

Ανάλυση και Μοντελοποίηση Ναυτιλιακών Δεδομένων με MongoDB

Δημήτρης Γαλατίδης
Πανεπιστήμιο Πειραιά
ME2409
Αθήνα, Ελλάδα
dimitriosgalatidis@yahoo.com

Άγγελος Τζούρτζης
Πανεπιστήμιο Πειραιά
ME2434
Αθήνα, Ελλάδα
aggelos.tzurtzis@gmail.com



Περίληψη

Στόχος της παρούσας εργασίας είναι η σχεδίαση και υλοποίηση μιας μη σχεσιακής βάσης δεδομένων στη MongoDB για τη διαχείριση ναυτιλιακών δεδομένων μεγάλης κλίμακας, αξιοποιώντας τις δυνατότητες του NoSQL μοντέλου για αποδοτική αποθήκευση και ανάκτηση πληροφοριών. Η υλοποίηση περιλάμβανε τη μετατροπή και εισαγωγή δεδομένων από διαφορετικές πηγές, τον σχεδιασμό της βάσης με έμφαση στην εμφώλευση, καθώς και τη βελτιστοποίηση των ερωτημάτων μέσω χωρικών και χρονικών ευρετηρίων. Για την αξιολόγηση της απόδοσης, πραγματοποιήθηκαν πειραματισμοί με διαφορετικές δομές ευρετηρίων, μεγέθη δεδομένων και τύπους ερωτημάτων, εστιάζοντας στη σύγκριση σχεσιακών, χωρικών και χωρο-χρονικών αναλύσεων. Τα αποτελέσματα ανέδειξαν τη σημασία των σύνθετων ευρετηρίων στη μείωση του χρόνου εκτέλεσης. Συμπερασματικά, η MongoDB αποδείχθηκε αποτελεσματική για τη διαχείριση μεγάλων όγκων ναυτιλιακών δεδομένων, με δυνατότητες βελτίωσης μέσω προηγμένων τεχνικών καταναμημένης αποθήκευσης και βελτιστοποίησης ερωτημάτων.

1 Εισαγωγή

Η διαχείριση μεγάλων συνόλων δεδομένων αποτελεί μία από τις μεγαλύτερες προκλήσεις της σύγχρονης εποχής, ιδιαίτερα σε τομείς που απαιτούν αποδοτική αποθήκευση, αναζήτηση και ανάλυση δεδομένων. Ο ναυτιλιακός τομέας παράγει τεράστιους όγκους δεδομένων, τα οποία περιλαμβάνουν θέσεις πλοίων, πληροφορίες τροχιών, γεωγραφικά και μετεωρολογικά δεδομένα. Η επεξεργασία αυτών των δεδομένων με συμβατικές σχεσιακές βάσεις δεδομένων συχνά δεν είναι η βέλτιστη λύση, καθώς απαιτείται μεγάλη ευελιξία και υψηλή απόδοση για την υποστήριξη χωρικών και χωρο-χρονικών ερωτημάτων.

Στην παρούσα εργασία, εξετάζεται η υλοποίηση μιας μη σχεσιακής βάσης δεδομένων στο σύστημα MongoDB, με στόχο τη

διαχείριση ναυτιλιακών δεδομένων μεγάλης κλίμακας. Η MongoDB προσφέρει ένα δυναμικό μοντέλο δεδομένων βασισμένο σε έγγραφα, παρέχοντας τη δυνατότητα εμφώλευσης, εύκολης επέκτασης και αποδοτικής αποθήκευσης δεδομένων. Η ευελιξία αυτή επιτρέπει τη βελτιστοποίηση των ερωτημάτων και την κλιμάκωση της βάσης δεδομένων.

Η εργασία εστιάζει στην ανάλυση των δεδομένων, τη βελτιστοποίηση της δομής αποθήκευσης και τη δημιουργία κατάλληλων ευρετηρίων ώστε να υποστηρίζονται αποδοτικά:

- Ερωτήματα σχεσιακού τύπου, όπως η αναζήτηση πλοίων με βάση συγκεκριμένα κριτήρια.
- Χωρικά ερωτήματα, για τον εντοπισμό πλοίων εντός συγκεκριμένων γεωγραφικών περιοχών.
- Χωρο-χρονικά ερωτήματα, που συνδυάζουν χωρικές και χρονικές συνθήκες αναζήτησης.

Για την αξιολόγηση της προσέγγισης, πραγματοποιείται πειραματική ανάλυση της απόδοσης των ερωτημάτων υπό διαφορετικές διαμορφώσεις ευρετηρίων και διαφορετικούς όγκους δεδομένων.

2 Σχετικές Εργασίες και Αναφορά στη Βιβλιογραφία

Η παρούσα εργασία βασίστηκε σε βιβλιογραφία που σχετίζεται με τη διαχείριση μεγάλων δεδομένων, τη βελτιστοποίηση αποθηκευτικών δομών και την αποδοτική εκτέλεση χωρο-χρονικών ερωτημάτων.

Το *Piraeus AIS dataset* των *Tritsarolis et al.* παρέχει λεπτομερή περιγραφή του συνόλου δεδομένων που αξιοποιείται στην παρούσα εργασία. Η ανάλυση των δεδομένων αυτών αποτέλεσε σημαντικό σημείο αναφοράς, καθώς επέτρεψε την ποιοτική κατανόηση του προβλήματος και των προκλήσεων που σχετίζονται με την αποθήκευση και την επεξεργασία ναυτιλιακών δεδομένων μεγάλης κλίμακας.

Τόσο στο βιβλίο των *Bradshaw et al.* όσο και στον *Ramakrishnan και Gehrke*, αναλύεται η χρήση της MongoDB για την αποθήκευση

δεδομένων και η δημιουργία ευρετηρίων για τη βελτίωση της απόδοσης. Αναλύεται η διαχείριση σχεσιακών και μη σχεσιακών δεδομένων, με έμφαση στη βελτιστοποίηση ερωτημάτων.

Οι σημειώσεις του μαθήματος και οι εργαστηριακές ασκήσεις παρείχαν χρήσιμες γνώσεις για την οργάνωση δεδομένων στη MongoDB, με ιδιαίτερη έμφαση στην επιλογή της εμφώλευσης, ανάλογα με τις απαιτήσεις των ερωτημάτων και τις σχέσεις μεταξύ των δεδομένων. Οι θεωρητικές αυτές αρχές συνέβαλαν στις αποφάσεις για τη διαμόρφωση των συλλογών στη βάση μας, λαμβάνοντας υπόψη τόσο τις σχέσεις μεταξύ των δεδομένων όσο και την αποδοτικότητα των ερωτημάτων.

Η βιβλιογραφία συνέβαλε στην κατανόηση της μοντελοποίησης των συλλογών με βάση τις σχέσεις των δεδομένων:

- **Σχέσεις 1-1:** Η εμφώλευση σχετικών δεδομένων στο βασικό έγγραφο μειώνει την ανάγκη για επιπλέον αναζητήσεις και βελτιώνει την ταχύτητα ανάκτησης δεδομένων.
- **Σχέσεις 1-N:** Ανάλογα με το μέγεθος και τη συχνότητα ενημέρωσης των δεδομένων, εξετάζεται αν η εμφώλευση είναι αποδοτική ή αν απαιτείται ξεχωριστή συλλογή με ξένα κλειδιά.
- **Σχέσεις N-M:** Σε περιπτώσεις πολλών προς πολλά (*many-to-many*), απαιτείται ξεχωριστή συλλογή που διατηρεί αναφορές προς τις σχετικές οντότητες μέσω ξένων κλειδίων.

Οι αρχές αυτές εφαρμόστηκαν στον σχεδιασμό της βάσης, ώστε να διασφαλιστεί η αποδοτική αποθήκευση και εκτέλεση ερωτημάτων, λαμβάνοντας υπόψη τόσο τη δομή των δεδομένων όσο και τις απαιτήσεις των αναλύσεων που πραγματοποιήθηκαν στο πλαίσιο της παρούσας εργασίας.

3 Ανάλυση Δεδομένων και Ορισμός Ερωτημάτων

Το σύνολο δεδομένων που χρησιμοποιήθηκε περιλαμβάνει ναυτιλιακές πληροφορίες από το Μάιο του 2017 έως το Δεκέμβριο του 2019, με συνολικό μέγεθος **31.4 GB**. Η κύρια πηγή όγκου δεδομένων προέρχεται από τις δυναμικές θέσεις πλοίων, οι οποίες καταγράφουν διαδοχικές χωρο-χρονικές καταστάσεις κάθε τρία λεπτά.

Τα δεδομένα οργανώνονται στις ακόλουθες κατηγορίες:

- **Δυναμικά δεδομένα πλοίων:** Καταγραφή της θέσης, της ταχύτητας και του προσανατολισμού κάθε πλοίου σε διάφορες χρονικές στιγμές.
- **Στατικές πληροφορίες πλοίων:** Περιλαμβάνουν χαρακτηριστικά όπως η χώρα προέλευσης (σημαία) και ο τύπος του πλοίου.
- **Γεωγραφικά δεδομένα:** Πληροφορίες για λιμάνια, νησιά και άλλες τοποθεσίες, με αντίστοιχες συντεταγμένες.
- **Μετεωρολογικά δεδομένα:** Περιλαμβάνουν θερμοκρασία, υγρασία και άλλες καιρικές παραμέτρους ανά τρίωρο.
- **Σύνοψη τροχιών πλοίων:** Συμπυκνωμένες πληροφορίες για διαδρομές και κινήσεις των πλοίων.

Τα δεδομένα επεξεργάστηκαν κατάλληλα ώστε να εισαχθούν στην βάση επιτυχώς με βάση τον παρακάτω αρχιτεκτονικό σχεδιασμό.

4 Αρχιτεκτονικός Σχεδιασμός και Μοντελοποίηση Δεδομένων

Όπως φαίνεται στο Σχήμα 1, τα αρχικά δεδομένα είναι αποθηκευμένα σε **σχεσιακή μορφή**, όπου κάθε οντότητα αντιστοιχεί σε

ξεχωριστούς πίνακες. Για παράδειγμα, υπάρχει ένας πίνακας για τα πλοία και ένας για τους τύπους πλοίων, με σύνδεση μέσω πρωτεύοντων και ξένων κλειδίων.

Για τη μετάβαση σε **μη σχεσιακή βάση δεδομένων (MongoDB)**, απαιτείται **ανασχεδιασμός του μοντέλου δεδομένων**, με έμφαση στην **εμφώλευση (embedding)** πληροφοριών, ώστε να βελτιωθεί η απόδοση των ερωτημάτων.

Η διαδικασία σχεδιασμού πραγματοποιήθηκε με βάση τις αρχές που αναφέρονται στην προηγούμενη ενότητα. Για παράδειγμα, η σχέση των πλοίων με τους τύπους τους είναι **1-N** και επιλέχθηκε η **αντίστροφη εμφώλευση**, όπου η πληροφορία του τύπου του πλοίου εμφωλεύεται μέσα στο έγγραφο κάθε πλοίου. Με αυτήν την προσέγγιση, δημιουργήθηκε η συλλογή **vessels**.

Τα γεωγραφικά δεδομένα, όπως λιμάνια και νησιά, αντί να διαχωριστούν σε ξεχωριστές συλλογές, οργανώθηκαν σε μία ενιαία συλλογή, με μια παράμετρο *type* που προσδιορίζει το είδος κάθε εγγραφής. Έτσι, δημιουργήθηκε η συλλογή **geodata**.

Για την αποθήκευση των μετεωρολογικών δεδομένων, δημιουργήθηκε η συλλογή **weather_data**.

Τα δυναμικά δεδομένα συνδέονται με τα πλοία μέσω του *id* τους. Θα μπορούσαν να εισαχθούν στη βάση ακολουθώντας τη σχεσιακή λογική, επιτρέποντας τη σύνδεση και την ανάκτηση πληροφοριών μέσω της συλλογής **vessels**, ή να εμφωλευθούν πλήρως στις αντίστοιχες εγγραφές μαζί με τις πληροφορίες τους.

Η δεύτερη προσέγγιση θα καθιστούσε τη συλλογή των πλοίων περιττή, καθώς όλη η σχετική πληροφορία θα ενσωματωνόταν στα δυναμικά δεδομένα. Ωστόσο, κρίθηκε σκόπιμο να ενσωματωθούν μόνο το *id* και η χώρα του πλοίου, διατηρώντας τα δεδομένα των πλοίων σε ξεχωριστή συλλογή. Αυτή η επιλογή περιορίζει την επανάληψη της πληροφορίας και διατηρεί την αποδοτικότητα των σχεσιακού τύπου ερωτημάτων που αφορούν τα δεδομένα των πλοίων.

Το *id* επιτρέπει την ανάκτηση επιπλέον πληροφορίας από τη συλλογή των πλοίων, ενώ η χώρα συμπεριλήφθηκε καθώς αποτελεί πληροφορία που πιθανόν να ανακτάται συχνά. Με αυτήν την προσέγγιση, μειώνεται η ανάγκη για συνδέσεις (*joins*), εκτός αν είναι απολύτως απαραίτητες. Ακολουθώντας αυτή τη λογική, δημιουργήθηκε η συλλογή **dynamic_vessels**.

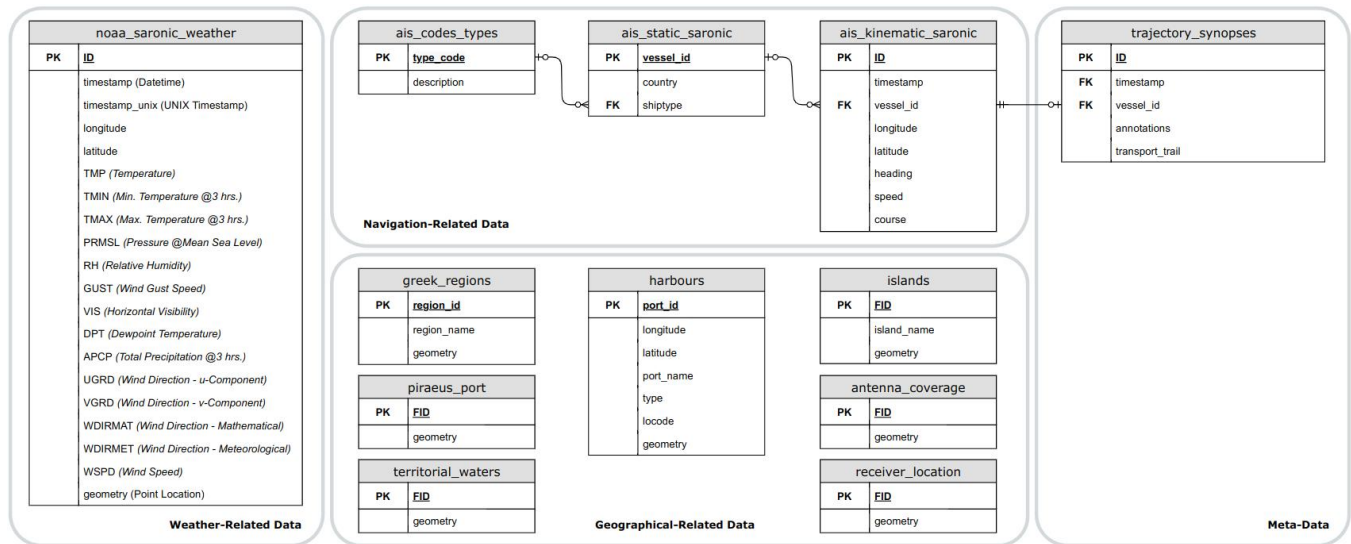
Με παρόμοια λογική, δημιουργήθηκε η συλλογή **synopses** για την αποθήκευση των δεδομένων τροχιών των πλοίων.

Όλες οι συλλογές, εκτός από τη συλλογή των πλοίων, περιλαμβάνουν χωρικά και χρονικά δεδομένα, επιτρέποντας την ανάκτηση πληροφορίας τόσο σε χωρικά όσο και σε χωροχρονικά ερωτήματα.

Οι συλλογές της βάσης συγκεντρώνονται στον Πίνακα 1.

Συλλογή	Περιγραφή
dynamic_vessels	Δυναμικά δεδομένα πλοίων.
geodata	Γεωγραφικά δεδομένα, όπως λιμάνια, νησιά.
vessels	Στατικά δεδομένα πλοίων, όπως όνομα, τύπος και σημαία.
weather_data	Μετεωρολογικά δεδομένα.
synopses	Δεδομένα τροχιών πλοίων.

Πίνακας 1: Περιγραφή των συλλογών δεδομένων στη MongoDB.



Σχήμα 1: Διάγραμμα της αρχικής σχεσιακής δομής δεδομένων.

5 Υλοποίηση και Τεχνική Διαδικασία

Για τις ανάγκες τις εργασίας χρησιμοποιήθηκε η εφαρμογή **MongoDB Compass** για την εγκατάσταση και διαχείριση της βάσης. Η εισαγωγή των δεδομένων στη βάση ήταν μια απαιτητική διαδικασία, καθώς τα δεδομένα προέρχονταν από διαφορετικές πηγές και είχαν ετερογενείς μορφές, όπως **CSV** και **Shapefile**. Για να καταστούν συμβατά με τη **MongoDB**, απαιτήθηκε μετατροπή σε **JSON** με την κατάλληλη δομή και περιεχόμενο.

Για τη μετατροπή και εισαγωγή των δεδομένων, χρησιμοποιήθηκε η γλώσσα προγραμματισμού **Python** με τις βιβλιοθήκες **pandas**, **pyshp** και **pytmongo**. Η διαδικασία περιελάμβανε τη φόρτωση των αρχείων, τον καθαρισμό και την αναδιαμόρφωση των δεδομένων, καθώς και τη μαζική εισαγωγή τους στη βάση με τη χρήση του `insert_many()` της **pytmongo**.

5.1 Φόρτωση Δομένων

Φόρτωση geodata

Τα γεωγραφικά δεδομένα περιλαμβάνουν πληροφορίες για λιμάνια, νησιά και άλλες τοποθεσίες σχετικές με τη ναυσιπλοΐα. Τα αρχεία ήταν αποθηκευμένα σε μορφή **Shapefiles** και μετατράπηκαν σε μορφή **JSON** για να εισαχθούν στη **MongoDB**.

Η αρχική μορφή των δεδομένων σε **Shapefile** περιείχε διαφορετικά αρχεία για κάθε κατηγορία γεωγραφικών δεδομένων. Προσαρμόστηκε το script εισαγωγής έτσι ώστε τα δεδομένα να εισαχθούν σε μια ενιαία συλλογή.

Ακολουθεί ένα δείγμα εγγράφου όπως εισήχθη αρχικά στη συλλογή **geodata**:

```
{
  "_id": ObjectId("679798cfe6230c61e61bbd30"),
  "id": "0",
  "type": "Feature",
  "properties": {
    "Lat": 37.93486,
    "Lon": 23.648899999999998,
```

```
    "Port Name": "ZEA",
    "type": "harbour",
    "locode": "GRMAZ"},
  "geometry": {
    "type": "Point",
    "coordinates": [23.648899999999998,
      37.93486]}}
```

Με βάση τα δεδομένα αυτά αποφασίστηκε να γίνουν οι κάτωθι αλλαγές. Καθότι φαίνεται ότι επαναλαμβάνεται η πληροφορία των συντεταγμένων, δημιουργήθηκε το χαρακτηριστικό `location` που θα περιείχε μόνο τις συντεταγμένες, απλοποιώντας τη δομή. Επιπλέον, κατηγορίες δεδομένων όπως τα νησιά, αρχικά αποθηκεύονταν ως **Multi Polygon**, περιλαμβάνοντας πολλαπλά σημεία που αναπαριστούσαν την περίμετρο του νησιού. Το script προσαρμόστηκε ώστε να υπολογίζει το κεντρικό σημείο του σχήματος και να αποθηκεύει μόνο ένα ζεύγος συντεταγμένων.

Καταργήθηκαν επίσης τα μοναδικά `id` των εγγράφων, που είχαν διαφορετική αρίθμηση ανά κατηγορία. Αντί αυτών, προστέθηκε το χαρακτηριστικό `type` που περιγράφει αν η εγγραφή αφορά λιμάνι, νησί ή άλλη γεωγραφική οντότητα.

Το τελικό έγγραφο διαμορφώθηκε ως εξής:

```
{
  "_id": ObjectId("679bd5cdd63de1c9d605e5e9"),
  "type": "island",
  "location": {
    "longitude": 23.14154282347064,
    "latitude": 37.85930883560959},
  "geometry_type": "point",
  "island_name": "Ovrios"}
```

Φόρτωση Μετεωρολογικών Δεδομένων

Τα μετεωρολογικά δεδομένα ήταν επίσης σε μορφή **Shapefile** και περιείχαν χρονοσειρές μετρήσεων σε διάφορες γεωγραφικές

τοποθεσίες. Πραγματοποιήθηκαν ορισμένες τροποποιήσεις για την ομοιομορφία και τη βελτιστοποίηση των δεδομένων.

Αρχικά, τα δεδομένα περιείχαν διπλή χρονική πληροφορία, μία σε μορφή δευτερολέπτων και μία ως συμβολοσειρά (*char type*). Επιλέχθηκε να διατηρηθεί η δεύτερη και να μετατραπεί σε μορφή **ISODate** για αποδοτικότερη αποθήκευση και επεξεργασία. Επιπλέον, οι συντεταγμένες, ομαδοποιήθηκαν στο πεδίο **location** για μεγαλύτερη συνοχή στη δομή των εγγράφων.

Η τελική μορφή του αρχείου είναι η εξής:

```
{  "_id": ObjectId("679bdee01fd467cd2345d916"),
  "type": "weather",
  "location": {
    "longitude": 23.04973,
    "latitude": 37.98541},
  "month": "may",
  "timestamp": ISODate("2017-05-01T00:00:00.000Z"),
  "TMP": 301.6,
  "TMIN": 301.6,
  "TMAX": 301.9,
  "PRMSL": 101251.7,
  "RH": 71.1,
  "GUST": 14.6,
  "DPT": 295.9,
  "WSPD": 12.822246293064254,
  "WDIRIMAT": 131.5221348020183,
  "WDIRMET": 138.47778651979817,
  "APCP": 0,
  "UGRD": -8.5,
  "VGRD": -9.6}
```

Φόρτωση Δεδομένων Πλοίων

Η εισαγωγή των δεδομένων πλοίων ακολούθησε παρόμοια διαδικασία, καθώς τα δεδομένα ήταν σε μορφή **CSV**. Η κύρια αλλαγή που έγινε αφορούσε την εμφώλευση της πληροφορίας του τύπου πλοίου (*ship type*) μέσα στο έγγραφο του πλοίου, αποφεύγοντας έτσι την ανάγκη ξεχωριστής συλλογής με τύπους πλοίων και επιταχύνοντας τα σχετικά ερωτήματα.

Ακολουθεί ένα δείγμα εγγράφου μετά την συγχώνευση:

```
{  "_id": "2fc08fa98ea89a4668a49f4ed238c4f4357e4565cad416358a452440c8e3a9a3",
  "country": "Denmark",
  "shiptype": {
    "code": 36,
    "description": "Sailing"}}
```

Φόρτωση Δυναμικών Δεδομένων

Τα δυναμικά δεδομένα πλοίων φορτώθηκαν στη βάση με ελάχιστες τροποποιήσεις. Η ημερομηνία μετατράπηκε σε μορφή **ISODate** για συμβατότητα με χρονικά ερωτήματα, ενώ η πληροφορία της χώρας κάθε πλοίου εμφωλεύτηκε μέσα στη δομή του εγγράφου, όπως είχε αποφασιστεί για αποδοτικότερη ανάκτηση δεδομένων.

Ακολουθεί ένα δείγμα εγγράφου μετά την προσαρμογή:

```
{  "_id": ObjectId("679bea334a54fc73eacee71e"),
```

```
"timestamp": ISODate("2017-08-01T03
:43:37.000+00:00"),
"vessel": {
  "vessel_id": "4
e7a9d2c3d62c3db3ab26ebab005e3515aa7
b18f26bc9c9db498c3b617be47e2",
  "country": "Panama"},
"location": {
  "longitude": 23.613966666667,
  "latitude": 37.880408333333},
"speed": 7.1,
"heading": 295,
"course": 294.4}
```

Φόρτωση Δεδομένων Σύνοψης τροχιών

Με παρόμοιο τρόπο φορτώθηκαν και τα δεδομένα των τροχιών κρατώντας την πληροφορία των αρχείων, αλλάζοντας μόνο τον τύπο του *timestamp*, και εμφωλεύοντας την πληροφορία της σημαίας του πλοίου.

Ακολουθεί ένα δείγμα εγγράφου μετά την προσαρμογή:

```
{  "_id": ObjectId("67a1d3db3510e74decc132b9"),
  "timestamp": ISODate("2017-05-09T21
:12:12.000+00:00"),
  "vessel": {
    "vessel_id": "6333
a4bed601c5bc568fdb1ef169d6147bf8b434
ca0f0787f1ef29cf41ad6229",
    "country": "Bermuda"},
  "location": {
    "longitude": 23.68235,
    "latitude": 37.929333333333},
  "heading": 49.79557984526107,
  "speed": 0.111631703381551,
  "annotations": ["STOP_START", "
CHANGE_IN_SPEED_END"],
  "transport_trail": [
    {
      "topic": "datacsv_saronikos_0",
      "timestamp": 1494364332}}]}
```

5.2 Εφαρμογή Ευρετηρίων

Για τη βελτίωση της απόδοσης των ερωτημάτων, εξετάστηκε η εφαρμογή διαφόρων τύπων ευρετηρίων, λαμβάνοντας υπόψιν τις απαιτήσεις για χωρικές, χρονικές και χωρο-χρονικές αναζητήσεις. Στις συλλογές που περιλαμβάνουν γεωγραφικά δεδομένα, αξιοποιήθηκε η δυνατότητα της MongoDB με το ευρετήριο (*2dsphere*) για την υποστήριξη ερωτημάτων με συντεταγμένες πάνω σε σφαιρική επιφάνεια. Η δημιουργία του ευρετηρίου στη συλλογή **geodata** έγινε ως εξής:

```
db.geodata.createIndex({ "location": "2dsphere" })
```

Με αυτήν την προσέγγιση, επιτρέπεται η αποδοτική εκτέλεση ερωτημάτων όπως η εύρεση λιμανιών ή νησιών σε συγκεκριμένη απόσταση από ένα δεδομένο σημείο. Για τις συλλογές που περιλαμβάνουν χρονοσειρές δεδομένων εξετάστηκε η εφαρμογή ευρετηρίου στο πεδίο *timestamp*. Η συλλογές που έχουν χρονικά δεδομένα είναι

αυτές του καιρού, των δυναμικών δεδομένων και των δεδομένων της τροχιάς. Η εισαγωγή τέτοιου index γίνεται με την παρακάτω εντολή:

```
db.dynamic_vessels.createIndex({ "timestamp": 1 })
```

Αυτό το ευρετήριο επιταχύνει ερωτήματα που φιλτράρουν δεδομένα βάσει συγκεκριμένων χρονικών διαστημάτων, όπως η εύρεση των μετρήσεων μιας συγκεκριμένης ημερομηνίας. Στις παραπάνω συλλογές εφόσον έχουν ταυτόχρονα γεωγραφικές και χρονικές πληροφορίες εξετάστηκε η δημιουργία σύνθετου (*compound*) ευρετηρίου συνδυάζοντας το *location* και το *timestamp* με την παρακάτω εντολή:

```
db.dynamic_vessels.createIndex({ "location": "2dsphere", "timestamp": 1 })
```

Αυτό το ευρετήριο επιτρέπει την αποδοτική εκτέλεση χωρο-χρονικών ερωτημάτων, όπως η εύρεση της θέσης ενός πλοίου σε μια συγκεκριμένη χρονική στιγμή ή η ανάλυση κινήσεων σε μια δεδομένη περιοχή και περίοδο. Ωστόσο, δεν μπορεί να χρησιμοποιηθεί αποκλειστικά για χρονικά ερωτήματα, καθώς το ευρετήριο απαιτεί την ύπαρξη χωρικών δεδομένων στο ερώτημα για να ενεργοποιηθεί (όντας πρώτο). Αντίθετα, μπορεί να αξιοποιηθεί πλήρως για χωρικά ερωτήματα χωρίς χρονικό περιορισμό.

Τα τελικά ευρετήρια πιο διατηρήθηκαν στη βάση αποφασίστηκαν ύστερα από τους πειραματισμούς που αναφέρονται στην επόμενη ενότητα.

5.3 Εξέταση Εφαρμογής Sharding

Κατά τη σχεδίαση της βάσης δεδομένων, εξετάστηκε η δυνατότητα εφαρμογής **sharding** για τη διαχείριση μεγάλων όγκων δεδομένων και τη βελτίωση της απόδοσης των ερωτημάτων. Το *sharding* αποτελεί τεχνική κατανομής δεδομένων σε πολλαπλούς διακομιστές (*shards*), επιτρέποντας την παράλληλη επεξεργασία ερωτημάτων και ενισχύοντας την κλιμακωσιμότητα του συστήματος.

Η αρχική πρόθεση ήταν η εφαρμογή του *sharding* στη συλλογή **dynamic_vessels**, καθώς αυτή περιέχει τον μεγαλύτερο όγκο δεδομένων, με συνεχή εισαγωγή νέων εγγραφών. Ως κλειδί κατακερματισμού (*shard key*) εξετάστηκε το πεδίο *timestamp*, προκειμένου να διασφαλιστεί η ομοιόμορφη κατανομή των δεδομένων στον χρόνο και να αποφευχθεί η υπερφόρτωση συγκεκριμένων *shards*.

Για την υλοποίηση του *sharding*, απαιτείται η ενεργοποίηση του μηχανισμού *sharding* στη βάση δεδομένων και η δημιουργία ευρετηρίου στο πεδίο που θα χρησιμοποιηθεί ως κλειδί κατακερματισμού, καθώς το *shard key* πρέπει υποχρεωτικά να διαθέτει δείκτη. Η διαδικασία που είχε προγραμματιστεί περιλαμβάνει τα παρακάτω βήματα:

```
sh.enableSharding("maritime")
db.dynamic_vessels.createIndex({ "timestamp": 1 })
sh.shardCollection("maritime.dynamic_vessels", { "timestamp": 1 })
```

Η εντολή *enableSharding* ενεργοποιεί το *sharding* στη βάση **maritime**, ενώ η εντολή *createIndex* δημιουργεί το απαραίτητο ευρετήριο στο πεδίο *timestamp*. Τέλος, η εντολή *shardCollection* εφαρμόζει το *sharding* στη συλλογή **dynamic_vessels** με βάση το επιλεγμένο κλειδί. Δυστυχώς, η υλοποίηση του *sharding* δεν κατέστη δυνατή λόγω τεχνικών δυσκολιών κατά την ενεργοποίηση του μηχανισμού σε *localhost* επίπεδο.

Η αξιολόγηση της απόδοσης επρόκειτο να πραγματοποιηθεί σε δύο βασικούς άξονες. Πρώτον, θα συγκρινόταν η ταχύτητα εκτέλεσης των ερωτημάτων πριν και μετά την ενεργοποίηση του *sharding*, ώστε να εκτιμηθεί η επιβάρυνση ή η βελτίωση της απόδοσης στη διαχείριση μεγάλου όγκου δεδομένων. Δεύτερον, θα αξιολογούνταν η ταχύτητα φόρτωσης δεδομένων στη βάση, μετρώντας τον χρόνο εισαγωγής δεδομένων πριν και μετά την εφαρμογή του *sharding*, ώστε να διαπιστωθεί αν η κατανεμημένη αποθήκευση επιταχύνει τη διαδικασία φόρτωσης.

6 Πειραματική Αξιολόγηση και Ανάλυση Απόδοσης

Στην παρούσα εργασία πραγματοποιήθηκαν πειραματισμοί με στόχο τη βελτίωση της απόδοσης, την κλιμακωσιμότητα και την αποδοτική διαχείριση του αυξανόμενου όγκου δεδομένων στη βάση.

Αρχικά, φορτώθηκαν στη βάση δεδομένων όλες οι συλλογές για το έτος 2017 και εξετάστηκε η ταχύτητα εκτέλεσης των ερωτημάτων υπό διαφορετικές διαμορφώσεις ευρετηρίων. Συγκεκριμένα, αξιολογήθηκαν οι εξής περιπτώσεις:

- Εφαρμογή χωρικού ευρετηρίου (*2dsphere*).
- Συνδυασμός χωρικού και χρονικού ευρετηρίου σε ξεχωριστά πεδία.
- Χρήση σύνθετου (*compound*) χωροχρονικού ευρετηρίου.

Στη συνέχεια, δεδομένης της εφαρμογής των πιο αποδοτικών ευρετηρίων, εξετάστηκε η επίδραση της αύξησης του όγκου δεδομένων στην απόδοση των ερωτημάτων. Συγκρίθηκαν οι εξής περιπτώσεις:

- Ταχύτητα εκτέλεσης ερωτημάτων με δεδομένα των ετών 2017-2018.
- Ταχύτητα εκτέλεσης ερωτημάτων με την προσθήκη των δεδομένων του 2019.

Η ανάλυση επικεντρώθηκε στον αντίκτυπο της αύξησης των δεδομένων στη διάρκεια εκτέλεσης των ερωτημάτων και στην αποτελεσματικότητα των ευρετηρίων, προκειμένου να αξιολογηθεί η κλιμακωσιμότητα της βάσης δεδομένων και η συνολική απόδοσή της.

Επιπλέον, πραγματοποιήθηκε πειραματισμός με πιο σύνθετα ερωτήματα σχεσιακού τύπου, τα οποία συνδυάζουν πληροφορίες από διαφορετικές συλλογές. Χαρακτηριστικά παραδείγματα περιλαμβάνουν:

- Υπολογισμός του αριθμού των πλοίων που πέρασαν εντός συγκεκριμένης απόστασης από ένα νησί κατά τη διάρκεια μιας χρονικής περιόδου.
- Ανάλυση των διαδρομών πλοίων που διήλθαν από συγκεκριμένη θαλάσσια περιοχή υπό συγκεκριμένες καιρικές συνθήκες (π.χ. ταχύτητα ανέμου πάνω από 10 m/s).
- Εντοπισμός πλοίων που εκτέλεσαν απότομες αλλαγές πορείας κοντά σε λιμάνια.

Ο στόχος αυτών των πειραμάτων ήταν να διερευνηθεί η απόδοση των σχεσιακών ερωτημάτων σε ένα μη σχεσιακό περιβάλλον, εξετάζοντας την αποτελεσματικότητα των ευρετηρίων και τις προκλήσεις που προκύπτουν από τη σύνθετη ανάλυση δεδομένων με τη MongoDB.

6.1 Ερωτήματα που εξετάστηκαν

Όπως αναφέρθηκε και στη δεύτερη ενότητα, η βάση σχεδιάστηκε και υλοποιήθηκε με στόχο την εκτέλεση ερωτημάτων τόσο σχεσιακού τύπου όσο και χωρικών και χωρο-χρονικών αναλύσεων. Για τους πειραματισμούς που πραγματοποιήθηκαν, επιλέχθηκαν τρία αντιπροσωπευτικά ερωτήματα, καθένα από τα οποία ανήκει σε μία από τις προαναφερθείσες κατηγορίες.

Σχεσιακού τύπου ερώτημα. Το ερώτημα αυτό αφορά την αναζήτηση πλοίων που φέρουν την ελληνική σημαία και των οποίων το όνομα περιέχει έναν συγκεκριμένο χαρακτήρα. Η αναζήτηση πραγματοποιείται στη συλλογή **vessels**

```
var startTime = new Date();
var results = db.vessels.find({
  "country": "Greece",
  "shiptype.description": { $regex: "A",
    $options: "i" }
}).toArray();
var endTime = new Date();
printjson(results);
print("Execution Time: " + (endTime - startTime) +
  " ms");
```

Το ερώτημα αναζητά πλοία των οποίων το όνομα περιέχει το γράμμα "Α", χωρίς ευαισθησία σε πεζά-κεφαλαία.

Χωρικό ερώτημα. Το ερώτημα αυτό εντοπίζει πλοία που πέρασαν σε απόσταση μικρότερη του ενός χιλιομέτρου από ένα συγκεκριμένο σημείο. Η αναζήτηση πραγματοποιείται στη συλλογή **dynamic_vessels**.

```
var startTime = new Date();

var results = db.dynamic_vessels.find({
  "location": {
    $near: {
      $geometry: {
        "type": "Point",
        "coordinates":
          [23.48947344541,
            37.83419543412]},
      $maxDistance: 1000}
    }
}).toArray();

var endTime = new Date();

printjson(results);
print("Execution Time: " + (endTime -
  startTime) + " ms");
```

Η εντολή *\$near* επιτρέπει την αναζήτηση πλοίων που βρίσκονται εντός μιας ακτίνας 1 km από το σημείο με συντεταγμένες (23.489, 37.834). Το *\$maxDistance* καθορίζει το εύρος της αναζήτησης σε μέτρα.

Χωρο-χρονικό ερώτημα. Το παρακάτω ερώτημα εκτελείται στη συλλογή **dynamic_vessels** συνδυάζοντας χωρικά και χρονικά φίλτρα.

```
var startTime = new Date();
var results = db.dynamic_vessels.find({
```

```
  "location": {
    $near: {
      $geometry: {
        "type": "Point",
        "coordinates":
          [23.48947344541,
            37.83419543412]},
      $maxDistance: 1000
    }
  },
  "timestamp": {
    $gte: ISODate("2017-06-01T00:00:00Z"),
    $lte: ISODate("2017-06-30T23:59:59Z")
  }
}).toArray();
var endTime = new Date();
printjson(results);
print("Execution Time: " + (endTime -
  startTime) + " ms");
```

Το ερώτημα φιλτράρει τα δεδομένα ώστε να περιλαμβάνουν μόνο πλοία που βρίσκονται εντός ακτίνας 1 km από το σημείο (23.489, 37.834) και των οποίων η χρονική σήμανση βρίσκεται μεταξύ 1ης και 30ής Ιουνίου 2017.

Τα αποτελέσματα από τα παραπάνω ερωτήματα αξιολογήθηκαν με βάση τους χρόνους εκτέλεσης, οι οποίοι μετρήθηκαν μέσω της διαφοράς *endTime - startTime*.

6.2 Εύρεση Βέλτιστου Ευρετηρίου

Για την εύρεση του βέλτιστου ευρετηρίου, πραγματοποιήθηκε ανάλυση των χρόνων εκτέλεσης των χωρικών και χωρο-χρονικών ερωτημάτων στη συλλογή **dynamic_vessels**, χρησιμοποιώντας τα δεδομένα του 2017. Η αξιολόγηση έγινε με βάση τόσο τον χρόνο απόκρισης των ερωτημάτων όσο και το μέγεθος των αντίστοιχων ευρετηρίων.

Στον πίνακα 2 παρουσιάζονται οι χρόνοι εκτέλεσης των ερωτημάτων καθώς και το μέγεθος των ευρετηρίων για διαφορετικούς τύπους ευρετηρίων.

Σύμφωνα με τα αποτελέσματα του πίνακα, το βέλτιστο ευρετήριο για τη συλλογή **dynamic_vessels** είναι το συνδυαστικό (*compound*) ευρετήριο που περιλαμβάνει τόσο το χωρικό *2dsphere* όσο και το χρονικό πεδίο *timestamp*. Αυτό επιτυγχάνει τον χαμηλότερο χρόνο εκτέλεσης στα χωρο-χρονικά ερωτήματα, μειώνοντας τον χρόνο από 1272ms σε μόλις 324ms, ενώ παράλληλα καταναλώνει λιγότερο αποθηκευτικό χώρο συγκριτικά με την ύπαρξη δύο ξεχωριστών ευρετηρίων.

Η απόδοση στα καθαρά χωρικά ερωτήματα παραμένει σχεδόν αμετάβλητη ανεξάρτητα από τη δομή του ευρετηρίου, καθώς η χρήση του *2dsphere* εξασφαλίζει υψηλή απόδοση σε όλες τις περιπτώσεις.

Η ύπαρξη δύο ξεχωριστών ευρετηρίων, ενός χωρικού και ενός χρονικού, δεν προσφέρει βελτίωση στην απόδοση των χωρο-χρονικών ερωτημάτων. Στην πραγματικότητα, ο χρόνος εκτέλεσης παραμένει ίδιος σαν να μην υπήρχε καθόλου χρονικό ευρετήριο. Αυτό συμβαίνει επειδή η MongoDB μπορεί να χρησιμοποιήσει μόνο ένα ευρετήριο κατά την εκτέλεση ενός ερωτήματος. Στην περίπτωση χωρο-χρονικών ερωτημάτων, το σύστημα επιλέγει πρώτα το χωρικό ευρετήριο για τον περιορισμό των δεδομένων, με αποτέλεσμα να

Πίνακας 2: Σύγκριση διαφορετικών τύπων ευρετηρίων στη συλλογή `dynamic_vessels`

Τύπος Ευρετηρίων	Χωρικό Ερώτημα (ms)	Χωρο-Χρονικό Ερώτημα (ms)	Μέγεθος Ευρετηρίου (MB)
Χωρικό (2dsphere)	1434	1272	1200
Σύνθετο (2dsphere, timestamp)	1398	324	882.5
2dsphere + timestamp ξεχωριστά	1405	1235	1777

μην μπορεί να εκμεταλλευτεί το ξεχωριστό χρονικό ευρετήριο για περαιτέρω φιλτράρισμα.

Τέλος, παρατηρήθηκε ότι η MongoDB διατηρεί στη μνήμη RAM τα αποτελέσματα των πρώτων ερωτημάτων, μειώνοντας τον χρόνο εκτέλεσης των επόμενων αναζητήσεων που αφορούν παρόμοια δεδομένα. Αυτό σημαίνει ότι μετά την πρώτη εκτέλεση, οι επόμενες γίνονται σημαντικά ταχύτερες λόγω της αποθήκευσης των δεδομένων στη RAM. Για μια πιο αντικειμενική σύγκριση της απόδοσης των ευρετηρίων, γινόταν επανεκκίνηση της βάσης δεδομένων (*restart*) πριν από κάθε εκτέλεση ερωτήματος.

Με δεδομένο ότι το συνδυαστικό χωρο-χρονικό ευρετήριο είναι και το πιο αποδοτικό, τόσο σε μέγεθος μνήμης όσο και σε χρόνους, εφαρμόστηκε και στις υπόλοιπες συλλογές που είχαν χωρικά και χρονικά δεδομένα.

6.3 Σύγκριση διαφορετικού μεγέθους βάσης

Για την ανάλυση της κλιμακωσιμότητας της βάσης δεδομένων, πραγματοποιήθηκε σύγκριση των χρόνων εκτέλεσης των ερωτημάτων σε βάσεις δεδομένων διαφορετικού μεγέθους.

Η ανάλυση επικεντρώθηκε στη μέτρηση των χρόνων εκτέλεσης των ερωτημάτων, καθώς και στη συσχέτιση αυτών με το συνολικό μέγεθος της βάσης δεδομένων. Στον πίνακα 3 παρουσιάζονται τα αποτελέσματα των μετρήσεων.

Πίνακας 3: Σύγκριση χρόνων εκτέλεσης ερωτημάτων σε διαφορετικά μεγέθη βάσης δεδομένων

Ερώτημα	2017-2018	2017-2019
Σχεσιακό ερώτημα	38 ms	40 ms
Χωρικό ερώτημα	4612 ms	15674 ms
Χωρο-χρονικό ερώτημα	472 ms	1071 ms

Όπως φαίνεται από τα αποτελέσματα, η αύξηση του μεγέθους της βάσης δεδομένων επηρεάζει διαφοροποιημένα την απόδοση των ερωτημάτων, ανάλογα με τον τύπο τους. Το σχεσιακό ερώτημα παρουσιάζει ελάχιστη διαφορά στον χρόνο εκτέλεσης, γεγονός που εξηγείται από το ότι η συλλογή `vessels` δεν επηρεάστηκε από την αύξηση των δεδομένων, καθώς δεν περιείχε νέες εγγραφές. Επομένως, η συνολική αύξηση του μεγέθους της βάσης δεν επηρεάζει την απόδοση ερωτημάτων που εκτελούνται σε στατικές συλλογές.

Αντίθετα, στο χωρικό ερώτημα παρατηρείται σημαντική αύξηση του χρόνου εκτέλεσης, καθώς το μέγεθος της συλλογής `dynamic_vessels` αυξήθηκε σημαντικά με την προσθήκη δεδομένων του 2019. Η MongoDB χρειάστηκε να αναζητήσει και να επεξεργαστεί πολύ μεγαλύτερο αριθμό εγγραφών, οδηγώντας σε επιβράδυνση της εκτέλεσης. Αυτό υποδηλώνει ότι το μέγεθος της συλλογής παίζει καθοριστικό ρόλο στην αποδοτικότητα των χωρικών ερωτημάτων, ειδικά όταν αυξάνεται ο όγκος των δεδομένων που πρέπει να σαρωθούν.

Το χωρο-χρονικό ερώτημα παρουσίασε μικρότερη αύξηση του χρόνου εκτέλεσης σε σύγκριση με το καθαρά χωρικό ερώτημα. Αυτό οφείλεται στην αποδοτικότητα του συνδυαστικού χωρο-χρονικού ευρετηρίου (*compound index*), το οποίο επιτρέπει το γρήγορο φιλτράρισμα των εγγραφών βάσει τόσο γεωγραφικών όσο και χρονικών περιορισμών.

6.4 Δοκιμή Ερωτημάτων Σύζευξης

Παρουσιάζονται δύο ερωτήματα σύζευξης που εκτελέστηκαν στη βάση, αξιοποιώντας τη δυνατότητα συσχετίσεων μεταξύ των συλλογών. Τα ερωτήματα αυτά συνδυάζουν χωρική και θεματική πληροφορία, επιτρέποντας την εξαγωγή σύνθετων συμπερασμάτων από τη βάση δεδομένων.

Το πρώτο ερώτημα αφορά τον εντοπισμό πλοίων που βρέθηκαν εντός απόστασης 5 χιλιομέτρων από ένα συγκεκριμένο νησί, χρησιμοποιώντας τη χωρική αναζήτηση με το `$nearSphere`.

```
const island = db.geodata.findOne(
  { "island_name": "Trakhyli" },
  { "location": 1, "_id": 0 });

db.dynamic_vessels.find({
  "location": {
    "$nearSphere": {
      "$geometry": {
        "type": "Point",
        "coordinates": [island.location.
          longitude, island.location.
          latitude]},
      "$maxDistance": 5000}}});
```

Το παραπάνω ερώτημα εκτελείται σε δύο βήματα. Αρχικά, γίνεται ανάκτηση των συντεταγμένων του νησιού "Trakhyli" από τη συλλογή `geodata`. Στη συνέχεια, πραγματοποιείται αναζήτηση στη συλλογή `dynamic_vessels` χρησιμοποιώντας το `$nearSphere`, το οποίο βρίσκει όλα τα πλοία που καταγράφηκαν εντός 5 χιλιομέτρων από το σημείο αυτό. Η χρήση του `$geometry` εξασφαλίζει ότι η αναζήτηση βασίζεται σε χωρικά δεδομένα, ενώ το `$maxDistance` καθορίζει τη μέγιστη απόσταση των πλοίων από το σημείο αναφοράς.

```
db.dynamic_vessels.aggregate([
  { "$lookup": {
    "from": "vessels",
    "localField": "vessel.vessel_id",
    "foreignField": "_id",
    "as": "vessel_info"}}},
  { "$unwind": "$vessel_info"},
  { "$match": {
    "vessel_info.shiptype.description": "
      Sailing",
```


Πίνακας 4: Τελική Εικόνα της Βάσης Δεδομένων στη MongoDB

Συλλογή	Μέγεθος Αποθήκευσης (GB)	Αριθμός Εγγράφων	Μέσο Μέγεθος Εγγράφου (B)	Μέγεθος Ευρετηρίων (GB)
dynamic_vessels	16.77	244M	254.00	8.79
geodata	0.000024	158	140.00	0.000073
synopses	1.39	15M	367.00	0.839
vessels	0.544	6.2K	169.00	0.000696
weather_data	0.028	614K	335.00	0.021

```
"timestamp": {
  "$gte": ISODate("2017-08-01T00:00:00Z"),
  "$lte": ISODate("2017-08-01T23:59:59Z")}}}});
```

Το δεύτερο ερώτημα χρησιμοποιεί το **Aggregation Framework** της MongoDB για την εκτέλεση μιας πιο σύνθετης αναζήτησης. Η διαδικασία αποτελείται από τρία στάδια:

- (1) **lookup**: Η εντολή αυτή εκτελεί ένα *join* μεταξύ της συλλογής **dynamic_vessels** και της **vessels**, συνδέοντας τις πληροφορίες των πλοίων με βάση το **vessel_id**.
- (2) **unwind**: Δεδομένου ότι η *\$lookup* δημιουργεί έναν πίνακα με τα αποτελέσματα της αντιστοίχισης, η *\$unwind* χρησιμοποιείται για να μετατρέψει τις εγγραφές σε μορφή επίπεδων εγγράφων.
- (3) **match**: Το τελικό φίλτρο εφαρμόζει τις συνθήκες επιλογής, κρατώντας μόνο τα πλοία τύπου "Sailing" που καταγράφηκαν την 1η Αυγούστου του 2017.

Αυτά τα πειράματα αποδεικνύουν ότι η MongoDB μπορεί να υποστηρίξει αποδοτικά τέτοια ερωτήματα ακόμα και σε ένα μη σχεσιακό περιβάλλον, αρκεί να γίνεται σωστός σχεδιασμός του σχήματος δεδομένων και των ευρετηρίων.

6.5 Τελική Εικόνα της Βάσης Δεδομένων

Στον Πίνακα 4 παρουσιάζεται η τελική εικόνα της βάσης δεδομένων με τα αναλυτικά μεγέθη κάθε συλλογής.

Η μεγαλύτερη συλλογή σε όγκο είναι η **dynamic_vessels**, η οποία περιλαμβάνει 244 εκατομμύρια έγγραφα και καταλαμβάνει 16.77 GB αποθηκευτικού χώρου, με συνολικό μέγεθος ευρετηρίων 8.79 GB. Αντίστοιχα, η συλλογή **synopses**, που περιέχει δεδομένα τροχιών, αποτελεί τη δεύτερη μεγαλύτερη σε όγκο, με 15 εκατομμύρια έγγραφα και συνολικό αποθηκευτικό χώρο 1.39 GB. Οι υπόλοιπες συλλογές, όπως τα στατικά δεδομένα πλοίων (**vessels**) και τα μετεωρολογικά δεδομένα (**weather_data**), είναι μικρότερες σε όγκο, με μέγεθος που κυμαίνεται από 28.31 MB έως 544.77 KB.

7 Συμπεράσματα και Μελλοντικές Κατευθύνσεις

Στην παρούσα εργασία, υλοποιήθηκε και αξιολογήθηκε μια μη σχεσιακή βάση δεδομένων στη MongoDB για τη διαχείριση ναυτιλιακών δεδομένων μεγάλης κλίμακας μέσω προσεκτικής μοντελοποίησης και επιλογής κατάλληλων ευρετηρίων. Οι πειραματικές μετρήσεις έδειξαν ότι η χρήση σύνθετων ευρετηρίων (*compound indexes*) που συνδυάζουν τη χωρική και τη χρονική πληροφορία

βελτιστοποιεί την απόκριση των χωρο-χρονικών ερωτημάτων, μειώνοντας σημαντικά τον χρόνο εκτέλεσης. Παράλληλα, η σωστή οργάνωση των συλλογών επέτρεψε τη διατήρηση της απόδοσης και για καθαρά σχεσιακού τύπου ερωτήματα, όπως η αναζήτηση πλοίων με συγκεκριμένα χαρακτηριστικά.

Η σύγκριση διαφορετικών προσεγγίσεων στη μοντελοποίηση των δεδομένων ανέδειξε τη σημασία της προσεκτικής επιλογής εμφώλευσης ανάλογα με τις ανάγκες των ερωτημάτων και το μέγεθος των δεδομένων. Επιπλέον, εξετάστηκε η εφαρμογή κατακερματισμού δεδομένων (*sharding*), ωστόσο τεχνικοί περιορισμοί δεν επέτρεψαν την πλήρη αξιολόγησή του.

Για μελλοντική διερεύνηση, προτείνονται οι εξής βελτιώσεις:

- **Εφαρμογή sharding**: Δοκιμή και αξιολόγηση του *sharding* σε ένα κατανεμημένο σύστημα, ώστε να εκτιμηθεί η απόδοσή του στην κλιμάκωση της βάσης δεδομένων.
- **Βελτιστοποίηση του μοντέλου δεδομένων**: Αναθεώρηση της επιλογής εμφώλευσης με βάση τις επιδόσεις διαφορετικών ερωτημάτων σε πραγματικά δεδομένα.
- **Διεύρυνση των ερωτημάτων**: Ανάπτυξη πιο σύνθετων χωρο-χρονικών αναλύσεων, όπως η παρακολούθηση μοτίβων κίνησης πλοίων και η αξιολόγηση της επίδρασης των μετεωρολογικών συνθηκών στη ναυσιπλοΐα.

Συμπερασματικά, η παρούσα εργασία παρείχε μια ολοκληρωμένη λύση για τη διαχείριση και ανάλυση ναυτιλιακών δεδομένων στη MongoDB, αποδεικνύοντας ότι οι κατάλληλες τεχνικές ευρετηρίασης και μοντελοποίησης μπορούν να βελτιστοποιήσουν την απόδοση τόσο σε χωρικά και χωρο-χρονικά ερωτήματα όσο και σε κλασικές σχεσιακές αναζητήσεις. Με την περαιτέρω εξέλιξη της υποδομής, η προσέγγιση αυτή μπορεί να επεκταθεί για τη διαχείριση ακόμα μεγαλύτερων όγκων δεδομένων σε πραγματικές συνθήκες.

Αναφορές

- [1] Shannon Bradshaw, Eoin Brazil και Kristina Chodorow. *MongoDB: The Definitive Guide: Powerful and Scalable Data Storage*. 3rd. O'Reilly Media, 2020. ISBN: 978-1491954461.
- [2] MongoDB. *MongoDB Manual*. URL: <https://www.mongodb.com/docs/manual/>.
- [3] Ramakrishnan Raghu και Johannes Gehrke. *Συστήματα Διαχείρισης Βάσεων Δεδομένων*. ΤΖΙΟΛΑΣ, 2011. ISBN: 9789604184118.
- [4] Andreas Tritsarolis, Yannis Kontoulis και Yannis Theodoridis. "The Piraeus AIS dataset for large-scale maritime data analytics". Στο: *Data in Brief* (2021). doi: 10.1016/j.dib.2021.107782.
- [5] Χ. Δουλκερίδης. *Διαφάνειες διαλέξεων μαθήματος Διαχείριση Δεδομένων για Σχεσιακές και Μη-σχεσιακές Βάσεις Δεδομένων*. 2024.
- [6] Χ. Δουλκερίδης και Ν. Κουτρουμάνης. *Διαφάνειες Εργαστηρίων MongoDB*. 2024.