ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
«ΠΛΗΡΟΦΟΡΙΚΗ»

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αυτόματη εξαγωγή σχήματος και δημιουργία DOLAR prototypes από άγνωστες πηγές δεδομένων

Δημήτριος Σ. Ανανάς

Επιβλέπων: **Αλέξης Δελής,** Καθηγητής
Κώστας Σαΐδης, Επιτετραμμένος Λέκτορας

ΑΘΗΝΑ

ΝΟΕΜΒΡΙΟΣ 2024

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αυτόματη εξαγωγή σχήματος και δημιουργία DOLAR prototypes από άγνωστες πηγές δεδομένων

Δημήτριος Σ. Ανανάς
Α.Μ.: cs2200001

ΕΠΙΒΛΕΠΩΝ: **Αλέξης Δελής,** Καθηγητής
Κώστας Σαΐδης, Επιτετραμμένος Λέκτορας

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ: **Αλέξης Δελής,** Καθηγητής,
Γιαννης Σμαραγδάκης, Καθηγητής

Νοέμβριος 2024

ΠΕΡΙΛΗΨΗ

Καθώς ο όγκος των πληροφοριών διευρύνεται συνεχώς και η διασυνδεσιμότητα των εφαρμογών αυξάνεται, οι σύγχρονες εφαρμογές πρέπει να χειρίζονται τη συνεχή πίεση χρήστη/επιχειρήσεων για να υποστηρίζουν νέες πηγές δεδομένων και σύνολα δεδομένων όσο το δυνατόν πιο απρόσκοπτα. Η ενσωμάτωση διαφορετικών συνόλων δεδομένων από διάφορες πηγές είναι μια σημαντική πρόκληση, καθώς τα δεδομένα μπορεί να προέρχονται από δομημένες βάσεις δεδομένων SQL, web API, NoSQL βάσεις δεδομένων, μη δομημένο περιεχόμενο και άλλα.

Για να επιτρέψουμε στις εφαρμογές να προσαρμόζονται εύκολα σε νέα σχήματα, σε αυτή τη εργασία, βασιζόμαστε στο DOLAR framework για να δημιουργήσουμε αυτόματα DOLAR prototypes, από άγνωστες πηγές δεδομένων. Η προσέγγισή μας περιλαμβάνει δύο μεθόδους (α) δειγματοληψία δεδομένων, όπου το σύνολο δεδομένων διασχίζεται για τη συλλογή δειγμάτων των δεδομένων και τη χρήση αυτών των δειγμάτων για τον προσδιορισμό της δομής των prototypes (τα πεδία και τους τύπους τους) (β) χαρτογράφηση σχήματος, όπου το σχήμα δεδομένων αντιστοιχίζεται αυτόματα στα αντίστοιχα πρωτότυπα που δημιουργούνται εν κινήσει.

Στο πλαίσιο αυτής της εργασίας, διερευνούμε την αποτελεσματικότητα της προτεινόμενης προσέγγισης για σύνολα δεδομένων JSON: η υλοποίηση μας καλύπτει τη δειγματοληψία δεδομένων JSON και τη χαρτογράφηση σχήματος MongoDB. Για την αξιολόγηση των αποτελεσμάτων, εισάγουμε μια σειρά μετρήσεων που αφορούν, (α) τον αριθμό, (β) το όνομα (γ) τον τύπο των παραγόμενων πεδίων κάθε μοντέλου, καθώς και (δ) αναδρομικό έλεγχο, συγκρίνοντας prototypes που δημιουργούνται αυτόματα με prototypes που έχουν δημιουργηθεί χειροκίνητα. Τα πειράματα που πραγματοποιήθηκαν δείχνουν ότι τόσο οι μέθοδοι δειγματοληψίας όσο και οι μέθοδοι χαρτογράφησης μπορούν να ταιριάζουν με το σχήμα που δημιουργήθηκε με το χέρι, όταν υπάρχει αρκετή διαθέσιμη πληροφορία στο δείγμα. Μελλοντικές εργασίες θα μπορούσαν να διερευνήσουν τη δημιουργία DOLAR prototypes από πρόσθετους τύπους συνόλων δεδομένων και/ή μορφών δεδομένων.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Μοντελοποίηση δεδομένων

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Αφαιρέσεις δεδομένων, ολοκλήρωση πληροφορίας, αντιστοίχιση μοντέλων

ABSTRACT

As the amount of information constantly expands, and the inter-connectivity of applications increases, modern applications need to handle the constant user / business pressure to support novel data sources and datasets as seamlessly as possible. Incorporating diverse datasets from various sources is a significant challenge, as the data may originate from structured SQL databases, web APIs, NoSQL datastores, unstructured content, and more.

To enable applications to easily adapt to new schemas, in this thesis, we build upon the DOLAR framework to create DOLAR prototypes from unknown data sources automatically. Our approach includes two methods (a) data sampling, where the dataset is traversed to collect samples of the data and use these samples to determine the structure of the prototypes (the fields and their types) (b) schema mapping, where the dataset schema is automatically mapped to corresponding prototypes created on the fly.

In the scope of this thesis, we investigate the effectiveness of the proposed approach for JSON datasets: our implementation covers JSON data sampling and MongoDB schema mapping. To evaluate the results, our experimentation introduces an array of metrics, namely, (a) number, (b) name, (c) type of the produced fields of each model and (d) recursive validation comparing automatically generated prototypes with manually created prototypes. Conducted experiments show that both our sampling and mapping methods can match the manually created schema rather precisely, when sufficient information is available in the samples. Future work could explore the generation of DOLAR prototypes from additional types of datasets and/or data formats.

SUBJECT AREA: Data modelling

KEYWORDS: Data modeling, data abstraction, information integration, model mapping

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

1. ΕΙΣΑΓΩΓΗ

Στον τομέα της ολοκλήρωσης των δεδομένων και της διαλειτουργικότητας των εφαρμογών, η δυσκολία της ομαλής ενσωμάτωσης διαφορετικών δεδομένων από διάφορες πηγές αποτελεί σημαντική πρόκληση για τις εφαρμογές. Αυτό γίνεται ακόμη πιο δύσκολο όταν πρέπει να εξασφαλίσουμε ότι τα δεδομένα παραμένουν συμβατά και εύκολα προσβάσιμα. Στη σημερινή εποχή που βασίζεται τόσο στα δεδομένα, από πολλές πηγές πληροφοριών,με δομημένες βάσεις δεδομένων και μη, όπως αρχεία κειμένου και API ιστού, η πολυπλοκότητα της ενσωμάτωσης αυτών των διαφορετικών συνόλων δεδομένων αποτελεί σημαντική δυσκολία. Η ανάγκη διευκόλυνσης της επικοινωνίας και της συνεργασίας μεταξύ διαφορετικών μορφών και δομών δεδομένων είναι πρωταρχικής σημασίας για την αποτελεσματική λήψη αποφάσεων και την βελτίωση των διαδικασιών. Επιπλέον, καθώς ο όγκος και η ποικιλία των δεδομένων συνεχίζουν να αυξάνονται εκθετικά, η ικανότητα ένωσης και εναρμόνισης δεδομένων από διαφορετικές πηγές γίνεται όλο και πιο κρίσιμη. Χωρίς αποτελεσματικές μεθόδους για τη διαλειτουργικότητα δεδομένων, οι οργανισμοί αντιμετωπίζουν τον κίνδυνο να χάσουν χρήσιμα δεδομένα,ή να αφιερώσουν μεγάλο χρονικό διάστημα για την εναρμόνιση τους. Έτσι, η αντιμετώπιση αυτών των προκλήσεων απαιτεί όχι μόνο τεχνικές λύσεις αλλά και ολοκληρωμένη κατανόηση των βασικών αρχών της μοντελοποίησης δεδομένων, των τεχνικών ενοποίησης τους και του εξελισσόμενου τοπίου των προτύπων και των μορφών δεδομένων.

Εκτός από την πρόκληση της ενσωμάτωσης γνωστών συνόλων δεδομένων, οι σύγχρονες εφαρμογές πρέπει επίσης να είναι προετοιμασμένες να χειρίζονται και άγνωστα μοντέλα δεδομένων. Αυτή η απαίτηση προκύπτει από τη δυναμική φύση των πηγών δεδομένων, όπου νέοι τύποι πληροφοριών ενδέχεται να εμφανιστούν απροσδόκητα. Ως εκ τούτου, οι αρχιτεκτονικές των εφαρμογών πρέπει να είναι ευέλικτες και προσαρμόσιμες, ικανές να φιλοξενούν διαφορετικές μορφές και μοντέλα δεδομένων χωρίς προηγούμενη γνώση ή μη αυτόματη παρέμβαση.

Αυτό απαιτεί την ανάπτυξη ισχυρών τεχνικών μοντελοποίησης δεδομένων που μπορούν να προσαρμοστούν δυναμικά σε νέα σχήματα και μορφές δεδομένων, επιτρέποντας στις εφαρμογές να απορροφούν, να επεξεργάζονται και να αναλύουν ροές δεδομένων απρόσκοπτα. Επιπλέον, η δυνατότητα αυτόματης ανακάλυψης και ερμηνείας άγνωστων συνόλων δεδομένων είναι απαραίτητη για τη διασφάλιση της επεκτασιμότητας και της ανθεκτικότητας των εφαρμογών που βασίζονται σε δεδομένα. Με την ενσωμάτωση μηχανισμών για το χειρισμό άγνωστων συνόλων δεδομένων, οι εφαρμογές μπορούν να προστατευτούν στο μέλλον έναντι απρόβλεπτων αλλαγών στις πηγές και τις μορφές δεδομένων, διασφαλίζοντας έτσι τη συνεχή αποτελεσματικότητά τους σε ένα συνεχώς μεταβαλλόμενο περιβάλλον.

Ωστόσο, τα δεδομένα σπάνια υπάρχουν μεμονωμένα. Συνήθως υπάρχουν ως μέρος πιο σύνθετων σχέσεων. Ως εκ τούτου, η επόμενη φάση της ανακάλυψης μοντέλων δεδομένων περιλαμβάνει τη διάκριση και την οριοθέτηση αυτών των σχέσεων, διευκρινίζοντας πώς τα ανόμοια στοιχεία δεδομένων συσχετίζονται και αλληλεπιδρούν μεταξύ τους.

Σε όλη αυτή την «εξερεύνηση», αναδύονται μοτίβα, προσφέροντας πολύτιμες γνώσεις για τη συμπεριφορά και τη δομή των δεδομένων. Αυτά τα μοτίβα όχι μόνο βοηθούν την κατανόηση των δεδομένων, αλλά παρέχουν επίσης καθοδήγηση για την αποκρυπτογράφηση της λογικής και της οργάνωσής τους.

Επιπλέον, η ανακάλυψη μοντέλων δεδομένων απαιτεί αξιολόγηση της ποιότητας των δεδομένων, που περιλαμβάνει παράγοντες όπως η πληρότητα, η ακρίβεια, και η

συνέπεια. Αυτή η διαδικασία αξιολόγησης είναι απαραίτητη για τη διασφάλιση της αξιοπιστίας και της ακεραιότητας των επακόλουθων αναλύσεων δεδομένων και προσπαθειών μοντελοποίησης.

Είναι σημαντικό ότι η ανακάλυψη μοντέλων δεδομένων είναι ένα επαναληπτικό ταξίδι, που χαρακτηρίζεται από συνεχή εξερεύνηση, ανάλυση και τελειοποίηση. Οι αναλυτές μπορεί να χρειαστεί να επανεξετάσουν και να αναθεωρήσουν την κατανόησή τους για το μοντέλο δεδομένων, καθώς συγκεντρώνουν πρόσθετες γνώσεις ή αντιμετωπίζουν νέες προκλήσεις.

Η εφαρμογή των προσεγγίσεων θα βασίζεται στο DOLAR framework [12],[13], το οποίο παρέχει υποστήριξη για το χειρισμό ετερογενών δεδομένων. Το DOLAR προσφέρει ένα σύνολο εργαλείων και βιβλιοθηκών για μοντελοποίηση δεδομένων, ενοποίηση και διαλειτουργικότητα, καθιστώντας το κατάλληλο για τους σκοπούς μας.

Στα πλαίσια της εργασίας αυτής, θα επικεντρωθούμε σε δύο προσεγγίσεις παραγωγής μοντέλων δεδομένων. Στην πρώτη προσέγγιση, θα εστιάσουμε στην δημιουργία σχήματος μέσω της δειγματοληψίας δεδομένων JSON (sampling). Σαρώνοντας όλα τα δεδομένα, θα εντοπίσουμε τους τύπους δεδομένων και τη σχέση τους. Στη δεύτερη προσέγγιση, θα προσπαθήσουμε να εξάγουμε ένα σχήμα μιας MongoDB βάσης δεδομένων. Αυτή η διαδικασία αποτελεί μια πιο σύνθετη διαδικασία και περιλαμβάνει την προσαρμογή των πληροφοριών που θα λάβουμε απο την (μη σχεσιακή) βάση δεδομένων σε μια “δομημένη” μορφή σχήματος. Τέλος, θα προσπαθήσουμε να περιορίσουμε την πληροφορία που λάβαμε ώστε να κρατήσουμε την απολύτως απαραίτητη. Και οι δύο προσεγγίσεις στοχεύουν στη συλλογή περιεκτικών πληροφοριών που είναι απαραίτητες για τη δημιουργία ενός συνεκτικού σχήματος. Συνδυάζοντας τεχνικές sampling και mapping, μπορούμε να κατασκευάσουμε ένα ενοποιημένο σχήμα που αντανακλά την ποικιλομορφία των υποκείμενων δεδομένων.

Για την αξιολόγηση των αποτελεσμάτων, θα χρησιμοποιήσουμε συγκεκριμένες μετρήσεις, με βάση τον αριθμό και τον τύπο των πληροφοριών που συλλέγονται, ενώ θα χρησιμοποιηθούν κάποια σχήματα που δημιουργήθηκαν χειροκίνητα για σύγκριση με τα δικά μας.

Η εργασία είναι δομημένη ως εξής:

Στο δεύτερο κεφάλαιο, θα ασχοληθούμε με τη **Μοντελοποίηση των δεδομένων**.Θα εμβαθύνουμε στις πλεο διαδεδομένες τεχνικές μοντελοποίησης δεδομένων, συμπεριλαμβανομένων αντικειμενοστρεφών, σχεσιακών, γραφικών προσεγγίσεων, RDF. Τέλος, θα αναλύσουμε τις δυνατότητες μοντελοποίησης των δεδομένων που παρέχει το DOLAR.

Στο τρίτο κεφάλαιο, θα μιλήσουμε για τις **μορφές ανακάλυψης μοντέλων δεδομένων** στις οποίες θα εστιάσουμε στα πλαίσια αυτής της εργασίας. Περιλαμβάνει τεχνικές δειγματοληψίας και αντιστοίχισης. Εξετάζουμε την αποτελεσματικότητα αυτών των μεθόδων για την προσαρμογή άγνωστων συνόλων δεδομένων και τη διευκόλυνση της διαλειτουργικότητας μεταξύ των εφαρμογών.

Στο τέταρτο κεφάλαιο, θα περιγράψουμε λεπτομερώς την **υλοποίηση** της προτεινόμενης προσέγγισης μας σχετικά με τη **JSON δειγματοληψία**, επισημαίνοντας τις τεχνολογίες και τις μεθοδολογίες που χρησιμοποιούνται για την ενοποίηση διαφορετικών συνόλων δεδομένων.

Στο πέμπτο κεφάλαιο, συνεχίζουμε την περιγραφή της **υλοποίησης** μας εστιάζοντας στο κομμάτι της **MongoDB αντιστοίχισης**.

Στο έκτο κεφάλαιο, πραγματοποιούμε ολοκληρωμένες **δοκιμές και πειράματα** για να αξιολογήσουμε την απόδοση και την αποτελεσματικότητα των λύσεων που εφαρμόζουμε. Χρησιμοποιούμε μετρήσεις για την ποσοτική αξιολόγηση της αποτελεσματικότητας των προσεγγίσεων μας για την επίτευξη απρόσκοπτης ενοποίησης δεδομένων.

Στο έβδομο κεφάλαιο παρουσιάζουμε σχετικές εργασίες και βιβλιογραφία.

Στο ογδοο κεφάλαιο συνοψίζουμε τα ευρήματά μας, εξάγουμε **συμπεράσματα** με βάση τα ερευνητικά μας αποτελέσματα και προτείνουμε τρόπους για μελλοντική έρευνα και ανάπτυξη στον τομέα της μοντελοποίησης και της ενοποίησης των δεδομένων, καθώς και της διαλειτουργικότητας των εφαρμογών.

2. Μοντελοποίηση Δεδομένων

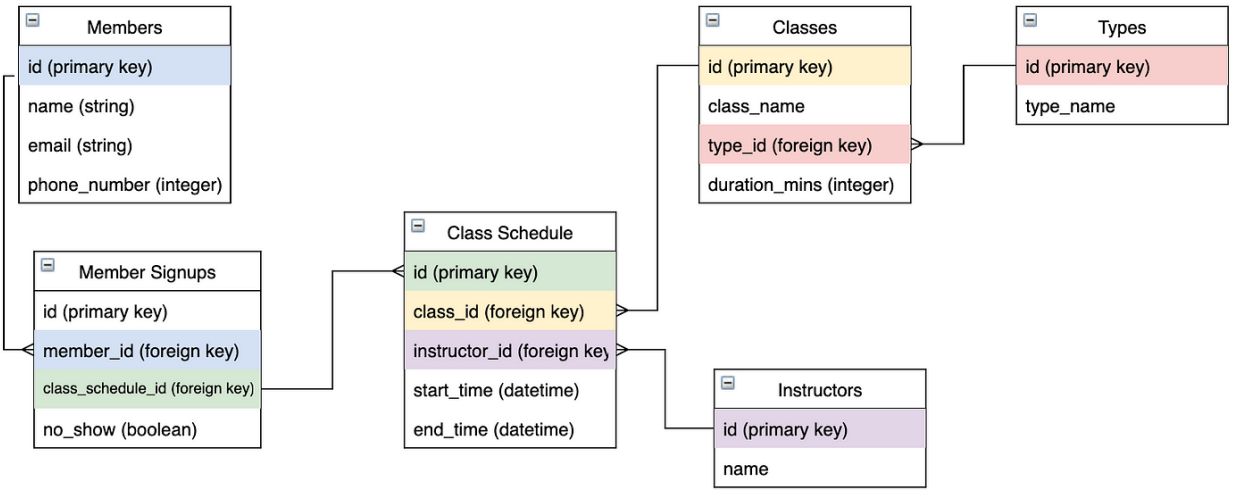
Αυτό το κεφάλαιο παρέχει μια σύνοψη των πιο κοινών τεχνικών μοντελοποίησης δεδομένων που εφαρμόζονται στην ανάπτυξη πληροφοριακών συστημάτων (όπως το Σχεσιακό μοντέλο, οι Γράφοι, το RDF και το Αντικειμενοστραφές μοντέλο) και καταλήγει με την παρουσίαση του DOLAR framework, το οποίο αποτέλεσε τη βάση για την υλοποίηση αυτής της εργασίας.

2.1 Σχεσιακό Μοντέλο (Relational Model)

Η μοντελοποίηση δεδομένων με σχεσιακό τρόπο είναι μια μεθοδολογία με κύρια εφαρμογή στο σχεδιασμό βάσεων δεδομένων, παρέχοντας ένα δομημένο πλαίσιο για την οργάνωση και τη διαχείριση δεδομένων στα συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS) [2] [5]. Ακολουθεί μια λεπτομερής εξερεύνηση των βασικών εννοιών, αρχών, πλεονεκτημάτων και περιορισμών του:

Βασικές Έννοιες:

- Πίνακες: Στο επίκεντρο της μοντελοποίησης σχεσιακών δεδομένων, οι πίνακες αντιπροσωπεύουν οντότητες ή σχέσεις εντός του τομέα ενδιαφέροντος. Κάθε πίνακας περιλαμβάνει γραμμές και στήλες, όπου οι γραμμές αντιπροσωπεύουν μεμονωμένες εγγραφές ή παρουσίες και οι στήλες αντιπροσωπεύουν χαρακτηριστικά ή ιδιότητες αυτών των εγγραφών.
- Κλειδιά: Οι σχέσεις μεταξύ των πινάκων δημιουργούνται μέσω κλειδιών, κυρίως πρωτευόντων κλειδιών (primary keys) και ξένων κλειδιών (foreign keys). Τα πρωτεύοντα κλειδιά προσδιορίζουν μοναδικά κάθε εγγραφή σε έναν πίνακα, ενώ τα ξένα κλειδιά δημιουργούν σχέσεις μεταξύ των εγγραφών σε διαφορετικούς πίνακες.



Εικόνα 1:Σχεδιακή Μοντελοποίηση Δεδομένων

Αρχές:

- Κανονικοποίηση: Θεμελιώδης αρχή στη μοντελοποίηση σχεσιακών δεδομένων, η κανονικοποίηση στοχεύει στην ελαχιστοποίηση του πλεονασμού και της εξάρτησης δεδομένων οργανώνοντας τα δεδομένα σε καλά δομημένους πίνακες.

Αυτή η διαδικασία περιλαμβάνει την αποσύνθεση μεγαλύτερων πινάκων σε μικρότερες, πιο διαχειρίσιμες οντότητες, διασφαλίζοντας την ακεραιότητα των δεδομένων και μειώνοντας τον κίνδυνο ανωμαλιών.

- Ιδιότητες ACID: Οι σχεσιακές βάσεις δεδομένων συμμορφώνονται με τις αρχές του ACID (Atomicity, Consistency, Isolation, Durability δηλαδή Ατομικότητα, Συνέπεια, Απομόνωση και Ανθεκτικότητα) για να διασφαλίζουν την ακεραιότητα και την αξιοπιστία των συναλλαγών. Αυτές οι ιδιότητες εγγυώνται ότι οι ενέργειες της βάσης δεδομένων επεξεργάζονται αξιόπιστα και με συνέπεια, ακόμη και σε περίπτωση αστοχιών ή σφαλμάτων του συστήματος.

Πλεονεκτήματα:

- Απλότητα: Η μοντελοποίηση σχεσιακών δεδομένων προσφέρει μια απλή και διαισθητική δομή, καθιστώντας εύκολη την κατανόηση και τη διαχείριση για προγραμματιστές και διαχειριστές βάσεων δεδομένων.
- Ισχυρές δυνατότητες ερωτημάτων (query): Οι σχεσιακές βάσεις δεδομένων παρέχουν ισχυρές δυνατότητες ερωτημάτων μέσω SQL (Structured Query Language), επιτρέποντας στους χρήστες να ανακτούν, να χειρίζονται και να αναλύουν αποτελεσματικά τα δεδομένα. Η “δηλωτική”φύση της SQL επιτρέπει στους χρήστες να εστιάζουν στον καθορισμό των δεδομένων που χρειάζονται, αντί στον τρόπο ανάκτησής τους.
- Ωριμότητα οικοσυστήματος: Οι σχεσιακές βάσεις δεδομένων διαθέτουν ένα ώριμο οικοσύστημα με εκτενή υποστήριξη, τεκμηρίωση και διαθέσιμα εργαλεία. Αυτή η ωριμότητα τα καθιστά κατάλληλα για ένα ευρύ φάσμα εφαρμογών, από έργα μικρής κλίμακας έως συστήματα εταιρικού επιπέδου.

Περιορισμοί:

- Πολυπλοκότητα με μεγάλα σύνολα δεδομένων: Καθώς αυξάνεται η πολυπλοκότητα των δεδομένων, η διαχείριση των σχέσεων μεταξύ πολλών πινάκων μπορεί να γίνει δύσκολη, οδηγώντας δυνητικά σε ζητήματα απόδοσης, όπως αργή εκτέλεση ερωτήματος ή διαμάχη (conflict) βάσης δεδομένων.
- Άκαμπτα σχήματα: Οι σχεσιακές βάσεις δεδομένων επιβάλλουν άκαμπτα σχήματα, τα οποία μπορεί να εμποδίσουν την ευελιξία στην αντιμετώπιση των εξελισσόμενων απαιτήσεων δεδομένων. Η τροποποίηση σχημάτων ή η προσθήκη νέων στηλών μπορεί να είναι περίπλοκη και μπορεί να απαιτήσει σημαντικό σχεδιασμό και χρόνο διακοπής λειτουργίας.

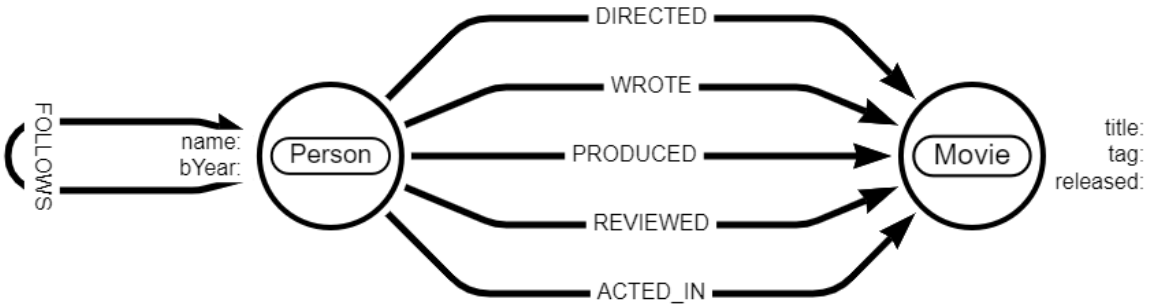
Συνοπτικά, η μοντελοποίηση σχεσιακών δεδομένων προσφέρει μια δομημένη προσέγγιση για την οργάνωση και τη διαχείριση δεδομένων. Τηρώντας βασικές αρχές και αξιοποιώντας τη δύναμη των σχεσιακών βάσεων δεδομένων, οι οργανισμοί μπορούν να δημιουργήσουν αποτελεσματικές και αξιόπιστες λύσεις αποθήκευσης δεδομένων για να καλύψουν τις επιχειρηματικές τους ανάγκες. Παρά τους περιορισμούς του, η απλότητα, οι δυνατότητες αναζήτησης και το ώριμο οικοσύστημα των σχεσιακών βάσεων δεδομένων τις καθιστούν δημοφιλή επιλογή για ένα ευρύ φάσμα εφαρμογών στη βιομηχανία λογισμικού.

2.2 Γράφοι

Η μοντελοποίηση δεδομένων με γράφους είναι μια θεμελιώδης μεθοδολογία στο σχεδιασμό της βάσης δεδομένων, που εστιάζει στην αναπαράσταση και την οργάνωση των δεδομένων ως ένα δίκτυο διασυνδεδεμένων κόμβων και ακμών [10] [11]. Ας εμβαθύνουμε στις βασικές έννοιες, αρχές, πλεονεκτήματα και περιορισμούς της μοντελοποίησης δεδομένων γραφήματος:

Βασικές Έννοιες:

- Κόμβοι: Οι κόμβοι αντιπροσωπεύουν οντότητες ή αντικείμενα εντός του τομέα δεδομένων. Κάθε κόμβος περιέχει ιδιότητες που περιγράφουν την οντότητα που αντιπροσωπεύει.
- Ακμές: Οι ακμές ορίζουν τις σχέσεις μεταξύ των κόμβων. Αντιπροσωπεύουν συνδέσεις ή συσχετίσεις μεταξύ οντοτήτων και μπορούν να έχουν ιδιότητες που περιγράφουν τη φύση της σχέσης.
- Γράφημα: Ένα γράφημα αποτελείται από μια συλλογή κόμβων και ακμών, που σχηματίζουν ένα δίκτυο που αντιπροσωπεύει τη δομή των δεδομένων και τις σχέσεις μεταξύ οντοτήτων.



Εικόνα 2: Γράφος

Αρχές:

- Δομή γραφήματος: Η μοντελοποίηση δεδομένων γραφήματος δίνει έμφαση στη δομή των σχέσεων μεταξύ οντοτήτων. Οι σχέσεις είναι σαφείς και αναπαρίστανται άμεσα ως ακμές μεταξύ κόμβων, επιτρέποντας πλούσια και πολύπλοκα μοντέλα δεδομένων.
- Διέλευση: Οι βάσεις δεδομένων γραφημάτων υποστηρίζουν αποτελεσματική διέλευση της δομής του γραφήματος, επιτρέποντας ισχυρές δυνατότητες αναζήτησης για πλοήγηση σχέσεων και ανακάλυψη μοτίβων μέσα στα δεδομένα.

Πλεονεκτήματα:

- Μοντελοποίηση με επίκεντρο τις σχέσεις: Η μοντελοποίηση δεδομένων γραφήματος υπερέχει στην αναπαράσταση και την αναζήτηση δεδομένων πλούσιων σε σχέσεις, όπως κοινωνικά δίκτυα, συστήματα προτάσεων. Οι σχέσεις είναι η πρώτη προτεραιότητα στις βάσεις δεδομένων γραφημάτων, επιτρέποντας αποτελεσματική διέλευση και ανάλυση.

- Απόδοση ερωτήματος: Οι βάσεις δεδομένων γραφημάτων προσφέρουν αποτελεσματική απόδοση για ερωτήματα (queries) που βασίζονται σε σχέσεις. Οι αλγόριθμοι διέλευσης, όπως η αναζήτηση κατά βάθος και η αναζήτηση κατά πλάτος, επιτρέπουν γρήγορη πλοήγηση στη δομή του γραφήματος, ακόμη και για μεγάλα και πολύπλοκα σύνολα δεδομένων.
- Ευελιξία σχήματος: Το ευέλικτο σχήμα βάσεων δεδομένων γραφημάτων διευκολύνει την ευέλικτη ανάπτυξη και επανάληψη. Οι προγραμματιστές μπορούν να προσαρμόσουν το μοντέλο δεδομένων στις εξελισσόμενες απαιτήσεις χωρίς να χρειάζονται πολύπλοκες μετεγκαταστάσεις σχημάτων (schema migration) ή διακοπές λειτουργίας.

Περιορισμοί:

- Επεκτασιμότητα: Ενώ οι βάσεις δεδομένων γραφημάτων υπερέχουν στη μοντελοποίηση και την αναζήτηση δεδομένων πλούσιων σε σχέσεις, ενδέχεται να αντιμετωπίσουν προκλήσεις επεκτασιμότητας για ορισμένους τύπους ερωτημάτων ή συνόλων δεδομένων. Η απόδοση μπορεί να υποβαθμιστεί καθώς αυξάνεται το μέγεθος και η πολυπλοκότητα του γραφήματος.
- Πολυπλοκότητα ερωτημάτων: Τα ερωτήματα περίπλοκων γραφημάτων που περιλαμβάνουν διελεύσεις μεγάλων γραφημάτων ή αντιστοίχιση μοτίβων σε πολλούς κόμβους και ακμές, ενδέχεται να απαιτούν εξελιγμένους αλγόριθμους και βελτιστοποιήσεις για την επίτευξη αποδεκτής απόδοσης.

Συνοπτικά, η μοντελοποίηση δεδομένων γραφήματος προσφέρει μια ισχυρή προσέγγιση για την αναπαράσταση και την αναζήτηση δεδομένων πλούσιων σε σχέσεις. Αξιοποιώντας δομές γραφημάτων και αλγόριθμους διέλευσης, οι οργανισμοί μπορούν να δημιουργήσουν αποτελεσματικά και ευέλικτα μοντέλα δεδομένων που αποτυπώνουν την πολυπλοκότητα των σχέσεων στον πραγματικό κόσμο. Ενώ οι βάσεις δεδομένων γραφημάτων υπερέχουν σε ορισμένες περιπτώσεις χρήσης, όπως τα κοινωνικά δίκτυα, ενδέχεται να αντιμετωπίσουν προκλήσεις επεκτασιμότητας για μεγάλα και πολύπλοκα σύνολα δεδομένων.

2.3 RDF

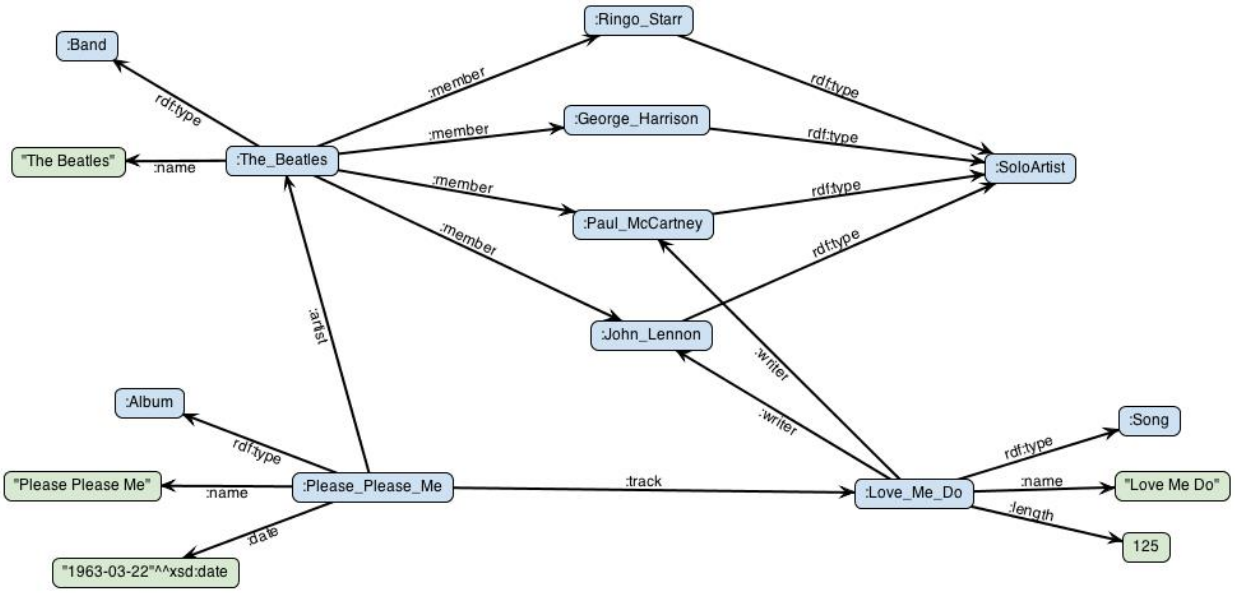
Το μοντέλο δεδομένων RDF είναι μια θεμελιώδης μεθοδολογία στις τεχνολογίες σημασιολογικού Ιστού, σχεδιασμένη για την αναπαράσταση και τη σύνδεση δεδομένων στον Ιστό [6] [7]. Ας διερευνήσουμε τις βασικές έννοιες, αρχές, πλεονεκτήματα και περιορισμούς του μοντέλου δεδομένων RDF:

Βασικές Έννοιες:

- Πόροι: Στο RDF, πόροι είναι οτιδήποτε μπορεί να αναγνωριστεί από ένα URI (Uniform Resource Identifier). Οι πόροι περιλαμβάνουν οντότητες, έννοιες ή πράγματα που περιγράφονται ή αναφέρονται σε δηλώσεις RDF.
- Τριπλέτες (triplets) : Τα βασικά δομικά στοιχεία του RDF είναι οι τριπλέτες, τα οποία αποτελούνται από υποκείμενο, κατηγόρημα και το αντικείμενο. Οι τριπλέτες αντιπροσωπεύουν δηλώσεις σχετικά με πόρους, με το υποκείμενο να

δηλώνει τον πόρο που περιγράφεται, το κατηγόρημα να αντιπροσωπεύει μια ιδιότητα ή σχέση και το αντικείμενο να αντιπροσωπεύει την τιμή ή έναν άλλο πόρο.

- Γραφήματα: Τα δεδομένα RDF αντιπροσωπεύονται τυπικά ως μια συλλογή διασυνδεδεμένων γραφημάτων, όπου κάθε γράφημα περιέχει ένα σύνολο τριπλών που περιγράφουν τις σχέσεις μεταξύ των πόρων.



Εικόνα 3: RDF

Αρχές:

- Συνδεδεμένα δεδομένα: Το RDF βασίζεται στην αρχή των συνδεδεμένων δεδομένων, η οποία προωθεί τη διασύνδεση δεδομένων στο διαδίκτυο μέσω της χρήσης τυποποιημένων μορφών και πρωτοκόλλων. Το RDF επιτρέπει τη δημιουργία δεσμών μεταξύ των πόρων, διευκολύνοντας την ανακάλυψη και την ενσωμάτωση σχετικών δεδομένων από διαφορετικές πηγές.
- Σημασιολογική διαλειτουργικότητα: Το RDF παρέχει ένα κοινό πλαίσιο για την αναπαράσταση και την ανταλλαγή δεδομένων σε μορφή αναγνώσιμη από μηχανή. Με την τήρηση των προτύπων RDF, οι δημιουργοί δεδομένων μπορούν να εξασφαλίσουν σημασιολογική διαλειτουργικότητα, επιτρέποντας στους καταναλωτές δεδομένων να κατανοήσουν και να ερμηνεύσουν το νόημα των δεδομένων.
- Επεκτασιμότητα: Το RDF έχει σχεδιαστεί για να κλιμακώνεται στο μέγεθος και την πολυπλοκότητα του ιστού, φιλοξενώντας μεγάλο όγκο δεδομένων και διαφορετικές πηγές δεδομένων. Η ευέλικτη δομή του που βασίζεται σε γραφήματα επιτρέπει την αναπαράσταση πολύπλοκων σχέσεων και την ενσωμάτωση δεδομένων από διαφορετικές πηγές.

Πλεονεκτήματα:

- Σημασιολογική αναπαράσταση: Το RDF επιτρέπει την αναπαράσταση δεδομένων σε σημασιολογική μορφή, όπου η σημασία των δεδομένων κωδικοποιείται σε

μορφή αναγνώσιμη από μηχανή. Αυτό διευκολύνει την αυτοματοποιημένη επεξεργασία, καθώς και την εξαγωγή ασφαλών συμπερασμάτων στα δεδομένα.

- Ενοποίηση συνδεδεμένων δεδομένων: Το RDF προωθεί την ενοποίηση δεδομένων από διαφορετικές πηγές μέσω της χρήσης τυποποιημένων αναγνωριστικών (URI) και μηχανισμών σύνδεσης. Αυτό επιτρέπει την ανακάλυψη και την εξερεύνηση διασυνδεδεμένων δεδομένων σε όλο τον ιστό.
- Ευελιξία: Η δομή που βασίζεται σε γράφημα του RDF παρέχει ευελιξία στην αναπαράσταση πολύπλοκων σχέσεων και εξελισσόμενων μοντέλων δεδομένων. Επιτρέπει την προσθήκη νέων ιδιοτήτων, σχέσεων και πηγών δεδομένων χωρίς να απαιτούνται αλλαγές σχήματος.

Περιορισμοί:

- Πολυπλοκότητα: Η μοντελοποίηση δεδομένων RDF μπορεί να είναι πολύπλοκη, ειδικά για χρήστες που δεν είναι εξοικειωμένοι με τις σημασιολογικές τεχνολογίες ιστού. Η κατανόηση των εννοιών RDF όπως τα URI, οι τριπλετες και τα γραφήματα μπορεί να απαιτεί μια καμπύλη εκμάθησης για τους νεοφερμένους.
- Απόδοση ερωτήματος: Καθώς τα σύνολα δεδομένων RDF αυξάνονται σε μέγεθος και πολυπλοκότητα, η απόδοση του ερωτήματος (query) μπορεί να μειωθεί αισθητά. Πολύπλοκα ερωτήματα που περιλαμβάνουν μεγάλα γραφήματα RDF ενδέχεται να απαιτούν τεχνικές βελτιστοποίησης για την επίτευξη αποδεκτών επιδόσεων.

Συνοπτικά, το μοντέλο δεδομένων RDF παρέχει ένα ισχυρό πλαίσιο για την αναπαράσταση και τη σύνδεση δεδομένων στον ιστό. Με τη μόχλευση τυποποιημένων μορφών, μηχανισμών σύνδεσης και σημασιολογικών αναπαραστάσεων, το RDF επιτρέπει την ενοποίηση, την ανακάλυψη και την ερμηνεία δεδομένων σε διάφορες πηγές και τομείς. Ενώ το RDF προσφέρει πολλά πλεονεκτήματα για την ενοποίηση δεδομένων και τη διαλειτουργικότητα, οι χρήστες θα πρέπει να προσέχουν την πολυπλοκότητά του και τις πιθανές επιδόσεις του όταν εργάζονται με μεγάλα ή πολύπλοκα σύνολα δεδομένων.

2.4 Αντικειμενοστραφές Μοντέλο(Object Oriented Model)

2.4.1 Κύρια χαρακτηριστικά

Στον τομέα της πληροφορικής και της μηχανικής λογισμικού, η δομή των δεδομένων είναι θεμελιώδης για τη δημιουργία επεκτάσιμων και αποτελεσματικών λύσεων λογισμικού. Η αντικειμενοστραφής μοντελοποίηση δεδομένων αποτελεί βασική μεθοδολογία σε αυτόν τον τομέα, παρέχοντας μια συστηματική προσέγγιση για το σχεδιασμό και την εφαρμογή συστημάτων λογισμικού[1][2][14]. Ας περιγράψουμε εν συντομία τις έννοιες που θα διερευνήσουμε:

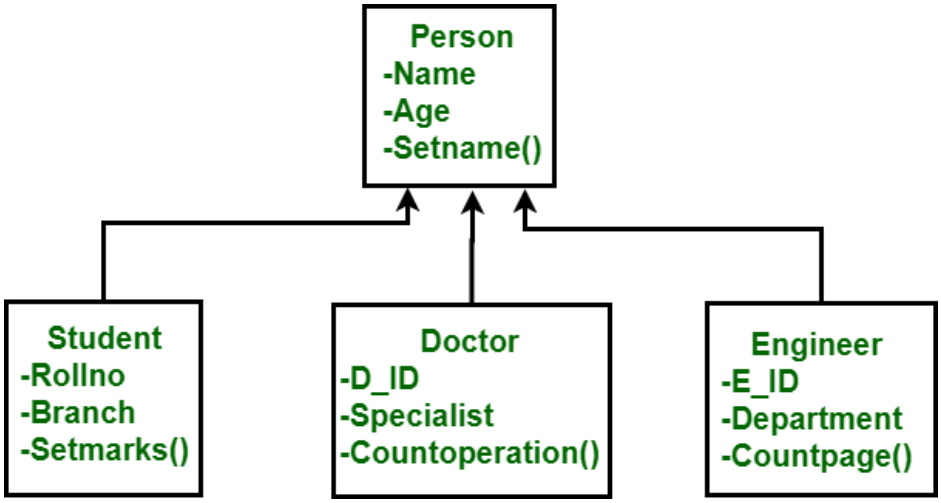
Ενθυλάκωση (Encapsulation): Η ενθυλάκωση χρησιμεύει ως θεμελιώδης αρχή στην αντικειμενοστραφή μοντελοποίηση δεδομένων, εστιάζοντας στη ομαδοποίηση δεδομένων και συμπεριφορών σε συνεκτικές μονάδες γνωστές ως αντικείμενα. Αυτό εξασφαλίζει περισσότερη ευελιξία, μειώνει την πολυπλοκότητα και ενισχύει την ακεραιότητα των δεδομένων στα συστήματα λογισμικού.

Κληρονομικότητα (Inheritance): Στην αντικειμενοστραφή μοντελοποίηση δεδομένων, η κληρονομικότητα διευκολύνει την επαναχρησιμοποίηση του κώδικα και δημιουργεί ιεραρχικές σχέσεις μεταξύ των στοιχείων λογισμικού. Αυτό επιτρέπει στις κλάσεις να κληρονομούν ιδιότητες και συμπεριφορές από γονικές κλάσεις, προωθώντας την επεκτασιμότητα και την οργάνωση κώδικα.

Πολυμορφισμός (Polymorphism): Ο πολυμορφισμός ενισχύει την ευελιξία και την επεκτασιμότητα στα συστήματα λογισμικού, επιτρέποντας την ομοιόμορφη αντιμετώπιση των αντικειμένων διαφορετικών κατηγοριών. Επιτρέπει δυναμικές αλληλεπιδράσεις και προσαρμοστικότητα, με αντικείμενα που παρουσιάζουν διαφορετικές συμπεριφορές με βάση τις συγκεκριμένες υλοποιήσεις κλάσης τους.

Αφαιρετικότητα(Abstraction): Η αφαιρετικότητα απλοποιεί τον σχεδιασμό λογισμικού εστιάζοντας σε βασικά χαρακτηριστικά ενώ κρύβει περιττές λεπτομέρειες. Επιτρέπει στους προγραμματιστές να δημιουργούν σαφή και συνοπτικά μοντέλα οντοτήτων του πραγματικού κόσμου, μειώνοντας το γνωστικό φορτίο και διευκολύνοντας την αποτελεσματική επικοινωνία μεταξύ των ομάδων ανάπτυξης.

Στοιχεία αντικειμενοστρεφούς μοντέλου δεδομένων:



Εικόνα 4: Αντικειμενοστραφής Μοντελοποίηση Δεδομένων

- **Αντικείμενα (Objects):** Ένα αντικείμενο είναι μια αφαιρετική εικόνα μιας οντότητας πραγματικού κόσμου ή μπορούμε να πούμε ότι είναι ένα παράδειγμα κλάσης. Το αντικείμενο ενσωματώνει δεδομένα και κώδικα σε μια ενιαία μονάδα που παρέχει δεδομένα με αφαιρετικό τρόπο, κρύβοντας τις λεπτομέρειες υλοποίησης από τον χρήστη.
- **Χαρακτηριστικα(Attributes):** Ένα χαρακτηριστικό περιγράφει τις ιδιότητες ενός αντικειμένου. Για παράδειγμα: Το αντικείμενο είναι STUDENT και τα χαρακτηριστικά του είναι Rollno, Branch, Setmarks() στην τάξη Student.
- **Μέθοδοι (Methods):** Η μέθοδος αντιπροσωπεύει τη συμπεριφορά ενός αντικειμένου. Με άλλα λόγια, αντιπροσωπεύει τη δράση του πραγματικού κόσμου.
- **Κλάση (Classes):** Μια κλάση είναι μια συλλογή παρόμοιων αντικειμένων με κοινή δομή, δηλαδή χαρακτηριστικά και συμπεριφορά, δηλαδή μεθόδους. Ένα αντικείμενο είναι ένα παράδειγμα κλάσης. Για παράδειγμα: Πρόσωπο, Φοιτητής, Γιατρός, Μηχανικός στο παραπάνω σχήμα.

Πλεονεκτήματα της αντικειμενοστραφούς μοντελοποίησης δεδομένων:

1. Ευελιξία και επαναχρησιμοποίηση: Τα εξαρτήματα μπορούν εύκολα να προσαρμοστούν και να επαναχρησιμοποιηθούν, βελτιώνοντας την αποδοτικότητα της ανάπτυξης.
2. Συντηρησιμότητα: Η ενθυλάκωση απλοποιεί τη βάση του κώδικα, καθιστώντας την ευκολότερη την κατανόηση και την τροποποίηση.
3. Επεκτασιμότητα: Τα συστήματα μπορούν να εξελίσσονται με την πάροδο του χρόνου χωρίς να θυσιάζονται οι επιδόσεις, εξυπηρετώντας την ανάπτυξη.
4. Καλύτερη συνεργασία: Παρέχει ένα κοινό πλαίσιο επικοινωνίας και επίλυσης προβλημάτων μεταξύ των ομάδων ανάπτυξης.

Μειονεκτήματα της αντικειμενοστραφούς μοντελοποίησης δεδομένων:

1. Πολυπλοκότητα: Τα συστήματα μεγάλης κλίμακας μπορεί να γίνουν δύσκολο να σχεδιαστούν και να διατηρηθούν λόγω αλληλεξαρτήσεων.
2. Καταλληλότητα εφαρμογής: Δεν μπορούν όλες οι εφαρμογές να επωφεληθούν από μια αντικειμενοστραφή προσέγγιση, ιδιαίτερα εκείνες που απαιτούν υψηλή απόδοση ή έλεγχο χαμηλού επιπέδου.
3. Δυσκολία εκμάθησης: Η μετάβαση από άλλα παραδείγματα μπορεί να απαιτεί σημαντική προσπάθεια για τους προγραμματιστές, επηρεάζοντας την παραγωγικότητα.

2.4.2 DOLAR

Το πλαίσιο DOLAR (Data Object Language And Runtime) προσφέρει μια ολοκληρωμένη λύση στις πολυπλοκότητες της επέκτασης της πληροφορίας [12] [13]. Με την εισαγωγή ενός εικονικού περιβάλλον χώρου πληροφοριών, το DOLAR αξιοποιεί τη δύναμη του αυτοματισμού και αφαίρεση για τη διευκόλυνση της ενσωμάτωσης νέων τύπων δεδομένων σε υπάρχουσες εφαρμογές. Αυτή η προσέγγιση μειώνει σημαντικά τις δυσκολίες επεκτασιμότητας, αυξάνοντας προσαρμοστικότητα του συστήματος.

Το DOLAR framework βασίζεται σε αντικειμενοστρεφείς αρχές και στοχεύει στην:

1. Απομόνωση της δομής των δεδομένων: Αυτό σημαίνει ευθυγράμμιση της λογικής οργάνωσης των δεδομένων με τις ανάγκες της εφαρμογής.
2. Προσαρμογή πρόσβασης φυσικών δεδομένων: Παρέχει συνδέσεις δικτύου ή βάσης δεδομένων σε αντικείμενα με τρόπο διαφανή για την εφαρμογή.
3. Παρουσίαση των αντικειμένων με αφηρημένο τρόπο: Ενσωματώνει απρόσκοπτα νέους τύπους αντικειμένων.
4. Απόκρυψη χειρισμού εσωτερικού αντικειμένου: Επιτρέπει την ομοιόμορφη τροποποίηση νέων αντικειμένων.

Με αυτόν τον τρόπο, το πλαίσιο DOLAR εκτελεί αυτές τις εργασίες άνετα και αποτελεσματικά, χωρίς να επιβάλλει σημαντικό χρόνο εκτέλεσης σε μια άκαμπτη, σκληρά κωδικοποιημένη υλοποίηση αυτών των χαρακτηριστικών.

2.4.2.1 Στοιχεία του DOLAR

Στα πλαίσια του DOLAR, εχουν δημιουργηθεί κάποια βασικά στοιχεία:

1. Τα εικονικά αντικείμενα DOLAR (DVO): είναι αντικείμενα εικονικού περιεχομένου που αντιπροσωπεύουν στοιχεία δεδομένων χωρίς εκτελέσιμο κώδικα, που περιλαμβάνουν σύνολα πεδίων, σχέσεων και σχήματα σύνθεσης (Stream Handles and Composition Schemes). Τα DVO απλοποιούν την εφαρμογή επιχειρηματικής λογικής εξαλείφοντας την δαπανηρή χειροκίνητη κωδικοποίηση, προσφέροντας εύκολες στην κατασκευή και διατήρηση προδιαγραφές που επιτρέπουν τη δημιουργία βοηθητικών προγραμμάτων καθορισμού δεδομένων για συγκεκριμένες εφαρμογές. Επιτρέπουν ομοιόμορφες διεπαφές συμβατές με υπηρεσίες για διαφορετικές δομές δεδομένων, διευκολύνοντας την απρόσκοπτη ενσωμάτωση νέων τύπων δεδομένων χωρίς τροποποιήσεις κώδικα μέσω σχημάτων σύνθεσης.
2. Ο Εικονικός Χώρος Πληροφοριών DOLAR, συμπεριλαμβανομένων των DVOStores, DVOIndexes και DOPSources οργανωμένων ιεραρχικά σε Domains DOLAR. Εξασφαλίζει αποτελεσματική και ομοιόμορφη πρόσβαση και τροποποίηση ετερογενούς περιεχομένου. Οι προγραμματιστές μπορούν να συνδέσουν DVO σε διάφορους χώρους αποθήκευσης δεδομένων, αυτοματοποιώντας εργασίες όπως π.χ. ο συγχρονισμός πρόσβασης.
3. Το ουδέτερο ως προς την υπηρεσία API DVO, ως βιβλιοθήκη κλάσης Java, προωθεί την επαναχρησιμοποίηση του DOLAR σε διαφορετικά περιβάλλοντα, υποστηρίζοντας τον χειρισμό δεδομένων άγνωστων εφαρμογών. Όπως τα συστήματα βάσεων δεδομένων, το DOLAR παρέχει ένα πλαίσιο γενικής χρήσης για την αποτελεσματική διαχείριση των χώρων πληροφοριών, την αφαίρεση λεπτομερειών για συγκεκριμένες υπηρεσίες για την προσαρμογή της ευέλικτης χρήσης εφαρμογών.

Στα πλαίσια της διπλωματικής αυτής εργασίας, θα εστιάσουμε λίγο περισσότερο στα εικονικά αντικείμενο DOLAR και στα DVO prototypes, τα οποία ορίζουν τις προδιαγραφές των DVOs.

2.4.2.2 DOLAR Prototypes

Τα DOLAR Prototypes επιτρέπουν στους σχεδιαστές να μοντελοποιούν ομοιόμορφα διάφορα ψηφιακά αντικείμενα, διασφαλίζοντας την αυτόματη συμμόρφωση με καθορισμένους τύπους. Τα Prototypes διευκολύνουν την ανάπτυξη ενοποιημένων υπηρεσιών Ψηφιακής Βιβλιοθήκης (DL) που βασίζονται στον ιστό, όπως η προσαρμοστική καταλογογράφηση, η μαζική απορρόφηση ψηφιακών αντικειμένων και οι αυτόματες μετατροπές περιεχομένου.

- Ορισμός: Τα DOLAR Prototypes είναι ορισμοί ψηφιακών τύπων αντικειμένων που προσδιορίζουν συστατικά μέρη και συμπεριφορές.
- Σκοπός: Διευκόλυνση ομοιόμορφης μοντελοποίησης διαφορετικών ψηφιακών αντικειμένων και διασφάλιση αυτόματης συμμόρφωσης με τον τύπο τους.
- Πλεονεκτήματα: Επιτρέπουν τη δημιουργία χρήσης ψηφιακών αντικειμένων που καθορίζονται από το χρήστη χωρίς προσαρμοσμένη ανάπτυξη, επιτρέποντας στις υπηρεσίες να λειτουργούν απευθείας σε όλους τους τύπους υλικού.

2.4.2.2.1 Στοιχεία των DOLAR prototypes:

1. Μεμονωμένα σύνολα μεταδεδομένων: Αναγνωρίζεται από ένα μοναδικό αναγνωριστικό και μια πολύγλωσση ετικέτα και περιγραφή.
2. Ειδικά στοιχεία: Ορίζεται από ένα αναγνωριστικό, επιθυμητές ετικέτες και περιγραφές και πρόσθετα χαρακτηριστικά συμπεριφοράς.
3. Αντιστοιχίσεις μεταξύ στοιχείων: Καθορίζει πιθανές αντιστοιχίσεις μεταξύ στοιχείων διαφόρων συνόλων μεταδεδομένων.

2.4.2.2.2 Σχέσεις σε DOLAR prototypes

Τα DVOP καθορίζουν τρεις τύπους σχέσεων:

1. Εσωτερικές σχέσεις: Τα ψηφιακά αντικείμενα αναφέρονται σε άλλα σχετικά αντικείμενα DL.
2. Δομικές σχέσεις: Μοντέλο σχέσεων «γονέα/παιδιού» μεταξύ ψηφιακών αντικειμένων που λειτουργούν ως δοχεία και των αντίστοιχων «παιδιών».
3. Εξωτερικές σχέσεις: Τα ψηφιακά αντικείμενα αναφέρονται σε εξωτερικές οντότητες παρέχοντας τις αντίστοιχες διευθύνσεις URL τους.

Ο αριθμός, η δομική σύσταση και οι σχέσεις των DVOPs είναι τα στοιχεία που θα μας απασχολήσουν κυρίως, στα πλαίσια αυτής της διπλωματικής εργασίας.

3. Αυτοματη Ανακάλυψη dolar μοντέλων

3.1 Ανάγκη για υποστήριξη πληροφορίας από νέες πηγές δεδομένων

Στην επιστήμη των υπολογιστών, η διαχείριση δεδομένων θέτει συνεχείς προκλήσεις καθώς ο όγκος τους αυξάνεται. Ο προσδιορισμός βασικών δεδομένων εν μέσω ανάπτυξης μπορεί να οδηγήσει σε πλεονάζουσα πληροφορία και ασυνέπειες στις δομές βάσεων δεδομένων που εξελίσσονται για να ανταποκρίνονται στις μεταβαλλόμενες απαιτήσεις. Ωστόσο, αυτή η εξέλιξη μπορεί να οδηγήσει σε αποκλίσεις μεταξύ του αρχικού σχήματος και της τρέχουσας κατάστασης της βάσης δεδομένων, μαζί με τη συσσώρευση παρωχημένων ή περιττών δεδομένων, εμποδίζοντας την αποτελεσματικότητα.

Παράλληλα, η εμφάνιση νέων μοντέλων δεδομένων, όπως οι βάσεις δεδομένων γράφων ή οι βάσεις δεδομένων που προσανατολίζονται σε έγγραφα, προσθέτει πολυπλοκότητα στη διαχείριση δεδομένων. Η μετάβαση από τις παραδοσιακές σχεσιακές βάσεις δεδομένων σε αυτά τα μοντέλα, καθώς και το αντίστροφο, απαιτεί προσεκτική μετεγκατάσταση και αναδιάρθρωση δεδομένων, διασφαλίζοντας τη συμβατότητα μεταξύ παλαιών και νέων μοντέλων αντιστοιχίζοντας τα χαρακτηριστικά και τις σχέσεις δεδομένων και ενημερώνοντας ανάλογα τα υπάρχοντα συστήματα.

Η μετάπτωση (migration) δεδομένων και η εξέλιξη του σχήματος είναι διαδικασίες που απαιτούν σχολαστικό σχεδιασμό και εκτέλεση. Η συντήρηση παραμένει ζωτικής σημασίας ακόμη και μετά τη μετεγκατάσταση, καθώς απαιτεί τακτικές αναθεωρήσεις και ενημερώσεις για να διατηρείται το μοντέλο δεδομένων ευθυγραμμισμένο με τις εξελισσόμενες επιχειρηματικές απαιτήσεις και τις τεχνολογικές εξελίξεις.

3.2 Αυτόματη παραγωγή DOLAR Μοντέλων

Στο πλαίσιο της εργασίας αυτής σκοπός μας είναι να δημιουργήσουμε και να “ανακαλύψουμε” την απαραίτητη πληροφορία από ένα άγνωστο σύνολο δεδομενων, με αυτόματο τρόπο. Παράλληλα, θέλουμε να προσδιορίσουμε τους τύπους αυτών των δεδομένων, καθώς και των τρόπο με τον οποίο η όλη πληροφορία είναι συνδεδεμένη.

Όπως έχουμε αναφέρει παραπάνω, οι προσεγγίσεις μας αφορούν την δειγματοληψία ενός JSON αρχείου και την αυτόματη αντιστοίχιση ενός MongoDB μοντέλου.

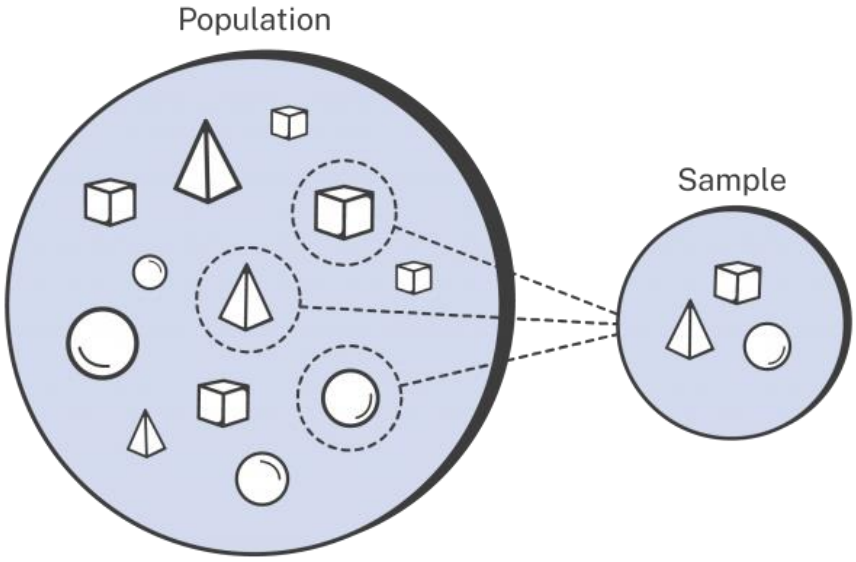
Και στις δύο περιπτώσεις, δίνεται εμφαση στον εντοπισμό του σωστού τύπου δεδομένων του κάθε πεδίου καθώς και στον προσδιορισμο των διασυνδέσεων του με τα υπόλοιπα στοιχεία.

3.2.1 Δειγματοληψία

Η δειγματοληψία είναι μια θεμελιώδης τεχνική στην ανάλυση δεδομένων που περιλαμβάνει την επιλογή ενός υποσυνόλου σημείων δεδομένων από έναν μεγαλύτερο πληθυσμό ή σύνολο δεδομένων. Ο στόχος της δειγματοληψίας είναι η εξαγωγή συμπερασμάτων για ολόκληρο τον πληθυσμό με βάση τις παρατηρήσεις από ένα υποσύνολο του δείγματος. Ουσιαστικά, η δειγματοληψία επιτρέπει στους ερευνητές και τους αναλυτές να μελετήσουν ένα διαχειρίσιμο τμήμα των δεδομένων, εξοικονομώντας έτσι χρόνο, πόρους και υπολογιστική ισχύ, ενώ παράλληλα αποκτούν σημαντικές γνώσεις.

Στο πλαίσιο της ανακάλυψης μοντέλων δεδομένων, η δειγματοληψία διαδραματίζει κρίσιμο ρόλο στην κατανόηση της υποκείμενης δομής και των προτύπων μέσα στο σύνολο δεδομένων. Αντί να αναλύεται ολόκληρο το σύνολο δεδομένων, το οποίο μπορεί να είναι απαγορευτικά μεγάλο και να απαιτεί πολλούς πόρους, η δειγματοληψία δίνει τη δυνατότητα στους ερευνητές να εστιάσουν τις προσπάθειές τους σε ένα αντιπροσωπευτικό υποσύνολο δεδομένων. Αυτό το υποσύνολο θα πρέπει να αντικατοπτρίζει με ακρίβεια τα χαρακτηριστικά και την κατανομή ολόκληρου του συνόλου δεδομένων, επιτρέποντας την αποτελεσματική ανάλυση και μοντελοποίηση.

Οι μεθοδολογίες δειγματοληψίας ποικίλλουν ανάλογα με τους συγκεκριμένους στόχους, τα χαρακτηριστικά του συνόλου δεδομένων και τους διαθέσιμους πόρους. Η επιλογή της μεθόδου δειγματοληψίας εξαρτάται από παράγοντες όπως το μέγεθος του πληθυσμού, το επιθυμητό επίπεδο ακρίβειας και η παρουσία διαστρωμάτωσης ή ομαδοποίησης εντός του συνόλου δεδομένων.



Εικόνα 5: Δειγματοληψία

Συνολικά, η δειγματοληψία αποτελεί μια πολύ σημαντική μέθοδος ανάλυσης δεδομένων, παρέχοντας ένα πρακτικό και αποτελεσματικό μέσο για την απόκτηση γνώσεων σχετικά με πολύπλοκα σύνολα δεδομένων, ενώ παράλληλα μετριάζει τις προκλήσεις που σχετίζονται με την ανάλυση μεγάλου όγκου δεδομένων. Χρησιμοποιώντας κατάλληλες τεχνικές δειγματοληψίας, οι ερευνητές μπορούν να αποκαλύψουν πολύτιμες πληροφορίες, να επικυρώσουν υποθέσεις και να ενημερώσουν τις διαδικασίες λήψης αποφάσεων σε διάφορους τομείς.

Στα πλαίσια της εργασίας αυτής, η δειγματοληψία θα πραγματοποιηθεί σε JSON αρχεία.

3.2.2 Αντιστοίχιση μοντέλων

Η αντιστοίχιση στη μοντελοποίηση δεδομένων αναφέρεται στη διαδικασία δημιουργίας σχέσεων μεταξύ διαφορετικών στοιχείων μέσα σε ένα μοντέλο δεδομένων. Αυτά τα στοιχεία θα μπορούσαν να περιλαμβάνουν οντότητες, χαρακτηριστικά, πίνακες, στήλες ή αντικείμενα, ανάλογα με τον τύπο του μοντέλου δεδομένων που χρησιμοποιείται.

Φανταστείτε τα δεδομένα σας ως ένα τεράστιο δίκτυο διασυνδεδεμένων κόμβων, που ο καθένας αντιπροσωπεύει μια συγκεκριμένη πληροφορία. Η αντιστοίχιση είναι η διαδικασία κατανόησης και καθορισμού των σχέσεων μεταξύ αυτών των κόμβων. Για παράδειγμα, σε ένα σύστημα διαχείρισης πελατειακών σχέσεων (CRM), ενδέχεται να έχει κόμβους που αντιπροσωπεύουν πελάτες, αγορές και αλληλεπιδράσεις. Η αντιστοίχιση θα καθόριζε τον τρόπο με τον οποίο κάθε πελάτης σχετίζεται με τις αγορές και τις αλληλεπιδράσεις του. Αυτή η διαδικασία αντιστοίχισης είναι ζωτικής σημασίας για διάφορους λόγους:

1. Ακεραιότητα δεδομένων: Η αντιστοίχιση διασφαλίζει ότι τα δεδομένα παραμένουν συνεπή και ακριβή σε διαφορετικά συστήματα και διαδικασίες. Ορίζοντας σαφείς σχέσεις μεταξύ των στοιχείων δεδομένων και μπορούν να αποτραπούν οι ασυνέπειες.

2. Αποτελεσματική διαχείριση δεδομένων: Η αντιστοίχιση επιτρέπει την αποτελεσματική ανάκτηση και χειρισμό δεδομένων. Όταν γνωρίζουμε πώς συνδέονται διαφορετικές οντότητες δεδομένων, μπορούμε να σχεδιάσουμε queries και αλγόριθμους για την πιο αποτελεσματική πλοήγηση στα δεδομένα.
3. Αποτελεσματική ενσωμάτωση πληροφορίας (integration): Στον σημερινό διασυνδεδεμένο κόσμο, οι οργανισμοί συχνά χρειάζεται να ενσωματώνουν δεδομένα από πολλαπλές πηγές. Η αντιστοίχιση διευκολύνει αυτήν την ενοποίηση παρέχοντας μια κοινή γλώσσα για την κατανόηση του τρόπου με τον οποίο τα διαφορετικά σύνολα δεδομένων σχετίζονται μεταξύ τους.
4. Καλύτερη κατανόηση: Η αντιστοίχιση συμβάλλει στη γεφύρωση του χάσματος μεταξύ των δομών τεχνικών δεδομένων και των επιχειρηματικών απαιτήσεων. Με την οπτικοποίηση των σχέσεων μεταξύ των στοιχείων δεδομένων, οι ενδιαφερόμενοι αποκτούν καλύτερη κατανόηση του τρόπου με τον οποίο τα δεδομένα μπορούν να αξιοποιηθούν για την υποστήριξη των επιχειρηματικών στόχων.

Στο πλαίσιο της εργασίας αυτής θα εστιάσουμε στη MongoDB, που είναι μια δημοφιλής βάση δεδομένων NoSQL που έχει επικρατήσει στον κόσμο της αποθήκευσης και διαχείρισης δεδομένων

3.3 Μετρικές Αποτελεσματικότητας δειγματοληψίας & αντιστοίχισης

Στα πλαίσια της εργασίας αυτής, χρειαζόμαστε κάποιες μετρητικές προκειμένου να βγάλουμε τα απαραίτητα συμπεράσματα για τις υλοποιήσεις μας.

Για να το κάνουμε αυτό, θα χρησιμοποιήσουμε το μοντέλα DVOP του DOLAR framework.

Συγκεκριμένα, θα πρέπει, αφού ολοκληρώσουμε την διαδικασία επιλογής της απαραίτητης πληροφορίας, να χρησιμοποιήσουμε το DOLAR προκειμένου να φτιάξουμε τα δικά μας Prototypes.

Για να έχουμε ακριβή εικόνα των αποτελεσμάτων μας, θα πρέπει να συγκρίνουμε αυτά που παρήχθησαν απο την υλοποίηση μας, με κάποια prototypes που θα δημιουργήσουμε η ίδιοι και θα περιέχουν την “σωστή” πληροφορία.

Για τον υπολογισμό της ομοιότητας των prototypes, έπρεπε να δημιουργηθούν συγκεκριμένες μετρητικές. Ιδιαίτερη σημασία έπρεπε να δοθεί στην κάλυψη όλων των δυνατών στοιχείων τα οποία τελικά επηρεάζουν το συνολικό αποτέλεσμα. Αποφασίστηκε να δοθεί σημασία σε τρεις βασικές κατηγορίες:

1. Τον αριθμό των πεδίων που βρίσκονται σε κάθε prototype. Στη συγκεκριμένη μετρητική, μας ενδιαφέρει απλά ένας αριθμός χωρίς περαιτέρω υπολογισμούς
2. Τα ονόματα των πεδίων. Η συγκεκριμένη μετρητική είναι πιο συγκεκριμένη απο την πρώτη, καθώς ελέγχει κατά πόσο τα πεδία του “source of truth” prototype υπάρχουν στο δημιουργημένο. Είναι επίσης υπερσύνολο της πρώτης μεθόδου, αφού εάν κάποιο πεδίο δεν υπάρχει (αν π.χ υπάρχουν λιγότερα πεδία), αυτό θα αποτυπωθεί και σε αυτή τη μετρητική.
3. Τους τύπους των πεδίων. Η συγκεκριμένη μετρητική είναι πιο συγκεκριμένη και απο τις δύο παραπάνω, καθώς εξαρτάται και από τον αριθμό αλλά και από τα ονόματα των πεδίων. Συγκρίνει τους τύπους των πεδίων του “source of truth” prototype σε σύγκριση με το δημιουργημένο.

Οι μετρήσεις των παραπάνω κατηγοριών χωρίζονται σε δύο υποκατηγορίες:

1. Την μέτρηση μόνο σε πρώτο επίπεδο. Σε αυτή την περίπτωση κάνουμε τους υπολογισμούς μόνο στο πρώτου επιπέδου prototype.
2. Την μέτρηση σε όλα τα επίπεδα. Στην περίπτωση αυτή, μετράμε αρχικα το πρώτου επιπέδου prototype, αλλά συνεχίζουμε και στα διασυνδεδεμένα με αυτό, προκειμένου να έχουμε πλήρη εικόνα.

4. Δειγματοληψία JSON δεδομένων για την αυτόματη παραγωγή DOLAR μοντέλων

Σε αυτό το κεφάλαιο, θα περιηγηθούμε στη διαδικασία εκτέλεσης της λειτουργίας δειγματοληψίας (Sampling) χρησιμοποιώντας ένα αρχείο JSON. Η δειγματοληψία είναι μια σημαντική διαδικασία που επιτρέπει την ανάλυση και την ανάκτηση δεδομένων από πολύπλοκες δομές JSON, επιτρέποντας την επιλογή αντιπροσωπευτικών υποσυνόλων για ανάλυση.

Σε αυτό το κεφάλαιο, θα εξηγήσουμε πώς λειτουργεί η δειγματοληψία και θα παρουσιάσουμε έναν λεπτομερή οδηγό για το πώς υλοποιήθηκε, χρησιμοποιώντας ένα αρχείο JSON. Θα αναλύσουμε τα βήματα που απαιτούνται για τη δημιουργία του ελάχιστου δυνατού συνόλου δεδομένων.

4.1 JSON

Στο πλαίσιο της εργασίας μας, στόχος μας είναι να διερευνήσουμε σχολαστικά τα περιεχόμενα ενός αρχείου JSON [3] για να εξαγάγουμε μόνο τις ζωτικής σημασίας πληροφορίες. Αυτό συνεπάγεται μια ολοκληρωμένη ανάλυση κάθε στοιχείου JSON για να διακρίνουμε τη συνάφειά του με τους ερευνητικούς μας στόχους. Στόχος μας είναι να διακρίνουμε και να απομονώσουμε τα στοιχεία που περιέχουν τα βασικά δεδομένα που απαιτούνται για τη μελέτη μας. Επιλέγοντας προσεκτικά αυτά τα στοιχεία, διασφαλίζουμε ότι η διαδικασία εξαγωγής δεδομένων μας είναι εστιασμένη και ευθυγραμμιζόμενη στενά με τους βασικούς στόχους της έρευνάς μας.

Το JSON (JavaScript Object Notation) είναι μια ελαφριά μορφή ανταλλαγής δεδομένων. Είναι εύκολο να τη διαβάσουν και να γράψουν οι άνθρωποι και εύκολο να αναλύσουν και να δημιουργήσουν οι μηχανές. Το JSON χρησιμοποιείται κυρίως για τη μετάδοση δεδομένων μεταξύ ενός διακομιστή και μιας εφαρμογής Ιστού, χρησιμεύοντας ως εναλλακτική της XML (eXtensible Markup Language).

Το JSON βασίζεται σε ένα υποσύνολο της γλώσσας προγραμματισμού JavaScript, αλλά είναι ανεξάρτητο από τη γλώσσα, που σημαίνει ότι μπορεί να χρησιμοποιηθεί σχεδόν με οποιαδήποτε γλώσσα προγραμματισμού. Αποτελείται από ζεύγη κλειδιών-τιμών όπου τα κλειδιά είναι συμβολοσειρές και οι τιμές μπορεί να είναι συμβολοσειρές, αριθμοί, πίνακες, αντικείμενα, booleans ή null.

4.1.1 Δομή, σύνταξη και χρήση JSON

Η απλότητα του JSON είναι μέρος της απηχησης του. Είναι εύκολο να γραφτεί, διαβάζεται και μεταφράζεται εύκολα μεταξύ των δομών δεδομένων που χρησιμοποιούνται από τις περισσότερες γλώσσες. Ας δούμε τι συνθέτει ένα αντικείμενο

JSON, τους τύπους δεδομένων που υποστηρίζει το JSON και άλλες λεπτομέρειες με τη σύνταξη αυτής της δημοφιλούς μορφής δεδομένων.

Τα αγκιστρα {} συγκρατούν αντικείμενα

Τα δεδομένα είναι σε ζεύγη κλειδιών, τιμών

Οι αγκύλες [] συγκρατούν πίνακες

Κάθε στοιχείο δεδομένων περικλείεται με εισαγωγικά εάν είναι χαρακτήρας ή χωρίς εισαγωγικά εάν είναι αριθμητική τιμή

```
{
    "name":"Katherine Johnson",
    "age":101,
    "orbital_mechanics": ["trajectories","launch
windows","emergency return paths"],
    "mathmatician": true,
    "last_location": null
}
```

Εικόνα 6: Δομή JSON

Τα κόμματα χρησιμοποιούνται για τον διαχωρισμό τμημάτων δεδομένων

4.1.2 Ποια είναι τα οφέλη του JSON;

Η αύξηση της δημοτικότητας του JSON συμπίπτει με την ανάγκη για ιστότοπους και εφαρμογές να μεταφέρουν πιο εύκολα και αποτελεσματικά δεδομένα από το ένα σύστημα στο άλλο. Ωστόσο, υπάρχουν πολλοί τρόποι με τους οποίους το JSON χρησιμοποιείται για κοινή χρήση δεδομένων, αποθήκευση ρυθμίσεων και αλληλεπίδραση με συστήματα. Η απλότητα και η ευελιξία του το καθιστούν εφαρμόσιμο σε πολλές διαφορετικές καταστάσεις.

Η πιο κοινή χρήση είναι η ανταλλαγή σειριακών δεδομένων μέσω σύνδεσης δικτύου. Ορισμένες άλλες συνήθεις χρήσεις του JSON περιλαμβάνουν δημόσια ή ιδιωτικά API, βάσεις δεδομένων NoSQL, περιγραφές σχημάτων, δημόσια δεδομένα ή εξαγωγές δεδομένων.

Τα οφέλη του JSON περιλαμβάνουν:

Συμπαγής, αποτελεσματική μορφή: Η σύνταξη JSON προσφέρει εύκολη ανάλυση δεδομένων και ακόμη πιο γρήγορη υλοποίηση.

Ευανάγνωστα: Τόσο οι άνθρωποι όσο και οι υπολογιστές μπορούν να ερμηνεύσουν γρήγορα τη σύνταξη με ελάχιστα σφάλματα.

Υποστηρίζεται ευρέως: Οι περισσότερες γλώσσες, λειτουργικά συστήματα και προγράμματα περιήγησης μπορούν να καταναλώνουν το JSON χωρίς μετατροπές, το οποίο επιτρέπει τη χρήση του χωρίς προβλήματα συμβατότητας.

Ξεκάθαροι τύποι δεδομένων: Είναι εύκολο να γίνει διάκριση μεταξύ τύπων δεδομένων και διευκολύνει την ερμηνεία των δεδομένων χωρίς να χρειάζεται να γνωρίζετε εκ των προτέρων.

Ευέλικτη μορφή: Το JSON υποστηρίζει ένα ευρύ φάσμα τύπων δεδομένων που μπορούν να συνδυαστούν για να εκφράσουν τη δομή των περισσότερων δεδομένων.

4.1.3 Πρακτικές εφαρμογές

Οι πρακτικές εφαρμογές και περιπτώσεις χρήσης του JSON είναι πολλές και ποικίλες. Ας εξετάσουμε μερικές από αυτές:

Μεταφορά Δεδομένων: Το JSON χρησιμοποιείται ευρέως για τη μεταφορά δεδομένων μεταξύ εφαρμογών. Οι υπηρεσίες web συχνά χρησιμοποιούν JSON για τη μεταφορά δεδομένων ανάμεσα στον client και τον server.

Σειριοποίηση Δεδομένων: Το JSON είναι κατάλληλο για τη σειριοποίηση και αποσειριοποίηση δεδομένων. Αυτό επιτρέπει την αποθήκευση και ανάκτηση δομημένων δεδομένων.

Ρύθμιση Συνδυασμών Δεδομένων: Συχνά χρησιμοποιείται για την αναπαράσταση συνδυασμών δεδομένων σε μορφή κλειδιού-τιμής.

Διαμοιρασμός Δομών Δεδομένων: Χρησιμοποιείται για τον απλό και κατανοητό διαμοιρασμό δομών δεδομένων μεταξύ διαφορετικών εφαρμογών.

Απόκτηση Δεδομένων από APIs: Οι υπηρεσίες API συχνά επιστρέφουν δεδομένα σε μορφή JSON. Αυτό καθιστά εύκολη την επεξεργασία και την ενσωμάτωση των δεδομένων σε διάφορες πλατφόρμες.

Διαμόρφωση Ρυθμίσεων: Το JSON συχνά χρησιμοποιείται για τη διαμόρφωση ρυθμίσεων εφαρμογών λογισμικού, καθώς παρέχει μια δομή που είναι εύκολα κατανοητή από τους προγραμματιστές και τους ανθρώπους.

Αυτές είναι μερικές μόνο από τις πολλές περιπτώσεις χρήσης του JSON. Η ευκολία στην ανάγνωση και συγγραφή του, καθώς και η δυνατότητα αναπαράστασης δομημένων δεδομένων, το καθιστούν ένα δημοφιλές μέσο ανταλλαγής πληροφοριών.

4.2 Σχεδιασμός και υλοποίηση συνάρτησης δειγματοληψίας με χρήση JSON

4.2.1 Πρώτη διαδικασία δειγματοληψίας

Το αρχικό βήμα της μεθοδολογίας περιλαμβάνει τη μετατροπή της δομής JSON σε οντότητες Groovy χρησιμοποιώντας ένα προϋπάρχον πακέτο. Αυτή η μετατροπή ουσιαστικά τροποποιεί τα δεδομένα ώστε να είναι εύκολη η ερμηνεία και η διαχείριση τους.

Μετά τη μετατροπή, ξεκινά η συστηματική διέλευση της δομής JSON, χρησιμοποιώντας αναδρομικές τεχνικές. Αυτή η διέλευση είναι απαραίτητη για την πλήρη ανάλυση των περιεχομένων του συνόλου δεδομένων και την κατηγοριοποίησή τους σε διακριτά μέρη.

Κατά τη διαδικασία διέλευσης, εξετάζεται κάθε αντικείμενο εντός της δομής JSON. Τα αντικείμενα που προσδιορίζονται ως τύπου “Map” θεωρούνται πιθανές οντότητες. Αυτές οι οντότητες καταχωρούνται και, στη συνέχεια, τα θυγατρικά τους στοιχεία αναλύονται και κατηγοριοποιούνται. Η αναδρομική φύση αυτής της διαδικασίας διασφαλίζει ότι όλα τα “child” στοιχεία διερευνώνται διεξοδικά και συνδέονται με τις μητρικές τους οντότητες.

Τα αντικείμενα που ταξινομούνται ως τύπος "List" αντιμετωπίζονται αρχικά ως “array”, δηλαδή μια συλλογή αντικειμένων ίδιου τύπου. Στη συνέχεια, επιχειρείται η εξερεύνηση

του του τύπου που εμπεριέχεται μέσα στη “List”. Κάθε στοιχείο στη λίστα καταχωρείται ως ξεχωριστό αντικείμενο και υποβάλλεται σε αναδρομική ανάλυση για κατηγοριοποίηση. Αυτό διασφαλίζει ότι η πληροφορία αντιπροσωπεύεται και συνδέεται σωστά μέσα στο σύνολο δεδομένων.

Για αντικείμενα που δεν ταιριάζουν στις κατηγορίες "Map" ή "List", θεωρούνται πρωτόγονα (primitive) αντικείμενα. Η αναγνώριση τύπου εκτελείται σε αυτά τα πρωτόγονα (π.χ. συμβολοσειρά, ακέραιος, long, boolean, ημερομηνία) και καταχωρούνται ως πεδία εντός προκαθορισμένων οντοτήτων.

Το τελικό αποτέλεσμα θα αποτελείται από έναν χάρτη, που θα περιέχει κάθε αντικείμενο (ή «μοντέλο») που βρέθηκε κατά τη διαδικασία δειγματοληψίας. Συνοπτικά, η μεθοδολογία συνδυάζει τεχνικές αναδρομικής διέλευσης με οντότητες Groovy για να αναλύει και να αναπαριστά συστηματικά δεδομένα JSON.

Πίνακας 1: JSON δειγματοληψία - Χειρισμός αντικειμένων

Τύπος αντικειμένου JSON	Χειρισμός, ή Αντιμετώπιση
Map	Καταγράφεται ως ξεχωριστή οντότητα. Τα θυγατρικά στοιχεια αναλύονται αναδρομικά.
Λίστα	Καταγράφεται κάθε στοιχείο ως ξεχωριστό αντικείμενο. Αναλύεται αναδρομικά.
Αλλα	Αντιμετωπίζεται ως ένα πρωτόγονο αντικείμενο. Προσδιορίζεται ο τύπος του και καταχωρείται ως πεδίο εντός της κατάλληλης οντότητας.

Αυτή η προσέγγιση διασφαλίζει μια ολοκληρωμένη κατανόηση της σύνθεσης και της ιεραρχίας του συνόλου δεδομένων, διευκολύνοντας την επακόλουθη ανάλυση και ερμηνεία. Ο παραπάνω πίνακας συνοψίζει τον χειρισμό διαφορετικών τύπων αντικειμένων JSON εντός της μεθοδολογίας. Αντικείμενα τύπου "Map" και "List" αντιμετωπίζονται διαφορετικά με βάση τη δομή τους, ενώ αντικείμενα που δεν ταιριάζουν σε αυτές τις κατηγορίες αντιμετωπίζονται ως πρωτόγονα αντικείμενα και επεξεργάζονται ανάλογα.

4.2.2 Εκκαθάριση περιττών πεδίων

Μόλις ολοκληρωθεί η αντιστοίχιση πεδίων και δημιουργηθεί ένας χάρτης όλων των αναγνωρισμένων πεδίων, είναι σημαντικό να αναγνωριστεί η πιθανότητα να συναντήσουμε διπλές καταχωρήσεις ή σχέσεις υποσυνόλου λόγω της διαδικασίας δειγματοληψίας. Αυτό σημαίνει ότι μπορεί να υπάρχουν περιπτώσεις όπου ορισμένα πεδία καταγράφονται πολλές φορές ή όπου ορισμένα πεδία είναι υποσύνολα άλλων. Για να διασφαλιστεί η ακρίβεια και η ακεραιότητα των δεδομένων, είναι απαραίτητο να διεξαχθεί μια λεπτομερής ανάλυση για τον εντοπισμό και την επίλυση τυχόν διπλότυπων ή υποσυνόλων που υπάρχουν στο σύνολο δεδομένων. Αυτό θα βοηθήσει στη βελτίωση του χάρτη και θα παρέχει μια πιο ακριβή αναπαράσταση του τοπίου των πεδίων.

4.2.2.1 Κατηγοριοποίηση αντικειμένων.

Το αρχικό βήμα στην αναλυτική μας διαδικασία περιλαμβάνει την κατηγοριοποίηση των αντικειμένων. Αυτή η διαδικασία ουσιαστικά συνεπάγεται τη σειριοποίηση κάθε αντικειμένου, μια διαδικασία που διευκολύνεται με την ταξινόμηση των πεδίων σε κάθε αντικείμενο σε αλφαβητική σειρά. Στη συνέχεια, η τιμή κατακερματισμού (hash value) για κάθε αντικείμενο υπολογίζεται προσθέτοντας επαναληπτικά το όνομα και τον τύπο κάθε πεδίου. Αυτή η μεθοδολογία έχει ως αποτέλεσμα τη δημιουργία μιας ολοκληρωμένης λίστας που περιλαμβάνει κλειδιά κατακερματισμού που αντιστοιχούν σε κάθε αντικείμενο.

Χρησιμοποιώντας αυτήν την τεχνική κατηγοριοποίησης, δημιουργούμε ένα δομημένο πλαίσιο που διευκολύνει την αποτελεσματική αναγνώριση και εκκαθάριση διπλών ή παρόμοιων αντικειμένων. Αυτή η οργανωμένη προσέγγιση απλοποιεί τη διαδικασία σύγκρισης, επιτρέποντας την ταχεία αναγνώριση αντικειμένων που μοιράζονται πανομοιότυπα ή παρόμοια χαρακτηριστικά. Κατά συνέπεια, η χρήση σειριακών κλειδιών κατακερματισμού χρησιμεύει ως πολύτιμος μηχανισμός για την ενίσχυση της σαφήνειας και της αποτελεσματικότητας των επόμενων αναλυτικών προσπαθειών.

4.2.2.2 Εκκαθάριση παρόμοιων πεδίων με χρήση hash value.

Στο πλαίσιο επεξεργασίας δεδομένων μας, η συγκεκριμένη διαδικασία παίζει καθοριστικό ρόλο στη βελτιστοποίηση της αναπαράστασης των αντικειμένων και στη μείωση του πλεονασμού εντός του συνόλου δεδομένων. Αυτή η μέθοδος λειτουργεί με τον εντοπισμό και την ενοποίηση παρόμοιων αντικειμένων με βάση τα μοναδικά αναγνωριστικά τους (hashes), βελτιστοποιώντας έτσι το σύνολο δεδομένων και ενισχύοντας τη συνοχή του.

Ο πρωταρχικός στόχος της διαδικασίας είναι η μείωση της πλεονάζουσας πληροφορίας εντός του συνόλου δεδομένων διατηρώντας παράλληλα βασικές πληροφορίες. Επιτυγχάνει αυτόν τον στόχο μέσω μιας σειράς βημάτων:

Type Normalization (κανονικοποίηση τύπου): Η μέθοδος φροντίζει ώστε κάθε πεδίο να έχει ένα συγκεκριμένο τύπο, με σκοπό τη διασφάλιση της συνέπειας στους τύπους δεδομένων μεταξύ των αντικειμένων. Αυτό το βήμα εναρμονίζει το σύνολο δεδομένων, αποφεύγοντας ασυνέπειες σε επόμενες λειτουργίες.

Αναγνώριση Μοναδικών Αντικειμένων: Η μέθοδος επαναλαμβάνεται μέσω της συλλογής αντικειμένων και χρησιμοποιώντας τις τιμές κατακερματισμού τους ως αναγνωριστικά. Για κάθε μοναδικό κατακερματισμό, επιλέγει την πρώτη εμφάνιση του αντίστοιχου αντικειμένου, ενοποιώντας πανομοιότυπα αντικείμενα και εξαλείφοντας τον πλεονασμό.

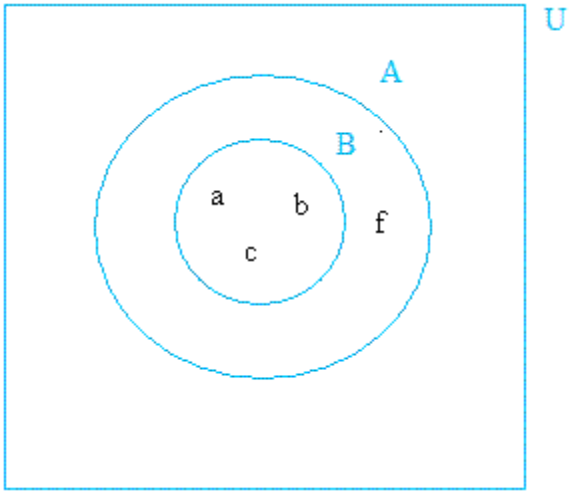
Βελτιστοποίηση περιορισμών (constraint optimization): Εκτός από την αφαίρεση αντικειμένων, η διαδικασία βελτιστοποιεί τους περιορισμούς (constraints) με πεδία αντικειμένων. Κάνοντας αναφορά μόνο της πρώτη εμφάνιση κάθε τύπου περιορισμού , η μέθοδος ελαχιστοποιεί τους περιττους, ενισχύοντας τη σαφήνεια και την αποτελεσματικότητα του συνόλου δεδομένων.

Κατασκευή Μοναδικών Αντικειμένων: Το μειωμένο σύνολο δεδομένων, που περιλαμβάνει μοναδικά αντικείμενα και βελτιστοποιημένους περιορισμούς, οργανώνεται σε μια συνεκτική δομή μοναδικών πρωτοτύπων. Αυτά τα πρωτότυπα χρησιμεύουν ως αντιπροσωπευτικά μοντέλα εντός του συνόλου δεδομένων, ενσωματώνοντας βασικά πεδία και χαρακτηριστικά.

Μέσω της συστηματικής μείωσης του πλεονασμού και της βελτιστοποίησης των περιορισμών, η διαδικασία ενισχύει τη συνοχή και την αποτελεσματικότητα των δεδομένων μας. Θέτει μια σταθερή βάση για μεταγενέστερη ανάλυση και ερμηνεία στο πλαίσιο της έρευνάς μας.

4.2.2.3 Αφαίρεση υποσυνόλων πεδίων

Η μέθοδος αφαίρεσης υποσυνολων αντικειμένων είναι ένα θεμελιώδες εργαλείο για τη βελτίωση των συλλογών αντικειμένων με την αφαίρεση περιττών καταχωρήσεων. Λειτουργεί μέσω μιας συστηματικής σύγκρισης αντικειμένων, εντοπίζοντας και εξαλείφοντας αυτά που περικλείονται εξ ολοκλήρου από άλλα.



Εικόνα 7: Υποσύνολο

Στόχος και Προσέγγιση: Ο πρωταρχικός στόχος της μεθόδου είναι να βελτιστοποιήσει την αναπαράσταση αντικειμένων εξαλείφοντας περιττές καταχωρήσεις. Αυτό επιτυγχάνεται με τα ακόλουθα βήματα:

1. Ολοκληρωμένη σύγκριση:
    - Η μέθοδος εξετάζει σχολαστικά κάθε αντικείμενο της συλλογής, συγκρίνοντάς το με άλλα.
    - Προσδιορίζει αντικείμενα που περιέχουν πλήρως το περιεχόμενο ενός άλλου, υποδεικνύοντας πλεονασμό.
2. Κατάργηση περιττών αντικειμένων:
    - Μόλις εντοπιστούν, τα περιττά αντικείμενα αφαιρούνται από τη συλλογή.
    - Οι σχετικοί περιορισμοί που συνδέονται με αντικείμενα που έχουν αφαιρεθεί καταργούνται επίσης για τη διατήρηση της ακεραιότητας των δεδομένων.

Συμβολή στη βελτίωση των δεδομένων: Με τη συστηματική κατάργηση περιττών αντικειμένων, η συγκεκριμένη διαδικασία βελτιώνει τη σαφήνεια και την αποτελεσματικότητα των συλλογών αντικειμένων. Διασφαλίζει ότι κάθε αντικείμενο συνεισφέρει μοναδικές πληροφορίες, διευκολύνοντας την πιο εστιασμένη και αποτελεσματική ανάλυση δεδομένων.

4.2.2.4 Συγχώνευση παρόμοιων αντικειμένων

Ο σκοπός αυτής της μεθόδου είναι να βελτιώσει τη συνοχή και την οργάνωση των δεδομένων μέσα στις συλλογές αντικειμένων με τη συγχώνευση αντικειμένων που μοιράζονται κοινά πεδία.

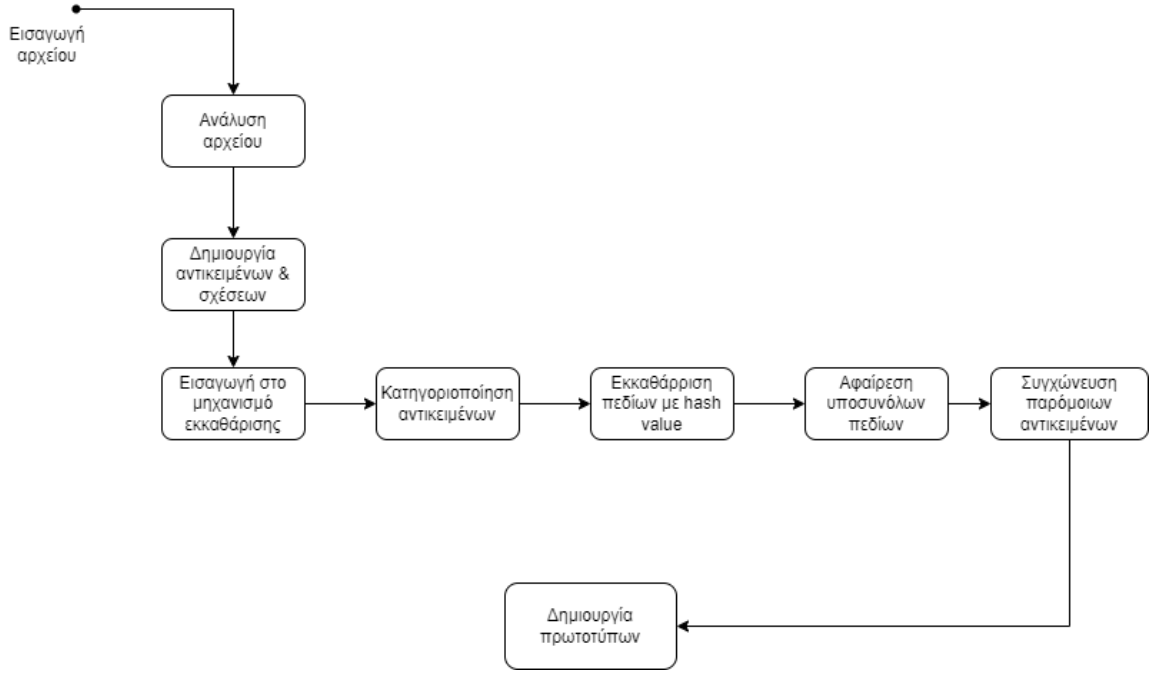
Για να επιτευχθεί αυτό, η μέθοδος εξετάζει συστηματικά τα πεδία κάθε αντικειμένου για να εντοπίσει κοινά σημεία. Στη συνέχεια συγχωνεύει τα δεδομένα που σχετίζονται με αυτά τα κοινά πεδία, μειώνοντας έτσι τον πλεονασμό και δημιουργώντας ένα πιο ενοποιημένο σύνολο δεδομένων.

Λεπτομερής Ανάλυση: Η ανάλυση ξεκινά με μια ολοκληρωμένη ανασκόπηση των πεδίων που υπάρχουν σε κάθε αντικείμενο. Μέσω αυτής της διαδικασίας, η μέθοδος προσδιορίζει πεδία που είναι κοινόχρηστα μεταξύ πολλών αντικειμένων, υποδεικνύοντας πιθανές περιοχές για ενοποίηση.

Μόλις εντοπιστούν κοινά πεδία, η μέθοδος συγχωνεύει προσεκτικά τα αντίστοιχα δεδομένα από σχετικά αντικείμενα. Αυτή η διαδικασία συγχώνευσης διασφαλίζει ότι το σύνολο δεδομένων διατηρεί μια περιεκτική αναπαράσταση πληροφοριών, ελαχιστοποιώντας παράλληλα την επικάλυψη.

Συνεισφορά στην τελειοποίηση των δεδομένων: Ενοποιώντας πληροφορίες από αντικείμενα με κοινά πεδία, η διαδικασία ενισχύει σημαντικά τη συνοχή και τη συνέπεια του συνόλου δεδομένων. Αυτή η βελτιστοποιημένη αναπαράσταση όχι μόνο απλοποιεί την ανάλυση δεδομένων αλλά προωθεί επίσης μια σαφέστερη κατανόηση της υποκείμενης δομής πληροφοριών.

Συνολικά, η συμβολή της μεθόδου στη βελτίωση των δεδομένων έγκειται στην ικανότητά της να ελαχιστοποιεί τον πλεονασμό διατηρώντας παράλληλα τον πλούτο του συνόλου δεδομένων. Μέσω της συστηματικής διαδικασίας συγχώνευσης, προωθεί την αποτελεσματικότερη εξερεύνηση και ανάλυση δεδομένων.



Εικόνα 8: Διάγραμμα ροής

4.2.3 Δημιουργία Πρωτοτύπων

Αφού ολοκληρωθεί η διαδικασία εκκαθάρισης και δημιουργίας όσο το δυνατόν πιο συνεκτικών και ολοκληρωμένων δεδομένων, ξεκινά η διαδικασία δημιουργίας των πρωτοτύπων.

Η μέθοδος απλοποιεί τη δημιουργία πρωτοτύπων για κάθε αντικείμενο (ObjectNode), διασφαλίζοντας μια δομημένη αναπαράσταση των δεδομένων. Στόχος του είναι να διατηρήσει τη συνέπεια και τη συνοχή στη συλλογή πρωτοτύπων, ενώ χειρίζεται αναφορές και περιορισμούς μέσα στα αντικείμενα.

Η μέθοδος επεξεργάζεται συστηματικά κάθε αντικείμενο για να δημιουργήσει το αντίστοιχο πρωτότυπο:

- Αρχικοποιεί το πρωτότυπο για το τρέχον αντικείμενο. Επαναλαμβάνοντας σε κάθε πεδίο, καθορίζει τον τύπο του πεδίου και τις σχετικές πληροφορίες.
- Για πεδία τύπου αναφοράς (ref), δημιουργεί αναδρομικά πρωτότυπα για αντικείμενα αναφοράς, επιλύοντας περιορισμούς.
- Τα πεδία προστίθενται στο πρωτότυπο με τις κατάλληλες πληροφορίες σχετικά με τον τύπο τους.
- Μόλις υποβληθούν σε επεξεργασία όλα τα πεδία, το πρωτότυπο σφραγίζεται, σηματοδοτώντας την ολοκλήρωση της διαδικασίας δημιουργίας.

Δημιουργώντας πρωτότυπα για κάθε ObjectNode, η μέθοδος εξασφαλίζει συνέπεια και συνοχή στη συλλογή πρωτοτύπων. Αντιπροσωπεύει με ακρίβεια τη δομή κάθε αντικειμένου, διαχειριζόμενη αποτελεσματικά τις αναφορές και τους περιορισμούς. Συνολικά, η μέθοδος συμβάλλει στη δημιουργία μιας δομημένης και συνεπούς αναπαράστασης δεδομένων, διευκολύνοντας την περαιτέρω ανάλυση και ερμηνεία εντός του πλαισίου επεξεργασίας δεδομένων.

5. Αυτόματη αντιστοίχιση (mapping) MongoDB δεδομένων σε DOLAR μοντέλα

Σε αυτό το κεφάλαιο, θα εξετάσουμε τη διαδικασία της αντιστοίχισης στο πλαίσιο του MongoDB. Η αντιστοίχιση (mapping) είναι η διαδικασία με την οποία συσχετίζουμε δεδομένα από μια εφαρμογή με αντίστοιχα δεδομένα στη βάση δεδομένων MongoDB.

Θα εξετάσουμε τα βασικά στοιχεία της αντιστοίχισης στη MongoDB, συμπεριλαμβανομένων των πεδίων, των τύπων δεδομένων και των σχέσεων μεταξύ δεδομένων. Θα δούμε επίσης πώς μπορούμε να δημιουργήσουμε μια αντιστοίχιση μεταξύ των δεδομένων της εφαρμογής μας και του MongoDB.

5.1 MongoDB

Η MongoDB είναι μια δημοφιλής βάση δεδομένων NoSQL που έχει επικρατήσει στον κόσμο της αποθήκευσης και διαχείρισης δεδομένων

Είναι ιδιαίτερα απλή στην αρχιτεκτονική της, παρουσιάζοντας μεγάλη ευελιξία, ταχύτητα και ευκολία. Είναι εύκολη στο χειρισμό της, ο οποίος μπορεί να γίνει είτε μέσω του MongoShell είτε μέσω τρίτων προγραμμάτων διαχείρισης βάσεων.

Κάθε νέα καταχώρηση στη MongoDB θεωρείται ως ένα νέο έγγραφο. Τα πεδία μπορεί να είναι είτε απλά είτε σύνθετα, είτε να εμπεριέχουν εμφωλευμένα έγγραφα. Η Mongo αποθηκεύει αυτά τα αρχεία στις λεγόμενες συλλογές (collections), αντίστοιχα με τους πίνακες στις σχεσιακές βάσεις δεδομένων. Η βάση δεδομένων αποτελείται ουσιαστικά από ένα σύνολο συλλογών. Κάθε βάση δεδομένων διαθέτει ξεχωριστό σύνολο αρχείων και δυνητικά μπορεί να περιέχει μεγάλο αριθμό από συλλογές, ενώ μία ενιαία εγκατάσταση MongoDB μπορεί να αποτελείται από πολλές βάσεις δεδομένων.

5.1.1 Πλεονεκτήματα και μειονεκτήματα της χρήσης της MongoDB

Αναφέρουμε μερικά πλεονεκτήματα της MongoDB

1. Ευελιξία στη Δομή των Δεδομένων: Η MongoDB επιτρέπει την αποθήκευση δεδομένων σε μορφή εγγράφου (document), όπως JSON ή BSON. Αυτό σημαίνει ότι μπορούμε να αποθηκεύουμε διαφορετικούς τύπους δεδομένων μέσα στην ίδια βάση, χωρίς την ανάγκη για προκαθορισμένο σχήμα. Αυτή η ευελιξία είναι εξαιρετικά χρήσιμη όταν οι απαιτήσεις σας για τα δεδομένα αλλάζουν συχνά.
2. Κλιμακωσιμότητα: Η MongoDB υποστηρίζει οριζόντια κλιμάκωση, επιτρέποντας τη διανομή των δεδομένων σε πολλούς διακομιστές ή clusters. Αυτό επιτρέπει τη διαχείριση μεγάλων όγκων δεδομένων και τη διασφάλιση υψηλής διαθεσιμότητας και απόδοσης.
3. Υποστήριξη για γεωγραφική αναζήτηση: Η MongoDB προσφέρει ενσωματωμένη υποστήριξη για γεωγραφικές αναζητήσεις, καθιστώντας την κατάλληλη για εφαρμογές που απαιτούν ανάλυση χωρικών δεδομένων, όπως εφαρμογές τοποθεσίας και χαρτογράφησης.

Ακολουθούν και μερικά μειονεκτήματα

1. Απόδοση σε σχέση με τις κλασικές Βάσεις Δεδομένων: Παρόλο που η MongoDB είναι αποδοτική για πολλές εφαρμογές, μπορεί να μην είναι τόσο γρήγορη όσο οι παραδοσιακές σχεσιακές βάσεις δεδομένων για συγκεκριμένες εργασίες. Είναι σημαντικό να εξετάζονται οι απαιτήσεις της εκάστοτε εφαρμογής πριν αποφασιστεί η χρήση της MongoDB.
2. Έλλειψη Ατομικότητας συναλλαγής βάσης δεδομένων: Η MongoDB δεν υποστηρίζει πλήρως ατομικότητα συναλλαγής βάσης δεδομένων όπως οι παραδοσιακές βάσεις δεδομένων. Αυτό μπορεί να οδηγήσει σε προβλήματα συγκρούσεων δεδομένων σε ορισμένες περιπτώσεις, εάν δεν γίνει σωστή διαχείριση κατά την αποθήκευση των δεδομένων.

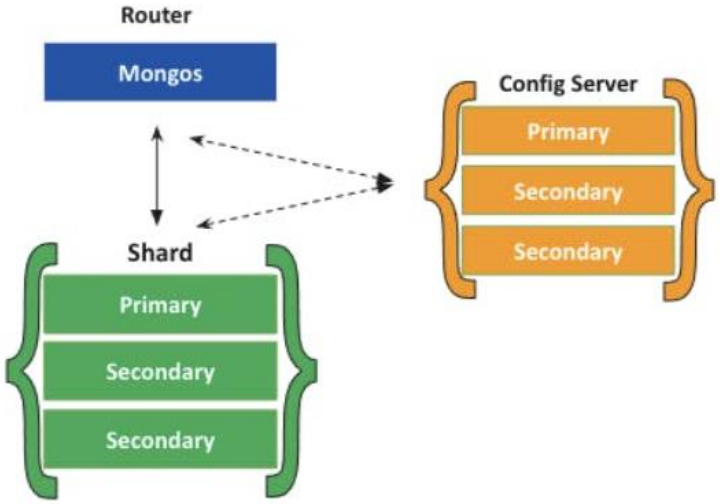
5.1.2 Αρχιτεκτονική της MongoDB

Η αρχιτεκτονική της MongoDB είναι σχεδιασμένη για να ανταποκρίνεται στις ανάγκες μιας σύγχρονης βάσης δεδομένων που λειτουργεί σε περιβάλλον υψηλής κίνησης και μεγάλων όγκων δεδομένων. Κάποια βασικά χαρακτηριστικά της αρχιτεκτονικής περιλαμβάνουν:

- Διακομιστές:
    - Η MongoDB λειτουργεί με ένα σύνολο διακομιστών που συνεργάζονται για την αποθήκευση και την ανάκτηση δεδομένων. Οι διακομιστές αυτοί είναι ανεξάρτητοι μεταξύ τους και μπορούν να εκτελούνται σε διάφορα φυσικά ή εικονικά μηχανήματα.

      Κάθε διακομιστής MongoDB είναι υπεύθυνος για τη διαχείριση ενός τμήματος της βάσης δεδομένων. Αυτή η κατανομή των δεδομένων σε διάφορους διακομιστές επιτρέπει την κλιμακωσιμότητα και την αύξηση της απόδοσης.
- Εγγραφές BSON:
    - Η MongoDB αποθηκεύει τα δεδομένα της σε μορφή εγγράφων BSON (Binary JSON). Το BSON είναι μια δυαδική μορφή που είναι πολύ αποδοτική όσον αφορά την αποθήκευση και την ανάκτηση των δεδομένων. Επιπλέον, το BSON υποστηρίζει πολλούς τύπους δεδομένων, συμπεριλαμβανομένων ακόμα και αναδρομικών δομών.
- Αυτοματισμός Διαμοιρασμός:

  Ο αυτοματισμός διαμοιρασμός είναι ένα σημαντικό χαρακτηριστικό της MongoDB που επιτρέπει τον αυτόματο και δυναμικό διαμοιρασμό των δεδομένων σε διάφορους διακομιστές. Αυτό επιτυγχάνεται μέσω του μηχανισμού διαμοιρασμού (sharding), όπου τα δεδομένα χωρίζονται σε τμήματα και ανατίθενται δυναμικά σε διάφορους διακομιστές. Ο διαμοιρασμός δεδομένων επιτρέπει την οριζόντια κλιμάκωση της MongoDB, προσφέροντας τη δυνατότητα να ανταποκριθεί σε αυξημένο φόρτο και να διαχειριστεί μεγαλύτερα σύνολα δεδομένων.



Εικόνα 9: MongoDB αρχιτεκτονική

5.1.2.1 Χρηση ORM σε MongoDB

Αντίστοιχα με τα ORM που έχουν δημιουργηθεί για τις σχεσιακές βάσεις δεδομένων, έχουν δημιουργηθεί και κάποιες υλοποιήσεις για μη σχεσιακές, και συγκεκριμένα για τη MongoDB. Παρακάτω παραθέτουμε κάποια ιδιαίτερα χαρακτηριστικά ενός MongoDB ORM:

1. **Μοναδικά Χαρακτηριστικά και Προκλήσεις**: Η φύση του MongoDB χωρίς σχήμα παρέχει ευελιξία, αλλά θέτει προκλήσεις στη διατήρηση της ακεραιότητας των δεδομένων και στην
2. **Αντιστοίχιση συλλογών και εγγράφων MongoDB**: Τα MongoDB ORM αντιστοιχίζουν συλλογές και έγγραφα σε αντικειμενοστραφή μοντέλα, βελτιστοποιώντας την πρόσβαση και τον χειρισμό δεδομένων. Αυτή η αντιστοίχιση επιτρέπει στους προγραμματιστές να συνεργάζονται με το MongoDB χρησιμοποιώντας γνωστά παραδείγματα προγραμματισμού, βελτιώνοντας την παραγωγικότητα και την αναγνωσιμότητα κώδικα.

Σε σύγκριση με παραδοσιακά ORM:

1. **Συγκριτική Ανάλυση**: Ο χειρισμός MongoDB ORM διαφέρει σημαντικά από τα παραδοσιακά ORM που χρησιμοποιούνται με σχεσιακές βάσεις δεδομένων. Ενώ τα παραδοσιακά ORM βασίζονται σε σταθερά σχήματα και ερωτήματα που βασίζονται σε SQL, τα MongoDB ORM λειτουργούν σε περιβάλλον χωρίς σχήμα με μια ξεχωριστή προσέγγιση ερωτημάτων και μοντελοποίηση δεδομένων που βασίζεται σε έγγραφα.
2. **Διαφορές στις προσεγγίσεις queries**: Τα queries στη MongoDB μεταφράζονται εσωτερικά σε MongoDB queries, προσφέροντας ευελιξία για την αναζήτηση σύνθετων δομών δεδομένων. Σε σύγκριση με τα παραδοσιακά ORM με ερωτήματα που βασίζονται σε SQL, τα MongoDB ORM παρέχουν μια πιο διαισθητική προσέγγιση για την εργασία με μοντέλα δεδομένων που βασίζονται σε έγγραφα.

5.1.2.2 Αντιστοίχιση σε MongoDB

Το βασικό πρόβλημα που δημιουργείται από τη φύση της MongoDB είναι ότι, εφόσον δεν υπάρχει ξεκάθαρο σχήμα στη βάση, δεν μπορεί και να υπάρξει ξεκάθαρος μηχανισμός προκειμένου να εξάγουμε εύκολα αυτή τη πληροφορία.

Ωστόσο, παρά την ευελιξία του σχήματος, μπορούμε και πάλι να αναλύσετε τη δομή των εγγράφων σε μια συλλογή για την εξαγωγή του σχήματος ή για να αποκτήσουμε πληροφορίες για το μοντέλο δεδομένων. Παρακάτω παραθέτουμε κάποιες κλασσικές προσεγγίσεις:

Εργαλεία ή βιβλιοθήκες: Μπορείτε να χρησιμοποιήσουμε εργαλεία ή βιβλιοθήκες που αναλύουν τα υπάρχοντα έγγραφα σε μια συλλογή και συμπεραίνουν ένα σχήμα με βάση τα πεδία και τους τύπους δεδομένων τους. Ένα τέτοιο είναι το MongoDB Compass. Είναι σημαντικό να αναφέρουμε ότι τέτοια εργαλεία πρακτικά “διαβάζουν” όλο το σχήμα και παραθέτουν στατιστικά συμπεράσματα για το σχήμα και τη φύση των πεδίων.

Aggregation Pipeline: Το πλαίσιο συγκέντρωσης του MongoDB επιτρέπει να εκτελούνται πολύπλοκες εργασίες επεξεργασίας δεδομένων, συμπεριλαμβανομένης της ανάλυσης της δομής των εγγράφων σε μια συλλογή. Στα πλαίσια της εργασίας αυτής, η υλοποίηση που έχει γίνει είναι με αυτή την τεχνική.

Map-Reduce: Η MongoDB υποστηρίζει λειτουργίες Map-Reduce, οι οποίες μπορούν να χρησιμοποιηθούν για την ανάλυση της δομής των εγγράφων σε μια συλλογή και την παραγωγή ενός σχήματος.

Επικύρωση σχήματος (scheme validation): Η MongoDB εισήγαγε την επικύρωση σχήματος στην έκδοση 3.2, η οποία επιτρέπει να ορίσουμε κανόνες επικύρωσης για συλλογές με βάση το σχήμα JSON.

Τέλος, αξίζει να αναφέρουμε ότι υπάρχουν πολλά διαθέσιμα εργαλεία και βιβλιοθήκες τρίτων που μπορούν να βοηθήσουν στην ανακάλυψη και ανάλυση σχημάτων στο MongoDB. Αυτά τα εργαλεία ενδέχεται να προσφέρουν δυνατότητες όπως οπτικοποίηση δομών εγγράφων, συμπερασματικά σχήματα και δημιουργία προφίλ δεδομένων.

5.2 Περιγραφή της διαδικασίας αντιστοίχισης

5.2.1 Δυναμικό σχήμα

Στο MongoDB, η αποθήκευση δεδομένων δεν περιορίζεται από μια άκαμπτη δομή, σε αντίθεση με τις παραδοσιακές βάσεις δεδομένων. Αυτό το χαρακτηριστικό, γνωστό ως δυναμικό σχήμα, σημαίνει ότι τα δεδομένα μπορούν να εξελιχθούν με την πάροδο του χρόνου, με νέα πεδία να προστίθενται ή υπάρχοντα να αλλάζουν δομή.

Η πρόκληση του δυναμικού σχήματος στο MongoDB: Αυτή η ευελιξία αποτελεί μια μοναδική πρόκληση. Χωρίς ένα σταθερό σχήμα, είναι δύσκολο να κατανοήσουμε την οργάνωση και τις σχέσεις μέσα στα δεδομένα. Σε αντίθεση με την παραδοσιακή διαδικασία αντιστοίχισης με προκαθορισμένα δομή, η πλοήγηση σε ένα δυναμικά μεταβαλλόμενο σύνολο δεδομένων είναι σαν εξερεύνηση αχαρτογράφητου εδάφους.

5.2.2 Λύση: Aggregate Functions

Ανακάλυψη του σχήματος δεδομένων με χρήση συναρτήσεων συγκεντρωτικών στοιχείων (Aggregate functions):

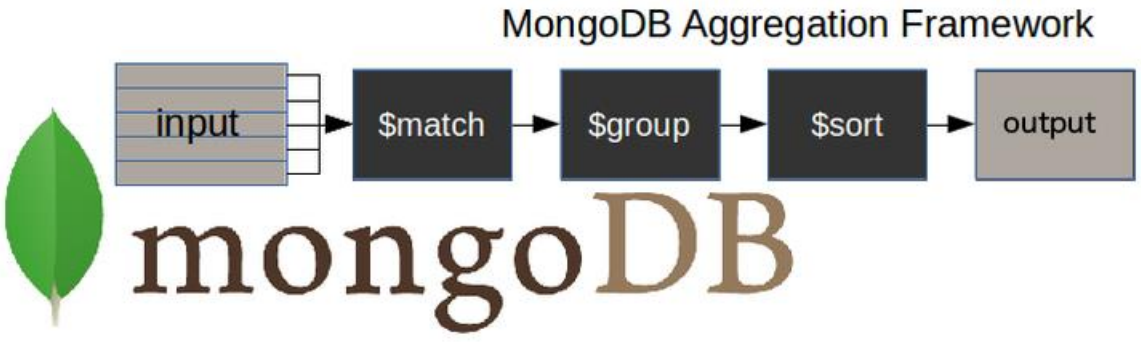
Για να αντιμετωπίσουμε αυτήν την πρόκληση, έχουμε αναπτύξει μια μέθοδο για να ανακαλύπτουμε δυναμικά τη δομή των δεδομένων. Κεντρικό στοιχείο αυτής της προσέγγισης είναι η χρήση των συγκεντρωτικών συναρτήσεων (aggregate functions) της MongoDB [8] [9]. Οι συγκεντρωτικές συναρτήσεις στη MongoDB είναι ευέλικτα εργαλεία που επιτρέπουν την εξελιγμένη ανάλυση και χειρισμό δεδομένων. Λειτουργούν σε συλλογές εγγράφων και μπορούν να εκτελέσουν ένα ευρύ φάσμα λειτουργιών, όπως ομαδοποίηση, φιλτράρισμα και μετατροπή δεδομένων.

Αξιοποιώντας τη δύναμη των aggregate functions, ξεκινάμε ένα ταξίδι εξερεύνησης δεδομένων. Χρησιμοποιούμε συναρτήσεις όπως $group, $project, $unwind και $lookup για να αναλύσουμε και να μελετήσουμε το σύνολο δεδομένων. Αυτές οι λειτουργίες μας επιτρέπουν να συγκεντρώνουμε δεδομένα από πολλά έγγραφα, να εξάγουμε συγκεκριμένα πεδία, να “ξετυλίγουμε” πίνακες για να αποκαλύψουμε ένθετες δομές και να πραγματοποιούμε συνδέσεις μεταξύ συλλογών.

Μέσω συστηματικής ανάλυσης που χρησιμοποιεί συναρτήσεις συγκεντρωτικών στοιχείων, αποκαλύπτουμε πολύτιμες γνώσεις σχετικά με το σχήμα και τη σύνθεση των δεδομένων. Προσδιορίζουμε τα βασικά πεδία, τους αντίστοιχους τύπους δεδομένων τους και τυχόν ένθετες δομές που μπορεί να περιέχουν.

Επιπλέον, οι αθροιστικές συναρτήσεις διευκολύνουν την ανακάλυψη μοτίβων και σχέσεων μέσα στο σύνολο δεδομένων. Μπορούμε να εντοπίσουμε κοινά σημεία, ανωμαλίες και τάσεις στα δεδομένα. Αυτή η βαθύτερη κατανόηση μας επιτρέπει να λαμβάνουμε τεκμηριωμένες αποφάσεις σχετικά με τη μοντελοποίηση δεδομένων και το σχεδιασμό σχημάτων.

Συνοπτικά, οι συναρτήσεις συγκεντρωτικών στοιχείων διαδραματίζουν κεντρικό ρόλο στην προσέγγισή μας στη αντιστοίχιση δυναμικών δεδομένων στο MongoDB. Μας παρέχουν τα απαραίτητα εργαλεία για την πλοήγηση στην πολυπλοκότητα ενός δυναμικού σχήματος, αποκαλύπτοντας πολύτιμες ιδέες και διευκολύνοντας την ανάπτυξη ολοκληρωμένων στρατηγικών αντιστοίχισης.



Εικόνα 10: MongoDB aggregation

5.2.3 Διαδικασία Mapping - Υλοποίηση

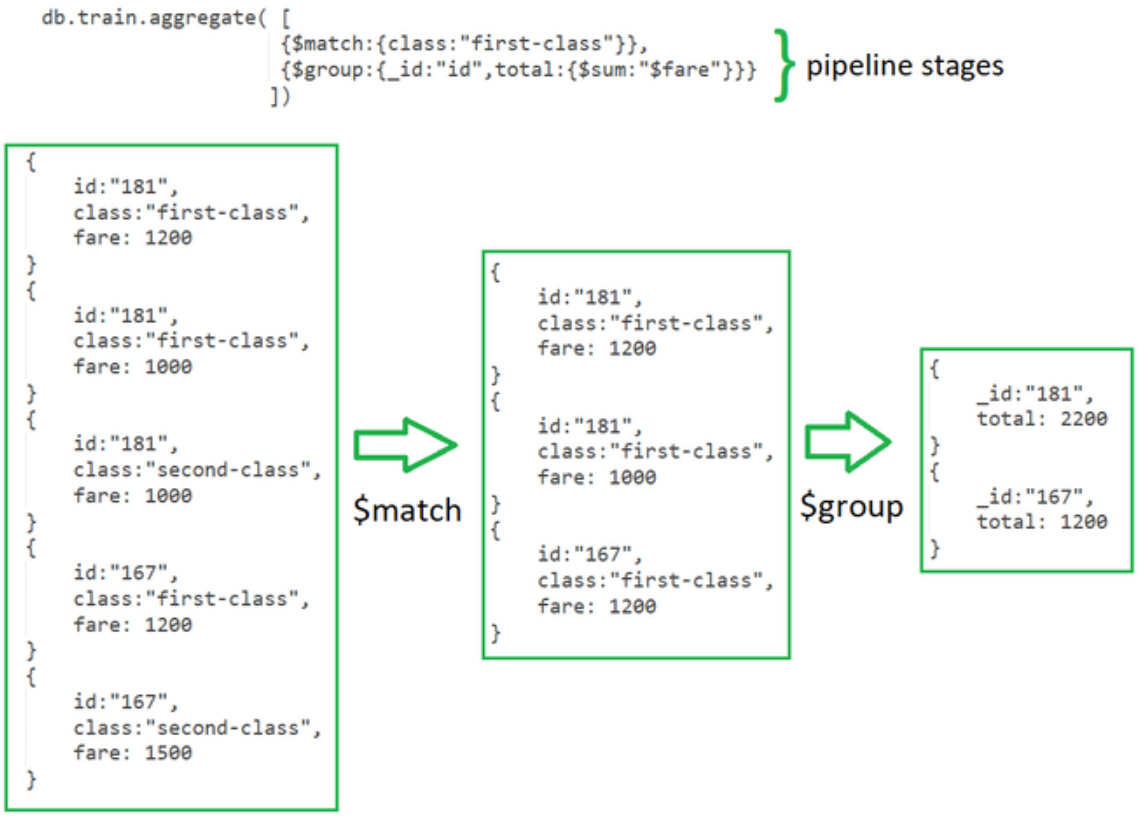
Επαναληπτική Διαδικασία Ανακάλυψης:

Η προσέγγισή μας στη αντιστοίχιση δυναμικών δεδομένων στη MongoDB περιλαμβάνει μια συστηματική και επαναληπτική διαδικασία εξερεύνησης και κατανόησης. Η διαδικασία είναι η εξής:

1. Αρχική αξιολόγηση: Ξεκινάμε αναλύοντας τη δομή των δεδομένων σε υψηλό επίπεδο. Αυτό περιλαμβάνει τον προσδιορισμό των βασικών πεδίων που υπάρχουν σε κάθε έγγραφο και των αντίστοιχων τύπων δεδομένων τους.
2. Εξερεύνηση ένθετων δομών: Στη συνέχεια, εμβαθύνουμε στα δεδομένα για να αποκαλύψουμε τυχόν ένθετες ή πολύπλοκες δομές. Αυτό περιλαμβάνει την εξερεύνηση πινάκων, αντικειμένων ή άλλων ένθετων πεδίων στα έγγραφα.
3. Καταγραφή μεταδεδομένων πεδίου: Καθώς περιηγούμαστε στα δεδομένα, καταγράφουμε σημαντικά μεταδεδομένα για κάθε πεδίο. Αυτό περιλαμβάνει λεπτομέρειες όπως ονόματα πεδίων, τύπους δεδομένων και τυχόν σχέσεις ή εξαρτήσεις μεταξύ πεδίων.
4. Δυναμική Προσαρμογή: Η προσέγγισή μας είναι ευέλικτη και προσαρμοστική. Καθώς συναντάμε νέα δεδομένα ή αλλαγές στη δομή δεδομένων, προσαρμόζουμε τη στρατηγική αντιστοίχιση ανάλογα. Αυτό διασφαλίζει ότι η διαδικασία μας παραμένει ακριβής και αντανακλά την εξελισσόμενη φύση των δεδομένων.

Πίνακας 2: Mongo Αντιστοίχιση - Χειρισμός Αντικειμένων

Τύπος πεδίου	Περιγραφή
Σύνθετα αντικείμενα	Αναπαριστά ένθετες δομές μέσα σε έγγραφα, που περιέχουν πολλαπλά πεδία.
Λίστες	Αντιπροσωπεύουν λίστες τιμών μέσα σε έγγραφα, που συχνά περιέχουν πολλαπλές εμφανίσεις του ίδιου πεδίου.
Πρωτόγονοι τύποι αντικειμένων	Αναπαριστά απλά πεδία με μεμονωμένες τιμές, όπως συμβολοσειρές, αριθμούς ή τιμές boolean.



Εικόνα 11:MongoDB aggregate function

Λεπτομέρειες Υλοποίησης:

Η μετάφραση της διαδικασίας ανακάλυψής μας σε πρακτική εφαρμογή περιλαμβάνει την ανάπτυξη κώδικα που αυτοματοποιεί τη διαδικασία αντιστοίχισης. Ακολουθεί μια επισκόπηση της προσέγγισής μας:

1. Ανάκτηση δεδομένων από τη MongoDB: Αλληλεπιδρούμε με τη MongoDB για να ανακτήσουμε τα δεδομένα που θέλουμε να αντιστοιχίσουμε. Αυτό συνήθως περιλαμβάνει την αναζήτηση συλλογών MongoDB για τη λήψη σχετικών εγγράφων.
2. Χρήση συγκεντρωτικών συναρτήσεων MongoDB:
    - Το MongoDB παρέχει ισχυρές λειτουργίες που μας επιτρέπουν να επεξεργαζόμαστε και να αναλύουμε δεδομένα με διάφορους τρόπους. Αξιοποιούμε αυτές τις λειτουργίες για να πραγματοποιήσουμε εξερεύνηση δεδομένων και να αποκαλύψουμε πληροφορίες σχετικά με τη δομή δεδομένων.
    - Η χρήση συγκεντρωτικών συναρτήσεων είναι παρόμοια με τη λήψη φωτογραφίας ενός συγκεκριμένου επιπέδου της συλλογής, αλλά για όλες τις επαναλήψεις. Αυτό μας δίνει τη δυνατότητα να αναλύσουμε τα δεδομένα ολοκληρωμένα και συστηματικά.
3. Επαναληπτικός αλγόριθμος αντιστοίχισης: Ο αλγόριθμος αντιστοίχισης μας, ακολουθεί μια επαναληπτική προσέγγιση παρόμοια με τη διαδικασία ανακάλυψής μας. Διασχίζει συστηματικά τα δεδομένα, προσδιορίζοντας πεδία, προσδιορίζοντας τύπους δεδομένων και χειρίζεται ένθετες δομές.

4. Καταγραφή αποτελεσμάτων αντιστοίχισης : Καθώς προχωρά ο αλγόριθμος αντιστοίχισης , καταγράφει τα ευρήματά του. Αυτό περιλαμβάνει τη δημιουργία ενός ολοκληρωμένου σχήματος που καταγράφει ονόματα πεδίων, τύπους δεδομένων και οποιαδήποτε άλλα σχετικά μεταδεδομένα.
5. Διαχείριση σφαλμάτων: Εφαρμόζουμε μηχανισμούς διαχείρισης σφαλμάτων για να διασφαλίσουμε την ευρωστία της διαδικασίας μας. Αυτό περιλαμβάνει το χειρισμό απροσδόκητων μορφών δεδομένων ή σφαλμάτων που μπορεί να προκύψουν κατά την ανάκτηση και επεξεργασία δεδομένων.
6. Συντήρηση και ενημερώσεις: Η εφαρμογή μας έχει σχεδιαστεί για να είναι συντηρήσιμη και προσαρμόσιμη. Ελέγχουμε τακτικά και ενημερώνουμε το σχήμα αντιστοίχισης μας για να προσαρμόζουμε τις αλλαγές στη δομή δεδομένων και να διασφαλίζουμε την ακρίβειά του με την πάροδο του χρόνου.

Στο τέλος της διαδικασίας, μας, το αποτέλεσμα θα αποτελείται από έναν χάρτη, που θα περιέχει κάθε αντικείμενο (ή «μοντέλο») που βρέθηκε κατά τη διαδικασία αντιστοίχισης.

Η προσέγγισή μας στη αντιστοίχιση δυναμικών δεδομένων στο MongoDB συνδυάζει μια συστηματική επαναληπτική διαδικασία ανακάλυψης με πρακτική εφαρμογή με χρήση κώδικα. Εξερευνώντας τα δεδομένα επαναληπτικά και προσαρμόζοντας δυναμικά τη στρατηγική αντιστοίχισης, είμαστε σε θέση να αντιστοίχισουμε με ακρίβεια τη συνεχώς εξελισσόμενη δομή των δυναμικών δεδομένων στο MongoDB. Μέσω της ισχυρής εφαρμογής και της συνεχούς συντήρησης, διασφαλίζουμε ότι το σχήμα αντιστοίχισης παραμένει αξιόπιστο και αντικατοπτρίζει τα υποκείμενα δεδομένα.

Μετά την επιτυχημένη διαδικασία αντιστοίχισης που περιγράφηκε παραπάνω, επιτυγχάνεται ένας ολοκληρωμένος χάρτης ταυτοποιημένων αντικειμένων. Ωστόσο, παρά τη μειωμένη εμφάνιση περιττών πεδίων, σε σύγκριση με τη διαδικασία δειγματοληψίας, η αναγνώριση δυνητικά εξωτερικών αντικειμένων παραμένει επιτακτική. Ανάλογη με την προσέγγιση που ακολουθήθηκε κατά τη φάση δειγματοληψίας JSON, έχει εφαρμοστεί μια μέθοδος για την εκκαθάριση των περιττών πεδίων. Αυτή η διαδικασία, ενώ εφαρμόζεται τόσο σε μεθοδολογίες δειγματοληψίας όσο και σε μεθοδολογίες αντιστοίχισης, χρησιμεύει στον εντοπισμό και την εξάλειψη περιττών αντικειμένων από το σύνολο δεδομένων.

Έτσι, η επακόλουθη διαδικασία αντικατοπτρίζει αυτή που περιγράφεται λεπτομερώς στις ενότητες 4.2.2 και 4.2.3. Ουσιαστικά, ο καθιερωμένος μηχανισμός εκκαθάρισης περιττών πεδίων ενσωματώνεται απρόσκοπτα στη φάση μετά τη αντιστοίχιση, διευκολύνοντας τη βελτίωση του συνόλου δεδομένων και τη δημιουργία των απαραίτητων πρωτοτύπων.

6. Αποτίμηση αποτελεσματικότητας αντιστοίχισης & δειγματοληψίας-Εκτέλεση Πειραμάτων

Στο κεφάλαιο αυτό, θα γίνει η παρουσίαση των πειραμάτων που εκτελέστηκαν, χρησιμοποιώντας τη JSON δειγματοληψία και τη Mongo αντιστοίχιση, όπως παρουσιάστηκαν στα προηγούμενα κεφάλαια.

6.1 Γενικές Παραδοχές εκτέλεσης πειραμάτων

1. Σε πολλές περιπτώσεις, ένα πεδίο δεν έχει κάποια τιμή, δημιουργώντας αδυναμία να προσδιορίσουμε με σιγουριά το είδος του. Αυτά τα πεδία, έχουμε αποφασίσει να τα χαρακτηρίσουμε ως TEXT.
2. Τα κενά πεδία συχνά δεν αναγνωρίζονται από τη Mongo, προκαλώντας δυσκολίες στις μετρήσεις.
3. Σε ορισμένες περιπτώσεις, το "Source of Truth" έχει ήδη χαρακτηρίσει ένα πεδίο ως έναν τύπο που δεν προκύπτει από τις μετρήσεις μας, οδηγώντας σε εσφαλμένες τιμές.

6.2 Αξιολόγηση Αποτελεσμάτων

6.2.1 Διαδικασία

Για την εκτέλεση των δοκιμών/μετρήσεων, χωρίζουμε την διαδικασία σε δύο μέρη.

Το δεύτερο μέρος είναι το κομμάτι του JSON sampling, για το οποίο χρειάζεται μόνο ένα αρχείο JSON, το οποίο πρέπει να τοποθετηθεί σε συγκεκριμένο φάκελο, στο πρότζεκτ.

Το δεύτερο μέρος, είναι το κομμάτι του Mongo mapping. Για την υποστήριξη αυτού το βήματος, χρειάζεται να ακολουθηθούν τα παρακάτω βήματα.

1. Εγκατάσταση της Mongo ως βάση δεδομένων (στο περιβάλλον που δημιουργήσαμε, χρησιμοποιήσαμε μια containerized βαση).
2. Δημιουργία και αρχικοποίηση της βάσης μας. Για το βήμα αυτό, χρησιμοποιήσαμε την εφαρμογή MongoDB Compass, η οποία διευκολύνει τη δημιουργία της βάσης.
3. Στο σημείο αυτό, μπορούμε να εισάγουμε στη βάση μας τα δεδομένα (σε μορφή CSV ή JSON).
4. Όταν η παραπάνω διαδικασία ολοκληρωθεί επιτυχως, μπορούμε πλέον να εισάγουμε τα στοιχεία της σύνδεσης μας στην εφαρμογή και να κάνουμε τις αντίστοιχες δοκιμές.

Για την επαλήθευση της ορθής λειτουργίας, έχουν δημιουργηθεί κάποια τεστ, τα οποία μετρούν την απόδοση των υλοποιήσεων με κάποιες μετρικές. Η διαδικασία που ακολουθείται περιγράφεται παρακάτω:

1. Δημιουργούμε τα prototypes, όπως περιγράφονται στα XML αρχεία, τα οποία και έχουν οριστεί σαν “source of truth” (πηγή της αλήθειας).

2. Εκτελούμε τη διαδικασία sampling/mapping, εισάγοντας σαν είσοδο είτε το αντίστοιχο αρχείο JSON (για την περίπτωση της δειγματοληψίας) είτε τη διεύθυνση(URL) της MongoDB.
3. Έχοντας δημιουργήσει τις κατάλληλες μετρικές, υπολογίζουμε το ποσοστό ομοιότητας του δημιουργημένου prototype σε συγκριση με το αρχικό.

6.2.2 Μετρικές επιβεβαίωσης ομοιότητας

Για τον υπολογισμό της ομοιότητας των prototypes, έπρεπε να δημιουργηθούν συγκεκριμένες μετρητικές. Ιδιαίτερη σημασία έπρεπε να δοθεί στην κάλυψη όλων των δυνατών στοιχείων τα οποία τελικά επηρεάζουν το συνολικό αποτέλεσμα. Αποφασίστηκε να δοθεί σημασία σε τρεις βασικές κατηγορίες:

1. Τον αριθμό των πεδίων που βρίσκονται σε κάθε prototype. Στη συγκεκριμένη μετρητική, μας ενδιαφέρει απλά ένας αριθμός χωρίς περαιτέρω υπολογισμούς
2. Τα ονόματα των πεδίων. Η συγκεκριμένη μετρητική είναι πιο συγκεκριμένη απο την πρώτη, καθώς ελέγχει κατά πόσο τα πεδία του “source of truth” prototype υπάρχουν στο δημιουργημένο. Είναι επίσης υπερσύνολο της πρώτης μεθόδου, αφού εάν κάποιο πεδίο δεν υπάρχει (αν π.χ υπάρχουν λιγότερα πεδία), αυτό θα αποτυπωθεί και σε αυτή τη μετρητική.
3. Τους τύπους των πεδίων. Η συγκεκριμένη μετρητική είναι πιο συγκεκριμένη και απο τις δύο παραπάνω, καθώς εξαρτάται και από τον αριθμό αλλά και από τα ονόματα των πεδίων. Συγκρίνει τους τύπους των πεδίων του “source of truth” prototype σε σύγκριση με το δημιουργημένο.

Οι μετρήσεις των παραπάνω κατηγοριών χωρίζονται σε δύο υποκατηγορίες:

1. Την μέτρηση μόνο σε πρώτο επίπεδο. Σε αυτή την περίπτωση κάνουμε τους υπολογισμούς μόνο στο πρώτου επιπέδου prototype.
2. Την μέτρηση σε όλα τα επίπεδα. Στην περίπτωση αυτή, μετράμε αρχικα το πρώτου επιπέδου prototype, αλλά συνεχίζουμε και στα διασυνδεδεμένα με αυτό, προκειμένου να έχουμε πλήρη εικόνα.

Σημαντικό είναι να αναφέρουμε, ότι πολλά από τα πεδία του συνόλου είναι κενά σε όλες τις περιπτώσεις. Αυτό δημιούργησε πρόβλημα καθώς δεν μπορούσε να προσδιοριστεί με σιγουριά ο τύπος των πεδίων, ενώ σε κάποιες περιπτώσεις δεν αναγνωρίζεται καθόλου από το σύστημα.

Για το λόγο αυτό αποφασίστηκε να χωρίσουμε τις μετρήσεις σε δύο ακόμα κατηγορίες:

1. Στη μέτρηση με την συμπερίληψη όλων των πεδίων (κενών και μη)
2. Στη μέτρηση με τον αποκλεισμό των κενών πεδίων.

Όλες οι μετρήσεις γίνονται σειριακά, και μετριέται το ποσοστό ομοιότητας των prototypes.

Περισσότερες πληροφορίας για το παραπάνω αναφέρονται στο 6.5.

6.2.3 Σύνολα Δεδομένων που χρησιμοποιήθηκαν

Τα σύνολα δεδομένων είναι μια ευγενική χορηγία της μη κερδοσκοπικής εταιρείας MONUMENTA και αφορούν την προστασία της φυσικής και αρχιτεκτονικής κληρονομιάς

Ελλάδας και Κύπρου. Συγκεκριμένα, στα δεδομένα βρίσκουμε υλικο από κτίρια, μνημεία, ιστορικά κτήρια και τοπία κ.α.

Για τη διαδικασία των μετρήσεων χρησιμοποιήθηκαν τρία σύνολα δεδομένων

6.2.3.1 Photos

Πρόκειται για ένα σύνολο δεδομένων μεσαίας πολυπλοκότητας, το οποίο αποτελείται από Prototypes. Η πληροφορία αφορά το μοντέλο μιας βάσης δεδομένων φωτογραφιών.

Πίνακας 3: Photos - Ονόματα & τύποι πεδίων

Όνομα πεδίου	Τύπος στο Παραγόμενο Prototype
parents	REF
code	TEXT
insertionNumber	TEXT
comments	TEXT
cataloguerComments	TEXT
dateFrom	DATE
dateTo	DATE
contentPresentation	TEXT
creator	REF(Person)
subjects	TEXT
people	EMBED(PersonRelator)
editor	REF(Person|CollectiveBody)
collectiveBodies	REF(CollectiveBody)
locations	TEXT
physicalSpecificationsTechnicalRequirements	TEXT
acquisitionProcess	TEXT
license	REF(License)
oldIdentifiers	TEXT
publish	BOOL
tekmiriomeno	BOOL
tekmiriosi_remarks	TEXT
date_tekmiriosi	TEXT
maintenanceStatus	TEXT

sortingStatus	TEXT
scanStatus	TEXT
scannedFlag	BOOL
researchTools	TEXT
copy	TEXT
city	REF(City)
address	TEXT
maintenanceNumber	TEXT
caption	TEXT
depictedSubject	TEXT
photoColor	TEXT
photographer	REF(Person)
types	REF(PhotoType)
characterization	REF(PhotoChar)
timestamps.createdAt	NUMBER
timestamps.modifiedAt	NUMBER
files.thumb	REF->.png file
files.hqFile	REF->.png file
files.webFile	REF->.png file
migration_files.filepath	TEXT
migration_files.filename	TEXT

6.2.3.2 External Reference

Πρόκειται για ένα σύνολο δεδομένων μεσαίας πολυπλοκότητας, το οποίο μοντελοποιεί μία βάση δεδομένων εξωτερικών παραπομπών.

Πίνακας 4: External Reference - Ονόματα & τύποι πεδίων

Όνομα πεδίου	Τύπος στο ”Source of truth” Prototype
parents	REF
text	TEXT
link	TEXT
description	TEXT
date	NUMBER

title	TEXT
people	REF/EMBED

6.2.3.3 Monument

Πρόκειται για ένα σύνολο δεδομένων πολύ μεγάλης πολυπλοκότητας το οποίο αποτελείται από 40+ μοναδικά Prototypes. Η πληροφορία αφορά την περιγραφή μιας βάσης δεδομένων σχετικά με μνημεία.

6.3 Εκτέλεση Πειραμάτων JSON Δειγματοληψίας - Αποτελέσματα

6.3.1 Photos

Παρακάτω παρατίθενται τα αποτελέσματα σε σχετικούς πίνακες:

Πίνακας 5: JSON δειγματοληψία - Photos

Μέτρηση	Ποσοστό Ομοιότητας
Αριθμός Πεδίων	100 %
Όνομα Πεδίων	100 %
Τύπος Πεδίων	67 %
Αναδρομικός Αριθμός Πεδίων	100 %
Αναδρομικό Όνομα Πεδίων	100 %
Αναδρομικός Τύπος Πεδίων	61 %

Όπως παρατηρούμε, υπάρχει υψηλός βαθμός ομοιότητας στον αριθμό και στο όνομα των πεδίων. Σχετικά χαμηλότερος είναι ο βαθμός στον τύπο πεδίων.

6.3.1.1 Αριθμός Πεδίων

Πίνακας 6: JSON δειγματοληψία - Photos - Αριθμός πεδίων

Παραγόμενο Prototype	”Source of truth” Prototype
43	43

6.3.1.2 Ονόματα Πεδίων

Στην περίπτωση αυτή έχουμε απόλυτη ταύτιση στα ονομάτα πεδίων μεταξύ του παραγόμενου και του ”Source of truth” Prototype.

6.3.1.3 Τύπος πεδίων

Πίνακας 7: JSON δειγματοληψία - Photos - Τύπος πεδίων

Όνομα πεδίου	Τύπος στο Δημιουργημένο Prototype	Τύπος στο Παραγόμενο Prototype
parents	REF	REF
code	TEXT	TEXT
insertionNumber	TEXT	TEXT
comments	TEXT	TEXT
cataloguerComments	TEXT	TEXT
dateFrom	**TEXT**	DATE
dateTo	**TEXT**	DATE
contentPresentation	TEXT	TEXT
creator	**TEXT**	REF(Person)
subjects	TEXT	TEXT
people	**TEXT**	EMBED(PersonRelator)
editor	**TEXT**	REF(Person|CollectiveBody)
collectiveBodies	**TEXT**	REF(CollectiveBody)
locations	TEXT	TEXT
physicalSpecificationsTechnicalRequirements	TEXT	TEXT
acquisitionProcess	TEXT	TEXT
license	**TEXT**	REF(License)
oldIdentifiers	TEXT	TEXT
publish	BOOL	BOOL
tekmiriomeno	**TEXT**	BOOL
tekmiriosi_remarks	TEXT	TEXT
date_tekmiriosi	TEXT	TEXT
maintenanceStatus	TEXT	TEXT
sortingStatus	TEXT	TEXT
scanStatus	TEXT	TEXT
scannedFlag	BOOL	BOOL
researchTools	TEXT	TEXT
copy	TEXT	TEXT
city	**TEXT**	REF(City)

address	TEXT	TEXT
maintenanceNumber	TEXT	TEXT
caption	TEXT	TEXT
depictedSubject	TEXT	TEXT
photoColor	TEXT	TEXT
photographer	REF	REF(Person)
types	REF	REF(PhotoType)
characterization	REF	REF(PhotoChar)
timestamps.createdAt	**TEXT**	NUMBER
timestamps.modifiedAt	**TEXT**	NUMBER
files.thumb	**TEXT**	REF->.png file
files.hqFile	**TEXT**	REF->.png file
files.webFile	**TEXT**	REF->.png file
migration_files.filepath	TEXT	TEXT
migration_files.filename	TEXT	TEXT

Παρατηρούμε ότι η ταύτιση είναι απόλυτη στο όνομα αλλά και στον αριθμό των πεδίων.

Υπάρχουν διαφοροποιήσεις σε αρκετά πεδία (σημειωμένο με κόκκινο), όπου στο “Source of Truth” χαρακτηρίζονται με διαφορετικούς τύπους σε σχέση με αυτούς από το δοθέν αρχείο. Η έλλειψη τιμών των πεδίων δεν μας επιτρέπει να χαρακτηρίσουμε σωστά τα παραπάνω πεδία.

6.3.1.4 Υπόλοιπες μετρήσεις

Αντίστοιχη είναι και η εξήγηση για την διαφοροποίηση στην μέτρηση με Αναδρομικό Τύπο Πεδίων όπου η απόδοση πέφτει λόγω της ελλιπής πληροφορίας.

Στις μετρήσεις με αναδρομικό αριθμό και όνομα πεδίων, δεν παρουσιάζεται κάποια ασυνέπεια

6.3.2 External Reference

Παρακάτω παρατίθενται τα αποτελέσματα σε σχετικούς πίνακες:

Πίνακας 8: JSON δειγματοληψία - External Reference

Μέτρηση	Ποσοστό Ομοιότητας
Αριθμός Πεδίων	100 %
Όνομα Πεδίων	100 %

Μέτρηση	Ποσοστό Ομοιότητας
Τύπος Πεδίων	86 %
Αναδρομικός Αριθμός Πεδίων	95 %
Αναδρομικό Όνομα Πεδίων	95 %
Αναδρομικός Τύπος Πεδίων	64 %

Όπως παρατηρούμε, υπάρχει υψηλός βαθμός ομοιότητας στον αριθμό και στο όνομα των πεδίων. Σχετικά χαμηλότερος είναι ο βαθμός στον τύπο πεδίων.

6.3.2.1 Αριθμός Πεδίων

Πίνακας 9: JSON δειγματοληψία - External Reference - Αριθμός πεδίων

Παραγόμενο Prototype	”Source of truth” Prototype
7	7

6.3.2.2 Ονόματα Πεδίων

Στην περίπτωση αυτή έχουμε απόλυτη ταύτιση στα ονόματα πεδίων μεταξύ του παραγόμενου και του ”Source of truth” Prototype.

6.3.2.3 Τύπος πεδίων

Πίνακας 10: JSON δειγματοληψία - External Reference - Τύπος πεδίων

Όνομα πεδίου	Τύπος στο Παραγόμενο Prototype	Τύπος στο ”Source of truth” Prototype
parents	TEXT	REF
text	TEXT	TEXT
link	TEXT	TEXT
description	TEXT	TEXT
date	NUMBER	NUMBER
title	TEXT	TEXT
people	REF/EMBED	REF/EMBED

Παρατηρούμε ότι η ταύτιση είναι απόλυτη στο όνομα αλλά και στον αριθμό των πεδίων.

Η μόνη διαφοροποίηση υπάρχει στο πεδίο “parents” όπου το “Source of Truth” το χαρακτηρίζει ως REF από το δοθέν αρχείο, δεν υπάρχει αρκετή πληροφορία που εξηγεί αυτό το χαρακτηρισμό.

6.3.2.4 Υπόλοιπες μετρήσεις

Αντίστοιχη είναι και η εξήγηση για την διαφοροποίηση στην μέτρηση με Αναδρομικό Τύπο Πεδίων όπου και η απόδοση πέφτει λόγω της ελλιπής πληροφορίας.

Για τις αναδρομικές μετρήσεις αριθμού και ονόματος πεδίου, φαίνεται αντίστοιχα να έχει δηλωθεί ένα παραπάνω πεδίο (description) στο δεύτερο επίπεδο, το οποίο όμως και δεν προκύπτει απο την δοθείσα πληροφορία.

6.3.3 Monument

Παρακάτω παρατίθενται τα αποτελέσματα σε σχετικούς πίνακες:

Πίνακας 11: JSON δειγματοληψία - Monument

Μέτρηση	Ποσοστό Ομοιότητας
Αριθμός Πεδίων	99 %
Όνομα Πεδίων	95 %
Τύπος Πεδίων	77 %
Αναδρομικός Αριθμός Πεδίων	99 %
Αναδρομικό Όνομα Πεδίων	99 %
Αναδρομικός Τύπος Πεδίων	65 %

Όπως παρατηρούμε, υπάρχει υψηλός βαθμός ομοιότητας στον αριθμό και στο όνομα των πεδίων. Σχετικά χαμηλότερος είναι ο βαθμός στον τύπο πεδίων.

Το παράδοξο των συγκεκριμένων μετρήσεων, είναι ότι ο βαθμός ομοιότητας στο όνομα πεδίων είναι μεγαλύτερος στην αναδρομική μέτρηση, από αυτή του πρώτου επιπέδου.

Λόγω της ιδιαίτερα αυξημένης πολυπλοκότητας του Monument dataset, παρατίθεται μόνο η ανάλυση αριθμών πεδίων.

6.3.3.1 Αριθμός Πεδίων

Πίνακας 12: JSON δειγματοληψία - Monument - Αριθμός πεδίων

Δημιουργημένο Prototype	”Source of truth” Prototype
149	151

6.4 Εκτέλεση Πειραμάτων Mongo Αντιστοίχισης- Αποτελέσματα

6.4.1 Photos

Παρακάτω παρατίθενται τα αποτελέσματα σε σχετικούς πίνακες:

Πίνακας 13: Mongo αντιστοίχιση - Photos

Μέτρηση	Ποσοστό Ομοιότητας
Αριθμός Πεδίων	100 %
Όνομα Πεδίων	100 %
Τύπος Πεδίων	67 %
Αναδρομικός Αριθμός Πεδίων	100 %
Αναδρομικό Όνομα Πεδίων	100 %
Αναδρομικός Τύπος Πεδίων	61 %

Παρατηρούμε ότι η ομοιότητα των πεδίων είναι ακριβώς η ίδια με αυτή στη JSON δειγματοληψία. Το γεγονός αυτό είναι πολύ σημαντικό, αν αναλογιστούμε ότι ο βαθμός δυσκολίας της υλοποίησης MongoDB αντιστοίχισης είναι σαφώς υψηλότερος.

6.4.1.1 Αριθμός Πεδίων

Πίνακας 14: Mongo αντιστοίχιση - Photos - Αριθμός πεδίων

Παραγόμενο Prototype	”Source of truth” Prototype
43	43

6.4.1.2 Ονόματα Πεδίων

Στην περίπτωση αυτη έχουμε απόλυτη ταύτιση στα ονομάτα πεδίων μεταξύ του παραγόμενου και του ”Source of truth” Prototype.

6.4.1.3 Τύπος πεδίων

Πίνακας 15: Mongo αντιστοίχιση - Photos - Τύπος πεδίων

Όνομα πεδίου	Τύπος στο Δημιουργημένο Prototype	Τύπος στο Παραγόμενο Prototype
parents	REF	REF
code	TEXT	TEXT
insertionNumber	TEXT	TEXT
comments	TEXT	TEXT
cataloguerComments	TEXT	TEXT
dateFrom	**TEXT**	DATE
dateTo	**TEXT**	DATE
contentPresentation	TEXT	TEXT
creator	**TEXT**	REF(Person)
subjects	TEXT	TEXT
people	**TEXT**	EMBED(PersonRelator)
editor	**TEXT**	REF(Person|CollectiveBody)
collectiveBodies	**TEXT**	REF(CollectiveBody)
locations	TEXT	TEXT
physicalSpecificationsTechnicalRequirements	TEXT	TEXT
acquisitionProcess	TEXT	TEXT
license	**TEXT**	REF(License)
oldIdentifiers	TEXT	TEXT
publish	BOOL	BOOL
tekmiriomeno	**TEXT**	BOOL
tekmiriosi_remarks	TEXT	TEXT
date_tekmiriosi	TEXT	TEXT
maintenanceStatus	TEXT	TEXT
sortingStatus	TEXT	TEXT
scanStatus	TEXT	TEXT
scannedFlag	BOOL	BOOL
researchTools	TEXT	TEXT
copy	TEXT	TEXT
city	**TEXT**	REF(City)

address	TEXT	TEXT
maintenanceNumber	TEXT	TEXT
caption	TEXT	TEXT
depictedSubject	TEXT	TEXT
photoColor	TEXT	TEXT
photographer	REF	REF(Person)
types	REF	REF(PhotoType)
characterization	REF	REF(PhotoChar)
timestamps.createdAt	**TEXT**	NUMBER
timestamps.modifiedAt	**TEXT**	NUMBER
files.thumb	**TEXT**	REF->.png file
files.hqFile	**TEXT**	REF->.png file
files.webFile	**TEXT**	REF->.png file
migration_files.filepath	TEXT	TEXT
migration_files.filename	TEXT	TEXT

Παρατηρούμε ότι η ταύτιση είναι απόλυτη στο όνομα αλλά και στον αριθμό των πεδίων.

Υπάρχουν διαφοροποιήσεις σε αρκετά πεδία (σημειωμένο με κόκκινο), όπου στο “Source of Truth” χαρακτηρίζονται με διαφορετικούς τύπους σε σχέση με αυτούς από το δοθέν αρχείο. Η έλλειψη τιμών των πεδίων δεν μας επιτρέπει να χαρακτηρίσουμε σωστά τα παραπάνω πεδία.

6.4.1.4 Υπόλοιπες μετρήσεις

Αντίστοιχη είναι και η εξήγηση για την διαφοροποίηση στην μέτρηση με Αναδρομικό Τύπο Πεδίων όπου η απόδοση πέφτει λόγω της ελλιπής πληροφορίας.

Στις μετρήσεις με αναδρομικό αριθμό και όνομα πεδίων, δεν παρουσιάζεται κάποια ασυνέπεια

6.4.2 External Reference

Παρακάτω παρατίθενται τα αποτελέσματα σε σχετικούς πίνακες:

Πίνακας 16: Mongo αντιστοίχιση - External Reference

Μέτρηση	Ποσοστό Ομοιότητας
Αριθμός Πεδίων	100 %
Όνομα Πεδίων	100 %

Μέτρηση	Ποσοστό Ομοιότητας
Τύπος Πεδίων	86 %
Αναδρομικός Αριθμός Πεδίων	95 %
Αναδρομικό Όνομα Πεδίων	95 %
Αναδρομικός Τύπος Πεδίων	64 %

Όπως παρατηρούμε, υπάρχει υψηλός βαθμός ομοιότητας στον αριθμό και στο όνομα των πεδίων. Σχετικά χαμηλότερος είναι ο βαθμός στον τύπο πεδίων.

6.4.2.1 Αριθμός Πεδίων

Πίνακας 17: Mongo αντιστοίχιση - External Reference - Αριθμός πεδίων

Παραγόμενο Prototype	”Source of truth” Prototype
7	7

6.4.2.2 Ονόματα Πεδίων

Στην περίπτωση αυτή έχουμε απόλυτη ταύτιση στα ονόματα πεδίων μεταξύ του παραγόμενου και του ”Source of truth” Prototype.

6.4.2.3 Τύπος πεδίων

Πίνακας 18: Mongo αντιστοίχιση - External Reference - Τύπος πεδίων

Όνομα πεδίου	Τύπος στο Παραγόμενο Prototype	Τύπος στο ”Source of truth” Prototype
parents	TEXT	REF
text	TEXT	TEXT
link	TEXT	TEXT
description	TEXT	TEXT
date	NUMBER	NUMBER
title	TEXT	TEXT
people	REF/EMBED	REF/EMBED

Παρατηρούμε ότι η ταύτιση είναι απόλυτη στο όνομα αλλά και στον αριθμό των πεδίων.

Η μόνη διαφοροποίηση υπάρχει στο πεδίο “parents” όπου το “Source of Truth” το χαρακτηρίζει ως REF από το δοθέν αρχείο, δεν υπάρχει αρκετή πληροφορία που εξηγεί αυτό το χαρακτηρισμό.

6.4.2.4 Υπόλοιπες μετρήσεις

Αντίστοιχη είναι και η εξήγηση για την διαφοροποίηση στην μέτρηση με Αναδρομικό Τύπο Πεδίων όπου και η απόδοση πέφτει λόγω της ελλιπής πληροφορίας.

Για τις αναδρομικές μετρήσεις αριθμού και ονόματος πεδίου, φαίνεται αντίστοιχα να έχει δηλωθεί ένα παραπάνω πεδίο (description) στο δεύτερο επίπεδο, το οποίο όμως και δεν προκύπτει απο την δοθείσα πληροφορία.

6.4.3 Monument

Παρακάτω παρατίθενται τα αποτελέσματα σε σχετικούς πίνακες:

Πίνακας 19: Mongo αντιστοίχιση - Monument

Μέτρηση	Ποσοστό Ομοιότητας
Αριθμός Πεδίων	99 %
Όνομα Πεδίων	95 %
Τύπος Πεδίων	80%
Αναδρομικός Αριθμός Πεδίων	99 %
Αναδρομικό Όνομα Πεδίων	99 %
Αναδρομικός Τύπος Πεδίων	64 %

Όπως παρατηρούμε, υπάρχει υψηλός βαθμός ομοιότητας στον αριθμό και στο όνομα των πεδίων. Σχετικά χαμηλότερος είναι ο βαθμός στον τύπο πεδίων.

Το παράδοξο των συγκεκριμένων μετρήσεων, είναι ότι ο βαθμός ομοιότητας στο όνομα πεδίων είναι μεγαλύτερος στην αναδρομική μέτρηση, από αυτή του πρώτου επιπέδου.

Λόγω της ιδιαίτερα αυξημένης πολυπλοκότητας του Monument dataset παρατίθεται μόνο η ανάλυση αριθμών πεδίων.

6.4.3.1 Αριθμός Πεδίων

Πίνακας 20: Mongo αντιστοίχιση - Monument - Αριθμός πεδίων

Παραγόμενο Prototype	”Source of truth” Prototype
149	151

6.5 Εκτέλεση πειραμάτων με εξαιρέσεις πεδίων—προβλημα με Null values.

Είναι σημαντικό να αναφέρουμε ότι για λόγους ορθότητας της πληροφορίας, δημιουργήθηκε μια υλοποίηση στην οποία αφαιρούμε τα πεδία που λανθασμένα απο το dataset, έχουν αναφερθεί ή οριστεί ως κάτι που δεν είναι (π.χ είναι TEXT αλλά ορίστηκαν ως REF).

Σκοπός αυτής της κίνησης είναι να επαληθεύσουμε την ορθότητα των μετρήσεων μας στα υπόλοιπα πεδία.

Το τελικό αποτέλεσμα και στις τρεις παραπάνω περιπτώσεις είναι 100% ομοιότητα, όπως αναμέναμε.

6.6 Συμπεράσματα από τα Αποτελέσματα των Πειραμάτων

Τα αποτελέσματα απο τις μετρήσεις που πραγματοποιήσαμε, φαίνονται συνοπτικά στον παρακάτω πίνακα:

Πίνακας 21: Συγκεντρωτικά αποτελέσματα

Μέτρηση	Photos (JSON)	Photos (Mongo)	External Reference (JSON)	External Reference (Mongo)	Monument (JSON)	Monument (Mongo)
Αριθμός Πεδίων	100%	100%	100%	100%	99%	99%
Όνομα Πεδίων	100%	100%	100%	100%	95%	95%
Τύπος Πεδίων	67%	67%	86%	86%	77%	80%
Αναδρομικός Αριθμός Πεδίων	100%	100%	95%	95%	99%	99%
Αναδρομικό Όνομα Πεδίων	100%	100%	95%	95%	99%	99%

Αναδρομικός Τύπος Πεδίων	61%	61%	64%	64%	65%	64%

Παρακάτω αναφέρουμε κάποια συμπεράσματα που προέκυψαν απο την εκτέλεση των πειραμάτων:

1.Απόδοση των Τεχνικών Εκτέλεσης Πειραμάτων

- Και οι δύο τεχνικές εκτέλεσης πειραμάτων, δηλαδή η δειγματοληψία JSON και η αντιστοίχιση MongoDB, παρήγαγαν σχεδόν βέλτιστα αποτελέσματα.
- Αυτό ισχύει παρά το γεγονός ότι το "source of truth" περιλαμβάνει πληροφορίες για συγκεκριμένα πεδία ή οντότητες, τις οποίες δεν διαθέτει η συλλογή JSON.
- Ακόμη και στη δική μας περίπτωση, τα ποσοστά ομοιότητας είναι αρκετά ενθαρρυντικά.
- Η μοναδική διαφοροποίηση που παρατηρήθηκε μεταξύ των δύο τεχνικών αφορά το Monument dataset (που είναι αρκετά σύνθετο), όπου η δειγματοληψία JSON είχε ελαφρώς καλύτερη απόδοση σε μια μετρική, ενώ αντίθετα η Mongo αντιστοίχιση είχε καλύτερη απόδοση σε μια άλλη.

2.Αποτελεσματικότητα των Τεχνικών Σύγκρισης:

- Η πιο ολοκληρωμένη και "αυστηρή" τεχνική σύγκρισης φαίνεται να είναι η σύγκριση των τύπων πεδίων.
- Αυτό αποδεικνύεται από το γεγονός ότι τα ποσοστά ομοιότητας με αυτή την τεχνική ήταν χαμηλότερα από εκείνα των άλλων τεχνικών σε όλες τις περιπτώσεις.
- Επίσης, η χρήση της αναδρομικής τεχνικής αποδείχθηκε πιο αυστηρή, οδηγώντας σε χαμηλότερα ποσοστά ομοιότητας, κάτι που ήταν αναμενόμενο.

3.Επίδραση της Πληροφορίας στα Ποσοστά Ομοιότητας

- Παρατηρήθηκε ότι όσο αυξάνεται η πληροφορία σε ένα σύνολο δεδομένων, τόσο αυξάνονται οι αναντιστοιχίες στα δεδομένα και μειώνονται τα ποσοστά ομοιότητας.
- Ωστόσο, απαιτείται περαιτέρω ανάλυση με τη χρήση περισσότερων συνόλων δεδομένων μεγάλου μεγέθους για να επαληθευτεί αυτό το συμπέρασμα.

4.Χρήση Τεχνικής εξαίρεσης Πεδίων

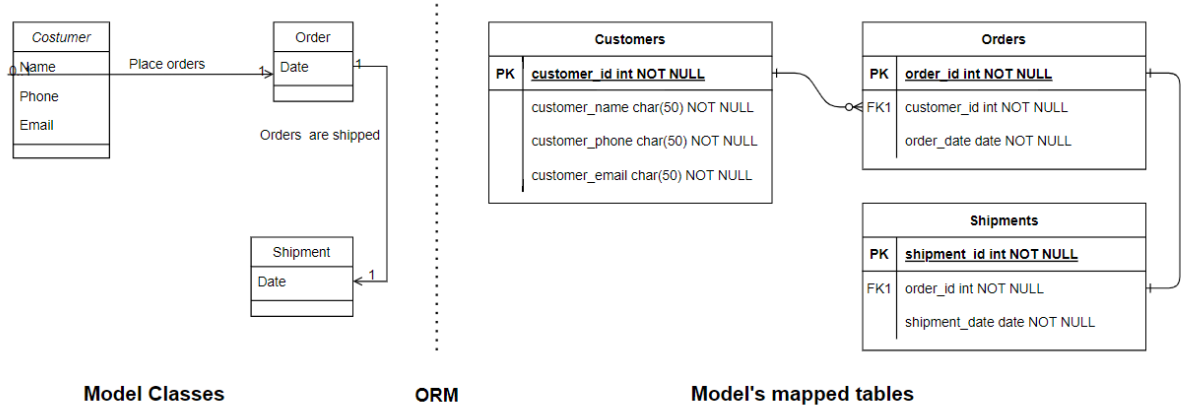
- Η τεχνική εξαίρεσης πεδίων, με την αφαίρεση των πεδίων που εντοπίστηκαν ως κενά (Null), αποδείχθηκε ιδιαίτερα σημαντική για την επιβεβαίωση της ορθότητας των συμπερασμάτων μας.
- Η εκτέλεση των πειραμάτων χωρίς τη σύγκριση αυτών των πεδίων συνέβαλε στην πιο ακριβή αξιολόγηση των αποτελεσμάτων.

7. Related Work

Ο σχεδιασμός και η ανάπτυξη συστημάτων πληροφοριών στις μέρες μας μπορεί να είναι πραγματικά μια δύσκολη εργασία. Στις περισσότερες περιπτώσεις, πρέπει να γνωρίζουμε πολλές γλώσσες προγραμματισμού και πλαίσια. Κάθε επίπεδο εφαρμογής διαθέτει τις δικές της τεχνολογίες. Και, ενώ ορισμένα μέρη των σύγχρονων εφαρμογών εξελίσσονται πολύ γρήγορα, άλλα χρησιμοποιούν τεχνολογία δεκαετιών. Για παράδειγμα, στις σχεσιακές βάσεις δεδομένων των περισσότερων εταιρειών, οι παλιές βάσεις δεδομένων εξακολουθούν να χρησιμοποιούν συγκεκριμένα παλιά πρότυπα αποθήκευσης της πληροφορίας.

Αυτό συμβαίνει γιατί οι σχεσιακές βάσεις δεδομένων είναι σταθερές, αξιόπιστες και αρκετά γρήγορες για τις περισσότερες εμπορικές εφαρμογές. Ως εκ τούτου, η χρήση τους είναι αρκετά διαδεδομένη και έχουν ισχυρές κοινότητες υποστήριξης. Σε αυτήν την αρχιτεκτονική, οι λογικές οντότητες παρατίθενται σε πίνακες που κρατούν τα χαρακτηριστικά κάθε οντότητας στις στήλες τους. Επίσης, οι πίνακες μπορούν να συσχετίζονται με άλλους για να μοντελοποιήσουν τη λογική σχέση μεταξύ των οντοτήτων. Χρησιμοποιούν έναν τυποποιημένο τρόπο προσθήκης, πρόσβασης, ενημέρωσης και διαγραφής δεδομένων, τη δομημένη γλώσσα ερωτημάτων (SQL).

Το ORM [17],[18], είναι ένα πλαίσιο που μπορεί να βοηθήσει και να απλοποιήσει τη μετάφραση μεταξύ των δύο παραδειγμάτων: αντικειμένων και σχεσιακών πινάκων βάσεων δεδομένων. Μπορεί να χρησιμοποιήσει ορισμούς κλάσεων (μοντέλα) για τη δημιουργία, τη διατήρηση και την παροχή πλήρους πρόσβασης στα δεδομένα των αντικειμένων και τη διατήρηση της βάσης δεδομένων τους. Το σχήμα δείχνει ένα απόσπασμα του τρόπου με τον οποίο ένα ORM θα αντιστοιχούσε ένα σύνολο κλάσεων σε σχεσιακούς πίνακες:



Εικόνα 12: Δομή ORM

Στη διατριβή "Εξαγωγή σχήματος και ανίχνευση δομικών ακραίων σημείων για μοντέλα δεδομένων NoSQL που βασίζονται σε JSON" [15] που συντάχθηκε από τους Meike Klettke, Uta Störl και Stefanie Scherzinger, εισάγεται μια νέα προσέγγιση για την εξαγωγή του σχήματος μιας βάσης δεδομένων NoSQL, ειδικά εστιασμένη σε συλλογές JSON εγγράφων που βρίσκονται συνήθως σε βάσεις δεδομένων όπως η MongoDB. Κεντρική θέση σε αυτή τη μέθοδο είναι η χρήση ενός γραφήματος αναγνώρισης δομής (SG), μιας εσωτερικής δομής δεδομένων γραφήματος που συνοψίζει αποτελεσματικά τα δομικά χαρακτηριστικά όλων των αποθηκευμένων εγγράφων. Μέσα σε αυτούς τους

δημιουργημένους γράφους, οι κόμβοι αντιπροσωπεύουν διάφορες ιδιότητες JSON, ένθετα αντικείμενα ή πίνακες, ενώ οι ακμές απεικονίζουν τις ιεραρχικές σχέσεις μεταξύ τους. Αξιοποιώντας αυτό το γράφημα αναγνώρισης δομής, ο αλγόριθμος εξαγωγής σχήματος αντλεί ένα ολοκληρωμένο σχήμα από τα έγγραφα JSON. Αυτή η διαδικασία εξαγωγής σχήματος όχι μόνο διευκολύνει τη βαθύτερη κατανόηση της δομής των δεδομένων αλλά επιτρέπει επίσης την ανάλυση της εξέλιξης του σχήματος με την πάροδο του χρόνου. Επιπλέον, η μέθοδος προσφέρει την ευελιξία εξαγωγής σχημάτων για συγκεκριμένα υποσύνολα εγγράφων, παρέχοντας πολύτιμες πληροφορίες για την οργάνωση των δεδομένων μέσα στη βάση δεδομένων.

Στην εργασία «Ανακάλυψη και οπτικοποίηση των σχημάτων βάσεων δεδομένων NoSQL» [16] από τους A. Hernández Chillón, F. Severino, D. Sevilla, and J. Molina, εφαρμόζεται μια προσέγγιση αντίστροφης μηχανικής που βασίζεται σε μοντέλα, για να συναχθούν σχήματα NoSQL για συστήματα προσανατολισμένα στη συγκέντρωση. Η προσέγγιση διακρίνεται από άλλες προτεινόμενες μεθόδους, όπως το JSONDiscoverer, με τα βασικά χαρακτηριστικά της:

A. Εξαγωγή όλων των εκδόσεων ή παραλλαγών κάθε οντότητας.

B. Ανακάλυψη όλων των σχέσεων μεταξύ των εξαγόμενων εκδόσεων οντοτήτων, συμπεριλαμβανομένων των συναθροίσεων και των αναφορών.

C. Εξέταση της επεκτασιμότητας και της απόδοσης στον αλγόριθμο συμπερασμάτων. Αντί να αποκτήσει ένα συνοπτικό ή κατά προσέγγιση σχήμα, η προσέγγιση καταγράφει όλες τις εκδόσεις οντοτήτων και τις σχέσεις τους.

Η στρατηγική ανακάλυψης σχήματος περιλαμβάνει τρία κύρια στάδια:

1. Λειτουργία Map-Reduce: Απευθείας πρόσβαση στη βάση δεδομένων για να αποκτήσετε το ελάχιστο σύνολο αντικειμένων JSON που απαιτούνται για τη διαδικασία εξαγωγής συμπερασμάτων.
2. Υπολογισμός σχημάτων αντικειμένων για κάθε αντικείμενο JSON, με αποτέλεσμα αρχέτυπα έκδοσης.
3. Ανάλυση εκδόσεων αντικειμένων για την ανακάλυψη οντοτήτων και σχέσεων, που αναπαρίστανται ως μοντέλο που συμμορφώνεται με ένα καθορισμένο μεταμοντέλο.

8. Συμπεράσματα

Συμπερασματικά, η παρούσα διπλωματική εργασία μελετά μηχανισμούς για την παραγωγή μοντέλων δεδομένων από άγνωστες πηγές. Συγκεκριμένα, εφαρμόσαμε έναν μηχανισμό αντιστοίχισης MongoDB και έναν μηχανισμό δειγματοληψίας JSON για την επίτευξη αυτού του στόχου. Τα ευρήματα της εργασιας δείχνουν ότι οι μηχανισμοί μας παρέχουν πολύ ακριβή αποτελέσματα σχετικά με τα πεδία, τους τύπους πεδίων και τον αριθμό των απαιτούμενων οντοτήτων.

Το Κεφάλαιο 2 ανέλυσε διάφορες τεχνικές μοντελοποίησης δεδομένων, παρέχοντας μια βάση για την έρευνά μας.

Στο Κεφάλαιο 3, μιλήσαμε για την χρήση του DOLAR και εισάγαμε το πλαίσιο της εργασίας, σχετικά με την αυτόματη ανακάλυψη DOLAR μοντέλων.

Το Κεφάλαιο 4, επικεντρώθηκε στον μηχανισμό ανακάλυψης μοντέλων δεδομένων δειγματοληψίας JSON.

Το Κεφάλαιο 5 περιέγραψε την αντίστοιχη διαδικασία για την αντιστοίχιση MongoDB μοντέλων.

Τέλος, το Κεφάλαιο 6 αξιολόγησε τις μετρήσεις που απέδειξαν τη λειτουργικότητα των προτεινόμενων μεθόδων μας, αποδεικνύοντας την υψηλή ακρίβειά τους.

Ωστόσο, η NoSQL φύση της MongoDB και τα πιθανά κενά πεδία στο JSON, εισάγουν τον περιορισμό της μη γνώσης των ακριβών τύπων δεδομένων για κενά πεδία. Παρόλα αυτά, τα αποτελέσματά μας συμβάλλουν σημαντικά στην κατανόηση της μοντελοποίησης δεδομένων, προσφέροντας μια αποτελεσματική προσέγγιση για την εξερεύνηση άγνωστων πηγών δεδομένων.

Η μελλοντική έρευνα θα πρέπει να διερευνήσει την παραγωγή μοντέλων δεδομένων για πρόσθετους άγνωστους πόρους, όπως άλλες βάσεις δεδομένων NoSQL, για περαιτέρω επικύρωση και επέκταση αυτών των ευρημάτων. Επιπλέον, μπορεί να γίνει περαιτέρω εργασία για τη συλλογή περισσότερων πληροφοριών σχετικά με τους τύπους δεδομένων, όπως το μέγιστο μέγεθος κάθε πεδίου. Παρά αυτούς τους περιορισμούς, αυτή η μελέτη παρέχει πολύτιμες γνώσεις για την ανακάλυψη μοντέλων δεδομένων και υπογραμμίζει τη σημασία της δημιουργίας ελάχιστων σχημάτων από άγνωστες πηγές.

Συνολικά, τα ευρήματα αυτής της εργασίας υπογραμμίζουν τη σημασία της παραγωγής ενός σχήματος δεδομένων που είναι ταυτόχρονα περιεκτικό και το ελάχιστο δυνατό. Αυτή η έρευνα προσφέρει μια σταθερή βάση για μελλοντικές μελέτες και πρακτικές εφαρμογές στον τομέα της μοντελοποίησης δεδομένων.

Παράρτημα

Βασικές λειτουργικότητες ORM

Ας περιγράψουμε τι συνεπάγονται συνήθως αυτές οι λειτουργίες:

Αντιστοίχιση οντοτήτων σε πίνακες: Περιλαμβάνει την αντιστοίχιση αντικειμενοστρεφών οντοτήτων/κλάσεων στον κώδικα της εφαρμογής σε πίνακες σε μια σχεσιακή βάση δεδομένων. Κάθε οντότητα αντιστοιχεί σε έναν πίνακα και κάθε χαρακτηριστικό της οντότητας αντιστοιχεί σε μια στήλη στον πίνακα. Οι σχέσεις μεταξύ οντοτήτων δημιουργούνται χρησιμοποιώντας περιορισμούς ξένων κλειδιών στο σχήμα της βάσης δεδομένων.

Καθορισμός σχέσεων:Ορίζει σχέσεις μεταξύ οντοτήτων, όπως συσχετίσεις ένα προς ένα, ένα προς πολλά και πολλά προς πολλά. Αυτές οι σχέσεις μεταφράζονται σε περιορισμούς βάσης δεδομένων, όπως ξένα κλειδιά, για την επιβολή της ακεραιότητας αναφοράς.

Διαχείριση σε ομάδες: Καθορίζει την ιδιότητα των σχέσεων, υποδεικνύοντας τον αριθμό των σχετικών οντοτήτων που μπορούν να συσχετιστούν μεταξύ τους. Οι περιορισμοί πολλαπλότητας, όπως 1:1, 1:M και M:N, ορίζουν τον αριθμό των περιπτώσεων που επιτρέπεται σε κάθε πλευρά της σχέσης.

Χαρτογράφηση κληρονομικότητας: Στον αντικειμενοστραφή προγραμματισμό, η κληρονομικότητα επιτρέπει στις κλάσεις να κληρονομούν ιδιότητες και συμπεριφορές από γονικές κλάσεις. Τα ORM frameworks υποστηρίζουν αυτήν την αντιστοίχιση κληρονομικότητας, όπου οι ιεραρχίες κληρονομικότητας αντιστοιχίζονται σε πίνακες βάσεων δεδομένων χρησιμοποιώντας στρατηγικές όπως η κληρονομικότητα ενός πίνακα, η κληρονομικότητα πίνακα κλάσεων ή η κληρονομικότητα ενωμένων πινάκων.

Σχολιασμοί ή διαμόρφωση μεταδεδομένων: Τα πλαίσια ORM παρέχουν μηχανισμούς στους προγραμματιστές για τον καθορισμό μεταδεδομένων αντιστοίχισης χρησιμοποιώντας σχολιασμούς, χαρακτηριστικά ή αρχεία διαμόρφωσης. Οι προγραμματιστές μπορούν να σχολιάσουν κλάσεις οντοτήτων ή να διαμορφώσουν τις λεπτομέρειες αντιστοίχισης μέσω προγραμματισμού για να καθορίσουν πώς αντιστοιχίζονται οι οντότητες σε πίνακες και στήλες βάσης δεδομένων.

Lazy Loading και Proxy Objects: Τα πλαίσια ORM υποστηρίζουν συχνά την αργή φόρτωση σχετικών οντοτήτων, όπου τα σχετικά αντικείμενα φορτώνονται από τη βάση δεδομένων μόνο όταν γίνεται πρόσβαση σε αυτά για πρώτη φορά. Τα αντικείμενα διακομιστή μεσολάβησης χρησιμοποιούνται για την αναπαράσταση οντοτήτων που φορτώνονται με “lazy” τρόπο, επιτρέποντας τη διαφανή φόρτωση σχετικών δεδομένων κατ' απαίτηση χωρίς ρητά ερωτήματα βάσης δεδομένων.

Δημιουργία σχήματος βάσης δεδομένων: Ορισμένα πλαίσια ORM προσφέρουν εργαλεία για την αυτόματη δημιουργία σχημάτων βάσης δεδομένων με βάση το μοντέλο οντότητας που ορίζεται στον κώδικα της εφαρμογής. Αυτή η δυνατότητα απλοποιεί τη ρύθμιση της βάσης δεδομένων και την εξέλιξη του σχήματος, επιτρέποντας στους προγραμματιστές να επικεντρωθούν στη μοντελοποίηση τομέα και όχι σε εργασίες διαχείρισης βάσης δεδομένων. Η αντιστοίχιση σχέσεων οντοτήτων είναι μια θεμελιώδης πτυχή των πλαισίων ORM, που επιτρέπει στους προγραμματιστές να εργάζονται με αντικειμενοστραφή μοντέλα τομέα ενώ αλληλεπιδρούν απρόσκοπτα με σχεσιακές βάσεις δεδομένων. Αφαιρεί την πολυπλοκότητα των αλληλεπιδράσεων βάσεων δεδομένων, επιτρέποντας αποτελεσματική πρόσβαση, χειρισμό και διαχείριση δεδομένων σε σύγχρονες εφαρμογές λογισμικού.

Πλεονεκτήματα ORM

Το πρώτο πλεονέκτημα είναι ότι μπορούμε να χρησιμοποιήσουμε τη γλώσσα του backend για να κωδικοποιήσουμε όλη την πρόσβαση στα δεδομένα. Για παράδειγμα, για να συγκεντρώσουμε μια λίστα προσώπων θα χρειαστεί να γράψουμε μια συνάρτηση παρόμοια με αυτή:

Algorithm 1: Return objects from the database using SQL

```
Input  : filter key, filter value
Output: list of persons
personList ← newstateList()
sql ← "select fullname, ocupation from persons where " +
 key + " = : value"
data ← query(sql, value)
while row ← data.next() do
    person ← new Person()
    person.fullname ← row.get('fullname')
    person.ocupation ← row.get('occupation')
    personList.add(person)
return personList
```

Εικόνα 13: Queries σε SQL

Αντίστοιχα, αν χρησιμοποιήσουμε ORM το αποτέλεσμα είναι το εξής:

Algorithm 2: Return objects from the database using an ORM

```
Input  : filter key, filter value
Output: list of persons
personList ← Person.query(key,"equals",value)
return personList
```

Εικόνα 14: Queries σε ORM

Η δεύτερη εικόνα, παρουσιάζει ένα query που είναι πολύ πιο ωραίο στην κωδικοποίηση και ανάγνωση. Αναφέρεται σε μια κλάση (δηλαδή Person) που ορίζει τα χαρακτηριστικά και τις μεθόδους που σχετίζονται με την επιχείρηση. Οι κλάσεις που ασχολούνται με αντικειμενοσχεσιακές αντιστοιχίσεις ονομάζονται Models. Αυτές οι κλάσεις χρησιμοποιούν ως πρότυπο, μια τάξη από τη βιβλιοθήκη κλάσεων μοντέλων του ORM. Με αυτόν τον τρόπο, κληρονομεί όλες τις μεθόδους που απαιτούνται για τη δημιουργία, την ανάκτηση, την ενημέρωση και τη διαγραφή από τη βάση δεδομένων.

Τα μοντέλα οριοθετούνται ασθενώς με την υπόλοιπη εφαρμογή.

Μειονεκτήματα ORM

Από την άλλη πλευρά, για να χρησιμοποιήσουμε σωστά το ORM, θα πρέπει να αφιερώσουμε αρκετό χρόνο στην εκμάθηση του. Παρόλο που χρησιμοποιεί την ίδια γλώσσα στην οποία βασίζεται ο κώδικάς μας, το ORM έχει τη δική του σημασιολογία και σύνταξη χρήσης.

Το ORM φροντίζει τελικά να μεταφράσει τον κώδικά μας σε SQL. Ωστόσο, παρόλο που τις περισσότερες φορές κάνει καλή δουλειά, δεν μπορεί να ανταγωνιστεί, σε σύνθετα ερωτήματα, την καλογραμμένη SQL.

Επίσης, ακόμα κι αν προσπαθήσει να δημιουργήσει τις καλύτερες δυνατές ρήτρες SQL, μπορεί να πάει στραβά. Ειδικά επειδή μερικές φορές είναι δύσκολο να καταλάβουμε τι κάνει το ORM στα παρασκήνια.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen, *Object-Oriented Modeling and Design*, Prentice-Hall, 1991.

[2] K. R. Dittrich, "Object-Oriented Data Model Concepts," in A. Dogac, M. T. Özsu, A. Biliris, and T. Sellis (eds), *Advances in Object-Oriented Database Systems*, NATO ASI Series, vol. 130. Springer, Berlin, Heidelberg, 1994. https://doi.org/10.1007/978-3-642-57939-4_2

[3] D. Crockford, "Introducing JSON," *json.org*, May 28, 2009. [Online]. Available: https://json.org.

[4] J. L. Harrington, "The Relational Data Model," in *Relational Database Design*, pp. 85-101, December 2009. doi: 10.1016/B978-0-12-374730-3.00005-X.

[5] G. Garani, "A Generalised Relational Data Model Approach in the Organisation and Management of Nested Data," *Journal Annals of Mathematics, Computing & Teleinformatics*, vol. 1, no. 4, pp. 17-23, 2006.

[6] Q. Tong, "Mapping Object-oriented Database Models into RDF(S)," *IEEE Access*, vol. 6, pp. 1-1, August 2018. doi: 10.1109/ACCESS.2018.2867152.

[7] O. Lassila and R. R. Swick, "Resource Description Framework (RDF): Model and Syntax," October 1997.

[8] D. Pasqualin, G. Souza, E. Buratti, E. Almeida, M. Didonet, M. Fabro, and D. Weingaertner, "A Case Study of the Aggregation Query Model in Read-Mostly NoSQL Document Stores," *Proceedings of the 2016 ACM International Conference on Information and Knowledge Management (CIKM)*, 2016. doi: 10.1145/2938503.2938546.

[9] "Aggregation," *MongoDB*. [Online]. Available: https://www.mongodb.com/resources/products/capabilities/aggregation

[10] "Graph Database," *Wikipedia*. [Online]. Available: https://en.wikipedia.org/wiki/Graph_database

[11] "Data Modelling," *Neo4j Developer*. [Online]. Available: https://neo4j.com/developer/data-modeling/

[12] K. Saidis, Y. Smaragdakis, and A. Delis, "DOLAR: Virtualizing Heterogeneous Information Spaces to Support their Expansion."

[13] K. Saidis, G. Pyrounakis, M. Nikolaidou, and A. Delis, "Digital Object Prototypes: An Effective Realization of Digital Object Types," in *Proceedings of the 10th European Conference on Digital Libraries*, Alicante, Spain, 2006.

[14] L. Cardelli and P. Wegner, "On Understanding Types, Data Abstraction, and Polymorphism," AT&T Bell Laboratories, Murray Hill, NJ 07974 and Dept. of Computer Science, Brown University, Providence, RI 02912.

[15] M. Klettke, U. Störl, and S. Scherzinger, "Schema extraction and structural outlier detection for JSON-based NoSQL data stores," *Journal*, vol. 2015, no. 3, pp. 425-444, March 2015.

[16] A. Hernández Chillón, F. Severino, D. Sevilla, and J. Molina, *Exploring the Visualization of Schemas for Aggregate-Oriented NoSQL Databases*, 2017.

[17] T. Kamada, *Visualizing Abstract Objects and Relations*, vol. 5, World Scientific, 1989.

[18] E. J. O'Neil, "Object/relational mapping 2008: Hibernate and the Entity Data Model (EDM)," in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, 2008.