενός συστήματος διαχείρισης περιουσιακών στοιχείων μέσω του συστήματος. Το μοντέλο δεδομένων άλλαξε συχνά και είχε πολλαπλές δομές δεδομένων, έχει προβολή έλεγχου πρόσβασης, που ενέπνευσαν την ιδέα της προσθήκης των ετικετών στο δίκτυο με στοιχεία, με λεζάντες ή σύνολα ιδιοτήτων, η μεγάλη έμφαση στις πληροφορίες σχέσης σε συνδυασμό με την ανάγκη υποστήριξης δεδομένων σχήματος άκαμπτου, που χρησιμοποιήθηκε αρχικά για την δημιουργία ενός εγγενούς συστήματος βάσης δεδομένων γραφημάτων ιδιοκτησίας. Από το 2007 και μετά αυτή η τεχνολογία άρχισε να παρέχεται για γενική χρήση ως σύστημα διαχείρισης βάσεων δεδομένων ανοιχτού κώδικα. Αρχικά το neo4j ενσωματώθηκε ως βιβλιοθήκη java. Το Tinkerpop Blueprints και Gremlin APIs, που αποτελούν μέρος του έργου Apache, προέρχονται από τις πρώιμες ενσωματώσεις του neo4j. Η αυξημένη χρήση οδήγησε στην έναρξη μιας γλώσσας ερωτήματος γράφων ιδιοτήτων που θα καταλάμβανε το ίδιο οικολογική θέση ως SQL. Η ανάπτυξη αυτής της γλώσσας είναι παράλληλη με τις αλλαγές στην υλοποίηση της. [4]

3.6 Τρέχουσες εξελίξεις

Η εξέλιξη της γλώσσας cypher γίνεται από μια ομάδα προγραμματιστών που συναινετικά βοηθά στην εξέλιξη της. Τρεις νέες δυνατότητες προστίθενται στην επομένη έκδοση της γλώσσας, τα πολλαπλά γραφήματα, η σύνθεση ερωτήματος και οι τύποι χρονικών δεδομένων π.χ. (ημερομηνία, ώρα). Αυτά αντικατοπτρίζουν την εμπειρία βιομηχανικής χρήσης της cypher, ιδιαίτερα στο neo4j και το SAP HANA γράφημα.

Πολλαπλά γραφήματα. Η cypher 9 υποθέτει ένα σιωπηρό ενιαίο παγκόσμιο γράφημα, που χρησιμοποιείται τόσο για ερωτήματα όσο και για λειτουργίες ενημέρωσης. Οι εφαρμογές μερικές φορές λειτουργούν με πολλά αποσυνδεδεμένα υποστοιχεία αυτού του παγκοσμίου γραφήματος. Τέτοια στοιχεία μπορούν να θεωρηθούν ξεχωριστά γραφήματα ιδιοτήτων. Ωστόσο η cypher δεν το κάνει ακόμα, γιατί περιλαμβάνει έναν μηχανισμό για τη ρητή αναφορά τέτοιων γραφημάτων και επομένως η γλώσσά δεν επιτρέπει εύκολα την εφαρμογή πράξεων σε συγκεκριμένα γραφήματα. Η πρόταση της cypher 10 για πολλαπλά γραφήματα είναι η εισαγωγή του ονόματος αναφοράς γραφημάτων, που αντιπροσωπεύουν γραφήματα, τα οποία βρίσκονται εξωτερικά και δημιουργήθηκαν από το ερώτημα ή από τα γραφήματα, τα οποία δημιουργήθηκαν από το προηγούμενο ερώτημα σε μια σύνθεση ερωτημάτων. Οι αναφορές γραφημάτων μπορούν να διαβιβαστούν ως επιχειρήματα και να επιστραφούν ως αποτελέσματα από ερωτήματα της cypher 10 και μπορούν να χρησιμοποιηθούν σε λειτουργίες συνόλου. Ως προς το όνομα τα γραφήματα συμβάλλουν στη φυσική κατάτμηση και στους μετασχηματισμούς γραφημάτων, τα οποία με τη σειρά τους ενεργοποιούν ερωτήματα σύνθεσης γράφων.

Σύνθεση ερωτήματος: Απαιτείται η μετάδοση πολλαπλών αναφορών γράφων, η επέκταση του υπάρχοντος μηχανισμού σύνθεσης πινάκων της cypher 9, όπου η ρήτρα προβολής ‘With’ μετατρέπει την έξοδο ενός ερωτήματος σε έναν πίνακα οδήγησης εισόδου για ένα ακόλουθο ερώτημα. Η πρόταση της cypher 10 για πολλαπλά γραφήματα προσθέτει τη δυνατότητα να περάσει ένας πίνακας γραφημάτων κατασκευής που αποτελείται από έναν ενιαίο πίνακα και πολλαπλά επώνυμα γραφήματα ως επιχειρήματα ερωτήματος. Ένας πίνακας γραφημάτων μπορεί επιπλέον να μεταβιβάσει πληροφορίες σχετικά με ποια από αυτά τα γραφήματα χρησιμοποιείται για ανάγνωση. Παρόμοιας ερώτησης το αποτέλεσμα είναι ένας πίνακας γραφημάτων. Αυτό επιτρέπει τη σύνταξη ερωτημάτων cypher ως αλυσίδας ερωτημάτων. Με την προσθήκη υποερωτημάτων

τέτοιες αλυσίδες ερωτημάτων μπορούν επίσης να διαμορφωθούν σε δέντρο. Στην cypher η πινακοειδής προβολή αποτελεσμάτων αναγνωρίζεται ως απαραίτητη για την εφαρμογή πρόσβασης στις τιμές ιδιοτήτων και κατά συνέπεια η cypher 10 είναι κλειστή κάτω από πίνακες γραφημάτων. Η cypher 10 εισάγει τα επώνυμα ερωτήματα , που απλοποιούν τη δημιουργία βιβλιοθηκών, επαναχρησιμοποιήσεις ερωτημάτων που μπορούν να συντίθεται σε διαφορετικά ερωτηματικά δέντρα και αποτελούν τη βάση για την προσφορά προβολής γραφημάτων .

Γλώσσες ερωτημάτων γράφων: Μπορεί να εκφραστεί κάποια πλοήγηση σε γράφημα σε SQL χρησιμοποιώντας αναδρομικά ερωτήματα. Βασική ιδιότητα όπως η προσβασιμότητα σε ένα γράφημα μπορεί να εκφραστεί αρκετά εύκολα, αλλά πολύπλοκα μοτίβα γραφημάτων που περιλαμβάνουν δεδομένα και η πλοήγηση γίνεται πολύ δυσκίνητα χρησιμοποιώντας την αναδρομή της SQL. Ακόμα ορισμένα χαρακτηριστικά της cypher , όπως πχ η μη επαναληψιμότητα των ακρών στην αντιστοίχιση σχεδίων δεν μπορεί γενικά να εκφραστεί σε SQL. Ελλείψεις της SQL ως γλώσσας ερωτημάτων γράφων οδήγησε σε πολύ μεγάλο όγκο έρευνας σχετικά με την κατάλληλη γλώσσα ερωτημάτων γραφημάτων. Ερωτήματα κανονικής διαδρομής (RPQs) με δυνατότητα έκφρασης μιας διαδρομής μεταξύ οποιονδήποτε δυο κόμβων ως κανονική έκφραση πάνω από τις ετικέτες ακμών, προστέθηκαν για πρώτη φορά από το 1987 και επεκτάθηκαν σε συνδετικό RPQs μέσω λήψης ενώσεων. Σχηματίζουν αυτές οι κατηγορίες ερωτημάτων τη βάση των γλωσσών , που μελετήθηκαν στη θεωρητική βιβλιογραφία. Τα RPQs και οι σχετικές γλώσσες επεκτάθηκαν σε ένα θεωρητικό μοντέλο γραφημάτων δεδομένων που μοιάζουν με γραφήματα ιδιοτήτων, τα οποία επίσης περιέγραψε το GXpath, μια επέκταση του Xpath με δοκιμές κόμβων. Η Sparql είναι μια τυπική γλώσσα που χρησιμοποιείται για την αναζήτηση rdf δεδομένων και προστέθηκε υποστήριξη για το RPQs στο SPARQL. Άλλες γλώσσες ερωτημάτων γραφημάτων βιομηχανικής ιδιοκτησίας όπως το psql της oracle που χρησιμοποιεί τη φόρμα SELECT-FROM-WHERE από την SQL και τη σύνταξη του μοτίβου γράφων από την cypher. Επιτρέπει RPQs μέσω της χρήσης της ρήτρας PATH ,που ορίζει ένα κανονικό μοτίβο διαδρομής. Δεν υποστηρίζει την εισαγωγή ή την ενημέρωση δεδομένων [4].

4. Γράφοι και AgensGraph

4.1 Γράφος

Κατά βάση ένα γράφημα περιλαμβάνει μια ομάδα κορυφών και ακμών. Με πιο προσιτή γλώσσα περιλαμβάνει ένα σύνολο κόμβων μαζί με τις συνδέσεις που τους συνδέουν. Αυτά τα γραφήματα συμβολίζουν οντότητες ως κόμβους και τις αλληλεπιδράσεις τους με τον κόσμο ως συνδέσεις. Αυτή η ευέλικτη και προσαρμόσιμη δομή μας επιτρέπει να προσομοιώσουμε μια ευρεία γκάμα σεναρίων. Αυτά τα σεναρία περιλαμβάνουν οτιδήποτε, από την ανάπτυξη ενός διαστημοπλοίου και τον σχεδιασμό συστημάτων οδικών δικτύων μέχρι την ανίχνευση των αλυσιδωτών αλυσίδων, την εξακρίβωση των προελεύσεων των τροφίμων, την εξέταση των ιατρικών βιογραφικών των πληθυσμών και πολλά άλλα.

Τα γραφήματα αποτελούν ανεκτίμητα εργαλεία για την κατανόηση μιας ευρείας ποικιλίας συνόλων δεδομένων σε διάφορους τομείς όπως η επιστήμη, η διοίκηση του κράτους και το εμπόριο. Αντίθετα από το δομημένο παράδειγμα των σχεσιακών βάσεων δεδομένων, ο πραγματικός κόσμος είναι περίπλοκος και αλληλένδετος, παρουσιάζοντας ταυτόχρονα συνεπή πρότυπα και ιδιοσυγκρασίες. Καθώς εμβαθύνουμε στην έννοια των γραφημάτων, αρχίζουμε να αναγνωρίζουμε την παρουσία τους σε αμέτρητα σεναρία. Για παράδειγμα, η Gartner αναγνωρίζει πέντε διακριτούς τύπους γραφημάτων στον επιχειρηματικό χώρο: κοινωνικά, πρόθεσης, κατανάλωσης, ενδιαφέροντος και κινητά. Η αξιοποίηση της δυναμικής αυτών των γραφημάτων αναγνωρίζεται για την παροχή μιας "βιώσιμα πλεονεκτικής ανταγωνιστικής προτεραιότητας."

Το τελευταίο διάστημα, πληθώρα εγχειρημάτων και προϊόντων έχουν εμφανιστεί για τη διαχείριση, την επεξεργασία και την ανάλυση γραφημάτων. Ο μεγάλος αριθμός αυτών των τεχνολογιών καθιστά δύσκολη την παρακολούθησή τους και των διαφορών τους, ακόμα και για εκείνους που είναι ενεργοί στον τομέα. Αυτή η ενότητα παρέχει ένα πλαίσιο υψηλού επιπέδου για την κατανόηση του αναπτυσσόμενου τοπίου των γραφημάτων.

Από μια ευρεία οπτική γωνία, μπορούμε να διακρίνουμε δύο βασικά τμήματα του πεδίου των γραφημάτων:

1. Τεχνολογίες που χρησιμοποιούνται κυρίως για τη συναλλαγματική διαδικτυακή αποθήκευση γραφημάτων, συχνά προσβάσιμη απευθείας και σε πραγματικό χρόνο από εφαρμογές. Αυτές οι τεχνολογίες αναφέρονται ως βάσεις δεδομένων γραφημάτων και αποτελούν το κύριο θέμα αυτής της διπλωματικής. Παρόμοια με το πλαίσιο των σχεσιακών βάσεων δεδομένων, μπορούν να αντιληφθούν ως το αντίστοιχο των "συνηθισμένων" συναλλαγματικών επεξεργασιών (OLTP) στον κόσμο των σχεσιακών βάσεων δεδομένων.

2. Τεχνολογίες που χρησιμοποιούνται κυρίως για την εκτέλεση της εξέλιξης των γραφημάτων σε ανάλυση εκτός σύνδεσης, συνήθως ως μια σειρά από διαδοχικά βήματα παρτίδας. Αυτές οι τεχνολογίες μπορούν να αναφερθούν ως μηχανές υπολογισμού γραφημάτων. Μπορούν να αντιληφθούν ότι ανήκουν στην ίδια κατηγορία με άλλα εργαλεία για την ανάλυση μεγάλων όγκων δεδομένων, όπως η εξόρυξη δεδομένων και η διαδικτυακή αναλυτική επεξεργασία (OLAP).

Ένα σύστημα για τη διαχείριση βάσεων δεδομένων βασισμένο σε γράφους, γνωστό ως γραφοβάση δεδομένων, είναι ένα διαδικτυακό σύστημα που έχει σχεδιαστεί για τον έλεγχο δεδομένων μέσω των μεθόδων: Δημιουργίας, Ανάγνωσης, Ενημέρωσης και Διαγραφής (CRUD), ταυτόχρονα προσφέροντας έναν προσανατολισμό σε δεδομένα τύπου γράφου. Αυτές οι βάσεις δεδομένων σχεδιάζονται κυρίως για λειτουργία εντός περιβαλλόντων συναλλαγματικών (OLTP), προσφέροντας βελτιστοποίηση για αποδοτική συναλλαγματική απόδοση. Η δομή τους σχεδιάζεται προσεκτικά για τη διασφάλιση της ακεραιότητας των συναλλαγών και τη συνεπή λειτουργική διαθεσιμότητα.

Κατά την εξερεύνηση των τεχνολογιών των γραφοβάσεων δεδομένων, είναι σημαντικό να ληφθούν υπόψη δύο βασικά χαρακτηριστικά:

1. Βασικός Μηχανισμός Αποθήκευσης:

Ορισμένες γραφοβάσεις χρησιμοποιούν μια εξειδικευμένη φυσική αποθήκευση που σχεδιάζεται για την αποτελεσματική διαχείριση και αποθήκευση δομών βάσης γράφου. Ωστόσο, είναι σημαντικό να σημειωθεί ότι όλες οι τεχνολογίες γραφοβάσεων δεδομένων δεν υιοθετούν αυτήν την προσέγγιση. Ορισμένες βάσεις δεδομένων μετατρέπουν τα δεδομένα του γράφου σε σχεσιακή βάση δεδομένων, σε βάση δεδομένων προς αντικείμενο ή άλλο ευέλικτο αποθηκευτικό μέσο.

2. Μηχανισμός Επεξεργασίας:

Ενώ ορισμένοι ορισμοί απαιτούν τη χρήση ανεξάρτητων δεικτών για μια γραφοβάση δεδομένων, η προοπτική μας είναι ελαφρώς πιο ευρεία. Κάθε βάση δεδομένων που παρουσιάζει στους χρήστες μια εμπειρία παρόμοια με αυτήν ενός γραφικού συστήματος (προσφέροντας CRUD λειτουργίες για μια δομή δεδομένων τύπου γράφου) μπορεί να θεωρηθεί αντίστοιχα ως μια βάση δεδομένων γράφου. Παρ' όλα αυτά, αναγνωρίζουμε τα σημαντικά πλεονεκτήματα της ανεξάρτητης ακεραιότητας των δεικτών και τα χρησιμοποιούμε συνεπώς.

Οι σχέσεις είναι πολύ σημαντικές στο μοντέλο δεδομένων γράφου. Αυτό αντίκειται σε άλλα συστήματα διαχείρισης βάσεων δεδομένων, όπου ο εντοπισμός συνδέσεων μεταξύ οντοτήτων απαιτεί συμπεράσματα όπως ξένα κλειδιά ή εκτελέσεις εκτός των κανονικών διαδικασιών, όπως η αντιστοίχιση-μείωση (map-reduce). Συνδυάζοντας τις απλές αφαιρέσεις των κόμβων και των σχέσεων σε συνδεδεμένες δομές, οι γραφοβάσεις δεδομένων μας δίνουν τη δυνατότητα να δημιουργούμε μοντέλα υψηλής πολυπλοκότητας που συνάδουν στενά με την περίπλοκη φύση του συγκεκριμένου πεδίου προβλημάτων. Τα αποτελέσματα αυτών των μοντέλων είναι πιο απλά και πιο ευέλικτα σε σύγκριση με αυτά που δημιουργούνται από παραδοσιακές σχεσιακές βάσεις δεδομένων και άλλες αποθηκευτικές λύσεις NOSQL (Not Only SQL).

Η προσαρμοστικότητα των γραφοβάσεων αναδύεται ως καθοριστικό πλεονέκτημα για προγραμματιστές και αρχιτέκτονες δεδομένων. Αυτή η προσαρμοστικότητα διευκολύνει τη σύνδεση δεδομένων σύμφωνα με τις μοναδικές απαιτήσεις του τομέα, επιτρέποντας στον σχηματισμό δομής και σχήματος να εξελίσσονται ταυτόχρονα με μια εμπλουτισμένη κατανόηση του χώρου προβλημάτων. Αντί να επιβάλλουν ακανόνιστες δομές από την αρχή, οι γραφοβάσεις ανταποκρίνονται απευθείας σε αυτήν την ανάγκη. Εκφράζουν και προσαρμόζονται ευέλικτα στις επιχειρησιακές ανάγκες, επιτρέποντας στην τεχνολογία πληροφορικής να συναρμολογείται με τον ρυθμό των επιχειρησιακών διαδικασιών.

Η από δημιουργίας προσθετική φύση των γραφημάτων αντιπροσωπεύει ένα εμφανές στοιχείο, που επιτρέπει την προσθήκη νέων σχέσεων, κόμβων, ετικετών και υπογραφών σε μια υπάρχουσα δομή χωρίς να επηρεάζονται τα τρέχοντα ερωτήματα ή η λειτουργία των εφαρμογών. Αυτό το χαρακτηριστικό ενισχύει σημαντικά την

παραγωγικότητα των προγραμματιστών και ενισχύει τη διαχείριση των κινδύνων των έργων. Η ενσωματωμένη ευελιξία του μοντέλου γράφου σημαίνει ότι δεν είναι απαραίτητος ο αναλυτικός μοντελοποιητικός σχεδιασμός του τομέα εκ των προτέρων, μια πρακτική που μπορεί να είναι επικίνδυνη αντιμετωπίζοντας τις αλλαγές στις απαιτήσεις της επιχείρησης. Επιπλέον, η προσθετική φύση των γραφημάτων οδηγεί σε λιγότερες μετακινήσεις, μειώνοντας έτσι το υπερβάλλον στη συντήρηση και τους συναφείς κινδύνους.

Όσον αφορά την ευελιξία, το επιθυμητό είναι να εξελίσσουμε το μοντέλο δεδομένων μαζί με το υπόλοιπο της εφαρμογής μας, χρησιμοποιώντας μια τεχνολογία που είναι αρμονισμένη με τις σύγχρονες πρακτικές ενδοσκόπησης και επαναληπτικής παράδοσης λογισμικού. Οι σύγχρονες γραφοβάσεις δεδομένων μας εξοπλίζουν για μεγάλη δυνατότητα ανάπτυξης και ευέλικτη συντήρηση συστημάτων. Η απαλλαγμένη από σχήμα φύση του μοντέλου δεδομένων γράφου, συνδυασμένη με τη δοκιμαστική φύση της διεπαφής εφαρμογής (API) και της γλώσσας ερωτημάτων της γραφοβάσης, ενισχύει τη δυνατότητά μας να εξελίσσουμε μια εφαρμογή με έλεγχο.

Την ίδια στιγμή, αξίζει να σημειώσουμε ότι η απαλλαγμένη από σχήμα φύση των γραφοβάσεων δεδομένων συνεπάγεται μια απόκλιση από τους συμβατικούς μηχανισμούς διακυβέρνησης δεδομένων που βασίζονται σε σχήμα, όπως αυτοί που συναντούμε στον κόσμο των σχέσεων. Ωστόσο, αυτό το χαρακτηριστικό δεν αντιπροσωπεύει έναν κίνδυνο. Αντιθέτως, αυτό καλεί μια πιο διαφανή και ενεργητική μορφή διακυβέρνησης. Αυτό περιλαμβάνει την προγραμματιστική εφαρμογή της διακυβέρνησης μέσω αυστηρού δοκιμαστικού ελέγχου, καθοδηγώντας τον σχηματισμό του μοντέλου δεδομένων, των ερωτημάτων και των επιχειρησιακών κανόνων που εξαρτώνται από το γράφο. Όπως περιγράφεται στο Κεφάλαιο 4, αυτή η πρακτική συμφωνεί αρμονικά με τις σύγχρονες ευέλικτες και δοκιμαστικές μεθοδολογίες ανάπτυξης λογισμικού. Επιτρέπει στις εφαρμογές που βασίζονται σε γραφοβάσεις δεδομένων να εξελίσσονται συμβατά με τις δυναμικές επιχειρησιακές περιβάλλοντος.

Οι γραφοβάσεις δεδομένων προσφέρουν μια σειρά σημαντικών οφελών:

1. Βελτιστοποιημένη Απόδοση: Οι ταχείς χρόνοι απόκρισης σε ερωτήματα είναι ζωτικής σημασίας για τις πλατφόρμες δεδομένων. Οι γραφοβάσεις ξεχωρίζουν στη διατήρηση της γρήγορης απόδοσης ακόμα και καθώς τα σύνολα δεδομένων διευρύνονται.

Χρησιμοποιώντας την *index-free adjacency*, περίπλοκες συνδέσεις μετατρέπονται σε γρήγορες διασταυρώσεις γραφημάτων, διασφαλίζοντας γρήγορη απόδοση ανεξάρτητα από το μέγεθος του συνόλου δεδομένων.

2. Απλοποιημένοι Κύκλοι Ανάπτυξης: Το μοντέλο δεδομένων γράφου αντιμετωπίζει την ειδική πρόκληση της ενσωμάτωσης μεταξύ αντικειμενικών και πινακικών μοντέλων στην ανάπτυξη λογισμικού. Αυτό δεν μειώνει μόνο τις περίπλοκες αναπτυξιακές διαδικασίες, αλλά συνδέει το χάσμα μεταξύ τεχνικών και επιχειρησιακών πεδίων. Τα κοινά μοντέλα ενισχύουν τη συνεργασία μεταξύ ειδικών θεματολογίας, αρχιτεκτόνων και προγραμματιστών, επηρεάζοντας άμεσα την ένταξή τους στην εφαρμογή.

3. Προσαρμοστική Ευελιξία στις Επιχειρηματικές Ανταποκρίσεις: Δυναμικές εφαρμογές υφίστανται αλλαγές λόγω εξελισσόμενων επιχειρηματικών συνθηκών, συμπεριφοράς των χρηστών και τεχνικών περιβαλλόντων. Οι παραδοσιακές μέθοδοι περιλαμβάνουν πολύπλοκες μεταφορές δεδομένων με τροποποιήσεις σχημάτων και μετασχηματισμούς δεδομένων. Ωστόσο, οι γραφοβάσεις δεδομένων, με την απαλλαγμένη από σχήμα ουσία τους και τις ευέλικτες δυνατότητες σχέσεων δεδομένων, εξελίσσονται παράλληλα με την εξέλιξη της επιχείρησης, μειώνοντας τον κίνδυνο και επιταχύνοντας τον χρόνο έως την αγορά.

4. Εφαρμογή σε Επιχειρήσεις: Κρίσιμες επιχειρηματικές εφαρμογές απαιτούν ανθεκτικότητα, επεκτασιμότητα και δυνατότητες συναλλαγών. Αν και κάποιες γραφοβάσεις δεδομένων είναι σχετικά νέες και λιγότερο ώριμες, συγκεκριμένες επιλογές στην αγορά προσφέρουν βασικά χαρακτηριστικά όπως συναλλαγές ACID (Ατομικότητα, Συνέπεια, Απομόνωση, Διάρκεια), υψηλή διαθεσιμότητα, οριζόντια επεκτασιμότητα ανάγνωσης και υποστήριξη για την αποθήκευση μεγάλου όγκου δεδομένων. Αυτά τα χαρακτηριστικά, σε συνδυασμό με την απόδοση και την ευελιξία, έχουν ωθήσει την υιοθέτηση γραφοβάσεων δεδομένων στις επιχειρήσεις, μετασχηματίζοντας τον τρόπο λειτουργίας των επιχειρησιακών μεθοδολογιών [5].

Τα συστήματα διαχείρισης γράφων είναι ουσιαστικές υποδομές για εφαρμογές που βασίζονται σε γράφους AI και ML. Αυτά τα συστήματα έχουν κερδίσει σημαντική προσοχή τόσο από ακαδημαϊκούς όσο και από βιομηχανικούς κύκλους. Παίζουν έναν

κρίσιμο ρόλο στην αποτελεσματική διαχείριση των δεδομένων των γράφων, καλύπτοντας τόσο τις άμεσες όσο και τις έμμεσες πληροφορίες. Το τοπίο περιλαμβάνει δύο κύριες κατηγορίες: συστήματα αναλυτικών γράφων, σχεδιασμένα για διττά υπολογισμούς σε ευρείες συστάδες (π.χ., GraphLab, Giraph, GraphX), και βάσεις δεδομένων γράφων, σχεδιασμένες για την αποθήκευση και τη γρήγορη ανάκτηση δεδομένων γράφων (π.χ., Neo4j, JanusGraph, ArangoDB). Οι τελευταίες επιλογές βασίζονται είτε στο μοντέλο ετικετοποιημένων ιδιοτήτων είτε στο πλαίσιο περιγραφής πόρων (RDF). Ενώ οι βάσεις δεδομένων RDF και οι αντίστοιχες βάσεις δεδομένων ιδιοτήτων όπως η Neo4j έχουν απολαύσει σημαντική αναγνώριση, ο βασικός προσανατολισμός του άρθρου αυτού είναι η αξιολόγηση των βάσεων δεδομένων γράφων που υποστηρίζουν το ετικετοποιημένο μοντέλο γράφων.

Πολλές βάσεις δεδομένων που επικεντρώνονται στους γράφους έχουν χρησιμοποιηθεί σε εταιρικά περιβάλλοντα για να διαχειριστούν αποτελεσματικά τα γρήγορα εξελισσόμενα δεδομένα δομών που χαρακτηρίζουν τα πεδία της τεχνητής νοημοσύνης. Παρότι μερικές δημοφιλείς βάσεις δεδομένων γράφων έχουν ήδη αποκτήσει μεγάλο μερίδιο στην αγορά και έχουν μελετηθεί ευρέως στις ακαδημαϊκές κοινότητες, τα τελευταία χρόνια μερικά νέα συστήματα βάσεων δεδομένων γράφων, όπως το AgensGraph, το TigerGraph και το LightGraph, έχουν αναπτυχθεί για να ικανοποιήσουν τις ανάγκες των υψηλού επιπέδου επιχειρηματικών απαιτήσεων, να μειώσουν την κατανάλωση πόρων και να βελτιώσουν την απόδοση. Ενθαρρυμένο από αυτές τις τάσεις πραγματοποιεί μια εμπειρική έρευνα στις σημαντικές βάσεις δεδομένων γράφων σε επιχειρησιακό επίπεδο. Λαμβάνει υπόψη τη Neo4j, μια ευρέως χρησιμοποιούμενη υπάρχουσα λύση, καθώς και νεότερες εναλλακτικές, όπως το AgensGraph, το TigerGraph και το LightGraph.

Η AgensGraph εμφανίζεται ως μια καινοτόμα βάση δεδομένων γράφων πολλαπλών μοντέλων της επόμενης γενιάς. Προσφέρει ομαλή προσαρμογή σε διάφορα μοντέλα δεδομένων, όπως γράφοι, σχεσιακά, έγγραφα και κλειδί-τιμή, ενώ χρησιμοποιεί τόσο την SQL όσο και τη Cypher για ερωτήματα. Αυτός ο μοναδικός συνδυασμός επιτρέπει τη σύγκλιση αυτών των γλωσσών σε ένα μόνο ερώτημα, καλύπτοντας ένα ευρύ φάσμα περιπτώσεων και αποδεικνύοντας ιδιαίτερα προσιτός για έμπειρους χρήστες SQL. Η AgensGraph εκμεταλλεύεται τη μηχανή αποθήκευσης PostgreSQL RDMS, αν και η διαχείριση εκτεταμένων δεδομένων μπορεί να οδηγήσει σε παρατεταμένους χρόνους φόρτωσης και εντατική κατανάλωση μνήμης. Σημαντικό είναι επίσης να αναφέρουμε ότι

σύνθετα ερωτήματα μπορεί να έχουν ως αποτέλεσμα την κατάληψη σημαντικού προσωρινού χώρου. Είναι σημαντικό να γνωρίζουμε ότι η AgensGraph παραμένει υπό ανάπτυξη και ενδέχεται να μην υποστηρίζει εξαντλητικά όλες τις γραμματικές της Cypher. Εντός αυτού του περιορισμού, οι συντάκτες αυτής της έρευνας αναλαμβάνουν τη δρομολόγηση της υλοποίησης περίπλοκων ερωτημάτων με τον επιδέξιο συνδυασμό της Cypher και της SQL, ειδικά στο πλαίσιο του περίπλοκου περιβάλλοντος του LDDB (Linked Data Benchmark Council Social Network Benchmark). [10]

4.2 AgensGraph

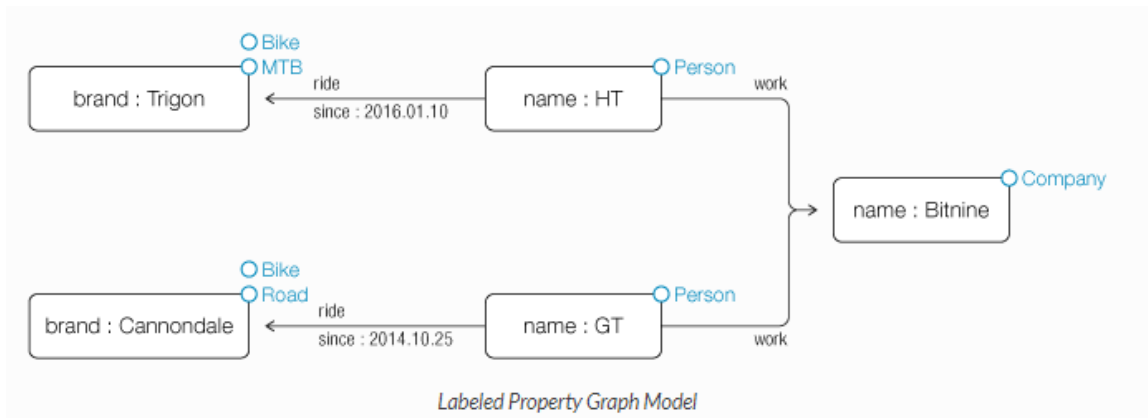
Η AgensGraph εμφανίζεται ως μια προηγμένη, ευέλικτη πολυμορφική βάση δεδομένων γράφων, σχεδιασμένη για σύγχρονα περίπλοκα περιβάλλοντα δεδομένων. Λειτουργώντας ως πολυμορφική βάση δεδομένων που ενσωματώνει το PostgreSQL RDBMS, αντιμετωπίζει άνετα τα μοντέλα δεδομένων τόσο σχεσιακά όσο και γραφημάτων με ευμετάβλητη συνύπαρξη. Αυτή η ενδυναμωμένη προσέγγιση επιτρέπει στους προγραμματιστές να συνδυάσουν τις παραδοσιακές σχεσιακές δομές με τα ευέλικτα μοντέλα γράφων εντός ενός μοναδικού συστήματος βάσης δεδομένων. Η AgensGraph εξαρτάται από την Ansi-SQL και το Open Cypher ως γλώσσες ερωτημάτων, δυνατότητα που επιτρέπει την ομαλή ένταξη των ερωτημάτων SQL και Cypher σε μία ενιαία εμπειρία ερωτημάτων.

Γνωστό για την ανθεκτικότητά του, τον πλούσιο σύνολο χαρακτηριστικών και την καταλληλότητα για τις επιχειρήσεις, το AgensGraph έχει σχεδιαστεί με προσοχή για την αντιμετώπιση πολύπλοκων συνδεδεμένων γραφικών δεδομένων. Η οπλοστασιοθήκη του περιλαμβάνει βασικά χαρακτηριστικά της βάσης δεδομένων, όπως συναλλαγές ACID, έλεγχο πολλαπλών εκδόσεων, αποθήκευση διαδικασιών, εκτελεστές, περιορισμοί, εξελεγμένη παρακολούθηση και ευέλικτο μοντέλο δεδομένων με υποστήριξη JSON. Επιπλέον, η AgensGraph αξιοποιεί το πλούσιο οικοσύστημα του PostgreSQL και μπορεί να επεκταθεί με εξωτερικά πρόσθετα όπως το PostGIS για τη διεύρυνση των δυνατοτήτων του.

Μερικά από τα Χαρακτηριστικά:

- Πολυμορφική υποστήριξη του μοντέλου δεδομένων γράφου ιδιοτήτων, του σχεσιακού μοντέλου δεδομένων και των εγγράφων JSON.
- Υποστήριξη της γλώσσας ερωτημάτων Cypher.
- Ολοκληρωμένο ερώτημα χρήσης SQL και Cypher.
- Διαχείριση αντικειμένων δεδομένων γράφου.
- Ιεραρχική οργάνωση ετικετών γράφου.
- Δείκτες ιδιοτήτων σε κόμβους και ακμές.
- Περιορισμοί: μοναδικότητας, υποχρεωτικότητας και ελέγχου.
- Συναλλαγές ACID.
- Συνδεσιμότητα με Hadoop.
- Ενεργό-ανενεργή αντιγραφή.

Η AgensGraph είναι μια πολυμορφική βάση δεδομένων που υποστηρίζει ταυτόχρονα τόσο το μοντέλο δεδομένων γράφου ιδιοτήτων όσο και το σχεσιακό μοντέλο.



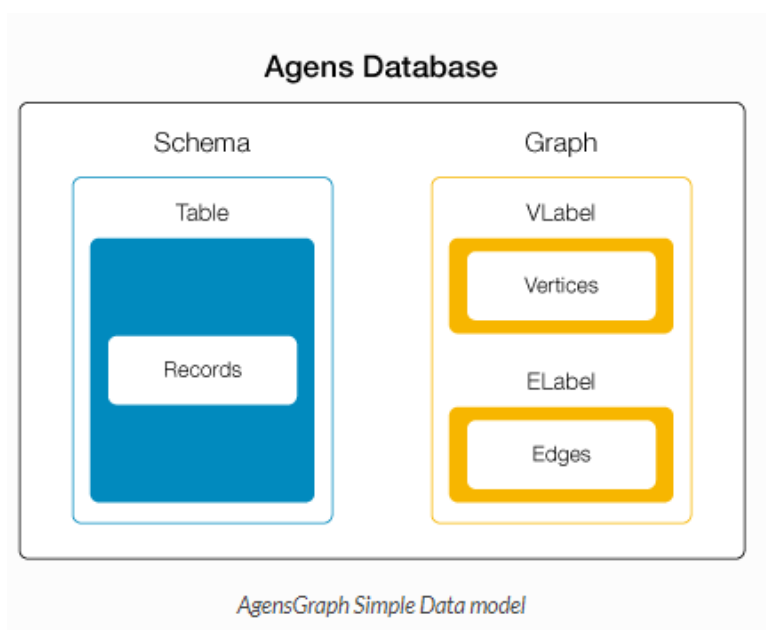
Εικόνα 2 source (https://bitmine.net/documentations/manual/quick_guide/agens_graph_quick_guide_html.html?ckattempt=1)

Το μοντέλο δεδομένων γράφου ιδιοτήτων περιλαμβάνει συνδεδεμένες οντότητες, οι οποίες μπορούν να έχουν οποιονδήποτε αριθμό γνωρισμάτων. Στην AgensGraph, μια οντότητα είναι γνωστή ως κόμβος. Οι κόμβοι μπορούν να έχουν αυθαίρετο αριθμό γνωρισμάτων και μπορούν να ταξινομούνται με ετικέτες. Οι ετικέτες χρησιμοποιούνται

για να ομαδοποιούν τους κόμβους προκειμένου να αναπαραστήσουν κάποιες κατηγορίες κόμβων, όπως για παράδειγμα τον ρόλο ενός ατόμου.

Οι ακμές είναι κατευθυνόμενες συνδέσεις μεταξύ δύο κόμβων. Οι ακμές μπορούν επίσης να έχουν γνωρίσματα και κατηγορικές ετικέτες όπως οι κόμβοι. Στην AgensGraph, μια ακμή πάντα έχει έναν αρχικό κόμβο και έναν τελικό κόμβο. Εάν ένα ερώτημα προσπαθήσει να διαγράψει έναν κόμβο, πρέπει πρώτα να διαγράψει όλες τις ακμές του. Στην AgensGraph, δεν μπορούν να υπάρχουν κατεστραμμένες ακμές.

Τα γνωρίσματα των ακμών και των κόμβων αναπαρίστανται στη μορφή JSON. Το JSON είναι ένα κείμενο μορφότυπο για τη σειριοποίηση ημι-δομημένων δεδομένων. Τα JSON αποτελούνται από έξι τύπους δεδομένων: συμβολοσειρές, αριθμούς, λογικές τιμές, κενό, αντικείμενα και πίνακες. Τα αντικείμενα AgensGraph αξιοποιούν πλήρως το μορφότυπο JSON αποθηκεύοντας πληροφορίες ως άτακτες συλλογές μηδέν ή περισσότερων ζευγών ονοματισμένης/τιμής. Ένα όνομα είναι μια συμβολοσειρά και μια τιμή μπορεί να είναι οποιοδήποτε προαναφερθέντος τύπου, συμπεριλαμβανομένων των ενσωματωμένων τύπων JSON. Η AgensGraph χρησιμοποιεί ειδικά το μορφότυπο JSONB. Διότι το JSONB είναι ένα αποσυνθεμένο δυαδικό μορφότυπο, επεξεργάζεται πολύ πιο γρήγορα από το κανονικό JSON, αλλά με το κόστος ενός ελαφρώς πιο αργού χρόνου εισόδου.



Εικόνα 3 source (https://bitnine.net/documentations/manual/quick_guide/agens_graph_quick_guide_html.html?ckattempt=1)

Εντός του πεδίου της AgensGraph, είναι εφικτή η δημιουργία πολλών βάσεων δεδομένων, καθεμία από τις οποίες περιλαμβάνει τη δυνατότητα για πολλαπλά σχήματα και γράφους. Αυτά τα σχήματα αντιστοιχούν σε σχέσεις πινάκων, ενώ τα αντικείμενα γράφου αποτυπώνουν τα δεδομένα του γράφου. Είναι αξιοσημείωτο ότι η διάκριση μεταξύ των ονομάτων των σχημάτων και των γράφων είναι ζωτικής σημασίας. Τα στοιχεία που αναφέρονται ως κόμβοι και ακμές ομαδοποιούνται εντός ετικετών, διαιρούμενες σε δύο κατηγορίες: ετικέτες κόμβων και ετικέτες ακμών. Σημαντικό είναι ότι παρά την παρουσία πολλαπλών γράφων σε μία βάση δεδομένων, μόνο ένας γράφος μπορεί να είναι ενεργός ταυτόχρονα.

Οι έννοιες των ετικετών αποτελούν βασικό στοιχείο, διευκολύνοντας την ομαδοποίηση κόμβων και ακμών. Αυτή η πλατφόρμα επιτρέπει στους χρήστες να δημιουργούν δείκτες ιδιοτήτων για όλους τους κόμβους υπό μια συγκεκριμένη ετικέτα. Η λειτουργικότητα αυτή επεκτείνεται στη δυνατότητα ελέγχου πρόσβασης για διάφορους τύπους χρηστών, ενώ οι ιεραρχίες ετικετών μπορούν να δημιουργηθούν προσδίδοντας κληρονομιά στις ετικέτες. Σημειώνεται πως υπάρχει μια προκαθορισμένη ετικέτα με το όνομα "ag_vertex" για τους κόμβους· εκεί αποθηκεύονται οι κόμβοι που δεν τους έχει καθοριστεί ειδική ετικέτα. Αντίθετα, κάθε ακμή συνοδεύεται πάντα από μια μοναδική ετικέτα.

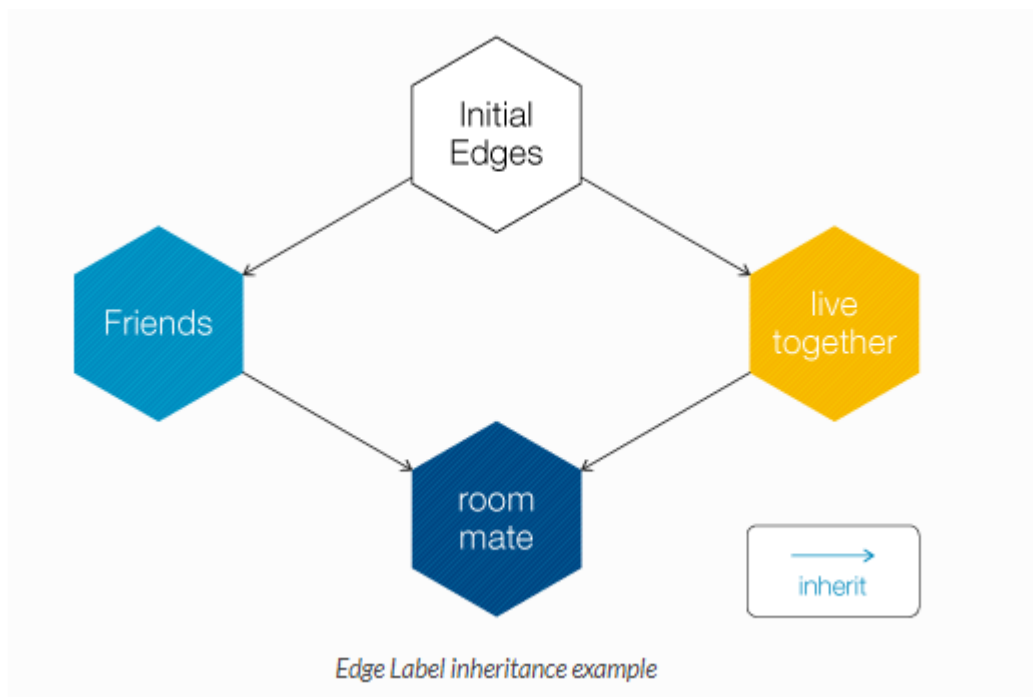
Οι ονομασίες που χρησιμοποιούνται για τις ετικέτες κόμβων και τις ετικέτες ακμών είναι VLABEL και ELABEL, αντίστοιχα:

VLABEL: Αυτός ο όρος σημαίνει την ετικέτα του κόμβου, που κατηγοριοποιεί τους κόμβους βάσει του ρόλου τους.

Κόμβος: Αντιπροσωπεύει τις οντότητες με τη δυνατότητα να φιλοξενούν χαρακτηριστικά.

ELABEL: Αντίθετα, ο όρος ELABEL αναφέρεται στην ετικέτα της ακμής, κατηγοριοποιώντας τις ακμές με βάση τους ρόλους τους.

Ακμή: Υποδεικνύει τις σχέσεις που συνδέουν διάφορες οντότητες.



Εικόνα 4 source (https://bitmine.net/documentations/manual/quick_guide/agens_graph_quick_guide_html.html?ckattempt=1)

Κάθε ετικέτα κληρονομεί μία ή περισσότερες ετικέτες. Ο παραπάνω πίνακας παρουσιάζει ένα παράδειγμα ιεραρχίας ετικετών ακμών. Η ιεραρχία των ετικετών είναι παρόμοια με την ιεραρχία κλάσεων στον αντικειμενοστραφή προγραμματισμό. Κάθε γονική ετικέτα περιλαμβάνει δεδομένα παιδικών ετικετών. Για παράδειγμα, δεδομένης της παραπάνω ιεραρχίας, αν μια ερώτηση ταιριάζει με την ακμή 'φίλοι', τα αποτελέσματα περιλαμβάνουν τα δεδομένα της ετικέτας 'συγκάτοικος' [6].

4.3 Deployment

Για την λειτουργία της AgensGraph απαιτούνται τουλάχιστον 4GB μνήμης RAM και 2.5GB χώρου δίσκου. Μπορεί να εκτελεστεί σε οποιαδήποτε γενική πλατφόρμα συμβατή με Unix, αλλά οι επίσημα πιστοποιημένες πλατφόρμες είναι η σειρά Linux (Centos, Ubuntu, RHEL) και η σειρά Windows (Windows Server 2008 64bit, Windows Server 2012 64bit, Windows 7 64bit).

Η λειτουργία της AgensGraph βασίζεται σε συγκεκριμένες διαμοιρασμένες μνήμες και ρυθμίσεις σημαφόρων για την εξασφάλιση ομαλής λειτουργίας. Κάθε αντίγραφο του διακομιστή απαιτεί έναν συγκεκριμένο όγκο κοινόχρηστης μνήμης

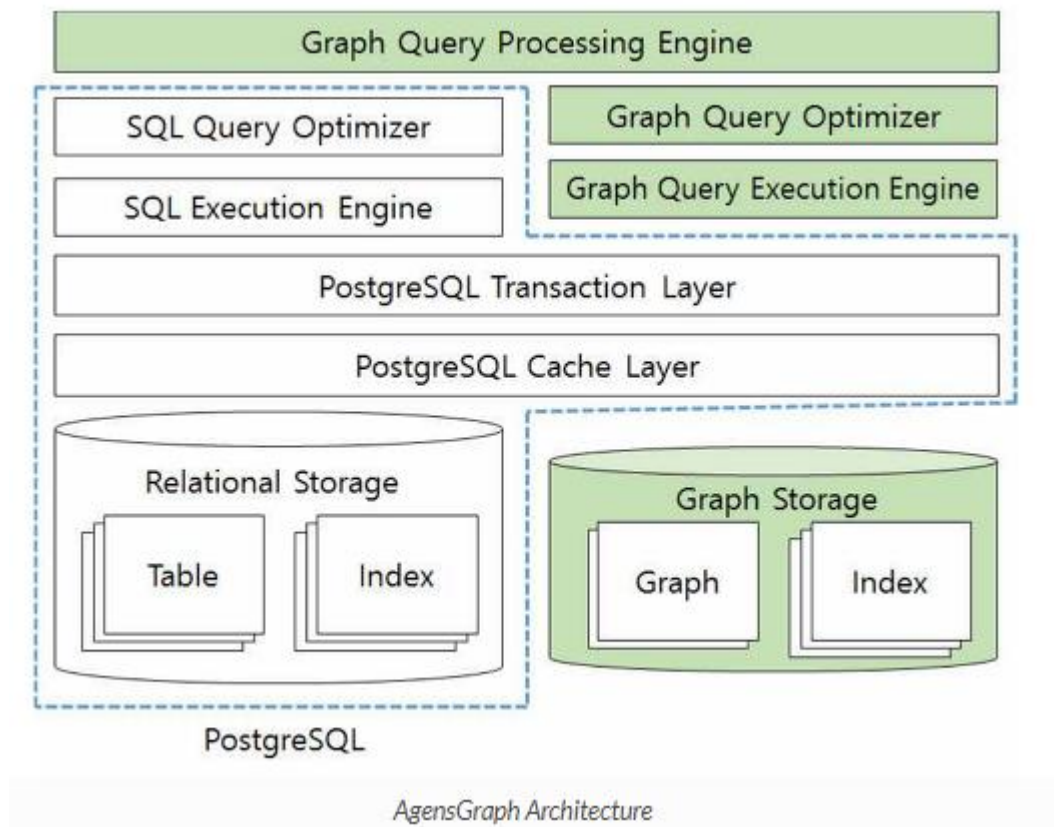
System V, συνήθως περίπου 48 bytes για πλατφόρμες 64-bit. Αυτή η εκχώρηση είναι γενικά κατάλληλη για σύγχρονα λειτουργικά συστήματα. Ωστόσο, καταστάσεις με πολλαπλά αντίγραφα του διακομιστή ή την συνταυτόχρονη χρήση κοινόχρηστης μνήμης System V από άλλες εφαρμογές ενδέχεται να απαιτούν προσαρμογές σε παραμέτρους όπως οι SHMMAX και SHMALL. Αξιοσημείωτο είναι ότι η SHMALL λειτουργεί με βάση της ευρείας σελίδας και όχι με βάση τον αριθμό των bytes σε περισσότερα συστήματα.

Σχετικά με την χρήση κοινόχρηστης μνήμης από το AgensGraph, γνωστή ως SHMMIN, το ελάχιστο μέγεθος είναι περίπου 32 bytes, το οποίο είναι απίθανο να προκαλέσει προβλήματα. Η χρήση σημαφόρων συνδέεται στενά με συνδέσεις και διεργασίες αυτοκάθαρσης. Το AgensGraph αναθέτει ένα σημαφόρο ανά σύνδεση και διεργασία αυτοκάθαρσης, οργανωμένο εντός συνόλων σημαφόρων. Το συνολικό πλήθος των σημαφόρων καθορίζεται από το SEMMNS, ενώ το SEMMNI περιορίζει τα ταυτόχρονα σύνολα σημαφόρων στο σύστημα.

Ενδέχεται να είναι απαραίτητο να βελτιστοποιηθεί η SEMMAP, προσεγγίζοντας την τιμή παρόμοια με το SEMMNS. Το SEMMAP καθορίζει το μέγεθος του χάρτη πόρων σημαφόρων, που διαχειρίζεται τις εγγραφές για τα παρακείμενα τμήματα σημαφόρων. Η σωστή διαχείριση αυτών των παραμέτρων εξασφαλίζει την διαθεσιμότητα πόρων και συντελεί στην αποτελεσματική λειτουργία του συστήματος.

4.4 Αρχιτεκτονική

Η AgensGraph είναι μια πολυμορφική βάση δεδομένων που αναπτύχθηκε με βάση το PostgreSQL. Η AgensGraph σχεδιάστηκε για να προσφέρει όλες τις λειτουργίες που παρέχει το PostgreSQL και να επεξεργάζεται ερωτήματα γράφων χρησιμοποιώντας τη γλώσσα Cypher. Εσωτερικά, διακρίνονται κυρίως στον επεξεργαστή του PostgreSQL και τον επεξεργαστή των γράφων, οι οποίοι χρησιμοποιούν κοινά τμήματα προσωρινής αποθήκευσης και συναλλαγών. Οι χρήστες μπορούν να επωφεληθούν από καλύτερη απόδοση και πιο βολική υλοποίηση εκτελώντας ταυτόχρονα ερωτήματα SQL και Cypher με μία μόνο ερώτηση.



Εικόνα 5 source (https://bitmine.net/documentations/manual/operation/english/agens_graph_operation_manual_html.html)

Η AgensGraph χρησιμοποιεί ένα μοντέλο πελάτη/εξυπηρετητή παρόμοιο με αυτό της PostgreSQL. Εντός αυτής της δομής, μια συνεδρία περιλαμβάνει πολλαπλές αλληλεπιδρούσες διεργασίες. Η διεργασία του εξυπηρετητή, με όνομα "postgres," φέρει την ευθύνη της διαχείρισης των αρχείων της βάσης δεδομένων, χειρίζεται τα αιτήματα σύνδεσης από τους πελάτες και διευκολύνει τη χρήση της βάσης δεδομένων από τους πελάτες. Στην πλευρά του χρήστη, οι διεργασίες των πελατών αναφέρονται σε διάφορες εφαρμογές, από προγράμματα βασισμένα σε κείμενο έως γραφικές διεπαφές ή ιστοσελίδες που προσπελούνται μέσω διακομιστή ιστού. Αυτές οι πελατικές εφαρμογές μπορούν να αναπτυχθούν από τους χρήστες ή να περιλαμβάνονται σε πακέτα διανομής.

Η AgensGraph προσαρμόζεται για την περίπτωση που ο πελάτης και ο εξυπηρετητής βρίσκονται σε διαφορετικούς φιλοξενητές εντός ενός περιβάλλοντος πελάτη/εξυπηρετητή. Η επικοινωνία μεταξύ αυτών των συστατικών συμβαίνει κυρίως μέσω της υποδομής δικτύου TCP/IP. Σημαντικό είναι να σημειώσουμε ότι, όταν ο πελάτης και ο εξυπηρετητής διαφέρουν, ο πελάτης δεν μπορεί να έχει άμεση πρόσβαση

σε ένα αρχείο βάσης δεδομένων στον φιλοξενητή του εξυπηρετητή. Αντίθετα, η πρόσβαση του πελάτη αφορά αρχεία που βρίσκονται στον δικό του φιλοξενητή.

Για τη δυνατότητα προσπέλασης από τους χρήστες, ο εξυπηρετητής του AgensGraph δημιουργεί μια νέα διεργασία για κάθε σύνδεση. Αυτό σημαίνει ότι η αλληλεπίδραση μεταξύ πελάτη και νεοσύστατης διεργασίας εξυπηρετητή συμβαίνει ανεξάρτητα από την κύρια διεργασία "postgres". Ουσιαστικά, η "postgres" λειτουργεί διαρκώς στον εξυπηρετητή για τη διαχείριση αιτημάτων σύνδεσης των πελατών και τη δημιουργία νέων διεργασιών νεοσύστατου εξυπηρετητή.

4.5 Ασφάλεια

Η επαλήθευση του πελάτη και η διαχείριση των ρόλων της βάσης δεδομένων είναι ζωτικής σημασίας για τη διατήρηση της ασφάλειας στο AgensGraph. Εμπνευσμένο από την τεχνική τεκμηρίωση του PostgreSQL, ειδικά όταν παρουσιάζονται χαρακτηριστικά στο πλαίσιο της Σχεσιακής Βάσης Δεδομένων (RDB) παρά την Σχεσιακή Βάση Δεδομένων Γράφων (GDB), το AgensGraph λαμβάνει αυτά τα ζητήματα σοβαρά υπόψη.

Καθώς οι εφαρμογές πελατών συνδέονται με το διακομιστή βάσης δεδομένων, ο προσεκτικός έλεγχος των επιτρεπόμενων χρηστών βάσης, που επιτρέπονται, είναι αναγκαία. Η επαλήθευση περιλαμβάνει τη δημιουργία αναγνωριστικών πελατών από το διακομιστή βάσης δεδομένων και τον προσδιορισμό εάν θα επιτραπούν συνδέσεις από εφαρμογές πελατών που χρησιμοποιούν τα ονόματα των χρηστών βάσης (ή χρήστες που εκτελούν τις εφαρμογές πελατών).

Η AgensGraph προσφέρει διάφορες μεθόδους επαλήθευσης πελατών, επιτρέποντας την επιλογή βάσει παραμέτρων όπως η διεύθυνση του πελάτη, η βάση δεδομένων και/ή ο χρήστης. Σημαντικό είναι ότι τα ονόματα των χρηστών της βάσης δεδομένων AgensGraph διαφέρουν λογικά από τα ονόματα των χρηστών του λειτουργικού συστήματος στο οποίο εκτελείται ο διακομιστής. Αν και είναι λογικό να συσχετίσουμε τα ονόματα των χρηστών της βάσης δεδομένων με τα ονόματα των χρηστών του λειτουργικού συστήματος, οι διακομιστές που είναι προσβάσιμοι απομακρυσμένα ενδέχεται να διαθέτουν πολλούς χρήστες βάσης δεδομένων χωρίς

αντίστοιχους τοπικούς λογαριασμούς λειτουργικού συστήματος, καθιστώντας τη συσχέτιση περιττή.

Η επαλήθευση του πελάτη υπακούει στο αρχείο διαμόρφωσης `pg_hba.conf`, το οποίο αποθηκεύεται στον κατάλογο δεδομένων του σμήνους βάσης δεδομένων. Η μορφή του `pg_hba.conf` είναι ένα σύνολο εγγραφών, όπου κάθε γραμμή είναι μια εγγραφή. Οι κενές γραμμές αγνοούνται και ο κείμενος που ακολουθεί μετά από το `"#"` αγνοείται επίσης. Κάθε εγγραφή, που δεν μπορεί να συνεχιστεί μετά από αλλαγή γραμμής, αποτελείται από αρκετά πεδία με διαχωριστικά όπως τα κενά και/ή τα `tab`. Χρησιμοποιώντας διπλά εισαγωγικά για τιμές πεδίων, μπορείτε να συμπεριλάβετε κενά στο πεδίο. Εάν χρησιμοποιείτε εισαγωγικά για λέξεις-κλειδιά της βάσης δεδομένων ή των χρηστών/διευθύνσεων στα πεδία της βάσης δεδομένων ή των χρηστών, τα λόγια θα χάσουν την ειδική τους σημασία.

Κάθε εγγραφή καθορίζει τον τύπο σύνδεσης που θα χρησιμοποιηθεί για συνδέσεις που ταιριάζουν με αυτές τις παραμέτρους, το εύρος διεύθυνσης IP του πελάτη (εάν είναι σχετικό), το όνομα της βάσης δεδομένων, το όνομα του χρήστη και τη μέθοδο επαλήθευσης. Η πρώτη εγγραφή που ταιριάζει με τον τύπο σύνδεσης, τη διεύθυνση του πελάτη, τη ζητηθείσα βάση δεδομένων και το όνομα του χρήστη χρησιμοποιείται κατά την επαλήθευση, χωρίς εναλλακτικές επιλογές ή αντικαταστάσεις.

4.6 Πιστοποίηση με Κωδικό Πρόσβασης

Στο εσωτερικό της AgensGraph, υπάρχουν τρεις διακριτοί μέθοδοι πιστοποίησης βασισμένες σε κωδικό πρόσβασης: `md5`, `scram-sha-256` και `password`. Αυτές οι προσεγγίσεις λειτουργούν με παρόμοιο τρόπο, με τη διαφορά ότι οι κωδικοί πρόσβασης μεταδίδονται είτε ως κατακερματισμένα MD5 είτε ως απλό κείμενο, ανάλογα με την επιλεγμένη μέθοδο.

Σε αυτό το πλαίσιο, το `scram-sha-256` εμφανίζεται ως η πιο αξιόπιστη επιλογή, ενώ το `md5` έχει ιστορικά ευρύτερη χρήση. Συνιστάται να επιλέξετε το `scram-sha-256` ή το `md5` για να ενισχύσετε την προστασία από πιθανές επιθέσεις "κατακράτησης"

κωδικών πρόσβασης, δίνοντας προτεραιότητα σε αυτές έναντι της κανονικής χρήσης κωδικών πρόσβασης. Εντούτοις, είναι σημαντικό να σημειωθεί ότι το md5 δεν μπορεί να χρησιμοποιηθεί σε συνδυασμό με το χαρακτηριστικό db_user_namespace. Ωστόσο, η χρήση ενός κωδικού παραμένει ασφαλής όταν η σύνδεση προστατεύεται με κρυπτογράφηση SSL.

Είναι απαραίτητο να διαφοροποιηθούν οι κωδικούς πρόσβασης των χρηστών της βάσης δεδομένων AgensGraph από τους κωδικούς πρόσβασης των χρηστών του λειτουργικού συστήματος. Οι κωδικοί πρόσβασης για κάθε χρήστη της βάσης δεδομένων αποθηκεύονται με ασφάλεια στον κατάλογο συστήματος pg_authid. Αυτοί οι κωδικοί πρόσβασης μπορούν να διαχειριστούν μέσω εντολών SQL, όπως το CREATE USER και το ALTER ROLE (παραδείγματος χάριν, CREATE USER foo WITH PASSWORD 'secret'). Σε περιπτώσεις όπου δεν έχει οριστεί κωδικός πρόσβασης για έναν χρήστη, ο αποθηκευμένος κωδικός πρόσβασης παραμένει κενός, με αποτέλεσμα η πιστοποίηση μέσω κωδικού πρόσβασης να αποτυγχάνει αυτόματα για τον συγκεκριμένο χρήστη. [8]

5. Σύγκριση

Το πείραμα που ακολουθεί έγινε από τους Wang et al [10]. Χρησιμοποιώντας τον δημιουργό δεδομένων LDBC, έχουμε τρία σύνολα δεδομένων με διαφορετικές κλίμακες. Όλα τα σύνολα δεδομένων έχουν την ίδια δομή. Οι στατιστικές των συνόλων δεδομένων περιλαμβάνονται στον πίνακα. Μπορούμε να διαπιστώσουμε ότι ο αριθμός των κορυφών και ακμών και το ακατέργαστο μέγεθος των συνόλων δεδομένων αυξάνονται σχεδόν γραμμικά με τον παράγοντα κλίμακας.

Στο πλαίσιο του LDBC SNB benchmark, ένα εκτεταμένο σύνολο από 54 ερωτήματα καλύπτει τόσο τα διαδραστικά όσο και τα ερωτήματα ευφυούς επιχειρησιακής λογικής. Οι υλοποιήσεις αυτών των ερωτημάτων σε τέσσερις διακριτές βάσεις δεδομένων γράφων προέρχονται είτε από επίσημες πηγές είτε αναπτύχθηκαν προσεκτικά από εμάς, όπως οι ερωτήσεις BI στην AgensGraph. Ένα υποκείμενο δεσμευτικός όρος για την βέλτιστη απόδοση των ερωτημάτων διαμορφώνει αυτήν την προσπάθεια, με στόχο τη μεγιστοποίηση της απόδοσης για κάθε βάση δεδομένων.

Όλες οι πειραματικές διεξαγωγές πραγματοποιούνται σε έναν ελεγχόμενο χώρο, με διπλούς επεξεργαστές Intel Xeon E5-2680 v2 2.80 GHz, 96 GB RAM και 960G NVMe SSD. Το περιβάλλον λειτουργίας είναι το Ubuntu 16.04.5. Η αξιολόγηση της απόδοσης χρησιμοποιεί τις γρήγορες δυνατότητες αποθήκευσης σε SSD NVMe, προεπιλεγμένη επιλογή για τα συστήματα βάσεων δεδομένων. Στο πλαίσιο της μεθοδολογίας, κάθε μικρό ερώτημα (IU και IS) επαναλαμβάνεται εκατό φορές, ενώ τα αντίστοιχα μεγάλα ερωτήματα (IC και BI) υποβάλλονται σε τριπλή επανάληψη. Ο μέσος χρόνος εκτέλεσης αποτελεί τη μετρική του αποτελέσματος. Επίσης, είναι σημαντικό να σημειώσουμε ότι ο χρόνος προεπιλεγμένης λήξης για την επεξεργασία των ερωτημάτων ορίζεται ως 1 ώρα. Το αποτέλεσμα αυτού αναπαρίσταται μέσω των συμβόλων "TO" και "OOM", που υποδεικνύουν τα περιστατικά λήξης χρονικού ορίου και έλλειψης μνήμης αντίστοιχα.

5.1 Εισαγωγή Δεδομένων

Λόγω του μεγάλου μεγέθους των ακατέργαστων δεδομένων, τα σύνολα δεδομένων εισήχθησαν μαζικά στα συστήματα βάσεων δεδομένων γράφων. Η εισαγωγή δεδομένων είναι επίσης μια συνηθισμένη διαδικασία στην παραγωγή. Η Neo4j εμφανίζει τον ταχύτερο χρόνο εισαγωγής δεδομένων, ακολουθούμενο από την AgensGraph. Ενδιαφέρον είναι ότι, ακόμα και αφού λάβουμε υπόψη τον χρόνο που απαιτείται για τη

δημιουργία δεικτών στην το συστήματα καταφέρνει να ξεπεράσει την AgensGraph όσον αφορά την αποδοτικότητα της μαζικής εισαγωγής δεδομένων. Τρομερή είναι η απόδοση της AgensGraph, η οποία περιορισμένη από την εξάρτησή της από την εξωτερική μηχανή αποθήκευσης PostgreSQL, αντιμετωπίζει σημαντικά πιο αργές ταχύτητες μαζικής εισαγωγής δεδομένων σε σχέση με τους συνοδοιπόρους του. Για το σετ δεδομένων DG100, η διαδικασία φόρτωσης δεδομένων της AgensGraph μπορεί να απαιτήσει σχεδόν μία ημέρα, μια μη λογική διάρκεια για συνηθισμένη χρήση.

Εστιάζοντας στα ακατέργαστα και φορτωμένα μεγέθη αποθήκευσης για τα τρία σύνολα δεδομένων στις δυο βάσεις δεδομένων. Η Neo4j καταλαμβάνει την πρωτοπορία στο να απαιτεί τον λιγότερο χώρο αποθήκευσης για τη διατήρηση των δεδομένων, ακόμα και λιγότερο από τον χώρο που καταλαμβάνουν τα ακατέργαστα δεδομένα. Η AgensGraph απαιτεί το μεγαλύτερο αποθηκευτικό χώρο.

Με βάση τα παραπάνω πειράματα που διενεργήθηκαν με μεθοδικό τρόπο, γίνεται προφανές ότι το Neo4j επιδεικνύει ανώτερη απόδοση στον τομέα της μαζικής εισαγωγής μεγάλου όγκου δεδομένων, αν και στον τομέα του αποθηκευτικού χώρου επιβάλλονται σημαντικές απαιτήσεις. Αντίθετα, η AgensGraph υστερεί σε σύγκριση, υποδηλώνοντας την ανάγκη για βελτίωση.

Για τα ερωτήματα IU, η AgensGraph έχει υψηλότερη αποδοτικότητα σε σχέση με την Neo4j, ενώ η AgensGraph επιδέχεται ειδικά στην εισαγωγή ακμών. ενώ υστερεί σε ταχύτητα στην ενημέρωση κορυφών. Από τα αποτελέσματα των ερωτημάτων IS, η AgensGraph επιτυγχάνουν εξαιρετικά αποτελέσματα σε επιλεγμένες περιπτώσεις, όπως τα IS 2 και IS 4, αλλά συνολικά υστερούν από την απόδοση του Neo4j. Οι AgensGraph και Neo4j εξακολουθούν να είναι εξειδικευμένες στις μικρές λειτουργίες.

Ενώ είναι πιθανό να συμβάλλουν διακυμάνσεις στη βελτιστοποίηση των δηλώσεων ερωτημάτων, η βασική πρόκληση βρίσκεται στις τεχνολογίες που διέπουν την αποθήκευση δεδομένων και την επεξεργασία ερωτημάτων. Για παράδειγμα, κατά την εκτέλεση του BI 16, η AgensGraph αντιμετωπίζει προβλήματα με την υπέρβαση του χρονικού ορίου ακόμα και στο σύνολο δεδομένων DG1. Μετά από πιο λεπτομερή έρευνα, διαπιστώθηκε ότι η AgensGraph δυσκολεύεται να ταιριάξει με αποτελεσματικότητα το πρότυπο (Πρόσωπο)-[:KNOWS*3..5]-(Πρόσωπο) εντός ενός λογικού χρονικού πλαισίου. Αυτή η κατάσταση ενδέχεται να σχετίζεται με τις λεπτομέρειες της δομής αποθήκευσης του ή τους μηχανισμούς εκτέλεσης ερωτημάτων.

- Το Neo4j είναι φιλικό προς τον χρήστη και το πιο αποδοτικό στην εισαγωγή δεδομένων. Ωστόσο, είναι κατάλληλο μόνο για μικρά ερωτήματα και μικρομετρημένα σύνολα δεδομένων, ενώ παρουσιάζει κακή απόδοση στην εκτέλεση περίπλοκων ερωτημάτων επιχειρηματικής νοημοσύνης.
- Η AgensGraph λειτουργεί καλά σε φορτία εργασίας που συνοδεύονται από SQL και απλές λειτουργίες ενημέρωσης και ερωτηματολογίας, ενώ εμφανίζει πολύ κακή απόδοση στην επεξεργασία περίπλοκων ερωτημάτων και στη διαχείριση μεγάλων συνόλων δεδομένων [10].

Στρέφοντας την προσοχή μας στην υβριδική επιλογή στην AgensGraph, η ένταξή της με την υποκείμενη σχεσιακή βάση δεδομένων εισάγει σημαντικό επίπεδο πολυπλοκότητας και τα συνακόλουθα κόστη για την ολοκληρωμένη οικοσύσταση της βάσης δεδομένων γράφων.

Οι δυο βάσεις δεδομένων προορίζονται για επιχειρηματικές εφαρμογές, προσφέροντας σταθερότητα, επαγγελματισμό και προσαρμοστικότητα για πραγματικά ερωτήματα επιχειρήσεων. Βασιζόμενοι στην εμπειρία των χρηστών, η Neo4j και AgensGraph προωθούν ένα πιο φιλικό προς τον χρήστη περιβάλλον, προσφέροντας υποστήριξη για ένα ευρύτερο φάσμα γραμματικών γλωσσών ερωτημάτων.

	Graph Model	Relational Model	Document Model	Key-value Store Model
AgensGraph	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>
ArangoDB	<input type="radio"/>	<input checked="" type="checkbox"/>	<input type="radio"/>	<input type="radio"/>
Couchbase	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="radio"/>	<input type="radio"/>
Neo4j	<input type="radio"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
OrientDB	<input type="radio"/>	<input checked="" type="checkbox"/>	<input type="radio"/>	<input type="radio"/>
CosmosDB	<input type="radio"/>	<input checked="" type="checkbox"/>	<input type="radio"/>	<input type="radio"/>
Neptune	<input type="radio"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Εικόνα 6: Βάσεις δεδομένων γράφων

5.3 Σύγκριση AgensGraph με Neo4j

Name	AgensGraph			Neo4j		
Description	Multi-model database supporting relational and graph data models and built upon PostgreSQL			Scalable, ACID-compliant graph database designed with a high-performance distributed cluster architecture, available in self-hosted and cloud offerings		
Primary database model	Graph DBMS			Graph DBMS		
	Relational DBMS					
DB-Engines Ranking	Score	0.07		Score	50.39	
	Rank	#389	Overall	Rank	#23	Overall
		#36	Graph DBMS		#1	Graph DBMS
		#161	Relational DBMS			
Website	bitnine.net			neo4j.com		
Technical documentation	bitnine.net/documentation			neo4j.com/docs		
Developer	Bitnine Global Inc.			Neo4j, Inc.		
Initial release	2016			2007		
Current release	2.1, December 2018			5.11, August 2023		
License	Open Source			Open Source		
Cloud-based only	no			no		
DBaaS offerings (sponsored links)				Neo4j Aura: Neo4j's fully managed cloud service: The zero-admin, always-on graph database for cloud developers.		
Implementation language	C			Java, Scala		
Server operating systems	Linux			Linux		
	OS X			OS X		
	Windows			Solaris		
				Windows		
Data scheme	depending on used data model			schema-free and schema-optional		
Typing	yes			yes		
XML support	no					
Secondary indexes	yes			yes		
SQL	yes			no		
APIs and other access methods	Cypher Query Language			Bolt protocol		
	JDBC			Cypher query language		
				Java API		
				Neo4j-OGM		
				RESTful HTTP API		
				Spring Data Neo4j		
				TinkerPop 3		

Supported programming languages	C	.Net
	Java	Clojure
	JavaScript	Elixir
	Python	Go
		Groovy
		Haskell
		Java
		JavaScript
		Perl
		PHP
		Python
		Ruby
		Scala
Server-side scripts	yes	yes
Triggers	no	yes
Partitioning methods	no, but can be realized using table inheritance	yes using Neo4j Fabric
Replication methods	Source-replica replication	Causal Clustering using Raft protocol
MapReduce	no	no
Consistency concepts	Immediate Consistency	Causal and Eventual Consistency configurable in Causal Cluster setup
		Immediate Consistency in stand-alone mode
Foreign keys	yes	yes
Transaction concepts	ACID	ACID
Concurrency	yes	yes
Durability	yes	yes
In-memory capabilities	no	
User concepts	fine grained access rights according to SQL-standard	Users, roles and permissions. Pluggable authentication with supported standards (LDAP, Active Directory,

DB-Engines Ranking: Μετρά τη δημοτικότητα των συστημάτων διαχείρισης βάσεων δεδομένων.

License: Εμπορικά ή Ανοικτού Κώδικα

DBaaS offerings: Βάση Δεδομένων ως Υπηρεσία

Typing: προκαθορισμένοι τύποι δεδομένων όπως το float ή ημερομηνία
XML support: Κάποια μορφή επεξεργασίας δεδομένων σε μορφή XML, π.χ. υποστήριξη για δομές δεδομένων XML και/ή υποστήριξη για XPath, XQuery ή XSLT.

Partitioning methods: Μέθοδοι για την αποθήκευση διαφορετικών δεδομένων σε διάφορους κόμβους.

Replication methods: Μέθοδοι για την πολλαπλή αποθήκευση δεδομένων με αντιγραφή σε πολλούς κόμβους (αποθήκευση με αντιγραφή για εξασφάλιση αντοχής σε βλάβες).

MapReduce: Προσφέρει μια διεπαφή προγραμματισμού εφαρμογών (API) για την καθορισμό μεθόδων χάρτη/ελαχίστων (Map/Reduce) που ορίζονται από τον χρήστη.

Consistency concepts: Μέθοδοι για τη διασφάλιση συνέπειας σε ένα κατανεμημένο σύστημα.

Transaction concepts: Υποστήριξη για τη διασφάλιση της ακεραιότητας των δεδομένων μετά από μη ατομικές επεμβάσεις στα δεδομένα [13].

Πλεονεκτήματα:

Η AgensGraph παρουσιάζει πολλά προτερήματα ως ένα καινοτόμο υβριδικό σύστημα βάσης δεδομένων, συνδυάζοντας αρμονικά τα ιδιαίτερα πλεονεκτήματα των βάσεων δεδομένων γράφων και των σχεσιακών βάσεων δεδομένων. Η ενσωματωμένη προσαρμοστικότητά του επιτρέπει την αναπαράσταση περίπλοκων σχέσεων δεδομένων, καθιστώντας το ιδιαίτερα κατάλληλο για εφαρμογές χαρακτηρισμένες από διασυνδεδεμένα στοιχεία. Παρέχοντας ισχυρές δυνατότητες για την διάβαση γράφων, η AgensGraph επιτρέπει την αποτελεσματική εξερεύνηση περίπλοκων δικτύων δεδομένων, συμβάλλοντας στην αναγνώριση νοηματικών προτύπων. Επιπλέον, οι χρήστες απολαμβάνουν τα οφέλη μιας ποικίλης σειράς γλωσσών ερωτημάτων, περιλαμβανομένου του κοινού Cypher και της SQL, το οποίο διαμορφώνει μια

ευέλικτη προσέγγιση για την ανάκτηση δεδομένων. Η συμπερίληψη προηγμένων τύπων δεδομένων, όπως JSON και γεωχωρικά δεδομένα, ενισχύει περαιτέρω την ευελιξία της, διευκολύνοντας την ομαλή ενσωμάτωση με σύγχρονες εφαρμογές. Ειδικότερα, το πολυμοντέλο προσέγγισης της ευκολύνει την διαχείριση δεδομένων αποδεχόμενη και τα μοντέλα δεδομένων γράφων και σχετικών δεδομένων μέσα σε μία μόνο βάση, ενισχύοντας την συνοχή και απλοποιώντας την αρχιτεκτονική εφαρμογών.

Μειονεκτήματα:

Ωστόσο, είναι αναγκαίο να αναγνωρίσουμε ότι η AgensGraph δεν είναι απαλλαγμένη από πολυπλοκότητες. Οι χρήστες που μεταβαίνουν από τις παραδοσιακές σχετικές βάσεις δεδομένων μπορεί να αντιμετωπίσουν μια περίοδο προσαρμογής κατά την κατανόηση των εννοιών των βάσεων δεδομένων γράφων και των γλωσσών ερωτημάτων. Οι περίπλοκες πτυχές της βελτιστοποίησης της απόδοσης, ιδιαίτερα για πολύπλοκα ερωτήματα που αφορούν μεγάλα σύνολα δεδομένων και περίπλοκες σχέσεις, μπορεί να αποδειχθούν πολύπλοκες.

Επιπλέον, η πόρων-επιβαρυντική φύση της διαχείρισης δεδομένων γράφων, σε συνδυασμό με τις πιθανές αυξημένες απαιτήσεις αποθήκευσης, θα μπορούσε να αποτελεί πρόκληση σε περιβάλλοντα με περιορισμένους πόρους. Επιπλέον, αξίζει να σημειωθεί ότι το οικοσύστημα που περιβάλλει την AgensGraph, συμπεριλαμβανομένης της κοινοτικής υποστήριξης και των ολοκληρωμένων λύσεων, μπορεί να μην έχει την ίδια εκτεταμένη φύση με πιο καθιερωμένες λύσεις βάσεων δεδομένων. Η πτυχή της επεκτασιμότητας, ιδιαίτερα όσον αφορά μεγάλα σύνολα δεδομένων και περιπτώσεις με υψηλή ταυτόχρονη χρήση, θα μπορούσε να αντιμετωπιστούν ως προκλήσεις σε σύγκριση με υπάρχουσες βάσεις δεδομένων γράφων. Επομένως, μια σοφή εκτίμηση του συγκεκριμένου πλαισίου χρήσης και των απαιτήσεων αποτελεί βασικό στοιχείο κατά την αξιολόγηση της καταλληλότητας της AgensGraph για ένα συγκεκριμένο εγχείρημα.

5.4 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΚΑΙ ΕΓΓΑΤΑΣΤΑΣΗ

Για την εκτέλεση των συγκρίσεων χρησιμοποίησα τον προσωπικό μου υπολογιστή με χαρακτηριστικά: Windows 10 Pro, επεξεργαστή AMD Ryzen 7 2700X Eight-Core 3.7 GHz, Ram 32 GB.

Neo4j:

Για την εγκατάσταση της neo4j πηγαίνουμε στο official site:

<https://neo4j.com/download/> και πατάμε κατέβασμα neo4j για υπολογιστή. Κάνουμε εγγραφή και μας δίνεται ένα κλειδί ενεργοποίησης. Μόλις κατεβεί το αρχείο το κάνουμε εγκατάσταση. Αφού τελειώσει η εγκατάσταση κάνουμε επικόλληση το κλειδί μέσα στην εφαρμογή και μπορούμε τώρα να κάνουμε καινούργια project. Στην συνέχεια κάνουμε εισαγωγή το σύνολο δεδομένων στο οποίο θα κάνουμε την σύγκριση μας.

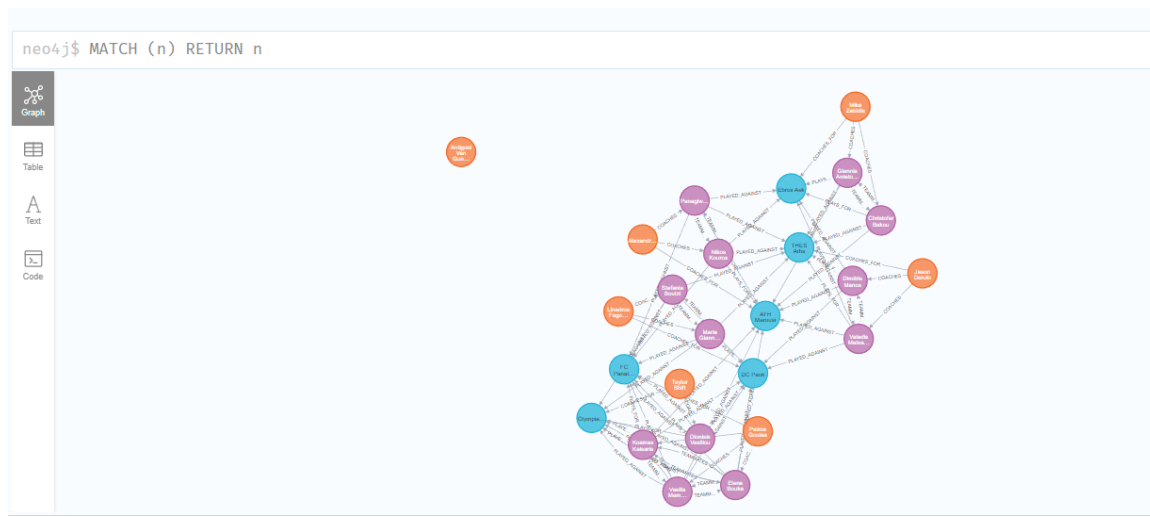
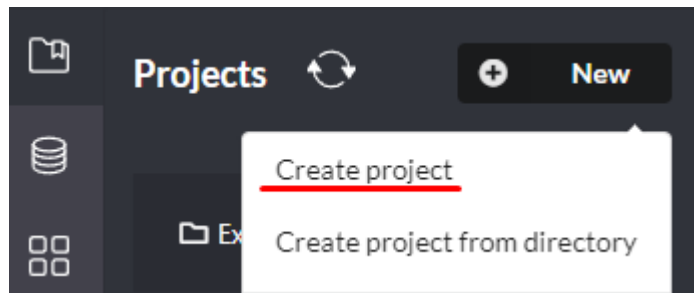
Get Neo4j Desktop

Get Started Now

Please fill out this form to begin your download

<input type="text"/>
First Name
<input type="text"/>
Last Name
<input type="text"/>
Email (Business Preferred)
<input type="text"/>
Company Name
<input type="text"/>
Phone Number
<input type="text"/>
Country/Territory
<input type="text"/>
By downloading you agree to the Neo4j License Agreement for Neo4j Desktop Software .
<input type="button" value="Download Desktop"/>
The information you provide will be used in accordance with the terms of our Privacy Policy

Εικόνα 7: Neo4j εγγραφή source (<https://neo4j.com/>)



Εικόνα 8: Model

AgensGraph:

Για την εγκατάσταση της AgensGraph πηγαίνουμε στο official site:

<https://bitnine.net/agensgraph-downloads/> και επιλέγουμε download. Στην συνέχεια επιλέγουμε την community Edition v2.13.0 για windows. Συμπληρώνουμε τα στοιχεία μας και μας στέλνουν στο email που δηλώσαμε έναν σύνδεσμο για κατέβασμα της AgensGraph στον υπολογιστή μας. Μόλις κατέβει κάνουμε εγκατάσταση ακολουθώντας και της οδηγίες από το official site. Για να έχουμε UI πρέπει να το κατεβάσουμε ξεχωριστά. Στο official site επιλέγουμε την browser έκδοση AgensBrowser 1.0,

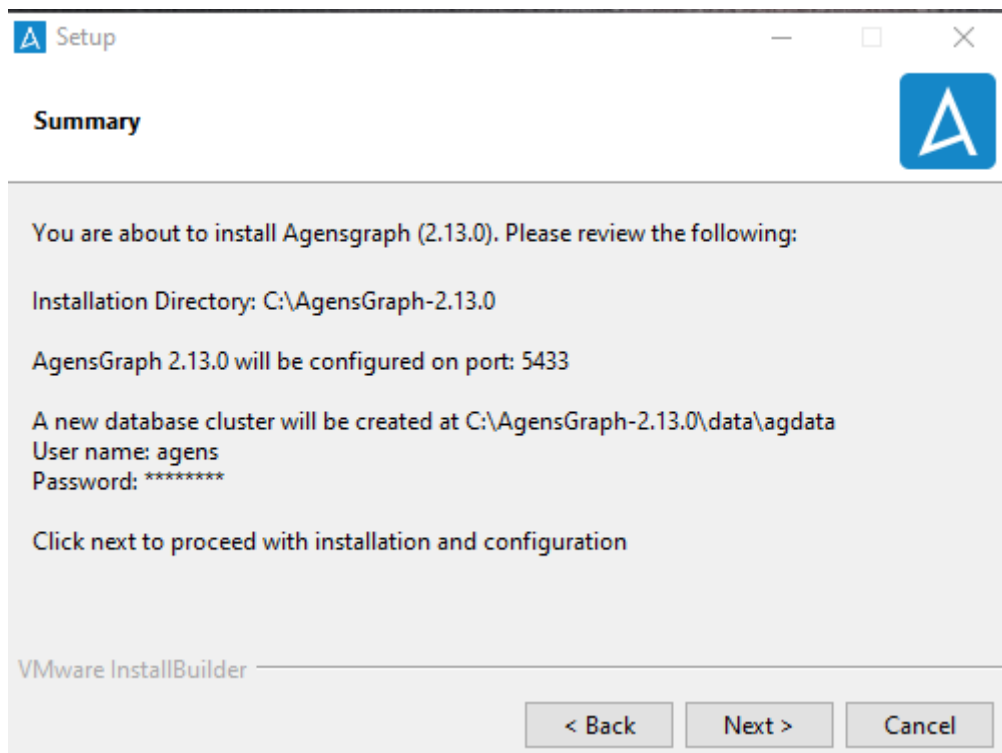
συμπληρώνουμε ξανά τα στοιχεία μας και μας στέλνουν σύνδεσμο για το κατέβασμα του browser. Προϋπόθεση είναι να έχουμε java 8. Για την εγκατάσταση του browser η διαδικασία είναι αρκετά πολύπλοκη και δημιουργεί μια αρνητική αίσθηση στον χρήστη που θα θέλει να δοκιμάσει μια καινούργια βάση δεδομένων.

Community Edition

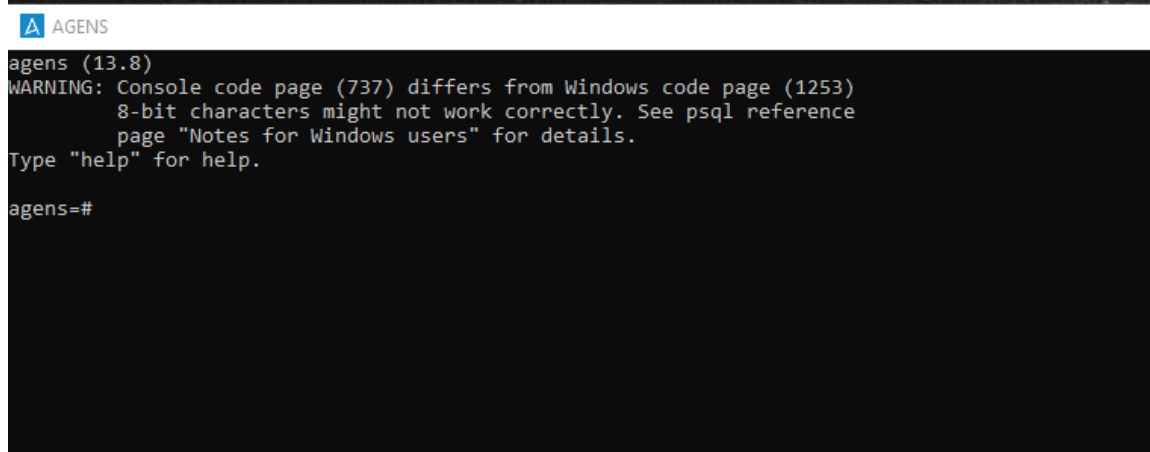
AgensGraph Community Edition is a multi-model graph database for businesses who seek a Fast, Solid, and Flexible environment for connected data with high relational compatibility. Combined with AgensBrowser, it offers priceless answers on key business issues with intuitive and effective visualization of the graph data.

v2.13.0 Linux	v2.13.0 Windows
Docker	MAC OS
AgensBrowser v1.0	AgensBrowser-Docker

Εικόνα 9: Σελίδα κατεβάσματος



Εικόνα 10: Περίληψη



Εικόνα 11: AgensGraph command line

```

** Package (compressed)
- agens-browser-web-1.0.jar (ver 1.0, compiled at 2018-03-02)
- agens-browser.config.yml
- readme.txt

** Requirement
- JRE 8 (JAVA 1.8)
- AgensGraph 1.2 or over (try connection test by your ID/PW to JDBC-URL and graph_path)
- Web browser : Chrome, Firefox, Safari, Edge (except IE)
- Open 'agens-browser.config.yml' and modify connection information for your environment.
  Please refer to the below manual:

** Manual
http://bitnine.net/documentations/agensbrowser-manual-1.0-en.html#installation-and-startup

```

Εικόνα 12: Απαιτήσεις AgensGraph

```

nba=# MATCH (n) RETURN n;
n
-----
player[3.1]{age: 30, name: "Kosmas Kaisaris", height: 1.70, number: 0, weight: 91}
player[3.2]{age: 30, name: "Vasilis Memmos", height: 1.85, number: 6, weight: 103}
player[3.3]{age: 27, name: "Elena Bouka", height: 1.65, number: 23, weight: 63}
player[3.4]{age: 29, name: "Dionisis Vasiliou", height: 1.91, number: 12, weight: 79}
player[3.5]{age: 30, name: "Valadis Matsakas", height: 1.80, number: 42, weight: 80}
player[3.6]{age: 26, name: "Dimitris Manos", height: 2.01, number: 6, weight: 109}
player[3.7]{age: 24, name: "Maria Giannelou", height: 1.63, number: 7, weight: 64}
player[3.8]{age: 30, name: "Stefania Boutzi", height: 1.64, number: 13, weight: 55}
player[3.9]{age: 26, name: "Giannis Antetokounmpo", height: 2.11, number: 34, weight: 110}
player[3.10]{age: 40, name: "Christofer Bakou", height: 1.87, number: 22, weight: 90}
player[3.11]{age: 27, name: "Nikos Kouros", height: 2.13, number: 21, weight: 130}
player[3.12]{age: 29, name: "Panagiwths Keramaris", height: 1.90, number: 22, weight: 87}
coach[4.1]{name: "Petros Goulas"}
coach[4.2]{name: "Taylor Shift"}
coach[4.3]{name: "Jason Derulo"}
coach[4.4]{name: "Linadros Fagomenos"}
coach[4.5]{name: "Mike Zabidis"}
coach[4.6]{name: "Alexandros Kotsis"}
coach[4.7]{name: "Antigoni Van Gundy"}
team[5.1]{name: "FC Panathinaikos"}
team[5.2]{name: "Olympiakos AE"}
team[5.3]{name: "THES Arhs"}
team[5.4]{name: "DC Paok"}
team[5.5]{name: "Ebros Aek"}
team[5.6]{name: "ATH Marousi"}
(25 rows)

```

Εικόνα 13: AgensGraph Model

The screenshot shows the Neo4j browser interface. At the top, the Cypher query is: `neo4j$ MATCH (p1:PLAYER)-[g1:PLAYED_AGAINST]-(opponent)-[g2:PLAYED_AGAINST]-(p2:PLAYER) WHERE p1 <> p2 AND g1.minutes > 30 AND g2.minutes > 30 WITH p1, p2, AVG(g1.points) AS avgPoints1, AVG(g1.assists) AS avgAssists1, AVG(g2.points) AS avgPoints2, AVG(g2.assists) AS avgAssists2, AVG(g1.rebounds) AS avgRebounds1, AVG(g2.rebounds) AS avgRebounds2`. Below the query, a table displays the results of the query, showing player matchups and their average statistics.

	p1.name	p2.name	avgPoints1	avgAssists1	avgRebounds1	avgPoints2	avgAssists2	avgRebounds2
1	"Maria Giannelou"	"Dionisis Vasiliou"	45.0	5.0	8.0	38.5	5.0	6.0
2	"Maria Giannelou"	"Nikos Kouros"	45.0	5.0	8.0	36.0	7.0	12.0
3	"Maria Giannelou"	"Panagiwths Keramaris"	45.0	5.0	8.0	22.0	1.0	7.0
4	"Maria Giannelou"	"Stefania Boutzi"	41.0	3.5	10.0	35.0	13.0	4.0
5	"Dionisis Vasiliou"	"Maria Giannelou"	38.5	5.0	6.0	45.0	5.0	8.0
6	"Dionisis Vasiliou"	"Stefania Boutzi"	38.5	5.0	6.0	38.0	13.0	4.0

Εικόνα 14: Query 13

5.5 Πείραμα

Αφού εγκαταστήσαμε τις βάσεις δεδομένων Neo4j και AgensGraph και κάναμε εισαγωγή των δεδομένων (25 labels, 25 nodes, 270 properties, 81 relationships) με την παρακάτω εντολή:


```

CREATE
(kosmas:PLAYER{name: 'Kosmas Kaisaris', age: 30, number: 0, height: 1.70, weight: 91}),
(vasilis:PLAYER{name: 'Vasilis Memmos', age: 30, number: 6, height: 1.85, weight: 103}),
(elena:PLAYER{name: 'Elena Bouka', age: 27, number: 23, height: 1.65, weight: 63}),
(dionisis:PLAYER{name: 'Dionisis Vasiliou', age: 29, number: 12, height: 1.91, weight: 79}),
(valadis:PLAYER{name: 'Valadis Matsakas', age: 30, number: 42, height: 1.80, weight: 80}),
(dimitris:PLAYER{name: 'Dimitris Manos', age: 26, number: 6, height: 2.01, weight: 109}),
(maria:PLAYER{name: 'Maria Giannelou', age: 24, number: 7, height: 1.63, weight: 64}),
(stefania:PLAYER{name: 'Stefania Boutzi', age: 30, number: 13, height: 1.64, weight: 55}),
(giannis:PLAYER{name: 'Giannis Antetokounmpo', age: 26, number: 34, height: 2.11, weight: 110}),
(christofer:PLAYER{name: 'Christofer Bakou', age: 40, number: 22, height: 1.87, weight: 90}),
(nikos:PLAYER{name: 'Nikos Kouros', age: 27, number: 21, height: 2.13, weight: 130}),
(panagiwths:PLAYER{name: 'Panagiwths Keramaris', age: 29, number: 22, height: 1.90, weight: 87}),

(petros:COACH{name: 'Petros Goulas'}),
(taylor:COACH{name: 'Taylor Shift'}),
(jason:COACH{name: 'Jason Derulo'}),
(linadros:COACH{name: 'Linadros Fagomenos'}),
(mike:COACH{name: 'Mike Zabidis'}),
(alexandros:COACH{name: 'Alexandros Kotsis'}),
(antigoni:COACH{name: 'Antigoni Van Gundy'}),

(panathinaikos:TEAM{name: 'FC Panathinaikos'}),
(olympiakos:TEAM{name: 'Olympiakos AE'}),
(arhs:TEAM{name: 'THES Arhs'}),
(paok:TEAM{name: 'DC Paok'}),
(aek:TEAM{name: 'Ebros Aek'}),
(marousi:TEAM{name: 'ATH Marousi'}),

(vasilis)-[:TEAMMATES]-> (kosmas),
(vasilis)<-[:TEAMMATES]- (kosmas),
(vasilis)-[:TEAMMATES]-> (elena),
(vasilis)<-[:TEAMMATES]- (elena),
(kosmas)-[:TEAMMATES]-> (elena),
(kosmas)<-[:TEAMMATES]- (elena),
(valadis)-[:TEAMMATES]-> (dimitris),
(valadis)<-[:TEAMMATES]- (dimitris),
(maria)-[:TEAMMATES]-> (stefania),
(maria)<-[:TEAMMATES]- (stefania),
(giannis)-[:TEAMMATES]-> (christofer),
(giannis)<-[:TEAMMATES]- (christofer),
(nikos)-[:TEAMMATES]-> (panagiwths),
(nikos)<-[:TEAMMATES]- (panagiwths),

(petros)-[:COACHES]->(vasilis),
(petros)-[:COACHES]->(elena),
(petros)-[:COACHES]->(kosmas),
(taylor)-[:COACHES]->(dionisis),
(jason)-[:COACHES]->(valadis),
(jason)-[:COACHES]->(dimitris),
(linadros)-[:COACHES]->(maria),
(linadros)-[:COACHES]->(stefania),
(mike)-[:COACHES]->(giannis),
(mike)-[:COACHES]->(christofer),
(alexandros)-[:COACHES]->(panagiwths),
(alexandros)-[:COACHES]->(nikos),

(vasilis)-[:PLAYS_FOR {salary: 40000000}]-> (panathinaikos),
(kosmas)-[:PLAYS_FOR {salary: 33000000}]-> (panathinaikos),
(elena)-[:PLAYS_FOR {salary: 38000000}]-> (panathinaikos),
(dionisis)-[:PLAYS_FOR {salary: 8000000}]-> (olympiakos),
(valadis)-[:PLAYS_FOR {salary: 50000000}]-> (arhs),

```

```

(dimitris)-[:PLAYS_FOR {salary: 26000000}]-> (arhs),
(maria)-[:PLAYS_FOR {salary: 45000000}]-> (paok),
(stefania)-[:PLAYS_FOR {salary: 42000000}]-> (paok),
(giannis)-[:PLAYS_FOR {salary: 47000000}]-> (aek),
(christofer)-[:PLAYS_FOR {salary: 43000000}]-> (aek),
(nikos)-[:PLAYS_FOR {salary: 40000000}]-> (marousi),
(panagiwths)-[:PLAYS_FOR {salary: 40000000}]-> (marousi),

(petros)-[:COACHES_FOR]->(panathinaikos),
(taylor)-[:COACHES_FOR]->(olympiakos),
(jason)-[:COACHES_FOR]->(arhs),
(linadros)-[:COACHES_FOR]->(paok),
(mike)-[:COACHES_FOR]->(aek),
(alexandros)-[:COACHES_FOR]->(marousi),

(vasilis)-[:PLAYED_AGAINST {minutes: 38, points: 32, assists: 6, rebounds: 6, turnovers: 2}]->
(olympiakos),
(kosmas)-[:PLAYED_AGAINST {minutes: 29, points: 16, assists: 12, rebounds: 11, turnovers: 16}]->
(olympiakos),
(elena)-[:PLAYED_AGAINST {minutes: 36, points: 27, assists: 2, rebounds: 8, turnovers: 1}]->
(olympiakos),
(dionisis)-[:PLAYED_AGAINST {minutes: 43, points: 42, assists: 7, rebounds: 8, turnovers: 4}]->
(panathinaikos),

(vasilis)-[:PLAYED_AGAINST {minutes: 23, points: 25, assists: 12, rebounds: 3, turnovers: 0}]->
(olympiakos),
(kosmas)-[:PLAYED_AGAINST {minutes: 20, points: 11, assists: 10, rebounds: 3, turnovers: 8}]->
(olympiakos),
(elena)-[:PLAYED_AGAINST {minutes: 30, points: 22, assists: 2, rebounds: 8, turnovers: 1}]->
(olympiakos),
(dionisis)-[:PLAYED_AGAINST {minutes: 35, points: 35, assists: 3, rebounds: 4, turnovers: 2}]->
(panathinaikos),

(vasilis)-[:PLAYED_AGAINST {minutes: 32, points: 18, assists: 3, rebounds: 6, turnovers: 1}]->
(paok),
(kosmas)-[:PLAYED_AGAINST {minutes: 26, points: 26, assists: 11, rebounds: 13, turnovers: 6}]->
(paok),
(elena)-[:PLAYED_AGAINST {minutes: 30, points: 26, assists: 7, rebounds: 18, turnovers: 3}]->
(paok),
(maria)-[:PLAYED_AGAINST {minutes: 43, points: 45, assists: 5, rebounds: 8, turnovers: 2}]->
(panathinaikos),
(stefania)-[:PLAYED_AGAINST {minutes: 46, points: 35, assists: 13, rebounds: 4, turnovers: 7}]->
(panathinaikos),

(maria)-[:PLAYED_AGAINST {minutes: 34, points: 37, assists: 2, rebounds: 12, turnovers: 1}]->
(olympiakos),
(stefania)-[:PLAYED_AGAINST {minutes: 46, points: 35, assists: 13, rebounds: 4, turnovers: 7}]->
(olympiakos),
(dionisis)-[:PLAYED_AGAINST {minutes: 26, points: 32, assists: 13, rebounds: 6, turnovers: 2}]->
(paok),

(valadis)-[:PLAYED_AGAINST {minutes: 44, points: 23, assists: 7, rebounds: 13, turnovers: 8}]->
(aek),
(dimitris)-[:PLAYED_AGAINST {minutes: 24, points: 16, assists: 2, rebounds: 12, turnovers: 0}]->
(aek),
(giannis)-[:PLAYED_AGAINST {minutes: 33, points: 26, assists: 16, rebounds: 18, turnovers: 5}]->
(arhs),
(christofer)-[:PLAYED_AGAINST {minutes: 46, points: 35, assists: 3, rebounds: 4, turnovers: 3}]->
(arhs),

(valadis)-[:PLAYED_AGAINST {minutes: 33, points: 28, assists: 6, rebounds: 3, turnovers: 3}]->
(marousi),

```

```

(dimitris)-[:PLAYED_AGAINST {minutes: 24, points: 18, assists: 4, rebounds: 11, turnovers: 1}]->
(marousi),
(nikos)-[:PLAYED_AGAINST {minutes: 25, points: 29, assists: 7, rebounds: 22, turnovers: 2}]->
(arhs),
(panagiwths)-[:PLAYED_AGAINST {minutes: 34, points: 18, assists: 13, rebounds: 4, turnovers: 0}]->
> (arhs),

(giannis)-[:PLAYED_AGAINST {minutes: 45, points: 36, assists: 5, rebounds: 12, turnovers: 3}]->
(marousi),
(christofer)-[:PLAYED_AGAINST {minutes: 35, points: 22, assists: 5, rebounds: 6, turnovers: 0}]->
(marousi),
(nikos)-[:PLAYED_AGAINST {minutes: 33, points: 23, assists: 3, rebounds: 10, turnovers: 3}]->
(aek),
(panagiwths)-[:PLAYED_AGAINST {minutes: 38, points: 23, assists: 4, rebounds: 5, turnovers: 1}]->
(aek),

(maria)-[:PLAYED_AGAINST {minutes: 29, points: 28, assists: 6, rebounds: 8, turnovers: 0}]->
(arhs),
(stefania)-[:PLAYED_AGAINST {minutes: 35, points: 17, assists: 10, rebounds: 8, turnovers: 5}]->
(arhs),
(valadis)-[:PLAYED_AGAINST {minutes: 37, points: 35, assists: 6, rebounds: 11, turnovers: 4}]->
(paok),
(dimitris)-[:PLAYED_AGAINST {minutes: 34, points: 27, assists: 4, rebounds: 8, turnovers: 0}]->
(paok),

(vasilis)-[:PLAYED_AGAINST {minutes: 32, points: 27, assists: 12, rebounds: 10, turnovers: 4}]->
(marousi),
(kosmas)-[:PLAYED_AGAINST {minutes: 25, points: 19, assists: 9, rebounds: 14, turnovers: 5}]->
(marousi),
(elena)-[:PLAYED_AGAINST {minutes: 32, points: 22, assists: 7, rebounds: 12, turnovers: 2}]->
(marousi),
(nikos)-[:PLAYED_AGAINST {minutes: 36, points: 36, assists: 7, rebounds: 12, turnovers: 0}]->
(panathinaikos),
(panagiwths)-[:PLAYED_AGAINST {minutes: 32, points: 22, assists: 1, rebounds: 7, turnovers: 0}]->
(panathinaikos);

```

τρέξαμε τα παρακάτω ερωτήματα :

- 1) Import Data
- 2) MATCH (n) RETURN n; (Εικόνα 8 & Εικόνα 13)
- 3) MATCH (player:PLAYER) RETURN player;
- 4) MATCH (player:PLAYER) RETURN player.name, player.height;
- 5) MATCH (player:PLAYER)

WHERE player.name = 'Vasilis Memmos'

RETURN player;
- 6) MATCH (player:PLAYER)

WHERE player.name <> 'Vasilis Memmos'

RETURN player;

- 7 MATCH (player:PLAYER)
WHERE (player.weight / (player.height * player.height)) > 25
RETURN player;
- 8 MATCH (player:PLAYER)
WHERE player.height >= 2
RETURN player
ORDER BY player.height DESC
LIMIT 3;
- 9 MATCH (vasilis:PLAYER {name: 'Vasilis Memmos'}) - [:TEAMMATES] ->
(teammate:PLAYER)
MATCH (teammate) - [contract:PLAYS_FOR] -> (:TEAM)
WHERE contract.salary >= 40000000
RETURN teammate;
- 10 MATCH (player:PLAYER) - [:PLAYS_FOR] - (:TEAM {name: 'FC
Panathinaikos'})
MATCH (player) - [gamePlayed:PLAYED_AGAINST] - (:TEAM)
RETURN player.name, AVG(gamePlayed.points) AS ppg
ORDER BY ppg DESC
LIMIT 1;
- 11 MATCH (t:TEAM) <- [p:PLAYS_FOR] - (player:PLAYER)
WITH t, AVG(p.salary) AS avgSalary
RETURN t.name, avgSalary
ORDER BY avgSalary DESC;
- 12
MATCH (p:PLAYER {name: 'Kosmas Kaisaris'}) - [:PLAYED_AGAINST] ->
(opponent)
WITH p, AVG(opponent.points) AS avgPoints, AVG(opponent.assists) AS
avgAssists, AVG(opponent.rebounds) AS avgRebounds
RETURN p.name, avgPoints, avgAssists, avgRebounds;
- 13
MATCH (p1:PLAYER) - [g1:PLAYED_AGAINST] -> (opponent) <-
[g2:PLAYED_AGAINST] - (p2:PLAYER)
WHERE p1 <> p2 AND g1.minutes > 30 AND g2.minutes > 30
WITH p1, p2, AVG(g1.points) AS avgPoints1, AVG(g1.assists) AS avgAssists1,
AVG(g1.rebounds) AS avgRebounds1,
AVG(g2.points) AS avgPoints2, AVG(g2.assists) AS avgAssists2,
AVG(g2.rebounds) AS avgRebounds2

```

RETURN p1.name, p2.name, avgPoints1, avgAssists1, avgRebounds1,
avgPoints2, avgAssists2, avgRebounds2
ORDER BY avgPoints1 DESC; (Εικόνα 14)

```

```

14 MATCH (p1:PLAYER)-[g1:PLAYED_AGAINST]->(opponent)<-
[g2:PLAYED_AGAINST]-(p2:PLAYER)
WHERE p1 <> p2
AND g1.minutes > 30
AND g2.minutes > 30
AND g1.points > 20
AND g2.points > 20
WITH p1, p2,
AVG(g1.points) AS avgPoints1,
AVG(g1.assists) AS avgAssists1,
AVG(g1.rebounds) AS avgRebounds1,
AVG(g2.points) AS avgPoints2,
AVG(g2.assists) AS avgAssists2,
AVG(g2.rebounds) AS avgRebounds2
WHERE avgPoints1 > 25 OR avgPoints2 > 25
RETURN p1.name, p2.name, avgPoints1, avgAssists1, avgRebounds1,
avgPoints2, avgAssists2, avgRebounds2
ORDER BY avgPoints1 DESC;

```

Οι χρόνοι είναι σε ms.

Query	Neo4j	AgensGraph	Records
Query 1	194	140,200	-
Query 2	22-7	12,187	25
Query 3	21-3	0,481	12
Query 4	2-2	0,459	12
Query 5	27-2	8,920	1

Query 6	29-2	0,628	11
Query 7	45-8	0,838	5
Query 8	53-8	0,994	3
Query 9	88-2	8,622	0
Query 10	84-2	11,387	1
Query 11	61-7	9,193	6
Query 12	49-2	1,551	1
Query 13	120-18	3,684	64
Query 14	135-13	4,046	48

Επιλέξαμε τα ερωτήματα αυτά βάσει ποικιλομορφίας για να δοκιμάσουμε διάφορες δυνατότητες των βάσεων. Ξεκινώντας από απλές αναζητήσεις, queries 1-4 για να μας φέρει όλα τα δεδομένα ή και επιλογή node και properties. Queries 5-6 χρησιμοποιήθηκε φιλτράρισμα ‘Where’. Queries 7-8 που απαιτούν υπολογισμούς στα attributes και ταξινομήσεις. Queries 9-10 φιλτράρουν attributes στα relationships. Από το 12 ερώτημα και μετά, έχουμε ομαδοποιήσεις. Queries 11-12 υπολογισμός μέσου όρου και Queries 13-14 περίπλοκα queries με πολλά φίλτρα, υπολογισμούς και συνθήκες.

Συμπέρασμα:

Από το συγκεκριμένο Dataset και τα ερωτήματα που τρέξαμε βλέπουμε ότι η AgensGraph είναι πολύ πιο γρήγορη από την Neo4j. Τα αποτελέσματα είναι σε αντίθεση με την παραπάνω έρευνα. Αυτό οφείλεται σε πολλούς παράγοντες όπως το μέγεθος των δεδομένων, την πολυπλοκότητα των ερωτημάτων, το σύστημα στο οποίο τρέξαμε τα ερωτήματα μας και άλλα. Πρέπει να αναφερθεί ότι το πείραμα στην Neo4j έγινε στον browser σε αντίθεση με την AgensGraph που έγινε στο command line. Έτσι οι χρόνοι της Neo4j επηρεάζονται. Ο πρώτος είναι ο χρόνος που έχει υπολογιστή το πρώτο

δεδομένο και ο δεύτερος χρόνος είναι ο χρόνος που χρειάστηκε να τελειώσουν όλα τα δεδομένα.

6 Μελλοντικές Επεκτάσεις

Για μελλοντική χρήση μπορεί να γίνει μια περαιτέρω διερεύνηση για τη βελτιστοποίηση της απόδοσης για πολύπλοκα ερωτήματα που περιλαμβάνουν μεγάλα σύνολα δεδομένων και περίπλοκες γραφικές σχέσεις, μπορεί να προσφέρει πολύτιμες εισηγήσεις. Η διερεύνηση προηγμένων τεχνικών ευρετηρίασης και στρατηγικών βελτιστοποίησης ερωτημάτων που προσαρμόζονται με προσοχή στην υβριδική φύση της AgensGraph μπορεί να συμβάλει στη βελτίωση της αποτελεσματικότητάς της και της κλιμάκωσης της. Επιπλέον, η εμβάθυνση στην κατανόηση και δημιουργία εργαλείων ή πλαισίων που βοηθούν στην απλοποίηση της μάθησης για χρήστες που μεταβαίνουν από τις σχετικές βάσεις δεδομένων. Αυτά τα εργαλεία θα μπορούσαν να περιλαμβάνουν εκπαιδευτικά μοντέλα, μηχανισμούς μετάφρασης ερωτημάτων ή βοηθήματα οπτικοποίησης που γεφυρώνουν αποτελεσματικά το χάσμα μεταξύ της SQL και των γλωσσών ερωτημάτων γράφων π.χ. cypher. Επίσης, η σφαιρική διερεύνηση πραγματικών περιπτώσεων χρήσης και εφαρμογών που εκμεταλλεύονται τις ξεχωριστές υβριδικές δυνατότητες της AgensGraph μπορεί να αποκαλύψει την πρακτική της αξία σε διάφορους τομείς. Αυτό θα περιλαμβάνει τον έλεγχο περιπτώσεων όπου τα μοντέλα δεδομένων γράφων και σχετικά δεδομένα συνυπάρχουν σε μία μόνο εφαρμογή, επισημαίνοντας τα πλεονεκτήματα και τις προκλήσεις που συνεπάγονται σε τέτοιου είδους υλοποιήσεις. Τέλος, μια αξιολόγηση της συμβατότητας και της ολοκλήρωσης της AgensGraph με αναδυόμενες τεχνολογίες, όπως πλατφόρμες νέφους(aws, google cloud...) μπορεί να υποδείξει τον δρόμο για την ομαλή της ενσωμάτωσης σε σύγχρονα, κλιμακούμενα αρχιτεκτονικά πλαίσια.

Βιβλιογραφία

- [1] Hector Garcia-Molina, Jeff Ullman, Jennifer Widom, 2015 Database Systems: The Complete Book by Pearson Education Inc.
- [2] Regina O . Obe and Leo S. Hsu, 2015 PostgreSQL: Up and Running by O'Reilly Media
- [5] Ian Robinson, Jim Webber & Emil Eifrem: Graph Databases 2nd Edition by O'Reilly Media
- [6]https://bitnine.net/documentations/manual/quick_guide/agens_graph_quick_guide_html.html
- [7] Pramid J. Sadalage and Martin Fowler: NoSQL Distilled: A Brief Guid to the Emerging World of Polyglot Persistence by Addison-Wesley Professional
- [8]https://bitnine.net/documentations/manual/operation/english/agens_graph_operation_manual_html.html
- [9]https://bitnine.net/documentations/manual/developer/english/agens_graph_developer_manual_html.html
- [10]Ran Wang, Zhengyi Yang, Wenjie Zhang and Xuemin Lim: (2020) An Empirical Study on Recent Graph Database Systems. Knowledge Science, Engineering and Management. KSEM 2020. Lecture Notes in Computer Science(), vol 12274. Springer, Cham.
- [11] Erling, O., et al.: (2015) The LDBC social network benchmark: interactive workload. In: Proceedings of the 2015 ACM SIGMOD, pp. 619–630
- [12] LDBC SNB task force: The LDBC social network benchmark. Technical report, LDBC (2019). https://ldbc.github.io/ldbc_snb_docs/ldbc-snb-specification.pdf
- [13] <https://db-engines.com/en/>: DB-Engines official site
- [14] <https://www.youtube.com/watch?v=8jNPelugC2s&t=1s> Neo4j
- [15] https://www.youtube.com/watch?v=GaFs3_nnyIU AgensGraph installation

Κυρώσεις για λογοκλοπή

Η λογοκλοπή είναι ένα πολύ σοβαρό παράπτωμα. Με απόφαση της ΓΣΕΣ φοιτητής που διαπιστώνεται ότι υποπίπτει σε λογοκλοπή κατά την εκπόνηση της διπλωματικής του εργασίας αποβάλλεται από το ΠΜΣ. Εάν έχει ήδη αποφοιτήσει ανακαλείται το Μεταπτυχιακό δίπλωμα Ειδίκευσης και προωθείται το θέμα στο Δικαστικό Γραφείο του Πανεπιστημίου για την έναρξη των ανάλογων νομικών διαδικασιών.