ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΕΠΙΣΚΟΠΗΣΗ ΣΥΣΤΗΜΑΤΩΝ ΔΙΑΧΕΙΡΙΣΗΣ ΔΕΔΟΜΕΝΩΝ ΜΕΓΑΛΗΣ ΚΛΙΜΑΚΑΣ

ΛΑΜΠΡΙΝΗ ΖΕΡΒΑ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Ιωάννης Κωνσταντίνου

Επίκουρος Καθηγητής

Λαμία 23/01 έτος 2023

ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΕΠΙΣΚΟΠΗΣΗ ΣΥΣΤΗΜΑΤΩΝ ΔΙΑΧΕΙΡΙΣΗΣ ΔΕΔΟΜΕΝΩΝ ΜΕΓΑΛΗΣ ΚΛΙΜΑΚΑΣ

ΛΑΜΠΡΙΝΗ ΖΕΡΒΑ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Ιωάννης Κωνσταντίνου

Επίκουρος Καθηγητής

Λαμία 23/01 έτος 2023

UNIVERSITY OF
THESSALY

SCHOOL OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE & TELECOMMUNICATIONS

LITERATURE REVIEW OF LARGE-SCALE DATA MANAGEMENT SYSTEMS

LAMPRINI ZERVA

FINAL THESIS

ADVISOR

Ioannis Konstantinou

Assistant Professor

Lamia 23/01 year 2023

«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις [(1)], που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

*1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάσθηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.*

*2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.*

3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια

4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία: ……10/03/2023…

Ο – Η Δηλ.

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον,

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου Ιωάννη Κωνσταντίνου για την καθοδήγηση που μου πρόσφερε καθώς τον χρόνο που διέθεσε δίνοντας μου χρήσιμες συμβουλές για την ολοκλήρωση της πτυχιακής μου εργασίας.

Επίσης, θα ήθελα να ευχαριστήσω όλους τους καθηγητές του Τμήματος Πληροφορικής και Τηλεπικοινωνιών για την συμβολή τους στην επιστημονική μου συγκρότηση στα χρόνια της φοίτησης μου στο Τμήμα.

Τέλος, θα ήθελα να πω ένα μεγάλο ευχαριστώ στην οικογένεια μου αλλά και στους φίλους μου για την ψυχολογική υποστήριξη σε όλο το διάστημα των σπουδών μου.

Περιεχόμενα

Περίληψη

Η συγκεκριμένη πτυχιακή εργασία έχει ως στόχο να παρουσιάσει τον ορισμό «Μεγάλα Δεδομένα», να αναφερθούν σε αυτά αλλά και στις βάσεις δεδομένων.

Στο πρώτο μέρος της εργασίας, αναφερόμαστε σε σημαντικούς ορισμούς των Big Data, σε τεχνικές που χρησιμοποιούνται αλλά και στις διάφορες γλώσσες προγραμματισμού.

Στο δεύτερο μέρος, αναφερόμαστε στις βάσεις δεδομένων και στα χαρακτηριστικά τους.

Στο τρίτο μέρος, παρουσιάζονται κάποιες NoSQL βάσεις δεδομένων που μπορούν να αναπτυχθούν στο υπολογιστικό νέφος και συγκεκριμένα η BangDB, η Voldemort, η LSM, η Tarantool, η Apache HBase, η Apache Cassandra, η ScyllaDB, η Cloudata, η CouchDB, η MondoDB, η ArangoDB, η Neo4J, η Titan, η AllegroGraph, και η WhiteDB.

Στο τέταρτο μέρος, παρουσιάζεται το σχήμα json το οποίο χρησιμοποιείται από την NoSQL.

Στο τέλος της εργασίας, παρουσιάζονται τα συμπεράσματα.

Λέξεις – Κλειδιά: Μεγάλα Δεδομένα, Βάσεις Δεδομένων, NoSQL, σχήμα json

Abstract

This thesis aims to present the definition of "Big Data", to refer to them and also to databases.

In the first part of the paper, we refer to important definitions of Big Data, techniques used and also the various programming languages.

In the second part, we refer to databases and their characteristics.

In the third part, some NoSQL databases that can be deployed in cloud computing are presented and specific BangDB, Voldemort, Tarantool, Apache HBase, Apache Cassandra, ScyllaDB, Cloudata, CouchDB, MondoDB, ArangoDB, Neo4J, Titan, AllegroGraph and WhiteDB.

In the fourth part, the json schema used by NoSQL is presented.

At the end of the paper, conclusions are presented.

Key words: Big Data, Databases, NoSQL, Schema json

Κατάλογος Εικόνων/Σχημάτων

Εικόνα 7: Παράδειγμα Key-Value (https://upload.wikimedia.org/wikipedia/commons/5/5b/KeyValue.PNG)

Εικόνα 8: Παράδειγμα Column Family Stores (https://i.imgur.com/j76Gqga.png)

Εικόνα 9: Document Databases (https://phoenixnap.com/kb/wp-content/uploads/2021/05/document-database-illustration.png)

Εικόνα 10: Graph Database (http://bi-insider.com/wp-content/uploads/2019/01/Graph-Database.png)

Εικόνα 11: BangDB

(https://mma.prnewswire.com/media/1798659/BangDB_Logo.jpg?p=twitter)

Εικόνα 12: Voldemort (https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSQCm5dJKVNLygGnaRX5WyMFbOmDKW8-Kwj0MHDtEdGQLr92MnjzX5BsgbP3Xm4bNBegNI&usqp=CAU)

Εικόνα 12: LSM Tree (https://upload.wikimedia.org/wikipedia/commons/thumb/f/f2/LSM_Tree.png/1200px-LSM_Tree.png)

Εικόνα 13: Tarantool (https://avatars.githubusercontent.com/u/2344919?s=200&v=4)

Εικόνα 14: Apache Hbase (https://d1.awsstatic.com/product-marketing/EMR/hbase-logo.e139b77f7031062f738f0fc28210e0ffa6ca26c8.png)

Εικόνα 15: Cassandra (https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.ovhcloud.com%2Fen-gb%2Fpublic-cloud%2Fapache-cassandra%2F&psig=AOvVaw1NvyxHuma2x8A8i0dY1da0&ust=1671038981255000&source=images&cd=vfe&ved=0CBAQjRxqFwoTCIDj_LmP9_sCFQAAAAAdAAAAABAY)

Εικόνα 16: Scylla (https://www.scylladb.com/wp-content/uploads/twitter-default.jpg)

Εικόνα 17: Cloudata (scylla (https://miro.medium.com/max/512/0*GJuHQcsSPku7sJVU.jpg))

Εικόνα 18: CoucheDB (https://blog.desdelinux.net/wp-content/uploads/2020/03/CouchDB-logo-1.png.webp)

Εικόνα 19: MongoDB (https://webimages.mongodb.com/_com_assets/cms/kuzt9r42or1fxvlq2-Meta_Generic.png)

Εικόνα 20: ArangoDB (https://www.devopsschool.com/blog/wp-content/uploads/2022/03/arangodb-01.png)

Εικόνα 21: Neo4j (https://dist.neo4j.com/wp-content/uploads/20170726233003/hello-world-neo4j-inc-company-name-change.png)

Εικόνα 22: Titan (https://www.datanami.com/wp-content/uploads/2016/06/TitanDB.png)

Εικόνα 23: AllegroGraph (https://w7.pngwing.com/pngs/838/1015/png-transparent-allegrograph-franz-inc-nosql-graph-database-others-blue-text-logo.png)

Εικόνα 24: WhiteDB (https://dbdb.io/media/twitter/whitedb.png)

Εικόνα 25: Έρευνα Indeed (https://www.simplilearn.com/ice9/free_resources_article_thumb/job-trends-indeed.png

Εισαγωγή

Ζούμε σε έναν σύγχρονο κόσμο όπου η τεχνολογία αναπτύσσεται διαρκώς, με αποτέλεσμα να υπάρχει ανάγκη για συλλογή και ανάκτηση δεδομένων. Ο όγκος αυτός είναι τεράστιος, επομένως εμφανίστηκε η ανάγκη ανάπτυξης για τη σωστή διαχείριση αυτών. Δημιουργήθηκε, λοιπόν, το πιο διαδεδομένο μοντέλο των συστημάτων διαχείρισης βάσεων δεδομένων, το οποίο είναι το σχεσιακό μοντέλο (RDBMS).

Το έτος 1970, άρχισαν να διαδίδονται τα νέα περί μοντέλων σχεσιακών δεδομένων, με αποτέλεσμα ο ανταγωνισμός να αυξάνεται συνεχώς. Ο ερευνητής EF Codd, πρότεινε ένα σχήμα βάσης δεδομένων, το οποίο δεν ήταν και ιδιαίτερο συνηθισμένο για εκείνη την εποχή. Η δουλειά του Codd βασίστηκε σε μια ιδέα της κανονικοποίησης δεδομένων , η οποία εξοικονομούσε χώρο αρχείων σε μονάδες δίσκου αποθήκευσης σε μια εποχή που τέτοια μηχανήματα θα μπορούσαν να είναι απαγορευτικά ακριβά για τις επιχειρήσεις.

Τα συστήματα αυτά διαχείρισης βάσεν δεδομένων προηγήθηκαν και έτσι θα έλεγε κανείς πως εκείνη η εποχή, ήταν η εποχή του RDBMS. Ενώ τα RDBMS λειτουργούσαν επίσης σε κεντρικούς και μεγάλους υπολογιστές, παρείχε ένα χαρακτηριστικό παράδειγμα, το DB2 της IBM το οποίο χρησιμοποιούταν και σε μεσαίες εφαρμογές υπολογιστών UNIX. Το RDBMS ήταν μια μεγάλη εφεύρευση για την κατανεμημένη αρχιτεκτονική του υπολογισμού πελάτη-διακομιστή , καθώς συνέδεε δεξαμενές μεμονωμένων προσωπικών υπολογιστών με διακομιστές αρχείων και βάσεων δεδομένων.

Προέκυψαν αρκετοί ανταγωνιστές, όπως οι Oracle, Ingres, Informix, Sybase, Unify, Progress και διάφοροι άλλοι. Έτσι, κυκλοφόρησαν τρία RDBMS με έναν κοινό στόχο. Να κυριαρχήσουν στις εμπορικές υλοποιήσεις. Συγκεκριμένα, η Oracle, το DB2 -το οποίο υπήρχε ήδη- και ο SQL Server της Microsoft.

Η κατανεμημένη πληροφορική απέκτησε μεγαλύτερη απήχηση και καθώς η αρχιτεκτονική νέφους άρχισε να χρησιμοποιείται περισσότερο, τα RDBMS συναντούσαν τον ανταγωνισμό με τη μορφή συστημάτων NoSQL . Τέτοια συστήματα ήταν συχνά ειδικά σχεδιασμένα για μαζική διανομή και υψηλή επεκτασιμότητα στο cloud. Όμως, ακόμη και στα πιο διαφορετικά και πολύπλοκα συστήματα cloud, η ανάγκη για κάποια εγγυημένη συνέπεια δεδομένων απαιτεί τα RDBMS να εμφανίζονται με κάποιο τρόπο, ή σχήμα. Επιπλέον, οι εκδόσεις των RDBMS έχουν αναδιαρθρωθεί σημαντικά για αναπαραγωγή του cloud. **[65]**

Από πολύ παλιά, υπήρχαν πολλά και μεγάλα δεδομένα. Τα πρώτα σημάδια των μεγάλων δεδομένων, εμφανίζονται το έτος 1663, όταν ο John Graunt ασχολήθηκε με ποσότητες πληροφοριών ενώ παράλληλα μελετούσε μία πανώλη, που στοίχειωνε την Ευρώπη εκείνη την εποχή. Ο Graunt ήταν ο πρώτος άνθρωπος που χρησιμοποίησε στατιστική ανάλυση δεδομένων.

Το 1943, εμφανίζεται η πρώτη μηχανή επεξεργασίας δεδομένων και αναπτύχθηκε από τους Βρετανούς για να αποκρυπτογραφήσει τους ναζιστικούς κώδικες κατά τη διάρκεια του Β' Παγκοσμίου Πολέμου. Αυτή η συσκευή, που ονομάστηκε Colossus, αναζήτησε μοτίβα στα

υποκλαπέντα μηνύματα με ρυθμό 5.000 χαρακτήρων ανά δευτερόλεπτο, μειώνοντας το χρόνο που χρειάστηκε η εργασία από εβδομάδες σε ώρες.

Στη συνέχεια, το 1965, η κυβέρνηση των Ηνωμένων Πολιτειών αποφάσισε να κατασκευάσει το πρώτο κέντρο δεδομένων που θα αποθηκεύει πάνω από 742 εκατομμύρια φορολογικές δηλώσεις και 175 εκατομμύρια σετ δακτυλικών αποτυπωμάτων. Αποφάσισαν να το κάνουν αυτό μεταφέροντας αυτά τα αρχεία σε μαγνητική ταινία υπολογιστή που έπρεπε να αποθηκευτούν σε μία μόνο θέση. Το έργο αργότερα εγκαταλείφθηκε, αλλά είναι γενικά αποδεκτό ως η αρχή της εποχής της ηλεκτρονικής αποθήκευσης δεδομένων.

Αργότερα, στις αρχές του 1800, το πεδίο των στατιστικών επεκτάθηκε για να συμπεριλάβει τη συλλογή και την ανάλυση δεδομένων. Το έτος 1880, ο κόσμος είδε για πρώτη φορά το πρόβλημα που επικρατεί σχετικά με την μεγάλη ποσότητα δεδομένων. Από την Αμερική, ενημέρωσαν πως θα χρειαστούν περίπου οκτώ χρόνια για να χειριστούν και να επεξεργαστούν τα δεδομένα που συλλέχθηκαν κατά το πρόγραμμα απογραφής εκείνο το έτος.

Το 1881, από το γραφείο απογραφής, ο Herman Hollerith εφηύρε τη Μηχανή Πινακοποίησης Hollerith που μείωσε την εργασία υπολογισμού. Καθ' όλη τη διάρκεια του 20ου αιώνα, τα δεδομένα εξελίχθηκαν με απροσδόκητη ταχύτητα, με αποτέλεσμα, τα μεγάλα δεδομένα να γίνουν ο πυρήνας της εξέλιξης. Εκείνη την εποχή δημιουργήθηκαν επίσης μηχανές για την αποθήκευση πληροφοριών καθώς και υπολογιστές. Το 1965, η κυβέρνηση των ΗΠΑ κατασκεύασε το πρώτο κέντρο δεδομένων, με σκοπό να αποθηκεύσει εκατομμύρια σετ δακτυλικών αποτυπωμάτων και φορολογικές δηλώσεις.

Ωστόσο, πηγαίνοντας αρκετά χρόνια πίσω, βρίσκουμε πως τα παλαιότερα παραδείγματα που έχουμε από ανθρώπους που αποθηκεύουν και αναλύουν δεδομένα είναι τα ραβδιά καταμέτρησης, χρονολογούνται από το 18.000 π.Χ. Συγκεκριμένα, το πρώτο στοιχείο αποθήκευσης προϊστορικών δεδομένων, φαίνεται να είναι το κόκκαλο Ishango.

Επίσης, οι αρχαίοι Αιγύπτιοι γύρω στο 300 π.Χ. ήδη προσπάθησαν να συλλάβουν όλα τα υπάρχοντα «δεδομένα» στη βιβλιοθήκη της Αλεξάνδρειας. Επιπλέον, η Ρωμαϊκή Αυτοκρατορία συνήθιζε να αναλύει προσεκτικά τις στατιστικές του στρατού της για να καθορίσει τη βέλτιστη κατανομή για τους στρατούς της.

Στη συνέχεια, το 2400 π.Χ., ήρθε ο άβακας. Η πρώτη ειδική συσκευή που κατασκευάστηκε ειδικά για την εκτέλεση υπολογισμών. Οι πρώτες βιβλιοθήκες εμφανίστηκαν επίσης περίπου αυτή την εποχή, αντιπροσωπεύοντας τις πρώτες μας προσπάθειες για μαζική αποθήκευση δεδομένων.

Με το πέρασα του χρόνου όμως, έπρεπε να υπάρξει μία σωστή διαχείριση των δεδομένων αυτών, σε μικρό χρονικό διάστημα και έτσι έπρεπε να αξιοποιηθούν μοντέλα τα όποια δεν είναι σχεσιακά. Για τον λόγο αυτό, παρουσιάστηκαν τα NoSQL συστήματα διαχείρισης βάσεων δεδομένων.

Μεταξύ 1989 και 1990 ο Tim Berners-Lee και ο Robert Cailliau δημιούργησαν τον Παγκόσμιο Ιστό και ανέπτυξαν HTML, URL και HTTP, όλα αυτά ενώ εργάζονταν για το CERN. Η εποχή του Διαδικτύου με την ευρεία και εύκολη πρόσβαση στα δεδομένα είχε αρχίσει και μέχρι το

1996 η αποθήκευση ψηφιακών δεδομένων είχε γίνει πιο οικονομική από την αποθήκευση πληροφοριών σε χαρτί.

Το 1998, ο Carlo Strozzi ανέπτυξε τη NoSQL, μια σχεσιακή βάση δεδομένων ανοιχτού κώδικα που παρείχε έναν τρόπο αποθήκευσης και ανάκτησης δεδομένων μοντελοποιημένων διαφορετικά από τις παραδοσιακές μεθόδους πινάκων που βρίσκονται στις σχεσιακές βάσεις δεδομένων.

Το 2000, το Διαδίκτυο έχει προσφέρει μοναδικές συλλογές δεδομένων και ευκαιρίες ανάλυσης δεδομένων. Με την επέκταση της διαδικτυακής κίνησης και διαφόρων εταιριών, όπως το Yahoo, το Amazon και το eBay άρχισαν να αναλύουν τη συμπεριφορά των πελατών εξετάζοντας ποσοστά κλικ, και αρχεία καταγραφής αναζήτησης. Αυτό άνοιξε έναν εντελώς νέο κόσμο δυνατοτήτων.

Το 2005, τα Big Data χαρακτηρίστηκαν από τον Roger Mougalas καθώς αναφέρθηκε σε ένα μεγάλο σύνολο δεδομένων που, εκείνη την εποχή, ήταν σχεδόν αδύνατο να διαχειριστεί και να επεξεργαστεί χρησιμοποιώντας τα διαθέσιμα εργαλεία που υπήρχαν. Την ίδια χρονιά δημιουργήθηκε το Hadoop, το οποίο μπορούσε να χειριστεί Big Data. Το Hadoop βασίστηκε σε ένα πλαίσιο λογισμικού ανοιχτού κώδικα και συγχωνεύτηκε με το MapReduce.

Τα Big Data έφεραν επανάσταση ολόκληρες βιομηχανίες. Είναι αποτέλεσμα της εποχής της πληροφορίας και αλλάζει τον τρόπο με τον οποίο οι άνθρωποι εργάζονται.

Σκοπός της εργασίας

Σκοπός της εργασίας είναι να κάνει μία εισαγωγή στις NoSQL βάσεις δεδομένων. Θα γίνει, αρχικά, μία αναφορά στον ορισμό των «Μεγάλων Δεδομένων» και στα βασικά μίας βάσης δεδομένων. Τι είναι, ποιος ο ορισμός της δομής δεδομένων, ποιες είναι οι κατηγορίες μίας βάσης δεδομένων. Θα γίνει περιγραφή διαφόρων δυνατοτήτων καθώς και πλεονεκτημάτων/μειονεκτημάτων. Στη συνέχεια, θα γίνει εμβάθυνση στο θέμα, δηλαδή στα NoSQL συστήματα. Τέλος, θα γίνει αναφορά στις διάφορες κατηγορίες του παραπάνω ορισμού και αναλυτική επεξήγηση των συστημάτων αυτών.

Κεφάλαιο 1ο

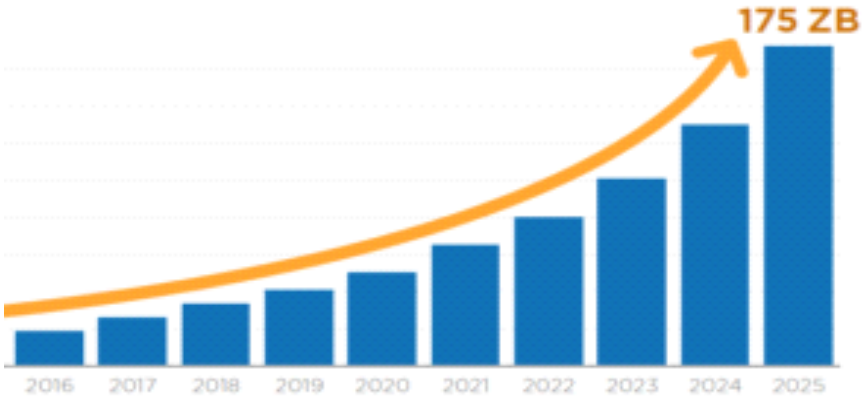
Big Data

1.1 Ορισμός

Τα Big Data είναι ένας όρος, ο οποίος αναφέρεται σε μεγάλου όγκου δεδομένα και τα οποία αυξάνονται διαρκώς, με γρήγορους ρυθμούς. Αυτό, έχει σαν αποτέλεσμα να χρειάζεται η δημιουργία νέων μοντέλων για την σωστή επεξεργασία και αποθήκευση τους. Το μέγεθος των Big Data αλλάζει συνεχώς. Παρατηρούμε πως από το 2012, κυμαίνονται από μερικές δεκάδες terabyte ως πολλά zettabyte. Για την αποκάλυψη πληροφοριών από σύνολα δεδομένων, επειδή είναι πολλά, διαφορετικά και περίπλοκα, τα Big Data, απαιτούν ένα σύνολο τεχνικών και τεχνολογιών με νέες μορφές ολοκλήρωσης. Όμως, οι τεχνικές και οι εφαρμογές που χρησιμοποιούνται σήμερα είναι ελλιπής, αφού αναφέραμε πως αναπτύσσονται συνεχώς. **[1] [2]**

Το 2020 εν καιρώ πανδημίας, η αύξηση των δεδομένων, δεν δείχνει σημάδια επιβράδυνσης. Η δημιουργία δεδομένων έκανε τεράστιο άλμα σύμφωνα με τα DataSphere και StorageSphere της IDC. Αναφέρθηκε πως, ο όγκος των ψηφιακών δεδομένων που θα δημιουργηθούν τα επόμενα πέντε χρόνια, θα είναι μεγαλύτερος από το διπλάσιο του όγκου των δεδομένων που δημιουργήθηκαν από την εμφάνιση της ψηφιακής αποθήκευσης.

Το παρακάτω γράφημα δείχνει ότι ο όγκος των πληροφοριών που δημιουργούνται παγκοσμίως αυξάνεται, αλλά και πως μέχρι το 2025 η IDC εκτιμάει πως 175 ZB θα δημιουργηθούν. **[3]**



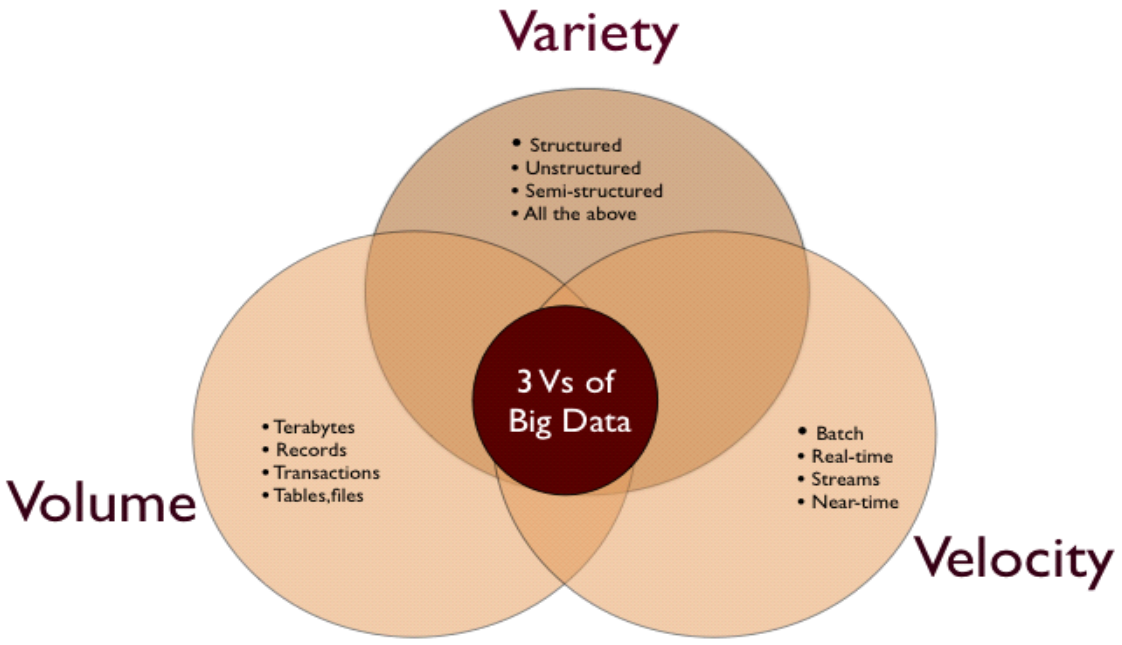
Εικόνα 1: Αύξηση των Big Data έως και το 1

1.2 Τα Vs των μεγάλων δεδομένων

Τα μεγάλα δεδομένα, είναι γνωστά για τον όγκο τους. Ένα περιβάλλον μεγάλων δεδομένων δεν χρειάζεται απαραίτητα να περιέχει μεγάλο όγκο δεδομένων. Για να περιγράψουμε καλύτερα και περισσότερο αποδοτικά τον όρο Big Data, χρησιμοποιήθηκαν τα "Variety", "Volume" και "Veracity". Όσον αφορά τα Vs των μεγάλων δεδομένων, για τα οποία θέλουμε να περιγράψουμε τις ιδιότητες ως προς τον όγκο, την ταχύτητα, την ποικιλία και την αξία, αναφερόμαστε ως «τρία Vs», «τέσσερα Vs» και «πέντε Vs».

Οι κατηγορίες των 3 Vs των Big Data περιλαμβάνουν μεγάλη ποικιλία δεδομένων και είναι:

- Variety: Μεγάλη ποικιλία δεδομένων
- Volume: Μεγάλος όγκος δεδομένων
- Velocity: Μεγάλη ταχύτητα όσον αφορά την ανάλυση των δεδομένων



Εικόνα 2: Γραφική αναπαράσταση των 3 Vs. 1

Υπάρχουν πολλοί τύποι δεδομένων οι οποίοι ίσως να χρειαστεί να αποθηκεύσουν και να διαχειριστούν συστήματα μεγάλων δεδομένων. Μπορούμε να αναφέρουμε πως ένα έργο ανάλυσης μεγάλων δεδομένων μπορεί να προβλέψει τις πωλήσεις ενός προϊόντος συγκρίνοντας δεδομένα προηγούμενων πωλήσεων, κλήσεις εξυπηρέτησης πελατών κα.

Η ταχύτητα αναφέρεται στην ταχύτητα όπου παράγονται τα δεδομένα και χρειάζεται να υποστούν επεξεργασία και ανάλυση. Η διαχείριση της ταχύτητας δεδομένων, είναι σημαντική αφού μπορεί να συμβάλλει στην μηχανική μάθηση και στην τεχνητή νοημοσύνη. **[4]**

1.3 Γιατί είναι σημαντικά τα Μεγάλα Δεδομένα;

Κάθε εταιρεία χρησιμοποιεί μεγάλα δεδομένα στα συστήματά τους έτσι, ώστε να βελτιώσουν οποιαδήποτε λειτουργία, να υπάρχει καλύτερη εξυπηρέτηση πελατών και οποιαδήποτε άλλη πράξη, η οποία μπορεί να αυξήσει τα έσοδα και τα κέρδη τους. Οποιαδήποτε επιχείρηση, τα χρησιμοποιεί ορθά, θα έχει και τα ανάλογα αποτελέσματα καθώς και θα έχουν ένα αποτελεσματικό πλεονέκτημα σε σχέση με τις υπόλοιπες εταιρίες, οι οποίες δεν τα έχουν αξιοποιήσει ακόμη.

Τα μεγάλα δεδομένα, έχουν πληροφορίες οι οποίες είναι σε μεγάλο βαθμό βοηθητικές καθώς μπορούν οι εταιρίες να τα χρησιμοποιήσουν και βελτιώσουν για παράδειγμα την διαφήμιση, τις προωθήσεις τους. Επίσης, μπορούν να χρησιμοποιηθούν και σε ιατρικά ζητήματα, για παράδειγμα στο να εντοπισθεί μία διάγνωση. Πολύ σημαντικό να αναφερθεί πως όλα τα ηλεκτρονικά αρχεία υγείας, οι ιστότοποι, τα μέσα κοινωνικής δικτύωσης κα, αποτελούν πληθώρα δεδομένα.

Ας αναφέρουμε κάποια παραδείγματα εταιρειών οι οποίες χρησιμοποιούν δεδομένα:

- Μεταφορικές εταιρείες: Για την σωστή διαχείριση της παράδοση των προϊόντων, οι οργανισμοί βασίζονται σε μεγάλα δεδομένα.
- Κλάδος ενέργειας: Τα μεγάλα δεδομένα βοηθούν κάθε εταίρεία με πετρέλαιο αλλά και φυσικό αέριο στο να εντοπίζονται πιθανές τοποθεσίες όπου μπορεί να πραγματοποιηθεί γεώτρηση ή και να παρακολουθούν τις λειτουργίες των αγωγών. Το ίδιο ισχύει και για τις επιχειρήσεις κοινής ωφέλειας το χρησιμοποιούν για την παρακολούθηση των ηλεκτρικών δικτύων.
- Οι εταιρείες χρηματοοικονομικών υπηρεσιών χρησιμοποιούν συστήματα μεγάλων δεδομένων για διαχείριση κινδύνου και ανάλυση δεδομένων αγοράς σε πραγματικό χρόνο.
- Ακόμη και οι κυβερνητικές χρήσεις περιλαμβάνουν την αντιμετώπιση καταστάσεων έκτακτης ανάγκης, την πρόληψη του εγκλήματος αλλά και τις πρωτοβουλίες έξυπνων πόλεων.

1.4 Παραδείγματα Μεγάλων Δεδομένων

Τα μεγάλα δεδομένα προέρχονται από διαφορετικές πηγές, όπως συστήματα επεξεργασίας συναλλαγών, βάσεις δεδομένων πελατών, έγγραφα, μηνύματα, ιατρικά αρχεία κα. Ακόμη και δεδομένα τα οποία δημιουργούνται από μηχανές, αρχεία καταγραφής δικτύου και διακομιστή και δεδομένα από αισθητήρες σε μηχανήματα κατασκευής, βιομηχανικό εξοπλισμό και συσκευές Διαδικτύου πραγμάτων.

Τα μεγάλα δεδομένα δεν αφορούν μόνο εσωτερικά συστήματα αλλά και εξωτερικά, όπως για παράδειγμα τις καιρικές συνθήκες, τις συνθήκες κυκλοφορίας, γεωγραφικές πληροφορίες κοκ. Πολλές εφαρμογές μεγάλων δεδομένων περιλαμβάνουν ροή δεδομένων όπου υποβάλλονται σε επεξεργασία και συλλέγονται συνεχώς.

1.5 Αποθήκευση και Επεξεργασία Μεγάλων Δεδομένων

Τι είναι η λίμνη δεδομένων; Ονομάζεται το "σημείο" όπου αποθηκεύονται τα μεγάλα δεδομένα. Στις αποθήκες δεδομένων υπάρχουν οι σχεσιακές βάσεις δεδομένων, ενώ στην λίμνη δεδομένων έχουμε τους διάφορους τύπους δεδομένων, όπως το Hadoop, υπηρεσίες αποθήκευσης cloud, βάσεις δεδομένων NoSQL και άλλες πλατφόρμες μεγάλων δεδομένων.

Πολλά περιβάλλοντα μεγάλων δεδομένων συνδυάζουν πολλαπλά συστήματα σε μια κατανεμημένη αρχιτεκτονική. Τα δεδομένα σε συστήματα μεγάλων δεδομένων μπορούν να παραμείνουν στην ακατέργαστη μορφή τους και στη συνέχεια να φιλτραριστούν και να οργανωθούν όπως χρειάζεται. Υπάρχει όμως και η περίπτωση να χρησιμοποιηθούν εργαλεία εξόρυξης δεδομένων καθώς και λογισμικό προετοιμασίας δεδομένων.

Για την επεξεργασία μεγάλων δεδομένων, είναι απαραίτητη η υπολογιστική ισχύς, αφού υπάρχουν μεγάλες απαιτήσεις. Τέτοιες τεχνολογίες είναι το Hadoop και η μηχανή επεξεργασίας Spark, οι οποίες θα αναφερθούν και θα αναπτυχθούν παρακάτω.

1.6 Η Άνοδος της SQL

Η International Business Machines Corporation (IBM), ανέπτυξε την γλώσσα SQL για να χειρίζεται σύνολα δεδομένων τα οποία είναι αποθηκευμένα σε RDBMS. Με την γλώσσα αυτή, η πρόσβαση και η τροποποίηση ήταν πιο γρήγορη χωρίς να γίνονται περίπλοκες οι εντολές που χρειαζόντουσαν. Θα μπορούσαμε να πούμε πως με ένα κλικ απλά, έχουμε πρόσβαση σε σύνολα δεδομένων. Σήμερα, οι περισσότεροι οργανισμοί εξακολουθούν να χρησιμοποιούν RDBMS με τον ένα ή τον άλλο τρόπο. **[7]**

Η SQL, αναπτύχθηκε από τους Donald D. Chamberlin και Raymond F. Boyce αφού έμαθαν για το σχεσιακό μοντέλου του Edgar F. Codd, στις αρχές του 1970. Η αρχική της ονομασία ήταν

SEQUEL και είχε ως στόχο να διαχειρίζεται και να ανακτά δεδομένα τα οποία είναι αποθηκευμένα στο αρχικό σύστημα διαχείρισης βάσεων δεδομένων. Ωστόσο, ήταν δύσκολο να χρησιμοποιηθεί. Το αρχικό όνομα SEQUEL στην πορεία άλλαξε, έμειναν μόνο τα σύμφωνα και μετονομάστηκε σε SQL, καθώς το SEQUEL υπήρχε ήδη ως ένα εμπορικό σήμα μίας υπάρχουσας εταιρίας.

Αφού δοκιμάστηκε η SQL σε διάφορους πελάτες για να προσδιορισθεί η χρησιμότητα και η πρακτικότητα του συστήματος, η IBM άρχισε να αναπτύσσει εμπορικά προϊόντα, τα οποία άρχισαν να κυκλοφορούν από το 1979 μέχρι και το 1983.

Στα τέλη του 1970, η Relational Software, Inc. -σήμερα η Oracle Corporation- είδε τις δυνατότητες των εννοιών από τους Codd, Chamberlin και Boyce και ανέπτυξε το δικό της RDBMS που βασίζεται, επίσης, σε SQL.

Μέχρι το 1986, υιοθετήθηκε επίσης ο ορισμός της γλώσσας SQL και από τότε μέχρι και το 2016, νέες εκδόσεις δημοσιεύθηκαν. **[73] [74]**

1.7 Γλώσσες Προγραμματισμού

Για την επεξεργασία και την ανάλυση των Μεγάλων Δεδομένων, χρησιμοποιούνται και αξιοποιούνται οι κατάλληλες γλώσσες προγραμματισμού. Η SQL, χωρίζεται σε δύο κατηγορίες. Την δηλωτική (declarative) και την διαδικαστική (procedural).

Με τον όρο δηλωτική, εννοούμε πως ο προγραμματιστής δηλώνει αυτό που θέλει και όχι τον τρόπο που θα το κάνει. Δηλαδή, το μόνο έχει να κάνει είναι να το δηλώσει στην βάση και η βάση θα «βρει» τον καλύτερο τρόπο για να λάβει το αποτέλεσμα. Αυτή είναι μια πολύ διαφορετική προσέγγιση από αυτή που μπορεί να έχει συνηθίσει ένας προγραμματιστής στη προσέγγιση στον προγραμματισμό. Με λίγα λόγια, παραχωρείται ο έλεγχος στη βάση δεδομένων. Αυτό λειτουργεί καλά, υπό την προϋπόθεση ότι οι προδιαγραφές της τελικής κατάστασης είναι σαφώς καθορισμένες και υπάρχει κατάλληλη διαδικασία υλοποίησης. Εάν πληρούνται και οι δύο αυτές προϋποθέσεις, ο δηλωτικός προγραμματισμός είναι πολύ αποτελεσματικός. **[66] [67] [68]**

Ποια όμως τα πλεονεκτήματα και τα μειονεκτήματα μίας δηλωτικής γλώσσας προγραμματισμού SQL;

Πλεονεκτήματα:

- Ο κώδικας είναι σύντομος και αποτελεσματικός
- Εύκολη βελτιστοποίηση καθώς η υλοποίηση ελέγχεται από έναν αλγόριθμο
- Δυνατότητα συντήρησης ανεξάρτητα από την ανάπτυξη εφαρμογών

Μειονεκτήματα:

- Μερικές φορές είναι δύσκολο να γίνει κατανοητό από εξωτερικούς ανθρώπους
- Βασισμένο σε ένα άγνωστο εννοιολογικό μοντέλο για τους ανθρώπους

Από την άλλη, η διαδικαστική SQL ή αλλιώς PL/SQL περιλαμβάνει συνθήκες και βρόχους. Γίνεται δήλωση σταθερών και μεταβλητών, συναρτήσεων. Χειρίζεται σφάλματα και υποστηρίζει πίνακες.

Ο σκοπός μιας συνάρτησης PL/SQL χρησιμοποιείται για τον υπολογισμό και την επιστροφή μιας μεμονωμένης τιμής. Αυτή η επιστρεφόμενη τιμή μπορεί να είναι μια τιμή όπως ένας αριθμός, μία ημερομηνία ή μια μεμονωμένη συλλογή στοιχείων/τιμών, δηλαδή πίνακες.

Ποια όμως τα πλεονεκτήματα και τα μειονεκτήματα μίας διαδικαστικής γλώσσας προγραμματισμού SQL;

Πλεονεκτήματα:

- Το ίδιο κομμάτι κώδικα χρησιμοποιείται επαναληπτικά, με αποτέλεσμα να υπάρχει υψηλή παραγωγικότητα.
- Η απόδοση είναι καλύτερη αλλά και γρήγορη. Αυτό συμβαίνει, γιατί οι διαδικασίες που αποθηκεύονται μεταγλωττίζονται μία και μοναδική φορά.
- Ο κώδικας που εκτελείται, αποθηκεύεται στην κρυφή μνήμη και έτσι οι απαιτήσεις της μνήμης μειώνονται.

Μειονεκτήματα:

- Ο εντοπισμός των σφαλμάτων καθίσταται δύσκολος και το να γίνει αποσφαλμάτωση είναι ακόμη πιο δύσκολη ή μπορεί και να μην γίνει καθόλου.
- Για την σύνταξη μίας καλύτερα αποθηκευμένης διαδικασίας, απαιτείται ακόμη ένας προγραμματιστής. Αυτό έχει σαν αποτέλεσμα μεγαλύτερο κόστος.

Υπάρχουν και άλλες γλώσσες προγραμματισμού, που συνδέονται με την βάσει. Τέτοιες είναι η Python, η Java, η R, η Scala.

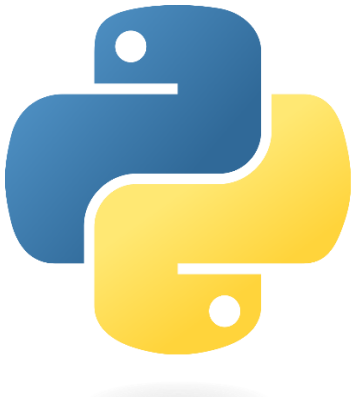
1. Python:

Η γλώσσα προγραμματισμού Python, δημιουργήθηκε το 1991 και είναι μία γλώσσα προγραμματισμού υψηλού επιπέδου. Υποστηρίζει αντικειμενοστρεφή και λειτουργικό προγραμματισμό. Θεωρείται μία σχετικά εύκολη γλώσσα η οποία διευκολύνει τους προγραμματιστές, καθώς μπορούν να εκφράσουν έννοιες σε μόλις λίγες γραμμές κώδικα. Διαθέτει πολλές βιβλιοθήκες και πολλές επίσης, συναρτήσεις.

Για την σύνδεση της με την βάση δεδομένων, μεταβιβάζουμε τις λεπτομέρειες της βάσεις δεδομένων όπως το όνομα του κεντρικού υπολογιστή, το όνομα χρήστη, τον κωδικό

πρόσβασης στην κλήση της μεθόδου και στην τέλος η μέθοδος επιστρέφει το αντικείμενο σύνδεσης.

Για να δημιουργηθεί η σύνδεση βάσης δεδομένων και Python, χρησιμοποιείται η μέθοδος connect(). Όπως επίσης, και το εργαλείο SQLAlchemy. Είναι μια βιβλιοθήκη που διευκολύνει την επικοινωνία μεταξύ προγραμμάτων Python και βάσεων δεδομένων. Τις περισσότερες φορές, αυτή η βιβλιοθήκη χρησιμοποιείται ως εργαλείο Object Relational Mapper (ORM) η οποία μεταφράζει κλάσεις Python σε πίνακες σε σχεσιακές βάσεις δεδομένων και μετατρέπει αυτόματα κλήσεις συναρτήσεων σε δηλώσεις SQL. **[70] [71]**



Εικόνα 3: Python 1

2. Java:

Η γλώσσα Java, είναι μία αντικειμενοστρεφής γλώσσα. Βασικό της χαρακτηριστικό είναι πως είναι ανεξάρτητη από το λειτουργικό σύστημα. Ότι πρόγραμμα είναι γραμμένο, δηλαδή, μπορεί να τρέξει σε οποιοδήποτε λειτουργικό σύστημα, είτε είναι Windows, Linux κοκ.

Η σύνδεση της Java με την βάση δεδομένων βοηθά στην αυτοματοποίηση της πρόσβασης στις βάσεις δεδομένων όπου ο προγραμματιστής έχει την δυνατότητα να χειριστεί απευθείας δεδομένα και να εργαστεί σε αυτά σε ένα αυτοματοποιημένο σενάριο. Με τη σύνδεση αυτή, επαληθεύονται τα αποτελέσματα, γίνεται να διαγραφούν

δεδομένα ακόμη και να ενημερωθούν συγκεκριμένα δεδομένα σύμφωνα με τις απαιτήσεις που έχει ο κάθε προγραμματιστής. **[71]**

Η Java μπορεί να συνδεθεί με την βάση δεδομένων μέσω της συνδεσιμότητα βάσεων δεδομένων Java Database Connectivity (JDBC). Είναι η προδιαγραφή Java μιας τυπικής διεπαφής προγραμματισμού εφαρμογών (API) που επιτρέπει στα προγράμματα Java να έχουν πρόσβαση σε συστήματα διαχείρισης βάσεων δεδομένων. Το JDBC API αποτελείται από ένα σύνολο διεπαφών και κλάσεων γραμμένων στη γλώσσα προγραμματισμού Java.

Εικόνα 4: Java 1

1.8 Τεχνικές Ανάλυσης Μεγάλων Δεδομένων

Όσο αναπτύσσονται οι τεχνολογίες, έχουν ως σκοπό να συγκεντρώνουν δεδομένα από πολλές και διαφορετικές πηγές και να γίνεται ανάλυση αυτών με τον πιο εύκολο και γρήγορο τρόπο. Οι τεχνικές και οι τεχνολογίες με το πέρασμα του χρόνου, αναπτύχθηκαν και αναπτύσσονται ακόμη, και έχουν ως σκοπό τους την συγκέντρωση των δεδομένων από διαφορετικές πηγές. Αυτές οι τεχνικές αναπτύχθηκαν από διάφορους κλάδους όπως αυτούς της στατιστικής, της επιστήμης των υπολογιστών και διάφορων άλλων.

Πλέον τα εργαλεία για την ανάλυση αυτών, είναι πολλά. Κάποια από αυτά είναι:

1. Clustering

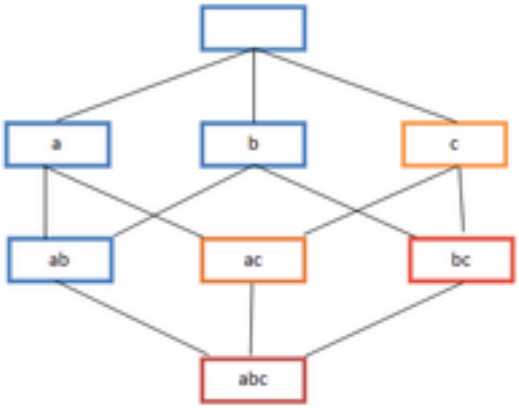
Όταν υπάρχει ένας τεράστιος όγκος δεδομένων, χρησιμοποιείται η παραπάνω τεχνική. Με βάση τα κοινά χαρακτηριστικά που υπάρχουν και τα κοινά γνωρίσματα, τα

ομαδοποιούμε κατάλληλα. Χρησιμοποιείται για διάφορους τομείς, όπως η αναγνώριση προτύπων , η ανάλυση εικόνας , η ανάκτηση πληροφοριών , η βιοπληροφορική , η συμπίεση δεδομένων , τα γραφικά υπολογιστών και η μηχανική μάθηση. **[61]**

Εικόνα 5: Clustering 1

2. Association Rule Learning

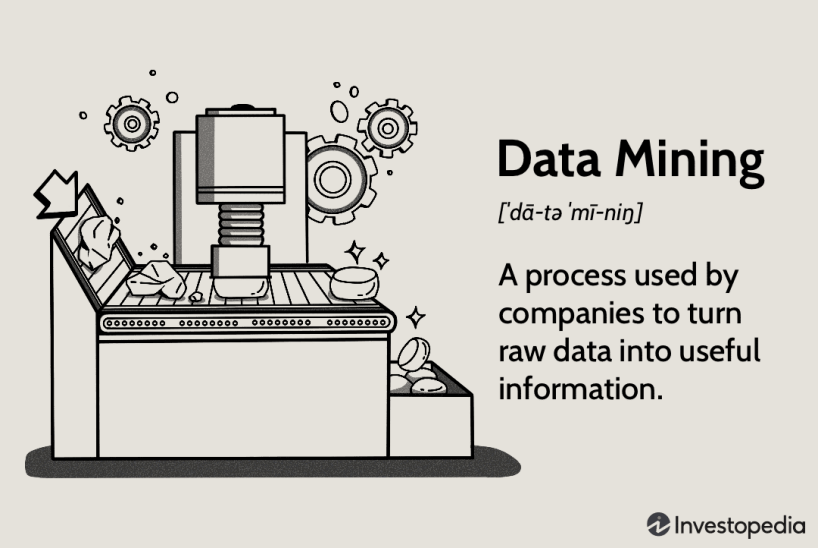
Με την τεχνική αυτή, μπορούμε να «ανακαλύψουμε» ζευγάρια μεταβλητών που σχετίζονται με κάποιο αποτέλεσμα. Γίνονται διάφορα τεστ, μέχρι να έχουμε το επιθυμητό αποτέλεσμα. Με αυτόν τον τρόπο, διώχνουμε τις μεταβλητές που δεν μας ενδιαφέρουν έτσι ώστε, να μην επηρεάζεται η ευστοχία της πρόβλεψης του αλγορίθμου. **[62]**



Εικόνα 6: Association Rule Learning 1

3. Data Mining

Η τεχνική αυτή, περιλαμβάνει μια σειρά από στατιστικά στοιχεία καθώς και την δυνατότητα να μάθει στον υπολογιστή να ξεχωρίζει μοτίβα μελετώντας datasets τα οποία υπάρχουν ήδη, και μετά την ανάλυση, μας παρέχει με πληροφορίες που θα ήταν αδύνατον να βρούνε οι εργαζόμενοι μόνοι τους χωρίς την ανάλυση. Το Data Mining, μας δίνει την δυνατότητα αυτά τα μοτίβα να προκύπτουν από ανάλυση ογκωδέστατων datasets που μεγαλώνουν καθημερινά. **[63]**



Εικόνα 7: Data Mining 1

Κεφάλαιο 2ο

Βάση Δεδομένων

2.1 Ορισμός

Μία βάση δεδομένων (database), είναι μία οργανωμένη συλλογή δεδομένων τα οποία αποθηκεύονται ηλεκτρονικά Οι μικρές βάσεις δεδομένων αποθηκεύονται σε συστήματα αρχείων, ενώ οι μεγάλες σε συμπλέγματα υπολογιστών. Με τον σχεδιασμό των βάσεων δεδομένων, έχουμε αποτελεσματική αποθήκευση δεδομένων όσον αφορά την ασφάλεια και το απόρρητο των ευαίσθητων δεδομένων καθώς και την ανοχή σφαλμάτων.

Το γνωστό σύστημα διαχείρισης βάσεων δεδομένων (DBMS), είναι ένα λογισμικό που αλληλεπιδρά με τους χρήστες και την βάση δεδομένων για την συλλογή των δεδομένων. Το σύστημα αυτό, είναι ένα σχεσιακό μοντέλο το οποίο κυριάρχησε την δεκαετία του 1980. Γνωστό και ως SQL. Ωστόσο, την δεκαετία του 2000, δημιουργήθηκε ένα μη σχεσιακό μοντέλο το οποίο ήταν και είναι περισσότερο αποτελεσματικό ως προς τα δεδομένα, το NoSQL, αφού χρησιμοποιεί διαφορετικές γλώσσες ερωτημάτων.

Τα τελευταία χρόνια, υπάρχει μεγάλη αύξηση όσον αφορά τα Big Data (Terabytes/Petabytes), με αποτέλεσμα να δημιουργούνται αρκετά προβλήματα σχετικά με την αποθήκευσή τους, αλλά και οι βάσεις δεδομένων να μην μπορούν να υποστηρίξουν τα μεγάλα όγκου δεδομένα. Για τον λόγο αυτόν, ιδρύθηκαν νέες βάσεις δεδομένων, οι οποίες δεν θα δυσκολεύονται στο να αποθηκεύουν τις μεγάλες αυτές ποσότητες. **[5]**

Εικόνα 8: Big Data 1

2.2 Δομή Δεδομένων

Οι βάσεις δεδομένων SQL βασίζονται σε πίνακες, ενώ οι βάσεις δεδομένων NoSQL είναι καταστήματα εγγράφων, κλειδιών-τιμών, γραφημάτων ή ευρείας στήλης.

Μερικά παραδείγματα βάσεων δεδομένων SQL περιλαμβάνουν MySQL , Oracle , PostgreSQL και Microsoft SQL Server . Τα παραδείγματα βάσεων δεδομένων NoSQL περιλαμβάνουν MongoDB , BigTable, Redis, RavenDB Cassandra, HBase, Neo4j και CouchDB.

2.3 Σύστημα Διαχείρισης Βάσεων Δεδομένων

Ένα Σύστημα Διαχείρισης Βάσεων Δεδομένων (DBMS) είναι ένα σύστημα διαχείρισης το οποίο υλοποιείται σε έναν υπολογιστή με πολλούς επεξεργαστές. Με την έννοια παράλληλο DBMS, έχουμε την διανομή του οριζόντιου κατακερματισμού ως προς έναν μεγάλο σχεσιακό πίνακα με πολλούς κόμβους, όπου μπορεί να γίνει οποιαδήποτε επεξεργασία, οποιαδήποτε στιγμή. Αυτό πραγματοποιείται με εκτέλεση βασικών εντολών SQL, σε οποιονδήποτε κόμβο. Υπάρχουν διαφορετικές αρχιτεκτονικές πολλών παράλληλων συστημάτων, όπου και η κάθε μία έχει τα θετικά της αλλά και τα αρνητικά της. Τα συστήματα παράλληλων βάσεων δεδομένων καταφέρνουν και βελτιώνουν τις επιδόσεις επεξεργασίας δεδομένων. Τα δεδομένα, στα κατανεμημένα συστήματα βάσεων αποθηκεύονται σε διαφορετικά συστήματα διαχείρισης βάσεων δεδομένων, τα οποία λειτουργούν εντελώς ανεξάρτητα.

Για να πραγματοποιείται συλλογή και αποθήκευση πληροφοριών, δημιουργήθηκαν διάφορα μοντέλα διαχείρισης βάσεων δεδομένων, τα οποία με το πέρασμα του χρόνου βελτιώθηκαν.

Τα κυριότερα μοντέλα βάσεων δεδομένων είναι:

1. Το σχεσιακό (relational model)
2. Το αντικειμενοστραφές (object – oriented model)
3. Το αντικειμενοσχεσιακό (object relational model)
4. Το δικτυακό (network model)
5. Το ιεραρχικό (hierarchical model)
6. Το NoSQL (not only relational)

Το **σχεσιακό μοντέλο**, αντιπροσωπεύει όλα τα δεδομένα σε πλοιάδες, ομαδοποιημένες σε σχέσεις. Περιγράφηκε για πρώτη φορά το 1969 και δεν υπάρχεο ιεραρχία. Για την αναζήτηση των δεδομένων, χρησιμοποιείται το περιεχόμενο των πινάκων χωρίς να χρειάζεται να ακολουθηθεί κάποια δενδρική διαδρομή.

Το **αντικειμενοστραφές μοντέλο**, χρησιμοποιείται για την κατασκευή αντικειμένων χρησιμοποιώντας μια συλλογή αντικειμένων που περιέχουν αποθηκευμένες τιμές των μεταβλητών που βρίσκονται μέσα σε ένα αντικείμενο. Η κάθε πληροφορία θεωρείται ως ανεξάρτητο αντικείμενο. Είναι, επίσης, σημαντικό να σημειωθεί ότι αποτελεί ένα αρκετά αποτελεσματικό μοντέλο σε εφαρμογές που υπάρχουν πολύπλοκες και δυναμικές δομές δεδομένων.

Το **αντικειμενοσχεσιακό μοντέλο**, είναι μία τεχνική προγραμματισμού, η οποία χρησιμοποιείται για να μετατραπούν δεδομένα μεταξύ συστημάτων τύπων, χρησιμοποιώντας αντικειμενοστραφείς γλώσσες προγραμματισμού. Μπορεί να είναι εύκολο για την χρήση του, αλλά η ταχύτητα απόκρισης τους είναι μικρή, κυρίως σε πολύπλοκες δομές δεδομένων.

Το **δικτυακό μοντέλο** έχει σχεδιαστεί, έτσι ώστε να είναι ευέλικτος ο τρόπος αναπαράστασης αντικειμένων και σχέσων. Δεν ξεκινάει υποχρεωτικά από την ρίζα (root).

Το **μοντέλο ιεραρχικής** βάσης δεδομένων, είναι ένα μοντέλο δεδομένων όπου τα δεδομένα είναι οργανωμένα σε μία δομή, η οποία μοιάζει με δέντρο. Έχουμε τις εγγραφές, ως δεδομένα όπου και συνδέονται μεταξύ τους, μέσω συνδέσμων. Ο τύπος μιας εγγραφής καθορίζει ποια πεδία περιέχει η εγγραφή. Μια εγγραφή είναι μια συλλογή πεδίων, με κάθε πεδίο να περιέχει μόνο μία τιμή. Για να ανακτηθούν δεδομένα από μια ιεραρχική βάση δεδομένων, πρέπει να διασχιστεί ολόκληρο το δέντρο ξεκινώντας από τον ριζικό κόμβο (root). Αυτό το μοντέλο αναγνωρίζεται ως το πρώτο μοντέλο βάσης δεδομένων που δημιουργήθηκε από την IBM το 1960.

Το **NoSQL μοντέλο** βάσεων δεδομένων, είναι ένα βασικό και χρήσιμο μοντέλο. Θα γίνει αναφορά παρακάτω. **[6]**

2.4 Θεώρημα CAP

Το θεώρημα CAP, που ονομάζεται επίσης και θεώρημα Brewer, είναι ένα θεμελιώδες θεώρημα στο πεδίο του σχεδιασμού συστημάτων. Παρουσιάστηκε για πρώτη φορά το 2000 από τον Eric

Brewer, κατά τη διάρκεια μιας ομιλίας για τις αρχές του κατανεμημένου υπολογισμού. Το 2002, οι καθηγητές δημοσίευσαν μια απόδειξη από το σκεπτικό του Brewer. Το θεώρημα αυτό, σημαίνει ότι κάθε κατανεμημένο κατάστημα δεδομένων, μπορεί να παρέχει μόνο δύο από τα ακόλουθα ή σε περίπτωση αποτυχίας δικτύου , πρέπει να γίνει μια επιλογή:

- Συνοχή (Consistency)
- Διαθεσιμότητα (Availability)
- Ανοχή κατάτμησης (Partition Tolerance)

Ένα κατανεμημένο σύστημα είναι μια συλλογή υπολογιστών που συνεργάζονται για να σχηματίσουν έναν ενιαίο υπολογιστή για τους τελικούς χρήστες. Όλα τα κατανεμημένα μηχανήματα λειτουργούν ταυτόχρονα. Οι χρήστες πρέπει να μπορούν να επικοινωνούν με οποιοδήποτε από τα κατανεμημένα μηχανήματα χωρίς να γνωρίζουν ότι είναι μόνο ένα μηχάνημα. Το δίκτυο κατανεμημένου συστήματος αποθηκεύει τα δεδομένα του σε περισσότερους από έναν μόνο κόμβο, χρησιμοποιώντας πολλαπλές φυσικές ή εικονικές μηχανές ταυτόχρονα.

Οι μικροϋπηρεσίες ορίζονται ως χαλαρά συνδεδεμένες υπηρεσίες που μπορούν να αναπτυχθούν, και να διατηρηθούν ανεξάρτητα. Περιλαμβάνουν τη δική τους στοίβα, βάση δεδομένων και επικοινωνούν μεταξύ τους μέσω ενός δικτύου. Οι μικροϋπηρεσίες έχουν γίνει ιδιαίτερα δημοφιλείς σε περιβάλλοντα cloud και χρησιμοποιούνται επίσης ευρέως σε κέντρα δεδομένων εσωτερικού χώρου.

Τι σημαίνουν όμως οι έννοιες της συνοχής, της συνέπεια και της ανοχής κατάτμησης;

Η **συνοχή** λαμβάνει την πιο πρόσφατη απάντηση ή ένα σφάλμα.

Η **διαθεσιμότητα** αφορά το κάθε αίτημα που λαμβάνει μία απάντηση.

Με την **ανοχή κατάτμησης**, το σύστημα συνεχίζει να λειτουργεί παρόλο που κάποια μηνύματα απορρίπτονται ή καθυστερούν.

Μπορούμε να έχουμε τους εξής τρεις διαφορετικούς συνδυασμούς:

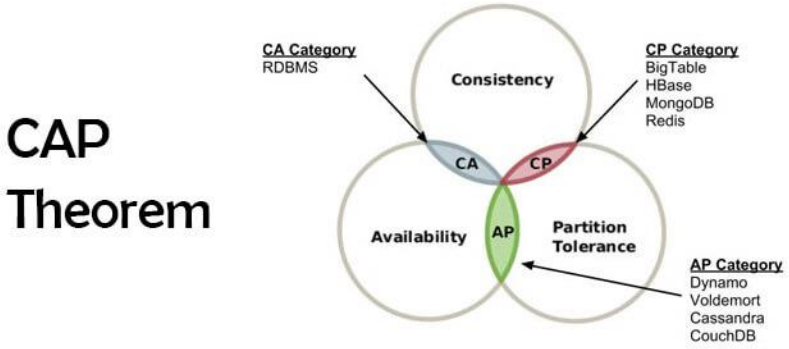
1. **CA:** Υπάρχει συνέπεια, διαθεσιμότητα αλλά όχι διαμοιρασμός. Με αυτόν τον τρόπο, εφόσον οι κόμβοι είναι online, τα δεδομένα διανέμονται με επιτυχία. Σε κάθε άλλη περίπτωση, τα δεδομένα δεν θα είναι ενημερωμένα.

2. **CP:** Υπάρχει συνέπεια και ανοχή στον διαμοιρασμό. Σε περίπτωση βλάβης, δεν υπάρχει συνεχής πρόσβαση στα δεδομένα.

3. **AP:** Υπάρχει ανοχή, διαθεσιμότητα στον διαμοιρασμό δεδομένων αλλά τα δεδομένα μπορεί και να μην είναι ενημερωμένα. Οι κόμβοι είναι online ακόμη και στην περίπτωση όπου δεν μπορούν να επικοινωνήσουν μεταξύ τους. Τα δεδομένα τους θα συγχρονίσουν όταν

ολοκληρωθεί ο διαμοιρασμός. Ωστόσο, το αποτέλεσμα είναι πως δεν είναι γνωστό εάν όλοι οι κόμβοι θα έχουν τα ίδια δεδομένα. **[8]**

Ωστόσο, υπάρχουν και κάποιες αρνητικές κριτικές, όσον αφορά το θεώρημα αυτό. Δεν καθορίζει ένα ανώτερο όριο για το χρόνο απόκρισης για διαθεσιμότητα, στην πράξη, υπάρχει ένα χρονικό όριο. Το θεώρημα CAP αγνοεί τον λανθάνοντα χρόνο , κάτι που είναι σημαντικό στην πράξη. Τα χρονικά όρια συχνά εφαρμόζονται στις υπηρεσίες. Κατά τη διάρκεια μιας κατάτμησης, εάν ακυρώσουμε ένα αίτημα, διατηρούμε τη συνέπεια αλλά χάνουμε τη διαθεσιμότητα.



Εικόνα 9: Θεώρημα CAP στην Κασσάνδρα 1

2.5 MapReduce

Σε ένα παράλληλο, κατανεμημένο αλγόριθμο, το MapRecude αποτελεί βοήθεια ως προς την επεξεργασία και την δημιουργία μεγάλων δεδομένων, καθώς είναι ένα μοντέλο προγραμματισμού. Είναι κατάλλο για την ανάπτυξη εφαρμογών, οι οποίες επεξεργάζονται μεγάλες ποσότητες δεδομένων σε συστοιχίες υπολογιστών (Clusters).

Είναι ένα πρότυπο προγραμματισμού που επιτρέπει τεράστια επεκτασιμότητα σε εκατοντάδες ή χιλιάδες διακομιστές σε ένα σύμπλεγμα Hadoop. Ως στοιχείο επεξεργασίας, το MapReduce είναι η καρδιά του Apache Hadoop. Υπάρχουν αρκετές βιβλιοθήκες, βέβαια, οι οποίες έχουν γραφτεί σε πολλές και διάφορες γλώσσες προγραμματισμού η κάθε μία με διαφορετικά επίπεδα βελτιστοποίησης αλλά η πιο δημοφιλή υλοποίηση ανοιχτού κώδικα που υποστηρίζει κατανεμημένες αναπαραστάσεις, θα έλεγε κανείς πως είναι το Apache Hadoop.

Ο όρος αυτός, αναφέρεται σε δύο ξεχωριστές εργασίες που εκτελούν τα προγράμματα Hadoop. Η πρώτη είναι η εργασία χάρτη, η οποία παίρνει ένα σύνολο δεδομένων και το μετατρέπει σε ένα άλλο σύνολο δεδομένων, όπου τα μεμονωμένα στοιχεία αναλύονται σε πλειάδες (ζεύγη κλειδιών/τιμών).

Πιο αναλυτικά, δηλαδή, μία συνάρτηση είναι το Map, η οποία χρησιμοποιεί ως αρχική διαδικασία ζεύγη κλειδιού - τιμής. Η δεύτερη συνάρτηση Reduce, συγχωνεύει όλες τις

ενδιάμεσες τιμές από το ίδιο κλειδί. Το σύνολο των δεδομένων, χωρίζεται σε υποσύνολα τα οποία επεξεργάζονται. Η συνάρτηση αυτή, ομαδοποιεί τα υποσύνολα που συνδέονται με το ενδιάμεσο κλειδί και τα στέλνει στην συνάρτηση Reduce. Η τελευταία συνάρτηση, δέχεται ένα ενδιάμεσα κλειδί και υποσύνολα τα οποία σχετίζονται με το κλειδί αυτό. Κανονικά, πρέπει να παραχθεί καμία ή μία τιμή εξόδου. **[9]**

Επεξεργάζεται παραλληλιζόμενα προβλήματα σε μεγάλα σύνολα δεδομένων, χρησιμοποιώντας κόμβους, οι οποίοι αναφέρονται ως σύμπλεγμα εάν βρίσκονται στο ίδιο τοπικό δίκτυο και χρησιμοποιούν παρόμοιο υλικό, ή πλέγμα εάν οι κόμβοι είναι μοιράζονται σε γεωγραφικά και διοικητικά κατανεμημένα συστήματα και χρησιμοποιούν διαφορετικό υλικό)

Ποια τα πλεονεκτήματά του όμως;

- Εύκολη χρήση
- Τα δεδομένα αποθηκεύονται σε αρχεία απλού κειμένου
- Υπάρχει ανεξαρτησία αποθήκευσης
- Έχει ανεκτικότητα σε τυχόν σφάλματα

2.6 Τι είναι η SQL injection και πως αντιμετωπίζεται

Με την «εισβολή», μπορούν να εξαγάγουν δεδομένα από την βάση δεδομένων αλλά και να εκτελέσουν κώδικα στα πλαίσια όπου γίνεται κάποια εφαρμογή, οποιαδήποτε στιγμή. Πραγματοποιούνται, δηλαδή, επιθέσεις με αποτέλεσμα να παραβιάσουν, για παράδειγμα, λογαριασμούς από χρήστες διαχείρισης με αποτέλεσμα να πάρουν κάθε έλεγχο. Τέτοιες επιθέσεις είναι ιδιαίτερα επικίνδυνες, καθώς τα καταστήματα δεδομένων NoSQL είναι συχνά μια καινοτομία για προγραμματιστές που είναι εξοικειωμένοι μόνο με προϊόντα σχεσιακής βάσης δεδομένων, γεγονός που αυξάνει τον κίνδυνο μη ασφαλούς κώδικα.

Λόγω του ότι πολλά «προϊόντα» του συστήματος NoSQL αναπτύσσονται ακόμη, έχει ως αποτέλεσμα να είναι αρκετά ανασφαλείς και έτσι να υπήρξαν αρκετές επιθέσεις injection. Όμως, με το πέρασμα του χρόνου, οι επόμενες εκδόσεις έχουν μία πολύ καλύτερα και μεγαλύτερη ασφάλεια.

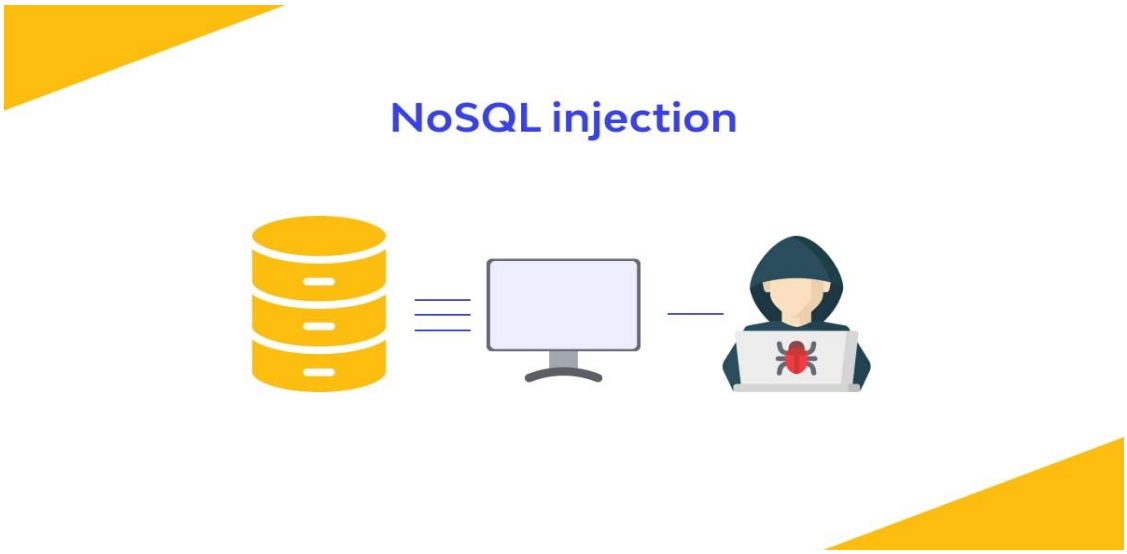
Πως αντιμετωπίζεται όμως μία επίθεση Injection; Ο καλύτερος τρόπος, είναι να αποφεύγεται η χρήση μη εξυγιαντικών εισόδων χρήστη στον κώδικα εφαρμογής, ειδικά κατά τη δημιουργία ερωτημάτων βάσης δεδομένων. Το MongoDB (θα αναφερθεί παρακάτω), για παράδειγμα, έχει ενσωματωμένες δυνατότητες για ασφαλή δημιουργία ερωτημάτων χωρίς JavaScript. Εάν χρειαστεί κάποιος να χρησιμοποιήσει την γλώσσα προγραμματισμού JavaScript σε ερωτήματα, θα ήταν χρήσιμο να επικυρώνεται και να κωδικοποιούνται σωστά όλες οι εισαγωγές χρήστη, εφαρμόζοντας τον κανόνα των ελάχιστων προνομίων. Επίσης, η καλή γνώση της γλώσσας θα συμβάλλει για την αποφυγή χρήσης ευάλωτων δομών.

Επιπλέον, μία εφαρμογή NoSQL injection είναι η επίθεση σε εφαρμογές web που έχουν δημιουργηθεί στη στοίβα MEAN (MongoDB, Express, Angular και Node). Κατά τη διαβίβαση

δεδομένων, οι εφαρμογές MEAN χρησιμοποιούν το JSON, το οποίο είναι το ίδιο πράγμα που χρησιμοποιείται από το MongoDB.

Με τον κώδικα JSON σε μια εφαρμογή MEAN μπορεί να επιτρέψει επιθέσεις injection σε μια βάση δεδομένων MongoDB.

Οι επιθέσεις injection NoSQL είναι αρκετά παρόμοιες με τις επιθέσεις injection SQL. Εκμεταλλεύονται την κακή εξυγίανση της εισόδου χρήστη κατά τη δημιουργία ερωτημάτων βάσης δεδομένων. Αυτό σημαίνει ότι τα ίδια εργαλεία για την προστασία από επιθέσεις injection SQL λειτουργούν επίσης για το NoSQL. **[13] [14] [15]**



Εικόνα 10: NoSQL Injection 1

Κεφάλαιο 3ο

Συστήματα NoSQL

3.1 Ορισμός

Με το πέρασμα των χρόνων, αναπτύσσονται όλο και περισσότερες εφαρμογές οι οποίες έχουν μεγάλες απαιτήσεις όσον αφορά τον αποθηκευτικό χώρο.

Το σύστημα NoSQL, είναι μία βάση δεδομένων η οποία παρέχει έναν μηχανισμό για να αποθηκεύνται και να ανακτώνται δεδομένα που μοντελοποιούνται από διάφορα μέσα τα οποία χρησιμοποιούνται στις σχεσιακές βάσεις δεδομένων. Αυτές οι βάσεις, είναι οι NoSQL. Χρησιμοποιούνται με έναν διαφορετικό τρόπο διαχείρισης των δεδομένων σε σχέση με μία σχεσιακή βάση δεδομένων.

Τα δεδομένα ενός συστήματος NoSQL, δεν έχουν δομημένη αρχιτεκτονική, με αποτέλεσμα να χρησιμοποιούν αποκλειστικά non - relational τρόπους οργάνωσης αλλά και ανάλυσης δεδομένων. **[10] [11]**

3.2 Σύντομη ιστορική Αναδρομή

Για αρκετά χρόνια, πολλές επιχειρήσεις χρησιμοποιούσαν συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS) για την αποθήκευση, επεξεργασία και ανάλυση κρίσιμων επιχειρηματικών πληροφοριών. Με την εμφάνιση των μεγάλων δεδομένων χρειάστηκε ένα νέο είδος βάσης δεδομένων. Οι σχεσιακές βάσεις δεδομένων, απαιτούσαν τεράστια ποσότητα συντήρησης. Για την σωστή διαχείριση του όγκου αυτού, έπρεπε να δημιουργηθεί μία μη σχεσιακή βάση δεδομένων. Η ιδέα αυτή, ήρθε το 1970 από τον Edgar Codd, έναν επιστήμονα υπολογιστών. Είχε σαν σκέψη να αρχειοθετήσει πληροφορίες σε πίνακες που περιέχουν σειρές και στήλες. Το σχέδιο αυτό είχε σαν αποτέλεσμα να γίνει ένα τεράστιο άλμα ως προς τα επίπεδα συστημάτων αρχείων που χρησιμοποιούσαν οι διάφοροι οργανισμοί. Τα RBDMS είναι εξαιρετικά στη διαχείριση φόρτου εργασίας συναλλαγών, τα οποία χρησιμοποιούνται πολύ σήμερα. **[10]**

3.3 NoSQL vs SQL

Όσο η SQL άλλο τόσο και η NoSQL αποτελούν σύγχρονες βάσεις δεδομένων. Ωστόσο, υπάρχουν βασικές διαφορές μεταξύ τους. **[16] [17]**

Οι πέντε πιο βασικές και κρίσιμες διαφορές είναι οι εξής:

1. Οι βάσεις δεδομένων SQL είναι σχεσιακές, σε αντίθεση με τις βάσεις δεδομένων NoSQL, οι οποίες είναι μη σχεσιακές.

2. Οι βάσεις δεδομένων SQL χρησιμοποιούν δομημένη γλώσσα ερωτημάτων και έχουν ένα προκαθορισμένο σχήμα. Οι βάσεις δεδομένων NoSQL έχουν δυναμικά σχήματα για μη δομημένα δεδομένα.
3. Οι βάσεις δεδομένων SQL μπορούν να κλιμακωθούν κατακόρυφα, ενώ οι βάσεις δεδομένων NoSQL μπορούν να κλιμακωθούν οριζόντια.
4. Οι βάσεις δεδομένων SQL βασίζονται σε πίνακες, ενώ οι βάσεις δεδομένων NoSQL είναι καταστήματα εγγράφων, κλειδιών-τιμών, γραφημάτων ή ευρείας στήλης.
5. Οι βάσεις δεδομένων SQL είναι καλύτερες για συναλλαγές πολλών σειρών, ενώ το NoSQL είναι καλύτερες για μη δομημένα δεδομένα όπως έγγραφα ή JSON.

3.4 Γιατί NoSQL;

Η χρήση σχεσιακών βάσεων δεδομένων, οδηγεί πολλές φορές σε προβλήματα που αφορούν την μοντελοποίηση δεδομένων καθώς και υπάρχουν μεγάλοι περιορισμοί σε ορισμένους servers που περιέχουν μεγάλες ποσότητες δεδομένων.

Πρέπει να αναφερθεί πως τα NoSQL συστήματα δεν ήρθαν για να αντικαταστήσουν τα σχεσιακά μοντέλα, αλλά για να τα συμπληρώσουν. Τα NoSQL συστήματα, είναι χρήσιμα σε μεγάλο βαθμό για όποιον εργάζεται με μεγάλη ποσότητα δεδομένων. Είναι, επίσης, χρήσιμα για ανάκτηση και ανταλλαγή δεδομένων μεταξύ μηχανημάτων καθώς και για την επεξεργασία συναλλαγών μεγάλου όγκου.

Η NoSQL, καθιστά «φτηνή» την αποθήκευση δεδομένων, όπως ιστορικά δεδομένα, αρχεία ηλεκτρονικού ταχυδρομείου, αρχεία καταγραφής και άλλα, καθώς μία σχεσιακή βάση δεδομένων, έχει μεγάλη ποσότητα δεδομένων, πολλούς πίνακες με αποτέλεσμα η απόδοση του συστήματος να μην είναι ίδια, με την αρχική. Παρόλο που τα NoSQL συστήματα έχουν πιο αδύναμα μοντέλα, μπορούν πιο εύκολα να θυσιάσουν την συνοχή για να έχουν μεγαλύτερη απόδοση. **[54]**

Με το πέρασμα του χρόνου όλο και περισσότερες απαιτήσεις δημιουργήθηκαν. Συγκεκριμένα:

- Απόδοση:

  Όσο το δυνατόν πιο χαμηλές οι καθυστερήσεις.

- Διαθεσιμότητα:

  Να είναι πάντα διαθέσιμη, όση εργασία και να υπάρχει αλλά ακόμη και στην περίπτωση όπου έχουν «πέσει» κάποιοι serves, τα δεδομένα να είναι διαθέσιμα.

- Κλιμάκωση:

  Να μπορεί να προστεθούν καινούργιοι servers, στην περίπτωση όπου υπάρχουν δεδομένα μεγάλου όγκου.

3.5 Πλεονεκτήματα NoSQL

Όπως όλες οι τεχνολογίες, έτσι και οι βάσεις δεδομένων NoSQL, προσφέρουν κάποια οφέλη αλλά ταυτόχρονα έχουν κάποιους περιορισμούς. Σε μια εποχή όπου οι σχεσιακές βάσεις δεδομένων χρησιμοποιούνται κυρίως για αποθήκευση και ανάκτηση δεδομένων, οι σύγχρονες τεχνολογίες Ιστού αποτελούσαν μια σημαντική πρόκληση με τη μορφή μη δομημένων δεδομένων. Οι σχεσιακές βάσεις δεδομένων δυσκολεύτηκαν ιδιαίτερα να αναπαραστήσουν υψηλή επεκτασιμότητα και έτσι έγιναν οι βάσεις δεδομένων NoSQL. Οι NoSQL βάσεις δεδομένων επικεντρώνονται στην αναλυτική επεξεργασία μεγάλης κλίμακας συνόλου δεδομένων, προσφέροντας αυξημένη επεκτασιμότητα και υψηλές επιδόσεις.

Πρέπει να αναφερθεί πως στα συστήματα RDBMS, όταν υπάρχει η επιθυμία για βελτίωση, θα προστεθεί περισσότερη RAM ή καλύτεροι επεξεργαστές. Σε αντίθεση με τα NoSQL συστήματα, όπου απλά γίνεται προσθήκη κόμβων για να υπάρχει καλύτερη και αποτελεσματικότερη επεξεργασία δεδομένων. **[53]**

Τα βασικότερα πλεονεκτήματα των βάσεων δεδομένων NoSQL είναι:

1. Ευέλικτο μοντέλο δεδομένων:

   Οι σχεσιακές βάσεις δεδομένων μπορούν να αποθηκεύσουν δεδομένα μόνο με δομημένο τρόπο. Σε αντίθεση με τις μη σχεσιακές βάσεις δεδομένων NoSQL, οι οποίες είναι ευέλικτες και μπορούν να αποθηκεύσουν οποιονδήποτε τύπο δεδομένων, είτε είναι δομημένα, είτε όχι.

2. Εξέλιξη μοντέλου δεδομένων:

   Με τις βάσεις δεδομένων NoSQL, υπάρχει η δυνατότητα ενημέρωσης των σχημάτων έτσι, ώστε να εξελίσσονται ανάλογα με τις απαιτήσεις και ταυτόχρονα να μην προκληθεί κάποια διακοπή λειτουργίας.

3. Ελαστική επεκτασιμότητα:

   Μπορούν να κλιμακωθούν για να "φιλοξενήσουν" οποιονδήποτε τύπο δεδομένων με χαμηλό κόστος παράλληλα.

4. Υψηλή απόδοση:

   Έχουν εξαιρετική απόδοση.

5. Ανοιχτού κώδικα:

   Λειτουργούν σε φτηνό υλικό και η απόδοσή τους είναι οικονομική.

Τα βασικότερα μειονεκτήματα των βάσεων δεδομένων NoSQL είναι:

1. Έλλειψη τυποποίησης:

   Δεν υπάρχουν κανόνες για τις βάσεις δεδομένων NoSQL. Ο σχεδιασμός και οι γλώσσες ερωτημάτων έχουν μεγάλες διαφορές μεταξύ τους, σε σχέση με τις παραδοσιακές βάσεις δεδομένων SQL.

2. Αντίγραφο ασφαλείας της βάσης δεδομένων:

   Τα αντίγραφα ασφαλείας είναι ένα μειονέκτημα στις βάσεις δεδομένων NoSQL. Υπάρχουν κάποιες βάσεις δεδομένων NoSQL, όπου έχουν ορισμένα εργαλεία για να δημιουργηθούν αντίγραφα ασφαλείας. Αυτά τα εργαλεία, όμως, δεν θεωρούνται και τα καταλληλότερα, καθώς δεν εξασφαλίζουν ομαλή και ολοκληρωμένη λύση στο να δημιουργηθούν τα αντίγραφα αυτά.

3. Υποστήριξη πολλαπλών πλατφορμών:

   Σε κάποια λειτουργικά συστήματα δεν εκτελούνται σωστά, για παράδειγμα στα Linux. Αυτό σημαίνει πως πρέπει να γίνουν αρκετές βελτιώσεις πάνω σε κάποια συστήματα, για την ομαλή λειτουργικότητα.

4. Κακή χρηστικότητα:

   Κάποια εργαλεία της, δεν είναι στην πραγματικότητα καθόλου χρήσιμα.

Advantages	**Disadvantage**
Simple in using Scalable	Immature
It does not need database administrators	Quick, flexible and high efficient
It performs with more Space	Difficult in maintenance
Huge range of data model	Not having standard query language
NoSQL, DBaaS gives like Riak, Cassandra is programmed for dealing with the failure of hardware.	Few NoSQL database are not having complaint

IV. QUERYING DIFFERENCE IN NOSQL DATABASE

Εικόνα 11 Πλεονεκτήματα vs Μειονεκτήματα 1

3.6 Κατηγορίες NoSQL συστημάτων

Όπως αναφέρθηκε και παραπάνω, υπάρχουν πολλοί και διαφορετικοί τύποι βάσεων δεδομένων NoSQL:

1. Key-values Stores:

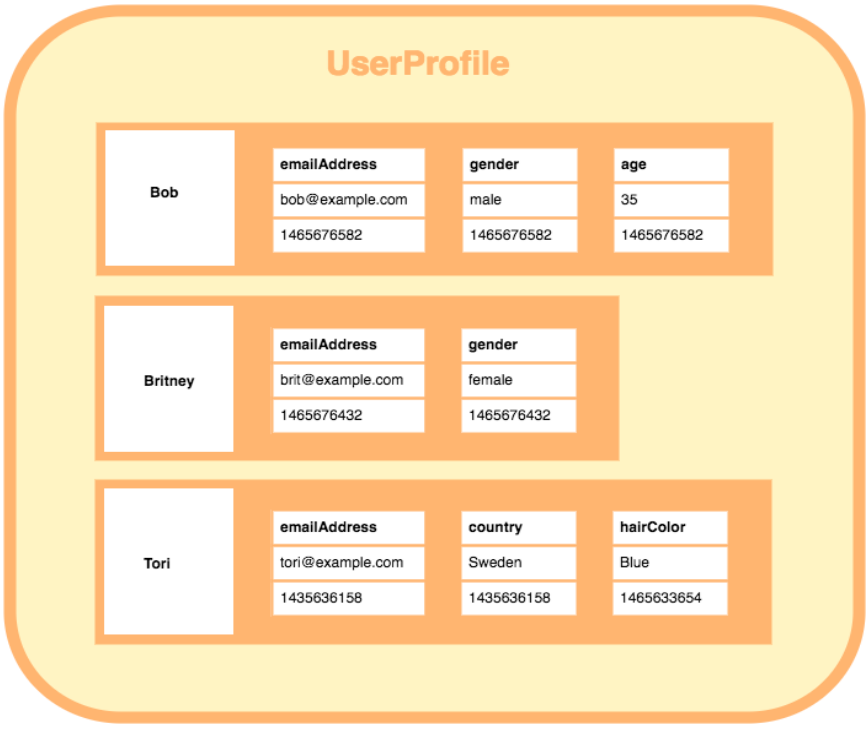
Μια βάση δεδομένων κλειδιού-τιμής είναι ένας τύπος μη σχεσιακής βάσης δεδομένων που χρησιμοποιεί μια απλή μέθοδο κλειδιού-τιμής για την αποθήκευση δεδομένων. Αποθηκεύει δεδομένα ως μια συλλογή ζευγών κλειδιών-τιμών στα οποία ένα κλειδί χρησιμεύει ως μοναδικό αναγνωριστικό. Τόσο τα κλειδιά όσο και οι τιμές μπορούν να είναι οτιδήποτε, είτε απλά αντικείμενα είτε πολύπλοκα σύνθετα αντικείμενα. Οι βάσεις δεδομένων κλειδιού-τιμής επιτρέπουν οριζόντια κλιμάκωση σε κλίμακες που άλλοι τύποι βάσεων δεδομένων δεν μπορούν να επιτύχουν. Για παράδειγμα, μία βάση δεδομένων NoSQL, εκχωρεί "διαμερίσματα" σε έναν πίνακα, εάν ένα υπάρχον "διαμέρισμα" γεμίσει και απαιτείται περισσότερος χώρος αποθήκευσης. **[18]**

Key	Value
K1	AAA,BBB,CCC
K2	AAA,BBB
K3	AAA,DDD
K4	AAA,2,01/01/2015
K5	3,ZZZ,5623

Εικόνα 12: Παράδειγμα Key-Value 1

2. Column Family Stores:

Η κατηγορία αυτή αναφέρετε στις βάσεις δεδομένων NoSQL που αποθηκεύουν δεδομένα σε εγγραφές και έχουν την ικανότητα να συγκρατούν πολύ μεγάλους αριθμούς δυναμικών στηλών. Οι στήλες μπορούν να περιέχουν μηδενικές τιμές και δεδομένα με διαφορετικούς τύπους δεδομένων. Επιπλέον, τα δεδομένα αποθηκεύονται σε κελιά ομαδοποιημένα σε στήλες δεδομένων και όχι ως σειρές δεδομένων. Η κατηγορία column family stores μπορούν να περιέχουν έναν σχεδόν απεριόριστο αριθμό στηλών που μπορούν να δημιουργηθούν κατά το χρόνο εκτέλεσης ή κατά τον καθορισμό του σχήματος. **[19]**



Εικόνα 13 Παράδειγμα Column Family Store 1

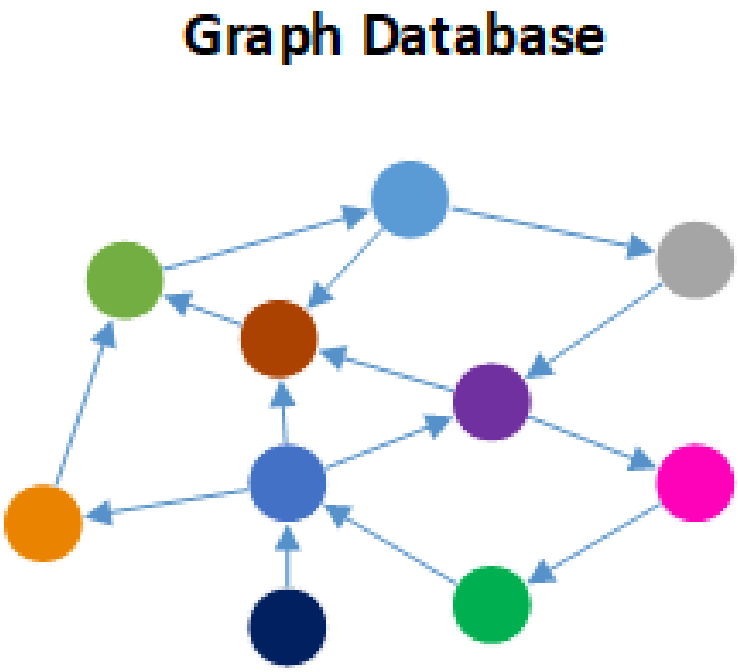
3. Document Databases:

Ένα document databases είναι πολύ διαφορετικό από άλλα μοντέλα δεδομένων, επειδή αποθηκεύει δεδομένα σε έγγραφα JSON, BSON ή XML. Μπορούμε, επίσης, να μετακινήσουμε έγγραφα κάτω από ένα έγγραφο. Δεν χρησιμοποιούνται σχήματα και αυτό βοηθάει στη διατήρηση των υπαρχόντων δεδομένων σε τεράστιους όγκους, επειδή δεν υπάρχουν απολύτως περιορισμοί στη μορφή και τη δομή της αποθήκευσης δεδομένων. **[20]**

Εικόνα 14 Document Databases 1

4. Graph Databases:

Είναι μια τεχνολογία για τη διαχείριση δεδομένων που έχει σχεδιαστεί για να χειρίζεται πολύ μεγάλα σύνολα δομημένων, ημι-δομημένων ή μη δομημένων δεδομένων. Επικεντρώνεται στις σχέσεις μεταξύ οντοτήτων και είναι σε θέση να συνάγει νέα γνώση από τις υπάρχουσες πληροφορίες. **[21]**



Εικόνα 15 Graph Database 1

Υπάρχουν πάρα πολλά συστήματα NoSQL. Παρακάτω θα αναφερθούν κάποια από αυτά:

Key-Value

3.6.1 BangDB

Το BangDB είναι κάτι περισσότερο από μια βάση δεδομένων NoSql και αποτελεί Key-Value. Έχει σχεδιαστεί για να βοηθά τους προγραμματιστές και τους χρήστες να επιλύουν προβλήματα με ευκολία και ταχύτητα και να δημιουργούν ισχυρές εφαρμογές. Επιτρέπει, επίσης, να ενσωματώνουν οποιοδήποτε είδος δεδομένων στο σύστημα να αναλύουν τα δεδομένα με απόλυτους τρόπους προκειμένου να λάβουν κάποιες ενέργειες για τη βελτιστοποίηση των επιχειρηματικών λειτουργιών. Όλα αυτά γίνονται με πολύ υψηλή.

Θα μπορούσε να θεωρηθεί, επίσης, ως μια πλατφόρμα συγκλίνουσας δεδομένων που έχει σχεδιαστεί για την ταχέως μεταβαλλόμενη τάση δεδομένων και επίσης για να είναι ανθεκτική στο να ανέχεται μελλοντικές αλλαγές και απαιτήσεις. Όταν πρόκειται να αντιμετωπίσουμε τα γρήγορα κινούμενα δεδομένα από συσκευές, αρχεία καταγραφής, ροές κλικ κ.λπ., πρέπει να έχουμε μηχανισμό ροής για την εύκολη απορρόφηση δεδομένων. Φέρνει την απαιτούμενη ευελιξία και ανθεκτικότητα όσον αφορά την αντιμετώπιση της ριπής παροχής δεδομένων και το σύστημα δεν θα αποτύχει λόγω της ανάγκης περισσότερης μνήμης.

Το BangDB έχει σχεδιαστεί για όλες τις απαιτήσεις και έχει αναπτυχθεί από την αρχή σε C/C++ για να ευθυγραμμίζεται με τις περιπτώσεις χρήσης, τις απαιτήσεις και τα σημεία πόνου των προγραμματιστών και των χρηστών. Εφόσον το BangDB μας επιτρέπει να ελέγχουμε τα δεδομένα και τον τρόπο που τα χειριζόμαστε, το ίδιο θα πρέπει να μεταφερθεί στους προγραμματιστές και τους χρήστες. **[22] [23] [24]**

Εικόνα 16 BangDB 1

3.6.2 Voldemort

Το σύστημα Voldemort, αποτελεί ένα κατανεμημένο σύστημα αποθήκευσης Key-Value. Χρησιμοποιεί προσωρινή αποθήκευση στη μνήμη για να καταργήσει ένα ξεχωριστό επίπεδο προσωρινής αποθήκευσης. Διαβάζει και γράφει κλίμακα οριζόντια . Το API αποφασίζει την αναπαραγωγή και την τοποθέτηση δεδομένων και φιλοξενεί ένα ευρύ φάσμα στρατηγικών για συγκεκριμένες εφαρμογές.

Τα δεδομένα αναπαράγονται αυτόματα στους διακομιστές και είναι χωρισμένα που σημαίνει ότι ένας μεμονωμένος διακομιστής περιέχει μόνο ένα μέρος των συνολικών δεδομένων. Κάθε κόμβος δεδομένων είναι ανεξάρτητος. Η σειριοποίηση με δυνατότητα σύνδεσης επιτρέπει εμπλουτισμένα κλειδιά και τιμές, συμπεριλαμβανομένων λιστών και πλειάδων. Με τα στοιχεία δεδομένων που έχουν εκδοθεί, μεγιστοποιείται η ακεραιότητα των δεδομένων. **[25] [26] [27]**

Εικόνα 17 Voldemort 1

3.6.3 Tarantool

Το Tarantool είναι μια πλατφόρμα υπολογιστών στη μνήμη με ένα ευέλικτο σχήμα δεδομένων, που χρησιμοποιείται καλύτερα για τη δημιουργία εφαρμογών υψηλής απόδοσης. Αποτελείται από δύο βασικά μέρη, το πρώτο είναι μια βάση δεδομένων στη μνήμη και το δεύτερο ένας διακομιστής εφαρμογής Lua. Διατηρεί τα δεδομένα στη μνήμη και εξασφαλίζει αντοχή σε σύγκρουση με καταγραφή και λήψη στιγμιότυπων πριν από την εγγραφή . Περιλαμβάνει διερμηνέα Lua και διαδραστική κονσόλα, αλλά δέχεται και συνδέσεις από προγράμματα σε πολλές άλλες γλώσσες.

Όλα τα δεδομένα διατηρούνται στη μνήμη RAM, και με την παραμονή των δεδομένων να διασφαλίζεται με την καταγραφή και τη λήψη στιγμιότυπων. Η αναπαραγωγή είναι ασύγχρονη και το να πάρει έναν διακομιστή Tarantool και να αναλάβει από έναν άλλο είναι δυνατή είτε από έναν διακομιστή αντιγραφής είτε από έναν διακομιστή "hot standby".

Το Tarantool χρησιμοποιεί ασύγχρονες εισόδους/εξόδους. Το αποτέλεσμα είναι ότι τα προγράμματα εφαρμογών ή οι αποθηκευμένες διαδικασίες πρέπει να γράφονται με γνώμονα τη συνεργατική πολλαπλή εργασία. **[29]**

Εικόνα 18 Tarantool 1

Column Family Stores

3.6.4 Apache HBase

Το σύστημα αυτό είναι ανοιχτού κώδικα και είναι γραμμένη σε γλώσσα Java. Αναπτύχθηκε ως μέρος του έργου Apache Hadoop του Apache Software Foundation και τρέχει πάνω από το HDFS (Hadoop Distributed File System). Είναι ανεκτικό σε σφάλματα και στο να αποθηκεύει μεγάλη ποσότητα μικρών δεδομένων, όπως για παράδειγμα η εύρεση 50 μεγαλύτερων στοιχείων σε μία ομάδα 2 δισεκατομμυρίων εγγραφών ή η εύρεση μη μηδενικών στοιχείων.

Το HBase λειτουργεί με φίλτρα Bloom ανά στήλη. Περιέχει πίνακες που χρησιμεύουν ως είσοδος και έξοδος για εργασίες MapReduce που εκτελούνται στο Hadoop και μπορούν να προσπελαστούν μέσω του API. Περιέχει πολλές και μεγάλες στήλες και έχει υιοθετηθεί ευρέως λόγω της καταγωγής του με το Hadoop και το HDFS. Τρέχει πάνω από το HDFS και είναι κατάλληλο για λειτουργίες γρήγορης ανάγνωσης και εγγραφής σε μεγάλα σύνολα δεδομένων με υψηλή απόδοση και χαμηλή καθυστέρηση εισόδου/εξόδου.

Το HBase δεν αντικαθιστά άμεσα μια κλασική βάση δεδομένων SQL, ωστόσο το έργο Apache Phoenix παρέχει ένα επίπεδο SQL για το HBase καθώς και πρόγραμμα το λεγόμενο JDBC που μπορεί να ενσωματωθεί με διάφορες εφαρμογές ανάλυσης και επιχειρηματικής ευφυΐας . Το έργο Apache Trafodion παρέχει μια μηχανή ερωτημάτων SQL με προγράμματα οδήγησης ODBC και JDBC και κατανεμημένη προστασία συναλλαγών ACID σε πολλαπλές δηλώσεις, πίνακες και σειρές που χρησιμοποιούν το HBase ως μηχανή αποθήκευσης.

Εξυπηρετεί πολλούς ιστότοπους που βασίζονται σε δεδομένα, αν και η πλατφόρμα μηνυμάτων του Facebook μετεγκαταστάθηκε από το HBase σε μία άλλη πλατφόρμα το 2018. Σε αντίθεση με τις σχεσιακές και τις παραδοσιακές βάσεις δεδομένων, το HBase δεν υποστηρίζει δέσμες ενεργειών SQL. Αντίθετα, το αντίστοιχο είναι γραμμένο σε Java, χρησιμοποιώντας ομοιότητα με μια εφαρμογή MapReduce. **[30] [31]**

Εικόνα 19 Apache Hbase 1

2.6.5 Apache Cassandra

Επίσης, μία ανοικτού τύπου βάσεων δεδομένων, η οποία είναι σχεδιασμένη για τεράστιο όγκο δεδομένων είτε χρησιμοποιούνται σε μεγάλα κατανεμημένα συστήματα, είτε σε υπολογιστές καθημερινής χρήσης.

Χρησιμοποιεί οριζόντια κλιμάκωση και είναι σε μεγάλο βαθμό αποτελεσματική σε περιπτώσεις που χρειάζεται σε μεγάλα κατανεμημένα δίκτυα (clusters). Η αρχιτεκτονική του, κάνει την δημιουργία και την επέκταση των δικτύων σχετικά απλή διαδικασία καθώς κάθε κόμβος είναι σχεδόν ίδιος με τους υπόλοιπους και δεν υπάρχουν σημεία συμφόρησης για να δημιουργήσουν καθυστερήσεις. Σημαντικό να αναφερθεί, πως δίνει η δυνατότητα στις επιχειρήσεις να προσθέσουν ακόμη περισσότερη χωρητικότητα για να φιλοξενήσουν επιπλέον δεδομένα.

Η γλώσσα ερωτημάτων της Cassandra είναι παρόμοια με την query language της SQL και έτσι οι προγραμματιστές μπορούν εύκολα να μετατρέψουν σε NoSQL μια σχεσιακή βάση δεδομένων.

Έχει καταπληκτική ιεαρχία στην μνήμη cache καθώς και προσεκτική τακτοποίηση δεδομένων στον δίσκο, όπου εγγυάται ασφάλεια και ταχύτητα. Η αρχιτεκτονική της αποθήκευσης των δεδομένων είναι παρόμοια με την δομή log-structured merge tree, όπου αναφέρθηκε και παραπάνω. Οι εγγραφές στέλνονται πρώτα στη μνήμη και στη συνέχεια σε έναν memtable στην μνήμη cache. Όταν ο memtable γεμίσει, τα δεδομένα πηγαίνουν στον δίσκο σε έναν Sorted String Table (SSTable) που είναι ένας ικανοποιητικός τρόπος αποθήκευσης μεγάλου αριθμού από ζεύγη key-value. Οι εγγραφές στον δίσκο είναι append-only, μπορείς να προσθέσεις μόνο δεδομένα και όχι να επεξεργαστείς ή να αφαιρέσεις. Και επειδή γίνονται με σειριακό τρόπο και όχι τυχαία, είναι πολύ αποτελεσματικές.

Με τον cluster, διανέμονται όλα τα δεδομένα σωστά στους κόμβους, μέσω της συνάρτησης κατακερματισμού hash. Οι εγγραφές στην Cassandra γράφονται σε πολλούς κόμβους ώστε τα δεδομένα να είναι ασφαλή σε περίπτωση κατάρρευσης κάποιου κόμβου. Ο κόμβος στον οποίο αντιγράφονται τα αρχικά δεδομένα λέγεται replica node. Εφαρμόζει την τεχνική Hinted Handoffs όπου αν κάποιος κόμβος δεν είναι διαθέσιμος οποιαδήποτε στιγμή, η Cassandra θα δημιουργήσει ένα στοιχείο το οποίο θα υποδεικνύει ότι η εγγραφή θα πρέπει να ξανασταλεί στον ανενεργό κόμβο. Όταν ο κόμβος επανέλθει τα δεδομένα θα σταλούν μέσω του hint στον κόμβο που είχε πρόβλημα αρχικά. **[32] [33]**

Εικόνα 20: Cassandra 1

2.6.6. ScyllaDB

Το ScyllaDB, είναι επίσης μία βάση NoSQL ανοιχτού κώδικα, προσανατολισμένη σε στήλη. Έχει μεγάλη απόδοση στην διαχείριση μεγάλου όγκου δεδομένων. Υπάρχουν αρκετές περιπτώσεις, όπου το Apache Cassandra, μπορεί να αντικατασταθεί από το συγκεκριμένο σύστημα. Η ScyllaDB έχει ξεπεράσει την Κασσάνδρα, τόσο ως προς την απόδοση όσο και ως προς το κόστος. Έχει την δυνατότητα να υποστηρίξει έως και ένα εκατομμύριο απόδοση σε ανάγνωση και εγγραφή. Παρόλο που υπάρχει αυτό το τεράστιο νούμερο, το σύστημα έχει μεγάλη απόδοση και σχεδόν καθόλου καθυστέρηση. Αυτοσυντονίζεται οπότε η χρήση του είναι αρκετά πιο εύκολη. Να σημειωθεί πως χρησιμοποιεί αρκετά λιγότερους υπολογιστικούς πόρους.

Η ScyllaDB είναι γραμμένη σε C++ μία επίσης αρκετά αποδοτική γλώσσα προγραμματισμού. Στη C++, η διαχείριση μνήμης μεταφορτώνεται στους προγραμματιστές. Αυτό είναι βοηθάει πολύ όταν οι προγραμματιστές που γράφουν τον κώδικα γνωρίζουν καλά τι κάνουν. Επίσης, χρησιμοποιεί ένα μοντέλο νήμα ανά πυρήνα το οποίο αυξάνει την ταχύτητα.

Όταν γίνεται κάποιο αίτημα ανάγνωσης, φαίνεται η πιο πρόσφατη εγγραφή ανεξάρτητα από τι. Υπάρχει πάντα μια απάντηση (χωρίς να υπάρχει κανένα σφάλμα) σε κάθε αίτημα. Εδώ, ένα αίτημα ανάγνωσης ενδέχεται να μην περιέχει την πιο πρόσφατη εγγραφή. Ακόμη και όταν υπάρχει ένα διαμέρισμα δικτύου μεταξύ των κόμβων, το σύστημα συνεχίζει να λειτουργεί. Σε ένα κατανεμημένο σύστημα, η ανοχή διαμερισμάτων είναι απαραίτητη, καθώς τα δίκτυα αποτυγχάνουν συνεχώς, γεγονός που δημιουργεί κατάτμηση μεταξύ των κόμβων. Χωρίς ανοχή κατάτμησης, δεν θα λειτουργούσε σωστά. Αυτό σημαίνει ότι πρέπει να επιλέξουμε μεταξύ συνέπειας και διαθεσιμότητας, θεωρώντας την ανοχή κατάτμησης ως προεπιλεγμένη εγγύηση. Σύμφωνα με το θεώρημα CAP, εγγυάται διαθεσιμότητα και ανοχή. Αυτό σημάινει, πως θα συνεχίσει να λειτουργεί ακόμη και με ένα διαμέρισμα δικτύου και πως κάθε ανάγνωση θα πραγματοποιηθεί χωρίς την εμφάνιση κάποιου σφάλματος. Ωστόσο, η απόκριση μπορεί να μην αντικατοπτρίζει πάντα τις πιο πρόσφατες ενημερώσεις. **[34]**

Εικόνα 21: Scylla 1

2.6.7 Cloudata

Οι βάσεις δεδομένων cloud είναι ακριβώς όπως οι παραδοσιακές βάσεις δεδομένων, αλλά δεν απαιτούν καμία εγκατάσταση και συντήρηση υποδομής. Οι βάσεις δεδομένων cloud εκτελούνται σε περιβάλλον υπολογιστικού νέφους σε αντίθεση με ένα περιβάλλον εσωτερικής εγκατάστασης. Η βάση αυτή, μπορεί να φιλοξενηθεί σε μία εικονική μηχανή που σε βασίζεται σε cloud. Οι εφαρμογές μπορούν στη συνέχεια να έχουν πρόσβαση σε όλα τα δεδομένα που είναι αποθηκευμένα σε μια βάση δεδομένων cloud μέσω ενός δικτύου από οποιαδήποτε συσκευή. Με τον πάροχο cloud, παρέχεται και διαχειρίζεται το υποκείμενο σύμπλεγμα βάσεων δεδομένων.

Σε μια αυτοδιαχειριζόμενη βάση δεδομένων, οι διαχειριστές συστήματος ή οι προγραμματιστές λογισμικού του οργανισμού είναι υπεύθυνοι για τη διαχείριση της βάσης δεδομένων. Αγοράζουν παρουσίες εικονικών μηχανών και εκτελούν τη βάση δεδομένων τους στις εικονικές μηχανές. Ο πάροχος cloud φροντίζει για την παροχή υποδομής. Σε μια αυτοδιαχειριζόμενη βάση δεδομένων, οι διαχειριστές συστήματος ή οι προγραμματιστές λογισμικού του οργανισμού είναι υπεύθυνοι για τη διαχείριση της βάσης δεδομένων. Αγοράζουν παρουσίες εικονικών μηχανών και εκτελούν τη βάση δεδομένων τους στις εικονικές μηχανές. Ο πάροχος cloud φροντίζει για την παροχή υποδομής.

Υπάρχει η δυνατότητα για οποιονδήποτε τύπο βάσης δεδομένων στο cloud. Και τις παραδοσιακές βάσεις δεδομένων SQL και τις πιο σύγχρονες βάσεις δεδομένων NoSQL. Το Mongo DB Atlas είναι μια βάση δεδομένων εγγράφων γενικής χρήσης που μπορεί να αναπτυχθεί σε οποιονδήποτε από τους σημαντικότερους παρόχους cloud, όπως το Amazon Web Services (AWS) , το Microsoft Azure και το Google Cloud.

Είναι πολύ σημαντικό να αναφερθεί πως η βάση δεδομέναν Cloudata, ειδοποιεί αυτόματα για ζητήματα απόδησης ώστε να μπορεί να γίνει βελτιστοποίηση και να πετύχει ο στόχος απόδοσης. Είναι αρκετά φθηνότερο να χρησιμοποιείται μία τέτοια βάση αλλά απαιτεί και λιγότερη χειρονακτική εργασία. **[35] [36] [37]**

Εικόνα 22: Cloudata 1

Document Store

2.6.8 Couch DB

Η Couch DB, είναι ανοιχτού τύπου NoSQL. Τα δεδομένα της, μπορούν να αποθηκεύονται χωρίς να έχει οριστεί η δομή της. Πολλές εταιρίες έχουν επιλέξει να δουλεύουν με το συγκεκριμένο σύστημα, καθώς αποτελεί μεγάλο βοήθημα στην διαχείριση μεγάλων και πολλών δεδομένων.

Μία βάση δεδομένων, η οποία τρέχει σε οποιονδήποτε server καθώς και μπορεί να λειτουργεί ταυτόχρονα σε αρκετές εικονικές μηχανές. Όταν τρέχει πάνω σε clusters, βελτιώνει όλες τις λειτουργίες και προσφέρει καλύτερη απόδοση και ικανότητα στο να διαχειριστεί.

Χρησιμοποιεί το πρωτόκολλο HTTP και το μοντέλο δεδομένων JSON, η οποία υποστηρίζεται από κάθε λογισιμικό. Όταν υπάρχουν πολλοί κόμβοι, τα δεδομένα αντιγράφονται έτσι ώστε σε περίπτωση που χρειαστούν να είναι διαθέσιμα, οποιαδήποτε στιγμή. Λόγω του ότι υποστηρίζει ένα συγκεκριμένο μοντέλο, μπορεί να χειρίζεται παράλληκα μεγάλο αριθμό χρηστών χωρίς να υπάρχει τυχόν σύγκρουση. Χρησιμοποιεί το προγραμματιστικό μοντέλο MapReduce το οποίο χρησιμοποιείται για την ανάπτυξη εφαρμογών που επεξεργάζονται παράλληλα τεράστιες ποσότητες δεδομένων. Ένα αρκετά σημαντικό μοντέλο για την δημιουργία και επεξεργασία μεγάλων δεδομένων. Λόγω του "εργαλείου" αυτού, το πρόβλημα χωρίζεται σε δύο "κομμάτια" και μπορεί να λυθεί πολύ πιο εύκολα από το να είναι ένα ενιαίο πρόβλημα. Επίσης, χρησιμοποιεί μία τεχνική με την οποία ο master είναι υπεύθυνος για τον χρονοπρογραμματισμό της εκτέλεσης των εργασιών και στην περίπτωση όπου καθυστερήσει ένας κόμβος, η εργασία θα ανατεθεί σε κάποιον άλλον κόμβο. Οι servers που υπάρχουν στην Couch DB, είναι κατάλληλοι είτε για κάποιον μικρό, είτε για κάποιον τεράστιο μεγέθους.

Τέλος, είναι σημαντικό να αναφερθεί και να τονιστεί ότι σε περίπτωση όπου η εφαρμογή σταματήσει να λειτουργεί, τα δεδομένα παραμένουν ασφαλή χωρίς να υπάρχει ο οποισδήποτε κίνδυνος. Στα κατανεμημένα συστήματα υπολογιστών, όταν υπάρχουν πολλοί κόμβοι, υπάρχει και ένα back up έτσι ώστε αν ζητηθεί να είναι διαθέσιμα τα δεδομένα. **[38] [39] [40]**

Εικόνα 23: CoucheDB 1

2.6.9 MongoDB

Θα έλεγε κανείς πως η Mongo DB, είναι η πιο γνωστή NoSQL βάση δεδομένων. Τρέχει σε διάφορα λειτουργικά συστήματα και αποτελεί document store βάση. Επιτρέπει στους χρήστες να δημιουργούν εφαρμογές, που κάποτε, χωρίς την βάση αυτήν δεν μπορούσαν. Η έκδοση της για cloud μπορεί να έχει εργαλεία για την διαχείριση, την επιδιόρθωση του λογισμικού, την παρακολούθηση, των back ups της βάσης δεδομένων και λειτουργεί σε ένα διαμοιρασμένο δίκτυο υπολογιστών.

Με το document store, τα δεδομένα αποθηκεύονται εύκολα, χρησιμοποιώντας ένα ευέλικτο μοντέλο εγγραφών. Δηλαδή, μπορούν να χρησιμοποιούν έναν ή περισσότερους πίνακες. Τα πεδία, μπορούν να διαφέρουν από έγγραφο σε έγγραφο και αυτό βοηθάει τους προγραμματιστές να εξελίξουν το μοντέλο των δεδομένων τους ταχύτατα, καθώς αλλάζουν οι απαιτήσεις των εφαρμογών. Είναι μία πολύ απλή βάση δεδομένων και επειδή χρησιμοποιεί JSON αρχεία, οι αλλαγές γίνονται γρήγορα και εύκολα, λόγω του δυναμικού τους σχεδιασμό. Οι διάφορες λειτουργίες γίνονται γρηγορότερα και οικονομικότερα. Έχει ενσωματωμένη λειτουργία κλιμάκωσης και μπορεί να χειρίζεται από μόνη της τα όποια προβλήματα παρουσιαστούν.

Μπορούν εύκολα και γρήγορα και να βρίσκουν δεδομένα χωρίς να χρειαστεί να χρησιμοποιήσουν επιπλέον κώδικα. Πρόκειται για μια τεχνολογική εξέλιξη όπου πολλές συσκευές μαζί αποτελούν ένα δίκτυο. Εφαρμόζει το σύστημα Internet of Things (IoT). Κάθε συσκευή ενσωματώνει ηλεκτρονικά μέσα, λογισμικό, αισθητήρες ώστε να επιτρέπει την σύνδεση και την ανταλλαγή δεδομένων μεταξύ τους, είτε σε τοπικό δίκτυο, είτε στο διαδίκτυο. Το IoT βοηθάει στην ανάπτυξη και στην εξέλιξη ακόμη και στην οικονομική ανάπτυξη.

Προσφέρει διαχείριση μεγάλου όγκου δεδομένων και το πιο σημαντικό, ασφάλεια των δεδομένων σε περίπτωση απάτης. Η βάση δεδομένων, "σπάει" σε μικρά κομμάτια, με αποτέλεσμα να είναι περισσότερο διαχειρίσιμη και ευκολότερη, άρα έτσι έχουμε καλύτερα διαχείριση του φόρτου εργασίας από δεδομένα αλλά και αποφυγή στο να χάσουμε δεδομένα.

Η δημοτικότητα του MongoDB μεταξύ των προγραμματιστών περιλαμβάνει το ευέλικτο μοντέλο δεδομένων και το διαισθητικό API του.

Υποστηρίζει ερωτήματα εύρους, πεδίου και κανονικής έκφρασης που μπορούν να επιστρέψουν πλήρη έγγραφα, συγκεκριμένα πεδία από εσωτερικά έγγραφα, ακόμη και δείγματα τυχαίων αποτελεσμάτων.

Διαθέτει υψηλή διαθεσιμότητα και έτσι επιτυγχάνεται μέσω σετ αντιγράφων, συμπεριλαμβανομένων πολλαπλών αντιγράφων δεδομένων. Το κύριο αντίγραφο χειρίζεται εγγραφές και κάθε αντίγραφο μπορεί να εξυπηρετήσει αιτήματα ανάγνωσης. Σε περίπτωση αποτυχίας του πρωτεύοντος αντιγράφου, ένα δευτερεύον αντίγραφο αναλαμβάνει ως πρωτεύον αντίγραφο.

Τέλος, υποστηρίζει διαφορετικούς τύπους ευρετηρίου, όπως μεμονωμένο πεδίο, πολλαπλό κλειδί (πίνακας), σύνθετο (πολλά πεδία), γεωχωρικό, κατακερματισμένο και κείμενο. Τα πεδία εγγράφου μπορούν να ευρετηριαστούν χρησιμοποιώντας τόσο πρωτεύοντες όσο και δευτερεύοντες δείκτες. **[55]**

Εικόνα 24: MongoDB 1

2.6.10 Arango DB

Η βάση αυτή, είναι μία ανοιχτού τύπου βάση δεδομένων με ευέλικτο μοντέλο σχετικό με κείμενα, γράφους και ζεήγη κλειδιών. Μπορεί να υποστηρίξει πολλά μοντέλα. Χρησιμοποιεί SQL και JavaScript, έτσι οι επιδόσεις της είναι αρκετά υψηλές. Το ευέλικτο μοντέλο που διαθέτει αλλά και το να συνδυάζει πολλούς τύπους δεδομένων, το κάνει ιδανικό για τις διάφορες εφαρμογές κοινωνικής δικτύωσης. Με το αρχείο JSON, βοηθάει την βάση δεδομένων να αποθηκεύει τα αρχεία και ταυτόχρονα η γλώσσα ερωτημάτων που χρησιμοποιεί βοηθάει στην τροποποίηση αλλά και στο να αποκτήσουν δεδομένα.

Μπορεί να χρησιμοποιηθεί ως μια υπηρεσία που συμβάλλει στο να δημιουργηθούν εφαρμογές διαδικτύου και ταυτόχρονα να δημιουργηθεί ένα περιβάλλον server για να τρέξει η εφαρμογή αυτή. Μπορεί να εκτελεί ερωτήματα σε πολλαπλά δεδομένα ή σε συλλογές δεδομένων. Μοιράζει δεδομένα σε διάφορους servers και δίνει την δυνατότητα να αποφασίσει εάν χρειάζεται υψηλή επίδοση. Ο λόγος που η ταχύτητα στην εγγραφή αλλά και στην ανάγνωση δεδομένων είναι τεράστια, είναι επειδή χρησιμοποιεί τις γνωστές cache memory, οι οποίες είναι τελευταίας τεχνολογίας αλλά έχουν και μεγάλη χωρητικότητα.

Όταν το επίπεδο των σφαλμάτων είναι υψηλό, τότε πρέπει να χρησιμοποιηθούν πολλαπλοί servers που να αποτυγχάνουν ανεξάρτητα ο ένας από τον άλλο. Συνήθως αντίγραφα από έναν server εκτελούνται σε ξεχωριστούς επεξεργαστές ενός κατανεμημένου συστήματος και χρησιμοποιούνται πρωτόκολλα που συνδέουν τις αλληλεπιδράσεις του χρήστη με τα αντίγραφα αυτά. Ωστόσο, εάν κάποια μέρη του συστήματος σταματήσουν να λειτουργούν για τον οποιονδήποτε λόγο, το σύστημα γενικά θα συνεχίσει να λειτουργεί κανονικά. **[41] [42] [43]**

Εικόνα 25: ArangoDB 1

Graph Data Base

2.6.11 Neo4J

Η Neo4j είναι μία ανοικτή τύπου βάσεων δεδομένων NoSQL. Είναι μία δημοφιλής βάση και η οποία όσων αφορά την βάση δεδομένων των γράφων είναι κυρίαρχη και έχει τεράστια ζήτηση στην αγορά. Μέσω αυτής, υπάρχει η δυνατότητα για back-up, αλλά και για την αντιμετώπιση διαφόρων προβλημάτων.

Η γλώσσα που χρησιμοποιείται είναι παρόμοια με την γλώσσα ερωτημάτων SQL, αλλά με κάποιες διαφορές έτσι ώστε να λειτουργεί αρμονικά στα γραφήματα. Με την Cypher, εύκολα κάποιος αντιλαμβάνεται καλύτερα την συγκεκριμένη βάση δεδομένων και μπορεί να την εφαρμόσει και πιο γρήγορα καθώς δεν υπάρχουν πολύπλοκα ερωτήματα, χωρίς πολλές σειρές κώδικα. Η βάση χρησιμοποιεί τους γνωστούς δείκτες (pointers) και τα δεδομένα αποθηκεύονται σε έναν πίνακα. Καταλαβαίνουμε, λοιπόν, πως είναι όλα μαζί και σε περίπτωση που χαθούν, μπορούν εύκολα να ανακτηθούν και πάλι.

Στο γράφημα, φαίνονται οι κόμβοι που υπάρχουν, οι οποίοι συνδέονται μεταξύ τους. Έτσι, δημιουργείτε το γράφημα. Οι κόμβοι είναι οι οντότητες. Δεν χρειάζεται να υπάρχουν σχέσεις μεταξύ των οντοτήτων για να δημιουργηθεί ένα γράφημα.

Γρήγορη ανάγνωση δεδομένων και ένα διαφορετικό και ευέλικτο σχήμα γραφικών το οποίο μπορεί να «ταιριάζει» σε όποια συνθήκη εμείς επιθυμούμε. **[44] [45] [46] [47]**

Εικόνα 26: Neo4j 1

2.6.12 Titan

Η Nosql βάση δεδομένων Titan, είναι μια κλιμακούμενη βάση δεδομένων γραφημάτων, η οποία έχει βελτιστοποιηθεί τόσο για την αποθήκευση και την αναζήτηση των γραφημάτων όπου περιέχουν εκατοντάδες δισεκατομμύρια κορυφές και ακμές κατανεμημένες σε ένα σύμπλεγμα πολλαπλών μηχανών. Υποστηρίζει την ταυτόχρονη πρόσβαση, σε πάρα πολλούς χρήστες που εκτελούν σύνθετες διασχίσεις γραφημάτων.

Μπορούμε να αναφέρουμε πως ένα από τα βασικά του χαρακτηριστικά είναι Ελαστική και γραμμική επεκτασιμότητα για μια αυξανόμενη βάση δεδομένων και χρηστών. Αναπαράγει δεδομένα και βρίσκει εύκολα σφάλματα τα οποία προκύπτουν. Υπάρχουν πολλά αντίγραφα ασφαλείας. Υποστηρίζει την συναλλαγή βάσεων δεδομένων, όπου προορίζονται να εγγυηθούν την εγκυρότητα των δεδομένων όποια σφάλματα προκύψουν, τις διακοπές ρεύματος και άλλες ατυχίες. Υποστηρίζει την ανάλυση δεδομένων παγκόσμιων γραφημάτων , αναφορές και πλατφόρμες μεγάλων δεδομένων, όπως είναι Apache Spark, Apache Hadoop κλπ. Διαθέτει αναζήτηση πλήρους κειμένων μέσω διαφόρων εργαλείων. **[48] [49]**

Εικόνα 27: Titan 1

2.6.13 AllegroGraph

Το AllegroGraph είναι μια βάση δεδομένων και ένα πλαίσιο εφαρμογής για τη δημιουργία λύσεων Enterprise Knowledge Graph που βασίζονται σε ένα τριπλό κατάστημα υψηλής απόδοσης. Τα δεδομένα και τα μεταδεδομένα μπορούν να διαχειριστούν, χρησιμοποιώντας Java, Python, Lisp και HTTP. Διαθέτει δυνατότητες ανάλυσης κοινωνικών δικτύων , γεωχωρικές και χρονικές. Καταλαβαίνουμε, λοιπόν, πως προσφέρει τεράστια οριζόντια επεκτασιμότητα. Μπορεί να εκτελεστεί σε πολλούς διακομιστές σε ευρέως διαχωρισμένες τοποθεσίες. Όταν οι πληροφορίες, τα δεδομένα, έχουν δομή γραφήματος, η συγκεκριμένη βάση είναι χρήσιμη και βοηθητική για την ανάκτηση και την αποθήκευση τους. Να σημειωθεί, πως το AllegroGraph υποστηρίζει MultiMaster Replication (MMR) το οποίο σημαίνει ότι επιτρέπει τον συγχρονισμό των αποθετηρίων σε κάθε κέντρο δεδομένων, ώστε να βλέπουν όλα τα ίδια δεδομένα (φυσικά επιτρέποντας χρόνο επικοινωνίας δικτύου).

Η παραπάνω εφαρμογές, είναι περισσότερο εύκολες όταν αφορά βάση δεδομένων γρραφημάτων, αφού δεν χρησιμοποιούνται πίνακες. Οι σχέσεις ένα προς πολλά και πολλά προς πολλά μοντελοποιούνται απευθείας, Σε γλώσσες ερωτημάτων όπως είναι η Sparql, οι σχέσεις γραφήματος προς αναζήτηση εκφράζονται απευθείας, Μπορούμε να προσθέσουμε νέα κατηγορήματα χωρίς να αλλάξουμε κάποιο σχήμα. Αυτό σημαίνει πως θα υπάρχει ευελιξία και προσαρμοστικότητα στη μοντελοποίηση δεδομένων. Με όλα τα παραπάνω, συμπεραίνουμε πως έχουμε μία πολύ αποτελεσματική μέθοδο. Το AllegroGraph είναι μια ισχυρή γενική βάση δεδομένων γραφημάτων.

Η ιδιότητα ανθεκτικότητας ορίζει ότι μόλις το σύστημα βάσης δεδομένων σηματοδοτήσει την επιτυχή ολοκλήρωση μιας συναλλαγής στην εφαρμογή, οι αλλαγές που γίνονται από τη συναλλαγή θα επιμείνουν ακόμη και με την παρουσία αστοχιών υλικού και λογισμικού. Όταν επιστρέψει η λειτουργία δέσμευσης του AllegroGraph, ο διακομιστής της βάσης δεδομένων θα έχει γράψει τις ενημερώσεις που έγιναν από τη συναλλαγή στο αρχείο καταγραφής συναλλαγών και θα περιμένει να ολοκληρωθεί η λειτουργία εισόδου/εξόδου καταγραφής. Επομένως, η εφαρμογή μπορεί να είναι σίγουρη ότι κάθε δεσμευμένη συναλλαγή θα έχει μόνιμη επίδραση στη μόνιμη κατάσταση της βάσης δεδομένων. **[50] [51]**

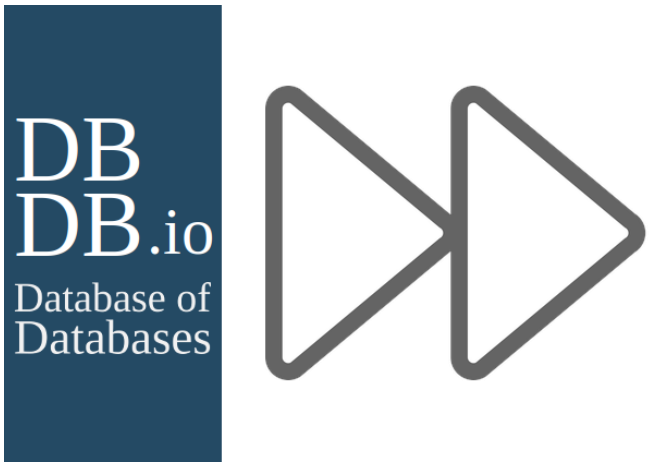
Εικόνα 28: AllegroGraph 1

2.6.14 WhiteDB

Αποτελεί μία ελαφριά βιβλιοθήκη βάσεων δεδομένων NoSQL, η οποία είναι γραμμένη σε C, όπου και λειτουργεί στην κύρια μνήμη. Τα δεδομένα διαβάζονται από την μνήμη αυτή και εγγράφονται, επίσης από αυτήν.

Τα δεδομένα διατηρούνται στην κοινόχρηστη μνήμη από προεπιλογή, καθιστώντας όλα τα δεδομένα προσβάσιμα σε ξεχωριστές διαδικασίες. Μπορούν να αποθηκευτούν συμβατικοί τύποι δεδομένων αλλά και δείκτες σε εγγραφές. Ο κάθε δείκτης, επιτρέπει την εξαιρετικά αποτελεσματική διέλευση σύνθετων δεδομένων.

Να σημειωθεί, επίσης, ότι υποστηρίζονται ευρετήρια, αρχεία .csv, .rdf, υπάρχει σύνδεση με την γλώσσα προγραμματισμού Python καθώς και βοηθητικά εργαλεία γραμμής εντολών. **[52]**



Εικόνα 29: WhiteDB 1

DataSource	Type	Open Source	Top Use Cases	Secutiry
BangDB	Key-Value Stores	No	AI, Text, Graph, Time-Series, Doc-Json, Large Files-objects	C/C++
Voldemort	Key-Value Stores	Limited Open Source	Json, BLOB ojects, XML documents	Java
Tarantool	Key-Value Stores	Yes	AI	C
Apache HBase	Column Family Store	Yes	Data volume, Sports, Medical, Hardware Environment	Java
Apache Cassandra	Column Family Store	Yes	Social analytics, real time analytics	Java
ScyllaDB	Column Family Store	Yes	Big Data Analytics, AI	C++
Cloudata	Column Family Store	No	AI, Big Data Analytics, Test and Development	Java
Couch DB	Document Stores	Yes	IoT, inventory	JavaScript
Mongo DB	Document Stores	Limited Open Source	IoT data managmen	C++
Arango DB	Document Stores	Yes		C++
Neo4J	Graph Database	Yes	Ai, master data, managment	Java, JavaScript, Go, and Python
Titan	Graph Database	Yes	Graph	C
AllegroGraph	Graph Database	No	Ai, master data, managment	Java, Python,
WhiteDB	Graph Database	No	Csv files, rdf	Python

3.7 Εξέλιξη του NoSQL

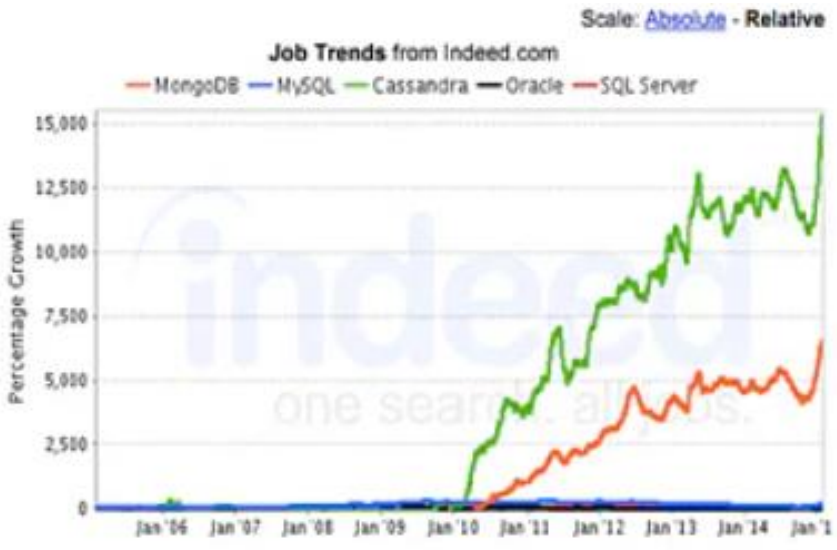
Τα συστήματα SQL και σχεσιακών βάσεων δεδομένων είναι «διασκορπισμένα» επειδή παρέχουν έναν καλό μηχανισμό γενικού σκοπού έτσι ώστε να υποστηρίζουν τις περισσότερες απαιτήσεις της διαχείρισης δεδομένων. Έχουν σχεδιαστεί για να είναι αξιόπιστα, ακριβή και χρήσιμα για προγραμματισμένες εφαρμογές. Ωστόσο, ορισμένες απαιτήσεις SQL και σχεσιακές μπορούν να τις καταστήσουν λιγότερο κατάλληλες για εφαρμογές που απαιτούν ευέλικτα δεδομένα και υψηλή ταχύτητα.

Για να καλυφτεί κάθε ανάγκη, δημιουργήθηκαν συστήματα βάσεων δεδομένων NoSQL. Είχαν προκύψει αρκετά προβλήματα σε διάφορες εταιρίες, όπως η Amazon με το DynamoDB , το Facebook και το Apache Cassandra και η Google με τη βάση δεδομένων BigTable.

Το Berkeley DB περιγράφηκε ευρέως ως μια ενσωματωμένη βάση δεδομένων που υποστήριζε στενά τις ανάγκες αποθήκευσης συγκεκριμένων εφαρμογών. Αυτό το λογισμικό ανοιχτού κώδικα παρείχε έναν απλό χώρο αποθήκευσης κλειδιών-τιμών. Δημιουργήθηκε το 1999.

Ο όρος NoSQL μπορεί να εφαρμοστεί σε ορισμένες βάσεις δεδομένων που προϋπήρχαν του συστήματος διαχείρισης σχεσιακών βάσεων δεδομένων αλλά πιο συχνά αναφέρεται στις βάσεις δεδομένων που δημιουργήθηκαν στις αρχές της δεκαετίας του 2000 με σκοπό τη ομαδοποίηση βάσεων δεδομένων μεγάλης κλίμακας σε εφαρμογές cloud και web.

Έτσι, οι άνθρωποι άρχισαν να κινούνται προς τις πιο φρέσκες και νεότερες ανάγκες δεδομένων, με αποτέλεσμα να συζητιέται όλο και περισσότερο ο όρος «Cassandra» και ο όρος «MongoDB» για την αντιμετώπιση αυτών των αναγκών.



Εικόνα 30: Έρευνα Indeed 1

Παρατηρώντας το παραπάνω διάγραμμα, παρατηρούμε πως δεν θα υπάρξει κάποια μεγάλη και ξαφνική αλλαγή στην αγορά. Για την ακρίβεια, προσέχουμε πως η αλλαγή αυτή θα «χτυπήσει» περισσότερο την MySQL. Ωστόσο, όσο περνάει ο καιρός κάθε RDBMS θα κινδυνεύει. Τα Big Data ανεβαίνουν όλο και περισσότερο, οπότε οι επιχειρήσεις θα βασίζονται όλο και περισσότερο στις βάσεις δεδομένων Hadoop και NoSQL. Να σημειωθεί, επίσης, πως δεν πρόκειται να υπάρξει κατάργηση των RDBMS από τα συστήματα NoSQL.

3.8 Το μέλλον του NoSQL

Πολλές είναι οι επιχειρήσεις οι οποίες κρύβουν έναν τεράστιο όγκο δεδομένων από πίσω τους, με αποτέλεσμα να εξετάζουν πολύ σοβαρά το θέμα της NoSQL. Σε μια έρευνα που πραγματοποιήθηκε, το 44% των επαγγελματιών της πληροφορικής δεν έχουν ακούσει για το σύστημα NoSQL. Μόνο το 1% ανέφερε ότι το NoSQL σύστημα αποτελεί μέρος της στρατηγικής τους κατεύθυνσης. Από τα παραπάνω, καταλαβαίνουμε πως η NoSQL έχει πάρει μία θέση στον κόσμο της πληροφορικής, αλλά θα πρέπει να συνεχίσει να εξελίσσεται για να πάρει τη μαζική έκκληση που πολλοί πιστεύουν ότι θα μπορούσε να έχει.

3.9 Απόδοση

Η απόδοση των βάσεων δεδομένων NoSQL συνήθως αξιολογείται χρησιμοποιώντας τη μέτρηση της απόδοσης και μετριέται σε δευτερόλεπτα. Η αξιολόγηση της απόδοσης πρέπει να δίνει προσοχή στα σωστά σημεία αναφοράς στον αναμενόμενο όγκος δεδομένων και οι ταυτόχρονοι φόρτοι εργασίας των χρηστών.

Μοντέλο Δεδομένων	**Εκτέλεση**	**Επεκτασιμότητα**	**Ευκαμψία**	**Πολυπλοκότητα**
Key–value store	Υψηλή	Υψηλή	Υψηλή	Μηδενική
Column-oriented store	Υψηλή	Υψηλή	Μέτρια	Χαμηλή
Graph database	Μεταβλητή	Μεταβλητή	Υψηλή	Χαμηλή

Document Database	Υψηλή	Μεταβλητή	Υψηλή	Μέτρια

Πίνακας 1: Απόδοση NoSQL

3.10 Σχέση Spark με NoSQL

Τι είναι το λεγόμενο Spark; Το Spark είναι ένα μεγάλο πλαίσιο επεξεργασίας δεδομένων που κάνει αναλύσεις, μηχανική εκμάθηση, επεξεργασία γραφημάτων και ακόμη περισσότερο που μπορεί να μην σχετίζονται με τον μεγάλο όγκο δεδομένων. Είναι παρόμοιο με το Map Reduce, και άλλα επίπεδα επεξεργασίας δεδομένων που είναι χτισμένα πάνω από το HDFS στο Hadoop. Όπως και το Hadoop, έτσι και το Spark, επικεντρώνεται στη βελτιστοποίηση, αλλά θα έλεγε κανείς πως είναι καλύτερο από πολλές απόψεις. Είναι πιο γρήγορο, πολύ πιο ωραίο στον προγραμματισμό και έχει καλούς συνδέσμους σχεδόν σε όλα.

Σε αντίθεση με το Hadoop, είναι πιο εύκολο να ξεκινήσει κάποιος να γράφει και να εκτελεί το Spark από τη γραμμή εντολών και στη συνέχεια να μπορεί να εκτελεί σε ένα πλήρες σύνολο δεδομένων.

Το Spark δεν είναι μία βάση δεδομένων. Είναι μια μηχανή επεξεργασίας δεδομένων. Διαβάζει μαζικά δεδομένα που είναι αποθηκευμένα κάπου όπως το HDFS ή ο διακομιστής Couchbase, επεξεργάζεται αυτά τα δεδομένα και στο τέλος, γράφει τα αποτελέσματά τους, ώστε να μπορούν να χρησιμοποιηθούν ξανά στην πορεία.

Πλέον, πολλοί οργανισμοί, χρησιμοποιούν Couchbase και Spark μαζί. Οι μεγάλες διαδικτυακές εφαρμογές που τρέχουν στο Couchbase τείνουν να έχουν πολλές από αυτές. Οι άνθρωποι δημιουργούν περισσότερα από αυτό κάθε μέρα όταν κάνουν αγορές στο διαδίκτυο ή στέλνουν ο ένας στον άλλο μηνύματα.

Το Spark παρέχει μοντέλα μηχανικής μάθησης, προβλέψεις, αποτελέσματα μεγάλων εργασιών ανάλυσης κ.λπ., και το Couchbase τα κλιμακώνει σε μεγάλο αριθμό χρηστών. Για παράδειγμα, ταξινομεί τις ανεπιθύμητες αλληλογραφίες για εφαρμογές επικοινωνίας σε πραγματικό χρόνο, περιλαμβάνει προγνωστικά αναλυτικά στοιχεία και μοντέλα ανίχνευσης απάτης για εφαρμογές για κινητές συσκευές που πρέπει να λάβουν άμεσες αποφάσεις αποδοχής ή απόρριψης πληρωμής . Επίσης, η αποθήκευση μεγάλου όγκου δεδομένων, τα οποία επεξεργάζονται αλλά και το Διαδικτύου των πραγμάτων.

Το κάθε ένα από τα παραπάνω, είναι βελτιστοποιημένο για τον φόρτο εργασίας που υπάρχει.

Ο διακομιστής Couchbase δημιουργήθηκε για να εκτελεί εφαρμογές που είναι γρήγορες, και εύκολες ως προς την διαχείριση.

Το Couchbase Spark Connector είναι ανοιχτού κώδικα και κάποια από τα πλεονεκτήματά του, είναι τα εξής:

- Είναι γρήγορο. Το Spark και το Couchbase, βρίσκονται κεντρικά στην μνήμη. Υποστηρίζει API υπο-έγγραφα και υπάρχει βελτιωμένη απόδοση.
- Είναι λειτουργικό. Υποστηρίζονται όλα τα RDD, τα DataFrames, τα σύνολα δεδομένων, η μείωση χάρτη και οι προβολές χωρικής προβολής, ακόμη και το DCP από τη Scala και την Java.

Επίλογος – Συμπεράσματα

Με βάση τα παραπάνω, καταλαβαίνουμε πως ο ορισμός Μεγάλα Δεδομένα, είναι πολύ «μεγάλος» και ο όγκος αυτός, αυξάνεται διαρκώς. Έτσι, συνεχώς επιχειρήσεις ψάχνουν τρόπους έτσι, ώστε αυτός ο όγκος να καλυφθεί και να μην επιβαρύνει την κάθε εργασία. Οι NoSQL βάσεις δεδομένων, είναι θα έλεγε κανείς πρόσφατες και έχουν αρκετά αποτελέσματα.

Πρέπει να σημειωθεί πως, η επιλογή για SQL ή NoSQL, εξαρτάται από τις εφαρμογές. Όμως, οι περισσότερες επιχειρήσεις, προτιμούν την βάση NoSQL, καθώς μπορεί να ανταπεξέλθει πιο εύκολα σε περιπτώσεις μεγάλου όγκου δεδομένων καθώς και είναι πιο εύκολη ως προς την χρήση του ανθρώπου,

Βιβλιογραφία

[1] https://inbusinessnews.reporter.com.cy/business/ict918/article/299566/big-data-kai-anaptyxi-epicheirimatikotitas

[2] https://el.wikipedia.org/wiki/%CE%9C%CE%B5%CE%B3%CE%AC%CE%BB%CE%B1_%CE%B4%CE%B5%CE%B4%CE%BF%CE%BC%CE%AD%CE%BD%CE%B1

[3] https://www.datanami.com/2022/01/11/big-growth-forecasted-for-big-data/

[4] https://en.wikipedia.org/wiki/Big_data#cite_note-22

[5] https://en.wikipedia.org/wiki/Database

[6] https://en.wikipedia.org/wiki/Database_model

[7] https://www.datastax.com/blog/evolution-nosql

[8] https://www.ibm.com/topics/cap-theorem

[9] https://www.talend.com/resources/what-is-mapreduce/

[10] https://en.wikipedia.org/wiki/NoSQL#Types_and_examples

[11] https://www.guru99.com/nosql-tutorial.html

[12] https://www.mongodb.com/nosql-explained

[13] https://www.imperva.com/learn/application-security/nosql-injection/

[14] https://book.hacktricks.xyz/pentesting-web/nosql-injection

[15] https://medium.com/rangeforce/nosql-injection-6514a8db29e3

[16] https://pandorafms.com/blog/nosql-vs-sql-key-differences/

[17] https://www.integrate.io/blog/the-sql-vs-nosql-difference/

[18] https://www.influxdata.com/key-value-database/

[19] https://www.techtarget.com/searchdatamanagement/tip/NoSQL-database-types-explained-Column-oriented-databases

[20] https://www.mongodb.com/document-databases

[21] https://towardsdatascience.com/introduction-to-nosql-graph-databases-fb2feac7a36

[22] https://db-engines.com/en/system/Bangdb

[23] https://github.com/sachin-sinha/BangDB

[24] https://github.com/sachin-sinha/BangDB

[25] https://en.wikipedia.org/wiki/Voldemort_(distributed_data_store)

[26] https://scholarworks.bridgeport.edu/xmlui/handle/123456789/1647

[27] https://www.project-voldemort.com/voldemort/

[28] https://en.wikipedia.org/wiki/Log-structured_merge-tree

[29] https://livetyping.com/en/portfolio/tarantool

[30] https://hbase.apache.org/

[31] https://thenewstack.io/a-look-at-hbase/

[32] https://en.wikipedia.org/wiki/Apache_Cassandra

[33] https://www.tutorialspoint.com/cassandra/cassandra_introduction.htm

[34] https://en.wikipedia.org/wiki/ScyllaDB

[35] https://www.techtarget.com/searchcloudcomputing/tip/Compare-NoSQL-database-types-in-the-cloud

[36] https://www.geeksforgeeks.org/introduction-to-nosql-cloud-database-services/

[37] https://journalofcloudcomputing.springeropen.com/nosqldatabase

[38] https://couchdb.apache.org/

[39] https://en.wikipedia.org/wiki/Apache_CouchDB

[40] https://docs.couchdb.org/en/3.2.2-docs/

[41] https://www.arangodb.com/

[42] https://en.wikipedia.org/wiki/ArangoDB

[43] https://github.com/arangodb/arangodb

[44] https://neo4j.com/developer/graph-db-vs-nosql/

[45] https://neo4j.com/blog/why-nosql-databases/

[46] https://neo4j.com/news/different-types-of-nosql-databases-and-when-to-use-them/

[47] https://neo4j.com/news/graph-databases-nosql-and-neo4j/

[48] https://www.trustradius.com/products/titan-graph-database/reviews#product-details

[49] https://github.com/distributedio/titan

[50] https://db-engines.com/en/system/AllegroGraph%3BMongoDB%3BOracle+NoSQL

[51] https://allegrograph.com/category/nosql/

[52] http://www.discoversdk.com/products/whitedb#/overview

[53] https://technologypoint.in/advantages-and-disadvantages-of-nosql-databases/

[54] https://www.geeksforgeeks.org/top-5-reasons-to-choose-nosql/

[55] https://www.opc-router.com/what-is-mongodb/

[56] https://medium.com/capital-one-tech/nosql-database-doesnt-mean-no-schema-a824d591034e

[57] https://en.wikipedia.org/wiki/Python_(programming_language)

[58] https://el.wikipedia.org/wiki/Java

[59] https://en.wikipedia.org/wiki/R_(programming_language)

[60] https://en.wikipedia.org/wiki/Scala_(programming_language)

[61] https://en.wikipedia.org/wiki/Cluster_analysis#Definition

[62] https://en.wikipedia.org/wiki/Association_rule_learning

[63] https://en.wikipedia.org/wiki/Data_mining

[64] https://www.academia.edu/57137600/%CE%97_%CE%B5%CF%86%CE%B1%CF%81%CE%BC%CE%BF%CE%B3%CE%AE_%CF%84%CF%89%CE%BD_%CE%BC%CE%B5%CE%B3%CE%AC%CE%BB%CF%89%CE%BD_%CE%B4%CE%B5%CE%B4%CE%BF%CE%BC%CE%AD%CE%BD%CF%89%CE%BD_Big_Data_%CF%83%CF%84%CE%B7%CE%BD_%CE%B1%CE%BD%CE%B8%CF%81%CF%89%CF%80%CE%B9%CF%83%CF%84%CE%B9%CE%BA%CE%AE_%CE%B2%CE%BF%CE%AE%CE%B8%CE%B5%CE%B9%CE%B1_%CF%85%CF%80%CF%8C_%CF%84%CE%BF_%CF%80%CF%81%CE%AF%CF%83%CE%BC%CE%B1_%CF%84%CE%BF%CF%85_%CE%93%CE%B5%CE%BD%CE%B9%CE%BA%CE%BF%CF%8D_%CE%9A%CE%B1%CE%BD%CE%BF%CE%BD%CE%B9%CF%83%CE%BC%CE%BF%CF%8D_%CE%B3%CE%B9%CE%B1_%CF%84%CE%B7%CE%BD_%CE%A0%CF%81%CE%BF%CF%83%CF%84%CE%B1%CF%83%CE%AF%CE%B1_%CE%94%CE%B5%CE%B4%CE%BF%CE%BC%CE%AD%CE%BD%CF%89%CE%BD

[65] https://en.wikipedia.org/wiki/Edgar_F._Codd

[66] https://softwareengineering.stackexchange.com/questions/200319/is-sql-declarative

[67] https://stackoverflow.com/questions/1619834/what-is-the-difference-between-declarative-and-procedural-programming-paradigms

[68] https://365datascience.com/tutorials/sql-tutorials/sql-declarative-language/

[69] https://www.geeksforgeeks.org/advantages-and-disadvantages-of-using-stored-procedures-sql/

[70] https://www.freecodecamp.org/news/connect-python-with-sql/

[71] https://en.wikipedia.org/wiki/SQLAlchemy

[72] https://onlineitguru.com/blog/how-do-java-and-sql-interact-with-each-other

[73] https://learnsql.com/blog/history-of-sql/

[74] https://www.softwebsolutions.com/resources/evolution-of-sql-the-journey-of-data.html