



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
UNIVERSITY OF PIRAEUS

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ
DEPARTMENT OF INFORMATICS

Ονοματεπώνυμα: Αγγελική Αποστόλου, Γλυκερία Καλαμάρα, Αντωνία Νικολέττα Γιαννουλάκη

Αριθμοί Μητρώου: Π19015, Π21042, Π19251

E-mails: aapostolou12@gmail.com, glykeriakalamara@gmail.com,
toniagiannoulaki@gmail.com

Εξεταζόμενο μάθημα: Συστήματα Διαχείρισης Βάσεων Δεδομένων

Επιβλέποντες Καθηγητές: Ιωάννης Θεοδωρίδης, Ιωάννης Κοντούλης

Ημερομηνία Παράδοσης: 06/02/2024

Περιεχόμενα

| | |
|---------------------------------------------------------------------|----|
| 1. Εκφώνηση | 5 |
| 2. Ερώτημα 1..... | 7 |
| 2.1 Υποερώτημα i | 7 |
| 2.1.1 Κώδικας..... | 7 |
| 2.1.2 Υποερώτημα i: Ανάλυση κώδικα | 7 |
| 2.1.3 Υποερώτημα i: Στιγμιότυπο από την εκτέλεση του κώδικα..... | 8 |
| 2.2 Υποερώτημα ii..... | 8 |
| 2.2.1 Κώδικας | 8 |
| 2.2.2 Υποερώτημα ii: Ανάλυση κώδικα..... | 9 |
| 2.2.3 Υποερώτημα ii: Στιγμιότυπο από την εκτέλεση του κώδικα | 9 |
| 2.3 Υποερώτημα iii | 10 |
| 2.3.1 Κώδικας | 10 |
| 2.3.2 Υποερώτημα iii: Ανάλυση κώδικα | 10 |
| 2.3.3 Υποερώτημα iii: Στιγμιότυπο από την εκτέλεση του κώδικα | 10 |
| 2.4 Υποερώτημα iv..... | 11 |
| 2.4.1 Κώδικας | 11 |
| 2.4.2 Υποερώτημα iv: Ανάλυση κώδικα..... | 11 |
| 2.4.3 Υποερώτημα iv: Στιγμιότυπο από την εκτέλεση του κώδικα | 12 |
| 2.5 Υποερώτημα v..... | 12 |
| 2.5.1 Κώδικας | 12 |
| 2.5.2 Υποερώτημα v: Ανάλυση κώδικα..... | 13 |
| 2.5.3 Υποερώτημα v: Στιγμιότυπο από την εκτέλεση του κώδικα | 14 |
| 3. Ερώτημα 2 | 15 |
| 3.1 Υποερώτημα i | 16 |
| 3.2 Υποερώτημα ii..... | 16 |
| 3.3 Υποερώτημα iii..... | 17 |
| 3.4 Υποερώτημα iv..... | 17 |
| 3.5 Υποερώτημα v..... | 18 |
| 4. Ερώτημα 3 | 19 |
| 4.1 Υποερώτημα i | 20 |
| 4.2 Υποερώτημα ii..... | 20 |
| 4.3 Υποερώτημα iii..... | 21 |

| | |
|---------------------------------------------------------------------|----|
| 4.4 Υποερώτημα iv..... | 21 |
| 4.5 Υποερώτημα v..... | 22 |
| 5. Ερώτημα 4..... | 23 |
| 5.1 Υποερώτημα i | 23 |
| 5.1.1 Κώδικας..... | 23 |
| 5.1.2 Υποερώτημα i: Ανάλυση κώδικα | 23 |
| 5.1.3 Υποερώτημα i: Στιγμιότυπο από την εκτέλεση του κώδικα..... | 24 |
| 5.2 Υποερώτημα ii..... | 24 |
| 5.2.1 Κώδικας..... | 24 |
| 5.2.2 Υποερώτημα ii: Ανάλυση κώδικα..... | 24 |
| 5.2.3 Υποερώτημα ii: Στιγμιότυπο από την εκτέλεση του κώδικα | 25 |
| 5.3 Υποερώτημα iii | 25 |
| 5.3.1 Κώδικας..... | 25 |
| 5.3.2 Υποερώτημα iii: Ανάλυση κώδικα | 25 |
| 5.3.3 Υποερώτημα iii: Στιγμιότυπο από την εκτέλεση του κώδικα | 26 |
| 5.4 Υποερώτημα iv..... | 26 |
| 5.4.1 Κώδικας..... | 26 |
| 5.4.2 Υποερώτημα iv: Ανάλυση κώδικα..... | 26 |
| 5.4.3 Υποερώτημα iv: Στιγμιότυπο από την εκτέλεση του κώδικα | 27 |
| 5.5 Υποερώτημα v..... | 27 |
| 5.5.1 Κώδικας..... | 27 |
| 5.5.2 Υποερώτημα v: Ανάλυση κώδικα..... | 28 |
| 5.5.3 Υποερώτημα v: Στιγμιότυπο από την εκτέλεση του κώδικα | 28 |
| 6. Ερώτημα 5 | 29 |
| 6.1 Υποερώτημα i | 30 |
| 6.2 Υποερώτημα ii..... | 30 |
| 6.2.1 Κώδικας..... | 30 |
| 6.2.2 Υποερώτημα ii: Ανάλυση Κώδικα..... | 30 |
| 6.2.3 Υποερώτημα ii: Στιγμιότυπο από την εκτέλεση του κώδικα | 31 |
| 6.2.4 Υποερώτημα ii: Διάγραμμα | 32 |
| 6.3 Υποερώτημα iii | 32 |
| 6.3.1 Κώδικας | 32 |
| 6.3.2 Υποερώτημα iii: Ανάλυση Κώδικα | 32 |

| | |
|---------------------------------------------------------------------|----|
| 6.3.3 Υποερώτημα iii: Στιγμιότυπο από την εκτέλεση του κώδικα | 33 |
| 6.3.4 Υποερώτημα iii: Διάγραμμα..... | 34 |
| 6.4 Υποερώτημα iv..... | 34 |
| 6.4.1 Κώδικας | 34 |
| 6.4.2 Υποερώτημα iv: Ανάλυση Κώδικα..... | 35 |
| 6.4.3 Υποερώτημα iv: Στιγμιότυπο από την εκτέλεση του κώδικα | 35 |
| 6.4.4 Υποερώτημα iv: Διάγραμμα | 36 |
| 6.5 Υποερώτημα v..... | 36 |
| 6.5.1 Κώδικας | 36 |
| 6.5.2 Υποερώτημα v: Ανάλυση Κώδικα..... | 37 |
| 6.5.3 Υποερώτημα v: Στιγμιότυπο από την εκτέλεση του κώδικα | 38 |
| 6.5.4 Υποερώτημα v: Διάγραμμα | 39 |

1. Εκφώνηση

Στο πλαίσιο της εργασίας του μαθήματος, θα εργαστείτε με ένα σύνολο πραγματικών δεδομένων σχετικό με τις θέσεις των πλοίων στον Πειραιά και τον Αργοσαρωνικό, όπως αυτές εκπέμπονται από τα πλοία και συλλέγονται από κατάλληλες κεραίες μέσω του Αυτόματου Συστήματος Αναγνώρισης (Automatic Identification System - AIS). Συγκεκριμένα, το σύνολο δεδομένων που θα χρησιμοποιήσετε αποτελείται από τρία (3) αρχεία CSV, ένα για κάθε πίνακα της ΒΔ. Ειδικότερα, το πρώτο αρχείο (Positions.csv) περιέχει στίγματα πλοίων για το χρονικό διάστημα 01/08/2019 - 30/08/2019, το δεύτερο (Vessels.csv) περιέχει στατικές πληροφορίες για τα πλοία (τύπο και σημαία) και, τέλος, το τρίτο (VesselTypes.csv) περιέχει σύντομη περιγραφή των διαφόρων τύπων πλοίων. Μπορείτε να κατεβάσετε το σύνολο δεδομένων από εδώ: <https://datastories.cs.unipi.gr/index.php/s/ZEM86Fe6i4FeJCI>.

- Positions (id, t, lon, lat, heading, course, speed, vessel_id) // τα στίγματα των πλοίων
- Vessels (id, flag, type) // πληροφορίες σχετικά με τα πλοία
- VesselTypes (code, description) // οι τύποι των πλοίων

Βάσει των παραπάνω, καλείστε να απαντήσετε στα ακόλουθα ερωτήματα 1-5. Προσοχή: κάθε φορά που εκτελείτε μία από τις παρακάτω επερωτήσεις θα δείχνετε τον χρόνο εκτέλεσης (πάντα θα εκτελείτε την επερώτηση τουλάχιστον δύο φορές και θα κρατάτε τον τελευταίο χρόνο – ο πρώτος χρόνος εκτέλεσης δεν είναι αντιπροσωπευτικός, γιατί τα buffers δεν έχουν προλάβει να αρχικοποιηθούν), καθώς και το πλάνο εκτέλεσης (χρησιμοποιώντας την εντολή EXPLAIN, screenshot). Σκοπός είναι κάθε φορά που άλλάζετε κάτι στη ΒΔ, με απώτερο στόχο να βελτιώσετε τους χρόνους εκτέλεσης, να παρατηρείτε αν υπάρχει βελτίωση και πόση είναι αυτή αλλά και να εξηγείτε τη βελτίωση αυτή με βάση τη θεωρία και το πλάνο εκτέλεσης. Η απάντηση στα ερωτήματα 1-5 πρέπει να γίνει με τη σειρά εμφάνισής τους, δηλαδή, θα απαντήσετε στο ερώτημα 2 αφού πρώτα έχετε απαντήσει στο 1 κοκ., έτσι ώστε οι αλλαγές που κάνατε (buffers, parallelism, κτλ.) στο ένα ερώτημα να συνεχίσουν να είναι ενεργές στα επόμενα.

Ερώτημα 1 (30 %)

Αφού φορτώσετε τα δεδομένα στην PostgreSQL (εντολή "COPY ... WITH CSV HEADER") και ανανεώσετε τα στατιστικά χρησιμοποιώντας την εντολή "VACUUM FULL ...", εκτελέστε τις παρακάτω επερωτήσεις (queries) χρησιμοποιώντας τις προεπιλεγμένες ρυθμίσεις της PostgreSQL και χωρίς να έχετε δημιουργήσει βοηθητικές δομές (π.χ. ευρετήρια).

i. Βρείτε τον αριθμό των στιγμάτων (lon, lat) ανά ημερολογιακή ημέρα και ταξινομήστε το αποτέλεσμα σε φθίνουσα σειρά (ως προς το πλήθος των στιγμάτων). Διευκρίνιση: ο συνδυασμός των χαρακτηριστικών lon, lat σε κάθε

εγγραφή του πίνακα Positions συνιστά τη θέση του πλοίου τη συγκεκριμένη χρονική στιγμή t.

ii. Βρείτε πόσα πλοία με ελληνική σημαία ανά τύπο πλοίου είναι καταγεγραμένα στη ΒΔ.

iii. Βρείτε ποια πλοία ανέπτυξαν κάποια στιγμή ταχύτητα άνω των 30 κόμβων , τι τύπου ήταν το κάθε πλοίο και πόσα ήταν αυτά τα πλοία ανά τύπο.

iv. Ειδικά για τα επιβατηγά πλοία (τύποι «passenger ...»), πόσα στίγματα καταγράφηκαν ανά ημέρα την περίοδο 14/08/2019 - 18/08/2019; TIP: description LIKE...

v. Ποια πλοία τύπου cargo ήταν 'αγκυροβολημένα' (ταχύτητα μηδέν) κάποια στιγμή μέσα στην περίοδο 15/08/2019 - 18/08/2019; Ποια για ολόκληρη την περίοδο 12/08/2019 - 19/08/2019;

Ερώτημα 2 (15 %)

Ρυθμίστε την PostgreSQL έτσι ώστε να χρησιμοποιεί ως buffer περισσότερη μνήμη από τη μνήμη RAM του υπολογιστή σας (ικανή ώστε να χωράει όσο γίνεται περισσότερο από το dataset, όλο αν είναι δυνατόν). Έπειτα, εκτελέστε πάλι τις παραπάνω επερωτήσεις και εξηγήστε τι παρατηρείτε. TIP: shared_buffers (π.χ. ALTER SYSTEM SET shared_buffers TO '256MB'; -- απαιτείται επανεκκίνηση του PostgreSQL server).

Ερώτημα 3 (15 %)

Ρυθμίστε την PostgreSQL έτσι ώστε να χρησιμοποιεί όλη την επεξεργαστική ισχύ του υπολογιστή σας. Έπειτα, εκτελέστε πάλι τις παραπάνω επερωτήσεις και εξηγήστε τι παρατηρείτε. TIP: max_parallel_workers_per_gather.

Ερώτημα 4 (20 %)

Δημιουργήστε τα κατάλληλα ευρετήρια στη ΒΔ για να τρέξουν οι παραπάνω επερωτήσεις πιο γρήγορα. Για κάθε ευρετήριο που θα δημιουργήσετε θα εξηγήσετε τους λόγους για τους οποίους επιλέξατε τον συγκεκριμένο τύπο ευρετηρίου, καθώς και το πώς βοηθάει στη βελτίωση του χρόνου εκτέλεσης. Αν κάποιο ευρετήριο δεν βελτιώσει την απόδοση, εξηγείστε γιατί.

Ερώτημα 5 (20 %)

Σπάστε το dataset σε shards/partitions χρησιμοποιώντας τη μέθοδο της επιλογής από αυτές που είδατε στις εργαστηριακές διαλέξεις (διαμέριση μέσω κληρονομικότητας μεταξύ πινάκων / δηλωτική διαμέριση). Υπάρχουν πολλοί τρόποι με τους οποίους μπορείτε να κάνετε το partitioning (π.χ. random, hash,

range, κτλ.), κάθε ομάδα θα επιλέξει μόνο έναν τρόπο και θα επιχειρηματολογήσετε για την επιλογή σας. Έπειτα, εκτελέστε πάλι τις παραπάνω επερωτήσεις. TIP: Σε κάθε πίνακα-παιδί μπορείτε να δημιουργήσετε τα κατάλληλα ευρετήρια για την περαιτέρω βελτίωση του χρόνου εκτέλεσης των ερωτημάτων.

2. Ερώτημα 1

2.1 Υποερώτημα i

2.1.1 Κώδικας

```
SELECT DATE_TRUNC('day', t) AS day, COUNT(*) AS count
FROM Positions
GROUP BY day
ORDER BY count DESC;
```

2.1.2 Υποερώτημα i: Ανάλυση κώδικα

Στην περίπτωση αυτή, καλούμαστε να βρούμε τον αριθμό των στιγμάτων (lon, lat) ανά ημερολογιακή ημέρα και να ταξινομήσουμε το αποτέλεσμα σε φθίνουσα σειρά.

Το παραπάνω κομμάτι κώδικα, εκτελεί μια επερώτηση στη βάση δεδομένων για να ανακτήσει τον αριθμό των εγγραφών για κάθε μέρα. Ο πίνακας που χρησιμοποιείται είναι ο Positions. Η συνάρτηση **DATE_TRUNC('day', t)** χρησιμοποιείται για να αποκοπεί η ώρα από το πεδίο "t" και να πάρει μόνο την ημερομηνία. Το **AS day** ορίζει το όνομα της στήλης που θα δημιουργηθεί για τα αποτελέσματα της συνάρτησης. Η στήλη **COUNT(*)** υπολογίζει τον αριθμό των εγγραφών για κάθε ημερομηνία. Οι εγγραφές ομαδοποιούνται ανά ημέρα, βάσει της στήλης "day" που δημιουργήθηκε από την συνάρτηση **DATE_TRUNC**. Τέλος, τα αποτελέσματα ταξινομούνται με φθίνουσα σειρά βάσει του αριθμού των εγγραφών.

2.1.3 Υποερώτημα i: Στιγμιότυπο από την εκτέλεση του κώδικα

The screenshot shows the pgAdmin 4 interface. The left sidebar is the Object Explorer, displaying the schema structure of the 'Vessels' database, including tables like 'positions' and 'vessels'. The main area is a query editor window titled 'postgres/postgres@Vessels'. It contains the following SQL query:

```
1 SELECT DATE_TRUNC('day', t) AS day, COUNT(*) AS count
2 FROM Positions
3 GROUP BY day
4 ORDER BY count DESC;
```

The results pane shows the output of the query:

| day | count |
|---------------------|--------|
| 2019-08-05 00:00:00 | 695757 |
| 2019-08-04 00:00:00 | 543604 |
| 2019-08-28 00:00:00 | 515392 |
| 2019-08-06 00:00:00 | 490819 |
| 2019-08-11 00:00:00 | 408012 |
| 2019-08-10 00:00:00 | 406090 |
| 2019-08-03 00:00:00 | 357393 |
| 2019-08-29 00:00:00 | 356825 |
| 2019-08-16 00:00:00 | 350259 |
| 2019-08-07 00:00:00 | 341773 |
| 2019-08-27 00:00:00 | 305732 |
| 2019-08-17 00:00:00 | 303885 |
| 2019-08-12 00:00:00 | 284402 |
| 2019-08-01 00:00:00 | 253844 |
| 2019-08-02 00:00:00 | 251133 |
| 2019-08-15 00:00:00 | 248170 |
| 2019-08-09 00:00:00 | 216958 |
| 2019-08-08 00:00:00 | 207113 |
| 2019-08-18 00:00:00 | 201904 |
| 2019-08-19 00:00:00 | 126262 |
| 2019-08-30 00:00:00 | 90467 |
| 2019-08-14 00:00:00 | 72994 |
| 2019-08-13 00:00:00 | 4427 |
| 2019-08-26 00:00:00 | 3436 |

Total rows: 24 of 24 | Query complete 00:00:01.688

2.2 Υποερώτημα ii

2.2.1 Κώδικας

```
SELECT
    vesseltypes.description, COUNT(*) AS number_of_ships, vessels.flag
FROM
    vessels
JOIN
    vesseltypes ON vessels.type = vesseltypes.code AND vessels.flag = 'Greece'
GROUP BY
    vesseltypes.description, vessels.flag;
```

2.2.2 Υποερώτημα ii: Ανάλυση κώδικα

Στην περίπτωση αυτή καλούμαστε να βρούμε πόσα πλοία με ελληνική σημαία ανά τύπο πλοίου είναι καταγεγραμμένα στην βάση δεδομένων.

Το παραπάνω κομμάτι κώδικα εκτελεί μια επερώτηση στη βάση δεδομένων και αποσκοπεί στην ανάκτηση πληροφοριών από τους πίνακες vessels και vesseltypes. Επιλέγονται τρεις στήλες για εμφάνιση στα αποτελέσματα:

- **vesseltypes.description:** Η περιγραφή του τύπου πλοίου.
- **COUNT(*) AS number_of_ships:** Ο αριθμός των πλοίων για κάθε τύπο.
- **vessels.flag:** Η σημαία του πλοίου.

Το Join συνδέει τους πίνακες vessels και vesseltypes όπου ο κωδικός του τύπου πλοίου (vessels.type) πρέπει ταιριάζει με τον κωδικό του τύπου πλοίου στον πίνακα vesseltypes (vesseltypes.code) και παράλληλα η σημαία να είναι ελληνική (vessels.flag = 'Greece'). Αυτό συμβαίνει για να μπορεί να πραγματοποιηθεί η ομαδοποίηση των αποτελεσμάτων πληρώντας τις προϋποθέσεις που δόθηκαν.

2.2.3 Υποερώτημα ii: Στιγμιότυπο από την εκτέλεση του κώδικα

The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows the database schema with tables: positions, vessels, and vesseltypes.
- Query Editor:** Displays the SQL query:SELECT
 vesseltypes.description, COUNT(*) AS number_of_ships, vessels.flag
FROM
 vessels
JOIN
 vesseltypes ON vessels.type = vesseltypes.code AND vessels.flag = 'Greece'
GROUP BY
 vesseltypes.description, vessels.flag;
- Data Output:** Shows the results of the query in a table:

| | description | number_of_ships | flag |
|----|---------------------------------------------------|-----------------|--------|
| 1 | Anti-pollution equipment | 1 | Greece |
| 2 | Cargo, all ships of this type | 14 | Greece |
| 3 | Cargo, Hazardous category D | 1 | Greece |
| 4 | Cargo, No additional information | 2 | Greece |
| 5 | Fishing | 6 | Greece |
| 6 | High speed craft (HSC), all ships of this type | 4 | Greece |
| 7 | High speed craft (HSC), Hazardous category B | 1 | Greece |
| 8 | High speed craft (HSC), No additional information | 3 | Greece |
| 9 | Not available (default) | 4 | Greece |
| 10 | Other Type, all ships of this type | 3 | Greece |
| 11 | Other Type, Hazardous category A | 1 | Greece |
| 12 | Other Type, no additional information | 4 | Greece |
| 13 | Passenger, all ships of this type | 40 | Greece |
| 14 | Passenger, Hazardous category A | 1 | Greece |
| 15 | Passenger, No additional information | 10 | Greece |
| 16 | Pilot Vessel | 4 | Greece |
| 17 | Pleasure Craft | 25 | Greece |
| 18 | Reserved for future use | 1 | Greece |
| 19 | Sailing | 29 | Greece |
| 20 | Search and Rescue vessel | 1 | Greece |
| 21 | Tanker, all ships of this type | 34 | Greece |
| 22 | Tanker, Hazardous category A | 10 | Greece |

2.3 Υποερώτημα iii

2.3.1 Κώδικας

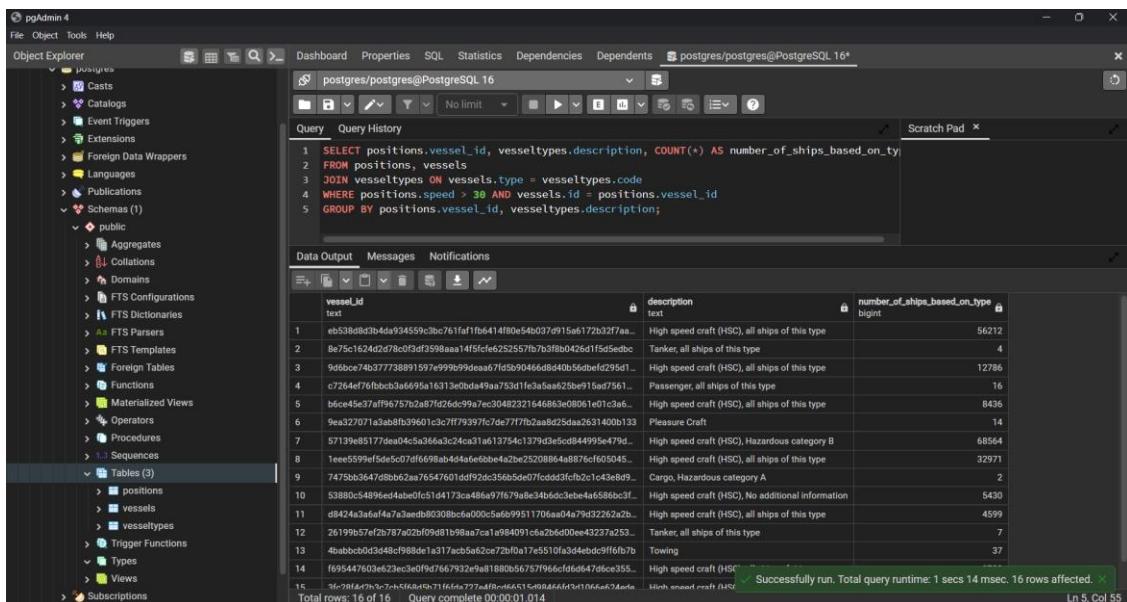
```
SELECT positions.vessel_id, vesseltypes.description, COUNT(*) AS number_of_ships_based_on_type
FROM positions, vessels
JOIN vesseltypes ON vessels.type = vesseltypes.code
WHERE positions.speed > 30 AND vessels.id = positions.vessel_id
GROUP BY positions.vessel_id, vesseltypes.description;
```

2.3.2 Υποερώτημα iii: Ανάλυση κώδικα

Στην περίπτωση αυτή καλούμαστε να βρούμε ποια πλοία ανέπτυξαν κάποια στιγμή ταχύτητα όνω των 30 κόμβων , τι τύπου ήταν το κάθε πλοίο και πόσα ήταν αυτά τα πλοία ανά τύπο.

Το παραπάνω κομμάτι κώδικα χρησιμοποιεί τους πίνακες **positions**, **vessels** και **vesseltypes**. Οι στήλες που εμφανίζονται είναι οι **vessel_id** από τον πίνακα **positions**, η **description** από τον πίνακα **vesseltypes** και το πλήθος των πλοίων (**number_of_ships_based_on_type**) με βάση τον τύπο τους. Χρησιμοποιούμε το **JOIN** σε συνδιασμό με το **WHERE** για να γίνει σύνδεση όλων των πινάκων έτσι ώστε να μπορέσουμε να πάρουμε τα **descriptions**(στοιχείο του 3ου πίνακα) των πλοίων που έχουν **speed**(στοιχείο του 1ου πίνακα) μεγαλύτερο από 30 κόμβους.

2.3.3 Υποερώτημα iii: Στιγμιότυπο από την εκτέλεση του κώδικα



The screenshot shows the pgAdmin 4 interface with the following details:

- File Object Tools Help**: The top menu bar.
- Object Explorer**: A tree view showing databases (postgres), schemas (public), tables (3), and other objects like casts, catalogs, event triggers, extensions, foreign data wrappers, languages, publications, and subscriptions.
- Dashboard Properties SQL Statistics Dependencies Dependents**: The tabs for the current connection.
- Query**: The active tab where the SQL query is entered.
- Data Output**: The tab where the results of the query are displayed.
- Messages Notifications**: Tabs for monitoring the database.
- Scratch Pad**: A temporary workspace for testing queries.
- SQL Query Content:**

```
1 SELECT positions.vessel_id, vesseltypes.description, COUNT(*) AS number_of_ships_based_on_type
2 FROM positions, vessels
3 JOIN vesseltypes ON vessels.type = vesseltypes.code
4 WHERE positions.speed > 30 AND vessels.id = positions.vessel_id
5 GROUP BY positions.vessel_id, vesseltypes.description;
```
- Results Table:** A table showing the output of the query. It has three columns: vessel_id, description, and number_of_ships_based_on_type. The data includes rows for High speed craft (HSC), Tanker, Passenger, and various cargo categories.
- Status Bar:** Shows "Successfully run. Total query runtime: 1 secs 14 msec. 16 rows affected." and "Ln 5, Col 55".

2.4 Υποερώτημα iv

2.4.1 Κώδικας

```
SELECT
    positions.vessel_id, date_trunc('day', t) AS day, COUNT(*) AS number_of_positions
FROM
    positions, vesseltypes
WHERE
    vesseltypes.description LIKE 'Passenger%' AND positions.t BETWEEN '2019-08-14' AND '2019-08-19'
GROUP BY
    positions.vessel_id, day
ORDER BY
    positions.vessel_id, day;
```

2.4.2 Υποερώτημα iv: Ανάλυση κώδικα

Στην περίπτωση αυτή καλούμαστε να βρούμε ειδικά για τα επιβατηκά πλοία, πόσα στίγματα καταγράφηκαν ανά ημέρα την περίοδο 14/08/2019 - 18/08/2019.

Στο παραπάνω κομμάτι κώδικα επιλέγονται τρεις στήλες για εμφάνιση των αποτελεσμάτων: `positions.vessel_id` για το αναγνωριστικό του πλοίου, `date_trunc('day', t) AS day` για την ημέρα (το `date_trunc` χρησιμοποιείται για να αναστραφεί η ημερομηνία στην αρχή της ημέρας), και `COUNT(*) AS number_of_positions` για τον αριθμό των θέσεων.

Η επερώτηση προέρχεται από δύο πίνακες: `positions` και `vesseltypes`. Εφαρμόζονται δύο συνθήκες φίλτρου: `vesseltypes.description LIKE 'Passenger%'` επιλέγονται μόνο οι εγγραφές με περιγραφή πλοίου που ξεκινά με "Passenger", και `positions.t BETWEEN '2019-08-14' AND '2019-08-19'` όπου επιλέγονται μόνο οι εγγραφές από 14/08/2019 έως και 18/08/2019.

Οι εγγραφές ομαδοποιούνται ανά πλοίο (`positions.vessel_id`) και ημέρα (`day`). Τέλος, τα αποτελέσματα ταξινομούνται βάσει του πλοίου (`positions.vessel_id`) και της ημέρας (`day`).

2.4.3 Υποερώτημα iv: Στιγμιότυπο από την εκτέλεση του κώδικα

```

SELECT
    positions.vessel_id, date_trunc('day', t) AS day, COUNT(*) AS number_of_positions
FROM
    positions, vesseltypes
WHERE
    vesseltypes.description LIKE 'Passenger%' AND positions.t BETWEEN '2019-08-14' AND '2019-08-19'
    /* an balw to orio stis 18/08 pairnei ta stoixeia ews kai tis 17/08 */
GROUP BY
    positions.vessel_id, day
ORDER BY
    positions.vessel_id, day;

```

| vessel_id | day | number_of_positions |
|--------------------------------------------------------------------|---------------------|---------------------|
| 01c8d531d4f3dbf4d2ab71f0018d47aa0db2358fb532bf184bbfbfd2b73927b1 | 2019-08-14 00:00:00 | 2430 |
| 01c8d531d4f3dbf4d2ab71f0018d47aa0db2358fb532bf184bbfbfd2b73927b1 | 2019-08-16 00:00:00 | 5640 |
| 04b6c84a518f833b7206a114ada7660d56888de1af6cabcb7bbe2c46ea6a7d... | 2019-08-15 00:00:00 | 4160 |
| 04b6c84a518f833b7206a114ada7660d56888de1af6cabcb7bbe2c46ea6a7d... | 2019-08-16 00:00:00 | 4300 |
| 04b6c84a518f833b7206a114ada7660d56888de1af6cabcb7bbe2c46ea6a7d... | 2019-08-17 00:00:00 | 10540 |
| 04b6c84a518f833b7206a114ada7660d56888de1af6cabcb7bbe2c46ea6a7d... | 2019-08-18 00:00:00 | 6290 |
| 04c3373354007a7d811c13943e2593acceca75d3eecd59c2f6e5c284a5b... | 2019-08-15 00:00:00 | 480 |
| 04c3373354007a7d811c13943e2593acceca75d3eecd59c2f6e5c284a5b... | 2019-08-16 00:00:00 | 9500 |
| 04c3373354007a7d811c13943e2593acceca75d3eecd59c2f6e5c284a5b... | 2019-08-17 00:00:00 | 140 |
| 05edaa0f038e80b4952faf8ea6ee12f2e95066896dc0bc91fdaea37c4912468... | 2019-08-16 00:00:00 | 4330 |
| 05edaa0f038e80b4952faf8ea6ee12f2e95066896dc0bc91fdaea37c4912468... | 2019-08-17 00:00:00 | 340 |
| 05edaa0f038e80b4952faf8ea6ee12f2e95066896dc0bc91fdaea37c4912468... | 2019-08-18 00:00:00 | 50 |
| 064f488f8edabd7beffcc207f05b7ece824c1c775b60a0fdbd82a3225285bc6... | 2019-08-17 00:00:00 | 6150 |
| 064f488f8edabd7beffcc207f05b7ece824c1c775b60a0fdbd82a3225285bc6... | 2019-08-18 00:00:00 | 4500 |
| 065809b408897ba8c70edf8ebd41aa8a52517815c144f952f25bc20eb79c6... | 2019-08-16 00:00:00 | 14190 |
| 065809b408897ba8c70edf8ebd41aa8a52517815c144f952f25bc20eb79c6... | 2019-08-17 00:00:00 | 8460 |
| 06f8e4d1a0a2052afa4031c3858930a597e039e228eb4f9b91784e2e729c6... | 2019-08-14 00:00:00 | 50 |
| 06f8e4d1a0a2052afa4031c3858930a597e039e228eb4f9b91784e2e729c6... | 2019-08-15 00:00:00 | 80 |
| 06f8e4d1a0a2052afa4031c3858930a597e039e228eb4f9b91784e2e729c6... | 2019-08-16 00:00:00 | 590 |
| 07e1134b0004056f0077445e0b02a0745a6c57a032a407661d4b1000 | 2019-08-15 00:00:00 | 5170 |

2.5 Υποερώτημα v

2.5.1 Κώδικας

Το ερώτημα αυτό χωρίζεται σε 2 υποερωτήματα :

- (α) Ποια πλοιά τύπου cargo ήταν 'αγκυροβολημένα' (ταχύτητα μηδέν) κάποια στιγμή μέσα στην περίοδο 15/08/2019 - 18/08/2019;
- (β)Ποια για ολόκληρη την περίοδο 12/08/2019 - 19/08/2019;

(α)

```

SELECT positions.vessel_id, positions.speed
FROM positions, vesseltypes
WHERE positions.t BETWEEN '2019-08-15' AND '2019-08-19' AND positions.speed = 0 AND vesseltypes.description LIKE 'Cargo%'
GROUP BY positions.vessel_id, positions.speed
ORDER BY positions.vessel_id;

```

(β)

```
WITH cte_example AS (
    SELECT positions.vessel_id, vesseltypes.description, positions.speed
    FROM positions
    JOIN vessels ON positions.vessel_id = vessels.id
    JOIN vesseltypes ON vessels.type = vesseltypes.code
    WHERE DATE(positions.t) BETWEEN '2019-08-12' AND '2019-08-19'
        AND vesseltypes.description LIKE 'Cargo%'
    GROUP BY positions.vessel_id, vesseltypes.description, positions.speed
    HAVING COUNT(DISTINCT DATE(positions.t)) = 8
    ORDER BY positions.vessel_id
)
SELECT *
FROM cte_example
WHERE speed = 0;
```

2.5.2 Υποερώτημα ν: Ανάλυση κώδικα

(α) Στην περίπτωση αυτή καλούμαστε να βρούμε ποια πλοία τύπου cargo ήταν 'αγκυροβολημένα' (ταχύτητα μηδέν) κάποια στιγμή μέσα στην περίοδο 15/08/2019 - 18/08/2019.

Το παραπάνω κομμάτι κώδικα θα εμφανίσει τα πεδία **vessel_id** και **speed** από τον πίνακα **positions**. Με την χρήση του **WHERE** επιλέγονται:

- Οι εγγραφές με ημερομηνία (t) μεταξύ '2019-08-15' και '2019-08-18'.
- Οι εγγραφές όπου η ταχύτητα (speed) είναι μηδέν.
- Οι εγγραφές όπου η περιγραφή του τύπου πλοίου (**vesseltypes.description**) είναι τύπου 'Cargo%'.

Στη συνέχεια, τα αποτελέσματα ομαδοποιούνται ανά **ID** πλοίου και ταχύτητα με τη χρήση του **GROUP BY**, και τέλος, ταξινομούνται με βάση το **ID** του πλοίου με το **ORDER BY**.

(β) Εδώ θα πρέπει να βρούμε ποια πλοία τύπου cargo ήταν 'αγκυροβολημένα' (ταχύτητα μηδέν) για ολόκληρη την περίοδο 12/08/2019 - 19/08/2019.

Στο παραπάνω κομμάτι κώδικα, με τη χρήση του **WITH** ορίζεται ένας προσωρινός πίνακας που ονομάζεται **cte_example** στον οποίο επιλέγονται στοιχεία και από τους 3 πίνακες. Οι συνθήκες σύζευξης και τα φίλτρα περιορίζουν τα αποτελέσματα σε εγγραφές όπου η ημερομηνία (**positions.t**) είναι μεταξύ 2019-08-12 έως και 2019-08-18 και ο τύπος πλοίου (**vesseltypes.description**) είναι "Cargo". Επιπλέον, χρησιμοποιείται η συνάρτηση **HAVING** για να επιλεγούν μόνο τις εγγραφές όπου έχουν 8 μοναδικές ημερομηνίες, προσδιορίζοντας έτσι τα πλοία που έχουν στίγματα για όλες τις ημέρες που επιλέξαμε. Τέλος, χρησιμοποιείται η **cte_example** για να επιλεγούν τα πλοία που έχουν μηδενική ταχύτητα κατά τη διάρκεια όλου του εύρους ημερομηνιών. Η χρήση της **WITH** καθιστά τον κώδικα πιο δομημένο και ευανάγνωστο, και επιτρέπει τη χρήση του προσωρινού πίνακα σε πολλαπλές επερωτήσεις χωρίς την ανάγκη επανάληψης του ίδιου κώδικα.

2.5.3 Υποερώτημα v: Στιγμιότυπο από την εκτέλεση του κώδικα

(α)

```

1 SELECT positions.vessel_id, positions.speed
2 FROM positions, vesseltypes
3 WHERE positions.t BETWEEN '2019-08-15' AND '2019-08-19' AND positions.speed = 0 AND vesseltype = 'Cargo'
4 GROUP BY positions.vessel_id, positions.speed
5 ORDER BY positions.vessel_id;
    
```

| vessel_id | speed |
|------------------------------------------------------------------|-------|
| 04b6c84a518fb33b7206a114ada7660d5888de1af6cabcb7bbe2c46ea6a7d... | 0 |
| 04c3373354007ad811c13943e2593acceac75d3eecd59c2f6c52845b... | 0 |
| 05edaa038e80b4952faf8ea6ee12f2e95066896dc0b91daea37c4912468... | 0 |
| 065809b408897ba8c70edf8ebd41a8a52517815c14f952f25b20eb79c6... | 0 |
| 06fb8e4d1a0a2052afa4031c3858930597eb39e228eb4f9b91784e2e729c6... | 0 |
| 075124bee049acfcc9277844362b240a743c46c5e7e0315ef97661d4b1220... | 0 |
| 08532cbd7297e77dca56d333376e2eb13ef3475cac18a2c10d1aae19cb50... | 0 |
| 0941beda1c9de6d2d4d26a27aeb7239d3f1cc13539f083d87266a6c4c0fa7... | 0 |
| 09e2bbead4dec076d56f8ed7507ce2f2e1a7912470d449d71e48d67d41... | 0 |
| 0a5af5b90a577ac62596445a3d3d89d9704d87f5aade7fc2c03187579... | 0 |
| 0acc5ceaf205b4b21fb5172d65c0de3ce02eea539d6831d56be7dc0f975bf... | 0 |
| 05f91db8c56b2f2e5c3d877e3ccc7d5e407179a329e6938fb355d42ba... | 0 |
| 0bee2882311e4202052c59ad326b4bf4d69b82270fe7bcf4723e27cb79f6a... | 0 |
| 0cad7fdce927fb794a0619eb7c527b47755c424d1ababf9537e22d35af... | 0 |

Total rows: 241 of 241 Query complete 00:00:03.361 Successfully run. Total query runtime

(β)

```

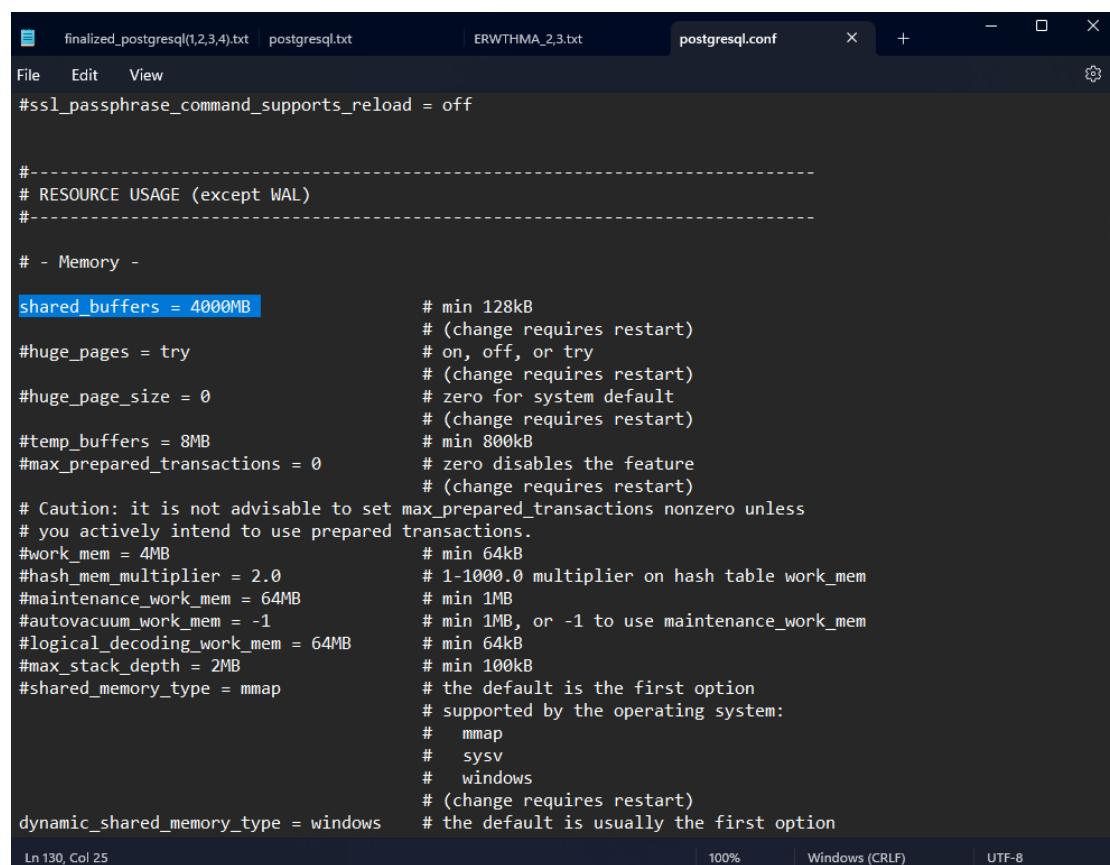
1 WITH cte_example AS (
2     SELECT positions.vessel_id, vesseltypes.description, positions.speed
3     FROM positions
4     JOIN vessels ON positions.vessel_id = vessels.id
5     JOIN vesseltypes ON vessels.type = vesseltypes.code
6     WHERE DATE(positions.t) BETWEEN '2019-08-12' AND '2019-08-19'
7     AND vesseltypes.description LIKE 'Cargo%'
8     GROUP BY positions.vessel_id, vesseltypes.description, positions.speed
9     HAVING COUNT(DISTINCT DATE(positions.t)) = 8
10    ORDER BY positions.vessel_id
11 )
12 SELECT *
13 FROM cte_example
14 WHERE speed = 0;
    
```

| vessel_id | description | speed |
|------------------------------------------------------------------|-------------------------------|-------|
| 80bc912679dfb6ff8574a0876f2ec400a43a5f4faef41d76c4345d28ca27c91a | Cargo, Hazardous categor... | 0 |
| d6b4ca21a3677d0cc9d9f8f3bcae8138af5a52feef087935eb09e93cd3... | Cargo, all ships of this type | 0 |

Total rows: 2 of 2 Query complete 00:00:04.682

3. Ερώτημα 2

Για το Ερώτημα 2 επεξεργαστήκαμε το αρχείο postgresql.conf και αλλάξαμε το shared_buffers σε 4000MB. Παρατηρούμε πως σε όλα τα υποερωτήματα ο χρόνος εκτέλεσης αυξάνεται σε διαφορετικό επίπεδο για το κάθε ένα. Αυτό οφείλεται στις διαφορές των ερωτημάτων, τον τρόπο με τον οποίο εκτελούνται τα ερωτήματα, την δομή της βάσης δεδομένων, τον όγκο των δεδομένων αλλά και το ίδιο το σύστημα μετά την αλλαγή που δέχτηκε. Συμπεραίνουμε πως τα ερωτήματά μας δεν επωφελήθηκαν από τα μεγαλύτερα shared buffers στο δικό μας σύστημα.



```
#ssl_passphrase_command_supports_reload = off

#-----
# RESOURCE USAGE (except WAL)
#-----

# - Memory -

shared_buffers = 4000MB          # min 128kB
#huge_pages = try
#huge_page_size = 0
#temp_buffers = 8MB
#max_prepared_transactions = 0
# Caution: it is not advisable to set max_prepared_transactions nonzero unless
# you actively intend to use prepared transactions.
#work_mem = 4MB
#hash_mem_multiplier = 2.0
#maintenance_work_mem = 64MB
#autovacuum_work_mem = -1
#logical_decoding_work_mem = 64MB
#max_stack_depth = 2MB
#shared_memory_type = mmap
#dynamic_shared_memory_type = windows
# the default is the first option
# supported by the operating system:
#   mmap
#   sysv
#   windows
# (change requires restart)
# the default is usually the first option
```

3.1 Υποερώτημα i

```

SELECT DATE_TRUNC('day', t) AS day, COUNT(*) AS count
FROM Positions
GROUP BY day
ORDER BY count DESC;

```

| day | count |
|---------------------|--------|
| 2019-08-05 00:00:00 | 69575 |
| 2019-08-04 00:00:00 | 543604 |
| 2019-08-28 00:00:00 | 515392 |
| 2019-08-06 00:00:00 | 490819 |
| 2019-08-11 00:00:00 | 480812 |
| 2019-08-10 00:00:00 | 406090 |
| 2019-08-03 00:00:00 | 357393 |
| 2019-08-29 00:00:00 | 356825 |
| 2019-08-18 00:00:00 | 350259 |
| 2019-08-07 00:00:00 | 341773 |
| 2019-08-27 00:00:00 | 305732 |
| 2019-08-17 00:00:00 | 303885 |
| 2019-08-12 00:00:00 | 284402 |
| 2019-08-01 00:00:00 | 253844 |
| 2019-08-02 00:00:00 | 251133 |
| 2019-08-13 00:00:00 | 248170 |
| 2019-08-09 00:00:00 | 216958 |
| 2019-08-08 00:00:00 | 207113 |
| 2019-08-18 00:00:00 | 201904 |
| 2019-08-19 00:00:00 | 126262 |
| 2019-08-30 00:00:00 | 90467 |
| 2019-08-14 00:00:00 | 72994 |
| 2019-08-11 00:00:00 | 4427 |
| 2019-08-26 00:00:00 | 3436 |

Total rows: 24 of 24 Query complete 00:00:03.056 Successfully run. Total query runtime: 3 secs 56 msec. 24 rows affected. Ln 4, col 21

3.2 Υποερώτημα ii

```

SELECT
    vesseltypes.description, COUNT(*) AS number_of_ships, vessels.flag
FROM
    vessels
JOIN
    vesseltypes ON vessels.type = vesseltypes.code AND vessels.flag = 'Greece'
GROUP BY
    vesseltypes.description, vessels.flag;

```

| description | number_of_ships | flag |
|----------------------------------------------------|-----------------|--------|
| Anti-pollution equipment | 1 | Greece |
| Cargo, all ships of this type | 14 | Greece |
| Cargo, Hazardous category D | 1 | Greece |
| Cargo, No additional information | 2 | Greece |
| Fishing | 6 | Greece |
| High speed craft (HSC), all ships of this type | 4 | Greece |
| High speed craft (HSC), Hazardous category B | 1 | Greece |
| High speed craft (HSC), No additional informati... | 3 | Greece |
| Not available (default) | 4 | Greece |
| Other Type, all ships of this type | 3 | Greece |
| Other Type, Hazardous category A | 1 | Greece |
| Other Type, no additional information | 4 | Greece |
| Passenger, all ships of this type | 40 | Greece |
| Passenger, Hazardous category A | 1 | Greece |
| Passenger, No additional information | 10 | Greece |
| Pilot Vessel | 4 | Greece |
| Pleasure Craft | 25 | Greece |
| Reserved for future use | 1 | Greece |
| Sailing | 29 | Greece |
| Search and Rescue vessel | 1 | Greece |
| Tanker, all ships of this type | 34 | Greece |
| Tanker, Hazardous category A | 10 | Greece |

Total rows: 28 of 28 Query complete 00:00:00.092 Successfully run. Total query runtime: 92 msec. 28 rows affected. Ln 8, col 40

3.3 Υποερώτημα iii

```

SELECT positions.vessel_id, vesseltypes.description, COUNT(*) AS number_of_ships_based_on_type
FROM positions, vessels
JOIN vesseltypes ON vessels.type = vesseltypes.code
WHERE positions.speed > 30 AND vessels.id = positions.vessel_id
GROUP BY positions.vessel_id, vesseltypes.description;

```

| vessel_id | description | number_of_ships_based_on_type |
|-----------|---------------------------------------------------|-------------------------------|
| 1 | High speed craft (HSC), all ships of this type | 32971 |
| 2 | Tanker, all ships of this type | 7 |
| 3 | High speed craft (HSC), No additional informat... | 10859 |
| 4 | Towing | 37 |
| 5 | High speed craft (HSC), No additional informat... | 5430 |
| 6 | High speed craft (HSC), Hazardous category B | 68564 |
| 7 | Cargo, Hazardous category A | 2 |
| 8 | Tanker, all ships of this type | 4 |
| 9 | Cargo, all ships of this type | 15 |
| 10 | High speed craft (HSC), all ships of this type | 12786 |
| 11 | Pleasure Craft | 14 |
| 12 | High speed craft (HSC), all ships of this type | 8436 |
| 13 | Passenger, all ships of this type | 16 |
| 14 | High speed craft (HSC), all ships of this type | 4599 |
| 15 | High speed craft (HSC), all ships of this type | 56212 |
| 16 | High speed craft (HSC), all ships of this type | 9782 |

Successful run. Total query runtime: 2 secs 317 msec. 16 rows affected. Ln 5, Col 55

3.4 Υποερώτημα iv

```

SELECT
    positions.vessel_id, date_trunc('day', t) AS day, COUNT(*) AS number_of_positions
FROM
    positions, vesseltypes
WHERE
    vesseltypes.description LIKE 'Passenger%' AND positions.t BETWEEN '2019-08-14' AND '2019-08-19'
    /* an bwto to orlo stis 18/08 pairnei ta stoxeia ews kai tis 17/08 */
GROUP BY
    positions.vessel_id, day
ORDER BY
    positions.vessel_id, day;

```

| vessel_id | day | number_of_positions |
|-----------|---------------------|---------------------|
| 1 | 2019-08-14 00:00:00 | 2430 |
| 2 | 2019-08-16 00:00:00 | 5640 |
| 3 | 2019-08-15 00:00:00 | 4160 |
| 4 | 2019-08-16 00:00:00 | 4300 |
| 5 | 2019-08-17 00:00:00 | 10540 |
| 6 | 2019-08-18 00:00:00 | 6290 |
| 7 | 2019-08-15 00:00:00 | 480 |
| 8 | 2019-08-16 00:00:00 | 9500 |
| 9 | 2019-08-17 00:00:00 | 140 |
| 10 | 2019-08-16 00:00:00 | 4330 |
| 11 | 2019-08-17 00:00:00 | 340 |
| 12 | 2019-08-18 00:00:00 | 50 |
| 13 | 2019-08-17 00:00:00 | 6150 |
| 14 | 2019-08-18 00:00:00 | 4500 |
| 15 | 2019-08-16 00:00:00 | 14190 |
| 16 | 2019-08-17 00:00:00 | 8460 |
| 17 | 2019-08-14 00:00:00 | 50 |
| 18 | 2019-08-15 00:00:00 | 80 |
| 19 | 2019-08-16 00:00:00 | 590 |
| 20 | 2019-08-15 00:00:00 | 5170 |

Successful run. Total query runtime: 12 secs 199 msec. 893 rows affected. Ln 11, Col 27

3.5 Υποερώτημα ν (α)

pgAdmin 4

File Object Tools Help

Object Explorer

Servers (4) PostgreSQL 13 PostgreSQL 14 PostgreSQL 16 PostgreSQL 17 Vessels Databases (1) PostgreSQL

Postgres Catalogs Event triggers Extensions Foreign Data Wrappers Languages Publications Schemas (1) public Aggregates Collations Domains FTS Configurations FTS Dictionaries FTS Parsers FTS Templates Foreign Tables Functions Materialized Views Operators Procedures Sequences Tables (3) positions vessels Columns

Dashboard Properties SQL Statistics Dependencies Dependents Processes postgres/postgres@Vessels*

Query Query History

```
1 SELECT positions.vessel_id, positions.speed
2 FROM positions, vesseltypes
3 WHERE positions.t BETWEEN '2019-08-15' AND '2019-08-19' AND positions.speed = 0 AND vesseltypes.description LIKE 'Cargo%'
4 GROUP BY positions.vessel_id, positions.speed
5 ORDER BY positions.vessel_id;
```

Data Output Messages Notifications

| vessel_id | speed |
|-----------|-------|
| 1 | 0 |
| 2 | 0 |
| 3 | 0 |
| 4 | 0 |
| 5 | 0 |
| 6 | 0 |
| 7 | 0 |
| 8 | 0 |
| 9 | 0 |
| 10 | 0 |
| 11 | 0 |
| 12 | 0 |
| 13 | 0 |
| 14 | 0 |
| 15 | 0 |
| 16 | 0 |
| 17 | 0 |
| 18 | 0 |
| 19 | 0 |
| 20 | 0 |
| 21 | 0 |
| 22 | 0 |
| 23 | 0 |
| 24 | 0 |

Total rows: 241 of 241 Query complete 00:00:05.785 Successfully run. Total query runtime: 5 secs 785 msec. 241 rows affected. Ln 5, Col 30

(β)

pgAdmin 4

File Object Tools Help

Object Explorer

Servers (4) PostgreSQL 13 PostgreSQL 14 PostgreSQL 16 PostgreSQL 17 Vessels Databases (1) PostgreSQL

Catalogs Event triggers Extensions Foreign Data Wrappers Languages Publications Schemas (1) public Aggregates Collations Domains FTS Configurations FTS Dictionaries FTS Parsers FTS Templates Foreign Tables Functions Materialized Views Operators Procedures Sequences Tables (3) positions vessels Columns

Dashboard Properties SQL Statistics Dependencies Dependents Processes postgres/postgres@Vessels*

Query Query History

```
1 WITH cte_example AS (
2     SELECT positions.vessel_id, vesseltypes.description, positions.speed
3     FROM positions
4     JOIN vessels ON positions.vessel_id = vessels.id
5     JOIN vesseltypes ON vessels.type = vesseltypes.code
6     WHERE DATE(positions.t) BETWEEN '2019-08-12' AND '2019-08-19'
7     AND vesseltypes.description LIKE 'Cargo'
8     GROUP BY positions.vessel_id, vesseltypes.description, positions.speed
9     HAVING COUNT(DISTINCT DATE(positions.t)) = 8
10    ORDER BY positions.vessel_id
11 )
12 SELECT *
13 FROM cte_example
14 WHERE speed = 0;
```

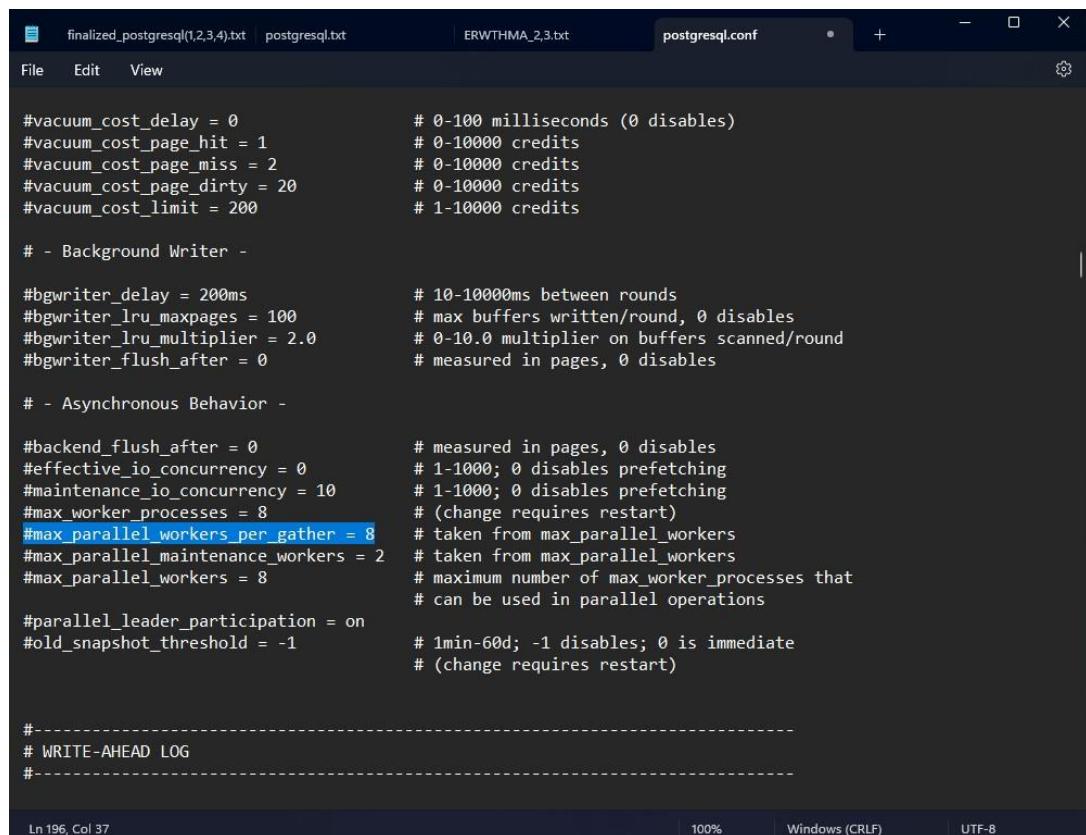
Data Output Messages Notifications

| vessel_id | description | speed |
|-----------|-------------------------------|-------|
| 1 | Cargo, Hazardous categor. | 0 |
| 2 | Cargo, all ships of this type | 0 |

Total rows: 2 of 2 Query complete 00:00:05.027 Successfully run. Total query runtime: 5 secs 27 msec. 2 rows affected. Ln 14, Col 17

4. Ερώτημα 3

Για το Ερώτημα 3 επεξεργαστήκαμε το αρχείο postgresql.conf και αλλάξαμε το max_parallel_workers_per_gather σε 8, χωρίς να επιστρέψουμε στην default τιμή τα shared_buffers. Παρατηρούμε πως σε όλα τα υποερωτήματα ο χρόνος εκτέλεσης μειώνεται σε σχέση με το Ερώτημα 2, εκτός από το υποερωτήμα 4 όπου ο χρόνος είναι πολύ κοντά με αυτόν του προηγούμενου ερωτήματος. Συμπεραίνουμε πως, συνολικά, τα ερωτήματά μας επωφελήθηκαν από το μεγαλύτερα max_parallel_workers_per_gather στο δικό μας σύστημα.



```
#vacuum_cost_delay = 0          # 0-100 milliseconds (0 disables)
#vacuum_cost_page_hit = 1       # 0-10000 credits
#vacuum_cost_page_miss = 2       # 0-10000 credits
#vacuum_cost_page_dirty = 20     # 0-10000 credits
#vacuum_cost_limit = 200        # 1-10000 credits

# - Background Writer -
#bgwriter_delay = 200ms         # 10-10000ms between rounds
#bgwriter_lru_maxpages = 100     # max buffers written/round, 0 disables
#bgwriter_lru_multiplier = 2.0   # 0-10.0 multiplier on buffers scanned/round
#bgwriter_flush_after = 0        # measured in pages, 0 disables

# - Asynchronous Behavior -
#backend_flush_after = 0         # measured in pages, 0 disables
#effective_io_concurrency = 0    # 1-1000; 0 disables prefetching
#maintenance_io_concurrency = 10  # 1-1000; 0 disables prefetching
#max_worker_processes = 8        # (change requires restart)
#max_parallel_workers_per_gather = 8
#max_parallel_maintenance_workers = 2
#max_parallel_workers = 8        # maximum number of max_worker_processes that
                                # can be used in parallel operations

#parallel_leader_participation = on
#old_snapshot_threshold = -1     # 1min-60d; -1 disables; 0 is immediate
                                # (change requires restart)

#-----
# WRITE-AHEAD LOG
#-----
```

4.1 Υποερώτημα i

The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows a tree view of databases (PostgreSQL 13, 14, 16) and a specific database named "Vessels".
- Query Editor:** Contains the following SQL query:


```
1 SELECT DATE_TRUNC('day', t) AS day, COUNT(*) AS count
2 FROM Positions
3 GROUP BY day
4 ORDER BY count DESC;
```
- Data Output:** Displays a table with 24 rows of data, showing the count of positions per day from August 5 to August 26, 2019.
- Messages:** Shows a success message: "Successfully run. Total query runtime: 2 secs 967 msec. 24 rows affected."
- Notifications:** Shows a message: "Ln 4, Col 21"

4.2 Υποερώτημα ii

The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows a tree view of databases (PostgreSQL 13, 14, 16) and a specific database named "Vessels".
- Query Editor:** Contains the following SQL query:


```
1 SELECT
2   vesseltypes.description, COUNT(*) AS number_of_ships, vessels.flag
3   FROM
4     vessels
5   JOIN
6     vesseltypes ON vessels.type = vesseltypes.code AND vessels.flag = 'Greece'
7   GROUP BY
8     vesseltypes.description, vessels.flag;
```
- Data Output:** Displays a table with 28 rows of data, showing the count of ships for each vessel type and flag, specifically for Greece.
- Messages:** Shows a success message: "Successfully run. Total query runtime: 83 msec. 28 rows affected."
- Notifications:** Shows a message: "Ln 8, Col 40"

4.3 Υποερώτημα iii

```

1 SELECT positions.vessel_id, vesselytypes.description, COUNT(*) AS number_of_ships_based_on_type
2   FROM positions, vessels
3      JOIN vesselytypes ON vessels.type = vesselypes.code
4     WHERE positions.speed > 30 AND vessels.id = positions.vessel_id
5   GROUP BY positions.vessel_id, vesselytypes.description;
    
```

| vessel_id | description | number_of_ships_based_on_type |
|---------------------------------------------------------------------|----------------------------------------------------|-------------------------------|
| 1eee559fref5de5d7c0df6698ab4d4a5e6bbbe42b2e25208864a876c6f605452... | High speed craft (HSC), all ships of this type | 32971 |
| 2619957ef2b7d7a2b2b9d1b598a7ca1w84091cda2b6000ea43237a23... | Tanker, all ships of this type | 7 |
| 3fc284d2b52c7c5f58d5b71ff6d727effcc6d515d98446fd3d1056e524ed... | High speed craft (HSC), No additional informati... | 10859 |
| 4ababcb05d344bf988de1a317ech5a62ce272bf0a17e51093a4e4bd9f6fb67b... | High speed craft (HSC), No additional informati... | 37 |
| 53886054995ed4b0ef516417ca3b699767a9b3496053ee4a65960cf... | High speed craft (HSC), No additional informati... | 5430 |
| 65719865177de04a593663c3ac3161754a17963e5cd5449959479dd... | High speed craft (HSC), Hazardous category B | 68564 |
| 7479b0547188b062a76547601dd926c356b58d7cddd3fc02c143eb09... | Cargo, Hazardous category A | 2 |
| 8e75f16242d728d03d3598aa7e45f5fe625255767b3894042d1565e5bcb... | Tanker, all ships of this type | 4 |
| 9a070294a294dc9f944bf403879732a799cc84ef4fc70003cd7befbfa1252d... | Cargo, all ships of this type | 16 |
| 96d0ce4b037773881957e99999deau6705904568944085d6d29541... | High speed craft (HSC), all ships of this type | 12786 |
| 9ea327071a3abfb9901c7f79397c7d8777f2a6d25da261400b133... | Pleasure Craft | 14 |
| b5ce45e37af967572a8767266c99a7e30482321646853a0061e01c34bc... | High speed craft (HSC), all ships of this type | 8436 |
| c7264ef7fb0cb3a695161313eb0d4fa9a7531f163a5a623b915d75612... | Passenger, all ships of this type | 16 |
| d8424a3fa4fa7a3ed80308bc00005a6b9951705a040479323262a2b... | High speed craft (HSC), all ships of this type | 4599 |
| e05386930469345993b6761f0b414905400376915e17203297a0bf... | High speed craft (HSC), all ships of this type | 56212 |
| f695447603e62ec3ef0f97d6679329a180b5675f966cf0d647d6ca355... | High speed craft (HSC), all ships of this type | 9782 |

Total rows: 16 of 16 | Query complete 00:00:2.161 | Ls 5, Col 27

Successfully run. Total query runtime: 2 secs 161 msec. 16 rows affected.

4.4 Υποερώτημα iv

```

1 SELECT
2   positions.vessel_id, date_trunc('day', t) AS day, COUNT(*) AS number_of_positions
3   FROM
4     positions, vesselytypes
5   WHERE
6     vesselytypes.description LIKE 'Passenger%' AND positions.t BETWEEN '2019-08-14' AND '2019-08-19'
7   /> an balw to orio stis 18/08 pairnei ta stoixeia eisw kai tis 17/08 */
8   GROUP BY
9     positions.vessel_id, day
10  ORDER BY
11    positions.vessel_id, day;
    
```

| vessel_id | day | number_of_positions |
|------------------------------------------------------------------|---------------------|---------------------|
| 1018d531454fb0f42ab71f1001847aa0b2358fb5329f1b40fb0f42b792761... | 2019-08-14 00:00:00 | 2430 |
| 1018d531454fb0f42ab71f1001847aa0b2358fb5329f1b40fb0f42b792761... | 2019-08-16 00:00:00 | 5640 |
| 0460c64a518f83307206a114ada7660556888de1afccabc7b0e245ea6a7d... | 2019-08-15 00:00:00 | 4160 |
| 0460c64a518f83307206a114ada7660556888de1afccabc7b0e245ea6a7d... | 2019-08-16 00:00:00 | 4300 |
| 0460c64a518f83307206a114ada7660556888de1afccabc7b0e245ea6a7d... | 2019-08-17 00:00:00 | 10540 |
| 0460c64a518f83307206a114ada7660556888de1afccabc7b0e245ea6a7d... | 2019-08-18 00:00:00 | 6290 |
| 04c33733400767d811c1942a293acceca75d3ee0de9c7fe62c9294b5... | 2019-08-15 00:00:00 | 480 |
| 04c33733400767d811c1942a293acceca75d3ee0de9c7fe62c9294b5... | 2019-08-16 00:00:00 | 9500 |
| 04c33733400767d811c1942a293acceca75d3ee0de9c7fe62c9294b5... | 2019-08-17 00:00:00 | 140 |
| 05edaa038e04952fa8eae172e950568946d0b91daee37491246... | 2019-08-16 00:00:00 | 4330 |
| 05edaa038e04952fa8eae172e950568946d0b91daee37491246... | 2019-08-17 00:00:00 | 340 |
| 05edaa038e04952fa8eae172e950568946d0b91daee37491246... | 2019-08-18 00:00:00 | 50 |
| 05edaa038e04952fa8eae172e950568946d0b91daee37491246... | 2019-08-17 00:00:00 | 6150 |
| 0644abffadab7deffcc20705a7ce4241c75909005058327295950c... | 2019-08-18 00:00:00 | 4500 |
| 0658094048897a8bc007e8d41aa8525178151449527250c0e79c... | 2019-08-16 00:00:00 | 14190 |
| 0658094048897a8bc007e8d41aa8525178151449527250c0e79c... | 2019-08-17 00:00:00 | 8460 |
| 0658094048897a8bc007e8d41aa8525178151449527250c0e79c... | 2019-08-14 00:00:00 | 50 |
| 0658094048897a8bc007e8d41aa8525178151449527250c0e79c... | 2019-08-15 00:00:00 | 80 |
| 0658094048897a8bc007e8d41aa8525178151449527250c0e79c... | 2019-08-16 00:00:00 | 990 |
| 073124be049afcf9278443625240a43c465e7e015ed97661d4b122... | 2019-08-15 00:00:00 | 5170 |

Total rows: 893 of 893 | Query complete 00:00:12.57 | Ls 11, Col 27

Successfully run. Total query runtime: 12 secs 157 msec. 893 rows affected.

4.5 Υποερώτημα ν

(α)

```

1 SELECT positions.vessel_id, positions.speed
2 FROM positions, vesseltypes
3 WHERE positions.t BETWEEN '2019-08-15' AND '2019-08-19' AND positions.speed = 0 AND vesseltypes.description LIKE 'Cargo%'
4 GROUP BY positions.vessel_id, positions.speed
5 ORDER BY positions.vessel_id;
    
```

| vessel_id | speed |
|-----------------------------------------------------------------|-------|
| 046c624a518f8330720a114ada7660d5688de1af5cabc7bbec2c4e65a7d... | 0 |
| 043377354007a7d811c139432593acceca75d3eecd59c6f5c284a5b... | 0 |
| 050d9a03860d04952ca8ae6172695068960c09119aae37c4912468... | 0 |
| 055809408897268c70df8e041aa8a55217815144a59723020e796... | 0 |
| 05884e1d1a029252fa0313858930a97e59228e4f991784e267296... | 0 |
| 07512abe049afcf927844320240a743246c5e70315ef97614691220... | 0 |
| 085322bd72977dc565d333373e2eb17ef7475ac8b2c101aaee9b50... | 0 |
| 0941beda1c9ded224d2ba27a77e7239d1cc13539083d872566a5c40f7... | 0 |
| 09720e0a40ec07d5d58f07407e22e1a7912470d4c6f774e4805704... | 0 |
| 0a5af5950a577ae62196444a3a33d0997974d875faade37c2c0317579... | 0 |
| 0a5cc5ace20384021b617266500e3ee2ea539d8831d56be7cd09750d... | 0 |
| 0b5f91dc5a607265c3707783ccc505a4b717a32794939b035542b... | 0 |
| 0be282311e4202052c5-a9a325b4d4469b82270fb7fc1723e27579f6... | 0 |
| 0caad7fdcece9278794a019eb7c52f74775c424d1a8abb9537e22d35a... | 0 |
| 0d9ca38ae464ca0bfcafa05681772cf9f1c047a70b46d13c50e6ff941c... | 0 |
| 0f509e842x1454dcba14-7652bc23a70472c442959b681aa727582... | 0 |
| 0f71697f000411fe9e2e52529f09f6/c71262681fce65ac26205c5d5959f... | 0 |
| 0f78f41ee223274608aad00c76aa4f7171561400018384744c2035660... | 0 |
| 0f615beed5c7f70f613d5b4347647d7f09c72334ec55470709f02e299... | 0 |
| 10799964ab44a542480780165765a2c4f733e89495fb3e82f2c660058... | 0 |
| 211659aa598063011b01173860c939229310985331146909360754... | 0 |
| 22244185a3e5b602aa03f979e6ce6017499c40d7954997cb8730587... | 0 |
| 2313065464046bd2930569f12230d6eff14c5a2f44c1499aa104098a8... | 0 |
| 2413666a5826534ddc46e6248f502111760d44e2a59788742ffccfcba64e... | 0 |

Total rows: 241 Query complete 00:00:05.137 Successfully run. Total query runtime: 5 secs 137 msec. 241 rows affected. Ln 5, Col 50

(β)

```

1 WITH cte_vessel AS (
2     SELECT positions.vessel_id, vesseltypes.description, positions.speed
3     FROM positions
4     JOIN vessels ON positions.vessel_id = vessels.id
5     JOIN vesseltype ON vessels.type = vesseltypes.code
6     WHERE DATE(positions.t) BETWEEN '2019-08-12' AND '2019-08-19'
7     AND vesseltypes.description LIKE 'Cargo%'
8     GROUP BY positions.vessel_id, vesseltypes.description, positions.speed
9     HAVING COUNT(DISTINCT DATE(positions.t)) = 8
10    ORDER BY positions.vessel_id
11)
12 SELECT *
13 FROM cte_vessel
14 WHERE speed = 0;
    
```

| vessel_id | description | speed |
|---------------------------------------------------------------|-------------------------------|-------|
| 80b912679df6ff857a0876f2ec400a43a54faef41d76c345d28ca2791a... | Cargo, Hazardous categor... | 0 |
| d5b4ca21a56700c9e9b03ca8138a54c52e6ff7935e69e93c03... | Cargo, all ships of this type | 0 |

Total rows: 2 of 2 Query complete 00:00:04.321 Successfully run. Total query runtime: 4 secs 321 msec. 2 rows affected. Ln 14, Col 17

5. Ερώτημα 4

Σύμφωνα με τα εργαστήρια των διαλέξεων, έχουμε 2 είδη ευρετηρίων που μπορούμε να χρησιμοποιήσουμε:

- Ευρετήρια B+ Δέντρων: Συνήθως χρησιμοποιούνται για ερωτήματα που περιλαμβάνουν συγκρίσεις (<, >, <=, >=, BETWEEN) και ταξινόμηση βάσει των τιμών τους. Είναι κατάλληλα για ερωτήματα που αφορούν συγκεκριμένες τιμές στήλης ή εύρος τιμών. Χρησιμοποιούνται και για τη σύζευξη (JOIN) και είναι πιο αποδοτικά σε σχέση με τα ευρετήρια κατακερματισμού σε ορισμένες περιπτώσεις.
- Ευρετήρια Κατακερματισμού: Χρησιμοποιούνται κυρίως για απλές αναζητήσεις βάσει του κλειδιού, δηλαδή όταν υπάρχει μια συγκεκριμένη τιμή για την οποία θέλουμε να εντοπίσουμε την αντίστοιχη εγγραφή. Είναι αποδοτικά για αναζητήσεις με συγκεκριμένα κλειδιά αναζήτησης(ερώτηση ταυτότητας), αλλά δεν υποστηρίζουν ερωτήματα που αφορούν εύρος τιμών(ερώτηση διαστήματος).

5.1 Υποερώτημα i

5.1.1 Κώδικας

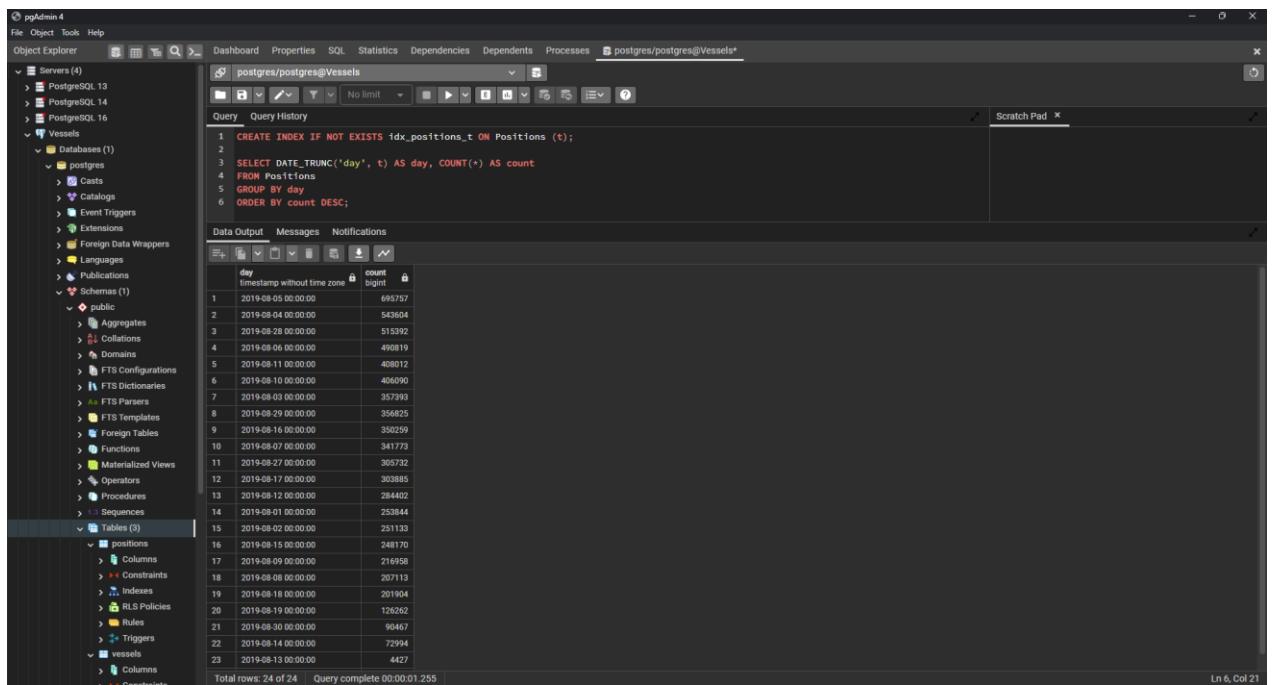
```
CREATE INDEX IF NOT EXISTS idx_positions_t ON Positions(t);

SELECT DATE_TRUNC('day', t) AS day, COUNT(*) AS count
FROM Positions
GROUP BY day
ORDER BY count DESC;
```

5.1.2 Υποερώτημα i: Ανάλυση κώδικα

Στο συγκεκριμένο ερώτημα χρησιμοποιούμε το ευρετήριο B+ δέντρο, επειδή επιτρέπουν στη βάση δεδομένων να αναζητά και να οργανώνει τα δεδομένα σε σχέση με την ημερομηνία πιο αποτελεσματικά. Κατά την εκτέλεση του ερωτήματος, το ευρετήριο B+ δέντρου περιορίζει τον αριθμό των εγγραφών που πρέπει να φιλτραριστούν, πετυχαίνοντας έτσι την αναζήτηση και την ομαδοποίηση των δεδομένων ανά ημερομηνία.

5.1.3 Υποερώτημα i: Στιγμιότυπο από την εκτέλεση του κώδικα



The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer tree shows a connection to 'postgres/postgres@Vessels' and lists databases, schemas, and tables under the 'Vessels' schema. In the center, a query window displays the following SQL code:

```
1 CREATE INDEX IF NOT EXISTS idx_positions_t ON Positions (t);
2
3 SELECT DATE_TRUNC('day', t) AS day, COUNT(*) AS count
4   FROM Positions
5  GROUP BY day
6 ORDER BY count DESC;
```

Below the code, the Data Output tab shows a table with 24 rows of data, representing the count of positions per day from August 5 to August 14, 2019. The table has columns 'day' (timestamp) and 'count' (bigint). The data is as follows:

| day | count |
|---------------------|--------|
| 2019-08-05 00:00:00 | 695757 |
| 2019-08-04 00:00:00 | 543604 |
| 2019-08-28 00:00:00 | 515392 |
| 2019-08-06 00:00:00 | 490819 |
| 2019-08-11 00:00:00 | 408012 |
| 2019-08-10 00:00:00 | 406090 |
| 2019-08-03 00:00:00 | 357393 |
| 2019-08-29 00:00:00 | 356825 |
| 2019-08-16 00:00:00 | 350259 |
| 2019-08-07 00:00:00 | 341773 |
| 2019-08-27 00:00:00 | 305732 |
| 2019-08-17 00:00:00 | 303885 |
| 2019-08-12 00:00:00 | 284402 |
| 2019-08-01 00:00:00 | 253844 |
| 2019-08-02 00:00:00 | 251133 |
| 2019-08-15 00:00:00 | 248170 |
| 2019-08-09 00:00:00 | 216958 |
| 2019-08-08 00:00:00 | 207113 |
| 2019-08-18 00:00:00 | 201904 |
| 2019-08-19 00:00:00 | 126262 |
| 2019-08-30 00:00:00 | 90467 |
| 2019-08-14 00:00:00 | 72994 |
| 2019-08-13 00:00:00 | 4427 |

Total rows: 24 of 24 Query complete 00:00:01.255 Ln 6, Col 21

5.2 Υποερώτημα ii

5.2.1 Κώδικας

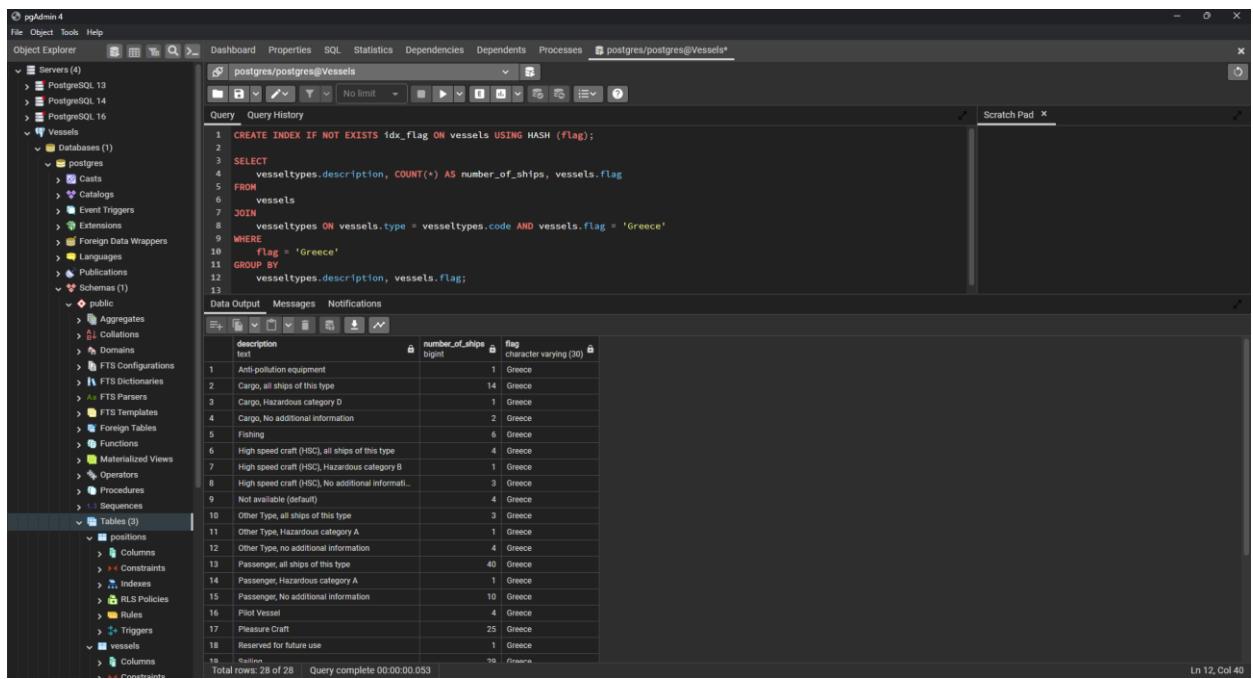
```
CREATE INDEX IF NOT EXISTS idx_flag ON vessels USING HASH (flag);

SELECT
    vesseltypes.description, COUNT(*) AS number_of_ships, vessels.flag
FROM
    vessels
JOIN
    vesseltypes ON vessels.type = vesseltypes.code
WHERE
    vessels.flag = 'Greece'
GROUP BY
    vesseltypes.description, vessels.flag;
```

5.2.2 Υποερώτημα ii: Ανάλυση κώδικα

Χρησιμοποιούμε ευρετήριο κατακερματισμού διότι οι αναζητήσεις με συγκεκριμένες τιμές είναι συχνά πιο αποτελεσματικές με ευρετήρια κατακερματισμού. Στη συγκεκριμένη περίπτωση, το ευρετήριο βοηθάει στη βελτίωση του χρόνου εκτέλεσης καθώς οδηγεί κατευθείαν τον υπολογιστή στα σωστά αποτελέσματα και επιτρέπει να βρει τα πλοία που έχουν ελληνική σημαία, χωρίς να χρειαστεί να ψάξει κάθε πλοίο ένα-ένα.

5.2.3 Υποερώτημα ii: Στιγμιότυπο από την εκτέλεση του κώδικα



```
CREATE INDEX IF NOT EXISTS idx_flag ON vessels USING HASH (flag);
SELECT vesseltypes.description, COUNT(*) AS number_of_ships, vessels.flag
FROM vessels
JOIN vesseltypes ON vessels.type = vesseltypes.code AND vessels.flag = 'Greece'
WHERE flag = 'Greece'
GROUP BY vesseltypes.description, vessels.flag;
```

| description | number_of_ships | flag |
|---------------------------------------------------|-----------------|--------|
| Anti-pollution equipment | 1 | Greece |
| Cargo, all ships of this type | 14 | Greece |
| Cargo, Hazardous category D | 1 | Greece |
| Cargo, No additional information | 2 | Greece |
| Fishing | 6 | Greece |
| High speed craft (HSC), all ships of this type | 4 | Greece |
| High speed craft (HSC), Hazardous category B | 1 | Greece |
| High speed craft (HSC), No additional information | 3 | Greece |
| Not available (default) | 4 | Greece |
| Other Type, all ships of this type | 3 | Greece |
| Other Type, Hazardous category A | 1 | Greece |
| Other Type, no additional information | 4 | Greece |
| Passenger, all ships of this type | 40 | Greece |
| Passenger, Hazardous category A | 1 | Greece |
| Passenger, No additional information | 10 | Greece |
| Pilot Vessel | 4 | Greece |
| Pleasure Craft | 25 | Greece |
| Reserved for future use | 1 | Greece |
| Railinn | 90 | Greece |

5.3 Υποερώτημα iii

5.3.1 Κώδικας

```
CREATE INDEX IF NOT EXISTS idx_positions_speed ON positions(speed);
CREATE INDEX IF NOT EXISTS idx_vessels_id ON vessels(type);

SELECT positions.vessel_id, vesseltypes.description, COUNT(*) AS number_of_ships_based_on_type
FROM positions, vessels
JOIN vesseltypes ON vessels.type = vesseltypes.code
WHERE positions.speed > 30 AND vessels.id = positions.vessel_id
GROUP BY positions.vessel_id, vesseltypes.description;
```

5.3.2 Υποερώτημα iii: Ανάλυση κώδικα

Στο ερώτημα αυτό, χρησιμοποιήσαμε 2 ευρετήρια, ένα για τη στήλη `speed` του πίνακα `positions` και ένα για τη στήλη `type` του πίνακα `vessels`. Το 1ο ευρετήριο είναι τύπου B+ δέντρου, καθώς βοηθάει στη γρήγορη ανάκτηση των θέσεων όπου η αναζήτηση γίνεται σε συνεχόμενα εύρη τιμών, όπως η ταχύτητα που υπερβαίνει τους 30 κόμβους.

Το 2ο ευρετήριο είναι επίσης τύπου B+ δέντρου, καθώς η δυνατότητα αναζήτησης σε συνεχόμενα εύρη (όπως η αναζήτηση των πλοίων με συγκεκριμένο τύπο) είναι αποτελεσματική με τη χρήση δέντρων B+.

5.3.3 Υποερώτημα iii: Στιγμιότυπο από την εκτέλεση του κώδικα

The screenshot shows the pgAdmin 4 interface with the following details:

- Servers:** PostgreSQL 13, PostgreSQL 14, PostgreSQL 16.
- Databases:** postgres, public.
- Tables:** Vessels, positions, vesseltypes.
- Query:**

```

CREATE INDEX IF NOT EXISTS idx_speed ON positions (speed);
CREATE INDEX IF NOT EXISTS idx_type ON vessels (type);

SELECT
    positions.vessel_id,
    vesseltypes.description,
    COUNT(*) AS number_of_ships_based_on_type
FROM
    positions
JOIN
    vessels ON positions.vessel_id = vessels.id
JOIN
    vesseltypes ON vessels.type = vesseltypes.code
WHERE
    positions.speed > 30
GROUP BY
    positions.vessel_id, vesseltypes.description;
  
```
- Data Output:** A table showing the results of the query. The table has columns: vessel_id, description, and number_of_ships_based_on_type.
- Messages:** Successfully run. Total query runtime: 852 msec. 16 rows affected.

| vessel_id | description | number_of_ships_based_on_type |
|------------------------------------------------------------------|---------------------------------------------------|-------------------------------|
| eb538c8d8b49a93459c9bcb61fa1f6414fb0e4bd237091a5172c3277a8bf | High speed craft (HSC), all ships of this type | 56212 |
| 74790b3647bb02aa76547601dd928c356656e07cc0d3fcfb2c1c43e8d9 | Cargo, Hazardous category A | 2 |
| 8e75cf162d20780cf3998aa14f9cfce25257b763980a26115d5edbc | Tanker, all ships of this type | 4 |
| 53880c5496ea44abe0fc1d4173ca46a7f679a8e4b5c3be4a6586e3cf3 | High speed craft (HSC), No additional informat... | 5430 |
| d84243a5af4a7e3aeb0b300bcbca00c5ad99511706a04a7932262a2b | High speed craft (HSC), all ships of this type | 4599 |
| 9d5bc574d377738891597e999299d9aa5a7fd59c046d8406556bfed295d1 | High speed craft (HSC), all ships of this type | 12786 |
| c7264ef7dbccb3a6695a16313e0bda49aa753d1fe3a5a623be915a7d5612 | Passenger, all ships of this type | 16 |
| b6ce0e3e7ff6767b2ca8f7e2dd99a9ec0402231646633e8061ed1c3a6c... | High speed craft (HSC), all ships of this type | 8436 |
| 26199557ff2787a220f0d81b99aa7ca19a40916c2b560de432772c53 | Tanker, all ships of this type | 7 |
| 4baabcc2d0348c48c9888de1a1317ac5a2c272b0a17e5510fa3d44bdcff9fb7b | Towing | 37 |
| f69544763eef23e3e9f07667932wf8a188056757996cfd6d6476e5355... | High speed craft (HSC), all ships of this type | 9782 |
| 3fc2814d2b3c7cb7f5b5d871f6d7274e4fc6655d584846dd3d1066e524ed | High speed craft (HSC), No additional informat... | 10859 |
| 9ea327071a3a8bf8980c3c1ff3997fc1d6777f702a8253da2331400b133 | Pleasure Craft | 14 |
| 9a07029a6294dcf984b483879732a79c864ef7c70003c87bef8a1252d | Cargo, all ships of this type | 15 |
| 57139e851770ea04c536da324ca31a613734c13793e3c5d844995e4795d... | High speed craft (HSC), Hazardous category B | 68564 |

Total rows: 16 of 16 Query complete 00:00:00.852 Lr 3, Col 1

5.4 Υποερώτημα iv

5.4.1 Κώδικας

```

CREATE INDEX IF NOT EXISTS idx_positions_day ON positions (DATE_TRUNC('day', t));

SELECT
    positions.vessel_id, DATE_TRUNC('day', t) AS day, COUNT(*) AS number_of_positions
FROM
    positions, vesseltypes
WHERE
    vesseltypes.description LIKE 'Passenger%' AND positions.t BETWEEN '2019-08-14' AND '2019-08-19'
GROUP BY
    positions.vessel_id, day
ORDER BY
    positions.vessel_id, day;
  
```

5.4.2 Υποερώτημα iv: Ανάλυση κώδικα

Σε αυτό το ερώτημα χρησιμοποιούμε ευρετήριο B+ δέντρου διότι βοηθά στην επιτάχυνση των αναζητήσεων βάσει ημερομηνιών, χωρίς να γίνει πλήρης σάρωση του πίνακα (όπως και στη συνάρτηση `DATE_TRUNC('day' , t)` που χρησιμοποιείται στο ευρετήριο αγνοεί τις ώρες, τα λεπτά και τα δευτερόλεπτα, επιτρέποντας έτσι την αναζήτηση σε συγκεκριμένες ημέρες) και κάνοντας τον κώδικα πιο αποδοτικό.

5.4.3 Υποερώτημα iv: Στιγμιότυπο από την εκτέλεση του κώδικα

```

CREATE INDEX IF NOT EXISTS idx_positions_day ON positions (DATE_TRUNC('day', t));
SELECT
    positions.vessel_id, DATE_TRUNC('day', t) AS day, COUNT(*) AS number_of_positions
FROM
    positions, vesseltypes
WHERE
    vesseltypes.description LIKE 'Passenger%' AND positions.t BETWEEN '2019-08-14' AND '2019-08-19'
GROUP BY
    positions.vessel_id, day
ORDER BY
    positions.vessel_id, day;
  
```

| vessel_id | day | number_of_positions |
|-----------|---------------------|---------------------|
| 1 | 2019-08-14 00:00:00 | 2430 |
| 2 | 2019-08-14 00:00:00 | 5640 |
| 3 | 2019-08-15 00:00:00 | 4160 |
| 4 | 2019-08-16 00:00:00 | 4300 |
| 5 | 2019-08-17 00:00:00 | 10540 |
| 6 | 2019-08-18 00:00:00 | 6290 |
| 7 | 2019-08-15 00:00:00 | 480 |
| 8 | 2019-08-16 00:00:00 | 9500 |
| 9 | 2019-08-17 00:00:00 | 140 |
| 10 | 2019-08-18 00:00:00 | 4330 |
| 11 | 2019-08-17 00:00:00 | 340 |
| 12 | 2019-08-18 00:00:00 | 50 |
| 13 | 2019-08-17 00:00:00 | 6150 |
| 14 | 2019-08-18 00:00:00 | 4500 |
| 15 | 2019-08-16 00:00:00 | 14190 |
| 16 | 2019-08-17 00:00:00 | 8460 |
| 17 | 2019-08-14 00:00:00 | 50 |
| 18 | 2019-08-15 00:00:00 | 80 |
| 19 | 2019-08-16 00:00:00 | 460 |

Total rows: 893 of 893 Query complete: 00:05.216 Ln 12, Col 30.

5.5 Υποερώτημα v

5.5.1 Κώδικας

(α)

```

CREATE INDEX IF NOT EXISTS idx_time_speed ON positions (t, speed);
CREATE INDEX IF NOT EXISTS idx_type ON vessels (type);

SELECT positions.vessel_id, positions.speed
FROM positions, vesseltypes
WHERE positions.t BETWEEN '2019-08-15' AND '2019-08-19' AND positions.speed = 0 AND vesseltypes.description LIKE 'Cargo%'
GROUP BY positions.vessel_id, positions.speed
ORDER BY positions.vessel_id;
  
```

(β)

```

CREATE INDEX IF NOT EXISTS idx_positions_t ON positions(t);
CREATE INDEX IF NOT EXISTS idx_vessel_type ON vessels(type);

WITH cte_example AS (
    SELECT positions.vessel_id, vesseltypes.description, positions.speed
    FROM positions
    JOIN vessels ON positions.vessel_id = vessels.id
    JOIN vesseltypes ON vessels.type = vesseltypes.code
    WHERE DATE(positions.t) BETWEEN '2019-08-12' AND '2019-08-19'
        AND vesseltypes.description LIKE 'Cargo%'
    GROUP BY positions.vessel_id, vesseltypes.description, positions.speed
    HAVING COUNT(DISTINCT DATE(positions.t)) = 8
    ORDER BY positions.vessel_id
)
SELECT *
FROM cte_example
WHERE speed = 0;
  
```

5.5.2 Υποερώτημα ν: Ανάλυση κώδικα

(α)

Χρησιμοποιούμε 2 ευρετήρια τύπου B+ δέντρου. Στο 1ο ευρετήριο δημιουργούμε για τις στήλες **t** (ημερομηνία) και **speed** του πίνακα **positions**. Η χρήση αυτού του ευρετηρίου βοηθάει στη βελτιστοποίηση των αναζητήσεων που βασίζονται ταυτόχρονα στο χρόνο και στην ταχύτητα και των αναζητήσεων που βασίζονται στον τύπο του πλοίου.

Στο 2ο ευρετήριο, δημιουργούμε για τη στήλη **type** του πίνακα **vessels**. Η χρήση του βοηθάει στη βελτιστοποίηση των αναζητήσεων που βασίζονται στον τύπο του πλοίου. Συγκεκριμένα, η συνθήκη **vesseltypes.description** LIKE '**Cargo%**' χρησιμοποιείται για την ανάκτηση πλοίων τύπου "Cargo", όπου με την χρήση του ευρετηρίου η απόδοση βελτιώνεται.

(β)

Χρησιμοποιούμε και εδώ 2 ευρετήρια τύπου B+ δέντρου. Το 1ο ευρετήριο το δημιουργούμε στη στήλη **t** (ημερομηνία) του πίνακα **positions**. Η χρήση του ευρετηρίου στη συγκεκριμένη στήλη βοηθάει στη βελτιστοποίηση των αναζητήσεων που βασίζονται σε ημερομηνίες. Η συγκεκριμένη αναζήτηση είναι στο **WHERE** όπου επιλέγονται οι εγγραφές του πίνακα **positions** με βάση την ημερομηνία που βρίσκεται μεταξύ '2019-08-15' έως και '2019-08-18'.

Το 2ο ευρετήριο το χρησιμοποιούμε στη στήλη **type** του πίνακα **vessels**. Χρησιμοποιείται για τη βελτιστοποίηση των αναζητήσεων που βασίζονται στον τύπο του πλοίου. Συγκεκριμένα, στο **WHERE**, επιλέγονται οι εγγραφές του πίνακα **vessels** όπου ο τύπος του πλοίου είναι "Cargo".

5.5.3 Υποερώτημα ν: Στιγμιότυπο από την εκτέλεση του κώδικα

(α)

```

CREATE INDEX IF NOT EXISTS idx_time_speed ON positions (t, speed);
CREATE INDEX IF NOT EXISTS idx_type ON vessels (type);

SELECT positions.vessel_id, positions.speed
FROM positions, vesseltypes
WHERE positions.t BETWEEN '2019-08-15' AND '2019-08-19' AND positions.speed = 0 AND vesseltypes.description LIKE 'Cargo%'

ORDER BY positions.vessel_id;
  
```

| vessel_id | speed |
|--------------------------------------------------------------------|-------|
| 0486c84a51ff813b1720a114da76405688d8e1af1ca8cb7b9e2b46e6a7d... | 0 |
| 04c373354007a88119349a2593acacecc793ce059265c5c84a5b... | 0 |
| 05eda9f038e049527ffefee127e95066896a0c0c91fddea71c4912468... | 0 |
| 06589f040389f70a7c70edbeda41a8dab02517815c144929bc20679c6... | 0 |
| 0686a4161a20252a94c931385939a597e99c28647991784a26729c6... | 0 |
| 071512aee4949ffcf9278436752d07a434c6c7e70315ef79816d01220... | 0 |
| 08332cd7297e77a5c6d333762e13e3475ac18b2c10daee49b50... | 0 |
| 0941beda1946d25442d7a677e672396f1c1339f083d8726646c4df97... | 0 |
| 0962bead4ec07e56fbfc7507c2d2e17a71247044c9d7eef8e67641... | 0 |
| 098afaf9b90577a7c2596445453c3d389f9704875a5a837fc2c03187579... | 0 |
| 10accacca2054b021f517265d5d0e02ee339f8831d56876d9795fb6... | 0 |
| 120591dc85692e52c7d87763cc7d5e4071179a3f29e938b3554d2ba... | 0 |
| 130be0232311e4203052c9e9a1264046682270fe7ec472e70b9f6a... | 0 |
| 140cad1fdec9278794a00b7ebc5770a775c4d1a8d937e27d55af... | 0 |
| 150dc2a88e46c4e0a7cb7c95e581722feff1cb4f7aef0e613c50effe41c8... | 0 |
| 1609f66d42a1454d05c1fe8379e23e74472c42e959b6d81a1d725f82... | 0 |
| 1707169f770a0431feef902e529f05967e1262588bec6e5ac26205c5b5659f6... | 0 |
| 180fc7801ee232246508aa0d37a8477471d54840601847c46c2b0566d00... | 0 |
| 190f51beed5c872f053d38d426470ffaf99927233642ec554703799992a299... | 0 |
| 201079954a9e94ad5426878016575a2c4ff733a9495fb7e2c2566d05... | 0 |
| 2111699aa598860367110317386c903222329d0a93316499f93a8d754... | 0 |
| 2212244856e3e50b03a30309e9c5b183890d40873994997a87305677... | 0 |

Total rows: 241 of 241 Query complete 00:00:02.294

(β)

The screenshot shows the pgAdmin 4 interface. In the Object Explorer on the left, under the 'Vessels' node, there is a 'Tables (3)' section which contains 'positions' and 'vessels'. The 'positions' table has columns: vessel_id (character varying(64)), description (text), and speed (double precision). The 'vessels' table has columns: vessel_id (character varying(64)), description (text), and speed (double precision). A query is run in the central 'Query' tab:

```
1 CREATE INDEX IF NOT EXISTS idx_positions_t ON positions(t);
2 CREATE INDEX IF NOT EXISTS idx_vessels_type ON vessels(type);
3
4 WITH cte_example AS (
5     SELECT positions.vessel_id, vesseltypes.description, positions.speed
6        FROM positions
7       JOIN vessels ON positions.vessel_id = vessels.id
8       JOIN vesseltypes ON vessels.type = vesseltypes.code
9      WHERE DATE(positions.t) BETWEEN '2019-08-12' AND '2019-08-19'
10     AND vesseltypes.description LIKE 'Cargo'
11   GROUP BY positions.vessel_id, vesseltypes.description, positions.speed
12   HAVING COUNT(DISTINCT DATE(positions.t)) = 8
13
14 )
15 SELECT *
16 FROM cte_example
17 WHERE speed = 0;
```

The results are displayed in the 'Data Output' tab:

| vessel_id | description | speed |
|-----------------------------------------------------------------|-------------------------------|-------|
| 80dc91269d50ff574b87f62ec400sa3a54faff41d76c4345d28ca27c91a | Cargo, Hazardous categor... | 0 |
| d5b4ca21a3677d0cc9d9fbf3bcad8138af5a54c52eefdb7935eb09e93cd3a.. | Cargo, all ships of this type | 0 |

Total rows: 2 of 2 Query complete 00:00:01.586 Successfully run. Total query runtime: 1 secs 586 msec. 2 rows affected. Ln 17, Col 17

6. Ερώτημα 5

Σύμφωνα με τις εργαστηριακές διαλέξεις, διδαχθήκαμε 2 βασικούς τύπους partitioning πινάκων:

- Διαμέριση μέσω κληρονομικότητας (Inheritance partitioning): η λογική που χρησιμοποιείται είναι ότι υπάρχει ένας κύριος πίνακας που ονομάζεται "γονικός πίνακας" (parent table), ενώ οι υπο-πίνακες που δημιουργούνται με το partitioning ονομάζονται "παιδιά" (child tables). Κάθε παιδί πίνακας «κληρονομεί» κάποια δεδομένα από τον γονέα.
- Δηλωτική Διαμέριση (Declarative Partitioning): χρησιμοποιεί δηλωτικές συντακτικές εντολές για τον καθορισμό των κανόνων κατακερματισμού, χωρίς να υπάρχει η ανάγκη για χρήση της κληρονομικότητας. Είναι πιο αφηρημένο σε σύγκριση με την κληρονομικότητα.

Μετά από ανάλυση, καταλήξαμε ότι στα υποερωτήματα, είναι προτιμότερο να χρησιμοποιηθεί Inheritance partitioning, διότι παρέχει μεγαλύτερη ευελιξία στο σχεδιασμό της δομής των υπο-πινάκων και μπορούμε να έχουμε διαφορετικές στήλες σε κάθε υπο-πίνακα, καθώς ο καθένας από αυτούς κληρονομεί τη δομή του γονικού πίνακα και μπορούμε να προσθέσουμε επιπλέον στήλες ή να τις αλλάξουμε. Επιπλέον, οποιαδήποτε αλλαγή γίνει στη δομή του γονικού πίνακα αλλάζει αυτόματα σε όλους τους πίνακες παιδιά.

6.1 Υποερώτημα i

Στο 1ο υποερώτημα δεν είναι απαραίτητο να γίνει **partitioning**. Δεν υπάρχει κάποιος περιορισμός ή κριτήριο (**constraint**) το οποίο μας βοηθά να χωρίσουμε τον πίνακα **positions**. Έτσι, θεωρήσαμε πως η διαίρεση του πίνακα σε μικρότερα τμήματα δεν προσφέρει κάποιο πλεονέκτημα, όπως ταχύτητα ή αποδοτικότητα, στο συγκεκριμένο ερώτημα.

6.2 Υποερώτημα ii

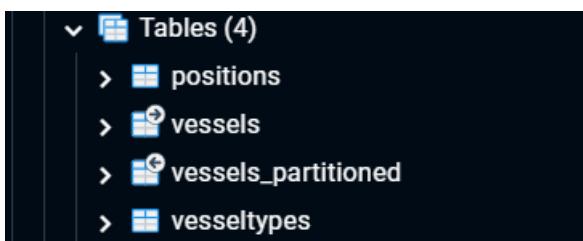
6.2.1 Κώδικας

```
CREATE TABLE vessels_partitioned(
    CHECK (flag = 'Greece%')
) INHERITS (vessels);

CREATE INDEX idx_vessels_partitioned_description ON vessels_partitioned(flag);

INSERT INTO vessels_partitioned (id, type, flag)
SELECT id, type, flag
FROM vessels
WHERE flag = 'Greece';
```

6.2.2 Υποερώτημα ii: Ανάλυση Κώδικα



Δημιουργείται ένας νέος πίνακα με το όνομα **vessels_partitioned**.

Χρησιμοποιούμε την εντολή **CHECK (flag = 'Greece%')** που δηλώνει έναν όρο ελέγχου που τσεκάρει ότι κάθε

εγγραφή στον **vessels_partitioned** πρέπει να έχει την στήλη **flag** με τιμή που ξεκινά με τη συμβολοσειρά '**Greece**'. Αυτός ο όρος θα χρησιμοποιηθεί για τον έλεγχο των εγγραφών που κατανέμονται στους υπο-πίνακες όταν χρησιμοποιείται κληρονομικότητα. Μετά χρησιμοποιούμε την εντολή **INHERITS (vessels)**: για να δηλώσουμε ότι ο πίνακας **vessels_partitioned** κληρονομεί από τον πίνακα **vessels**. Αυτό σημαίνει ότι ο πίνακας **vessels_partitioned** θα έχει όλες τις στήλες του **vessels** και θα κληρονομεί τις ιδιότητες του γονικού πίνακα. Δημιουργούμε ένα ευρετήριο με το όνομα **idx_vessels_partitioned_description**.

Τέλος, κάνουμε μεταφορά των δεδομένων με την εντολή **INSERT** από τον πίνακα **vessels** στον **vessels_partitioned** με ορό ελέγχου να μεταφερθούν τα δεδομένα μόνο με το **flag** Greece.

6.2.3 Υποερώτημα ii: Στιγμιότυπο από την εκτέλεση του κώδικα

```

INSERT INTO vessels_partitioned (id, type, flag)
SELECT id, type, flag
FROM vessels
WHERE flag = 'Greece';

```

Total rows: 16 of 16 | Query complete 00:00:00.074 | Ln 1, Col 1

Εδώ βλέπουμε ότι η μεταφορά δεδομένων έχει γίνει με επιτυχία και ο partitioned πίνακας είναι γεμάτος με δεδομένα και το **flag** των πλοίων αρχίζει με Greece.

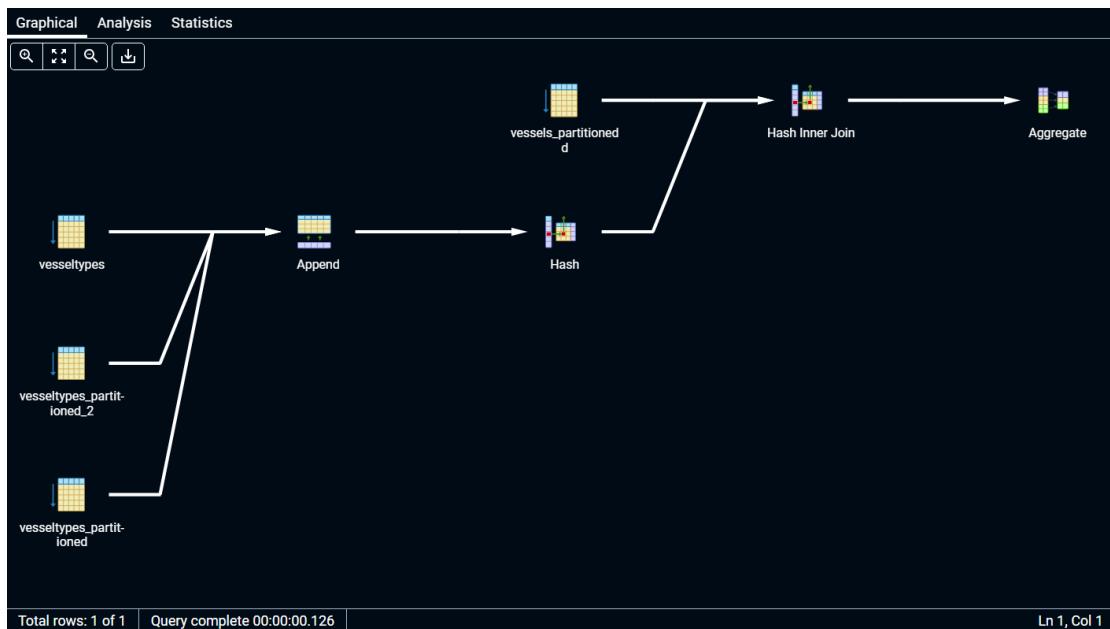
| description | number_of_ships | flag |
|--------------------------------------|-----------------|--------|
| Cargo, No additional information | 2 | Greece |
| Passenger, No additional information | 10 | Greece |
| Towing | 4 | Greece |
| Pilot Vessel | 4 | Greece |
| Tanker, all ships of this type | 34 | Greece |
| Cargo, all ships of this type | 14 | Greece |
| Tanker, Hazardous category A | 10 | Greece |
| Reserved for future use | 1 | Greece |
| Sailing | 29 | Greece |

Total rows: 28 of 28 | Query complete 00:00:00.144 | Ln 11, Col 1

Εδώ αντικαταστήσαμε τον πίνακα **vessels** με τον partitioned πίνακα **vessels_partitioned**.

6.2.4 Υποερώτημα ii: Διάγραμμα

Αν τρέξουμε την εντολή EXPLAIN ANALYZE έχουμε το εξής διάγραμμα:



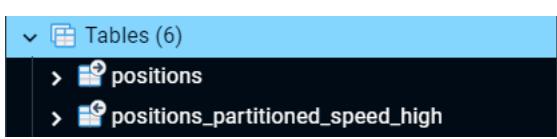
6.3 Υποερώτημα iii

6.3.1 Κώδικας

```
CREATE TABLE positions_partitioned_speed_high(
    CHECK (speed > 30)
) INHERITS (positions);
```

```
INSERT INTO positions_partitioned_speed_high (id, vessel_id, t, lon, lat, heading, course ,speed)
SELECT id, vessel_id, t, lon, lat, heading, course ,speed
FROM positions
WHERE speed > 30;
```

6.3.2 Υποερώτημα iii: Ανάλυση Κώδικα



Σε αυτό το ερώτημα, δημιουργούμε έναν partitioned πίνακα που κληρονομεί τις στήλες από τον πίνακα **positions**. Εκχωρούμε τα δεδομένα από τον πίνακα **positions** μέσω της εντολής **INSERT** στον πίνακα **positions_partitioned_speed_high** με την συνθήκη (όρος ελέγχου) ότι θα μπουν τα δεδομένα με το speed (ταχύτητα) να είναι μεγαλύτερη των 30 κόμβων.

6.3.3 Υποερώτημα iii: Στιγμιότυπο από την εκτέλεση του κώδικα

```

SELECT * FROM positions_partitioned_speed_high

```

| | t | lon | lat | heading | course | speed |
|---|---------------------|----------|----------|---------|--------|-------|
| 1 | 2019-08-04 07:39:06 | 23.62329 | 37.83368 | 179 | 181.8 | 30.7 |
| 2 | 2019-08-04 07:39:08 | 23.60928 | 37.93473 | 179 | 62.6 | 31.6 |
| 3 | 2019-08-04 07:39:09 | 23.62238 | 37.83339 | 179 | 181.6 | 30.7 |
| 4 | 2019-08-04 07:39:10 | 23.60962 | 37.93485 | 179 | 63.6 | 31.6 |
| 5 | 2019-08-04 07:39:13 | 23.62327 | 37.83283 | 178 | 181 | 30.6 |
| 6 | 2019-08-04 07:39:14 | 23.61028 | 37.9351 | 179 | 65 | 31.7 |
| 7 | 2019-08-04 07:39:14 | 23.62326 | 37.83255 | 178 | 180.5 | 30.5 |
| 8 | 2019-08-04 07:39:17 | 23.62326 | 37.83226 | 178 | 180.2 | 30.5 |

Total rows: 1000 of 209741 | Query complete 00:00:00.804 | Ln 1, Col 47

Εδώ βλέπουμε ότι η μεταφορά δεδομένων έχει γίνει με επιτυχία και ο partitioned πίνακας είναι γεμάτος με δεδομένα.

```

SELECT positions_partitioned_speed_high.vessel_id, vesseltypes.description, COUNT(*) AS number_of_ships_based_on_type
FROM positions_partitioned_speed_high, vessels
JOIN vesseltypes ON vessels.type = vesseltypes.code
WHERE positions_partitioned_speed_high.speed > 30 AND vessels.id = positions_partitioned_speed_high.vessel_id
GROUP BY positions_partitioned_speed_high.vessel_id, vesseltypes.description;

```

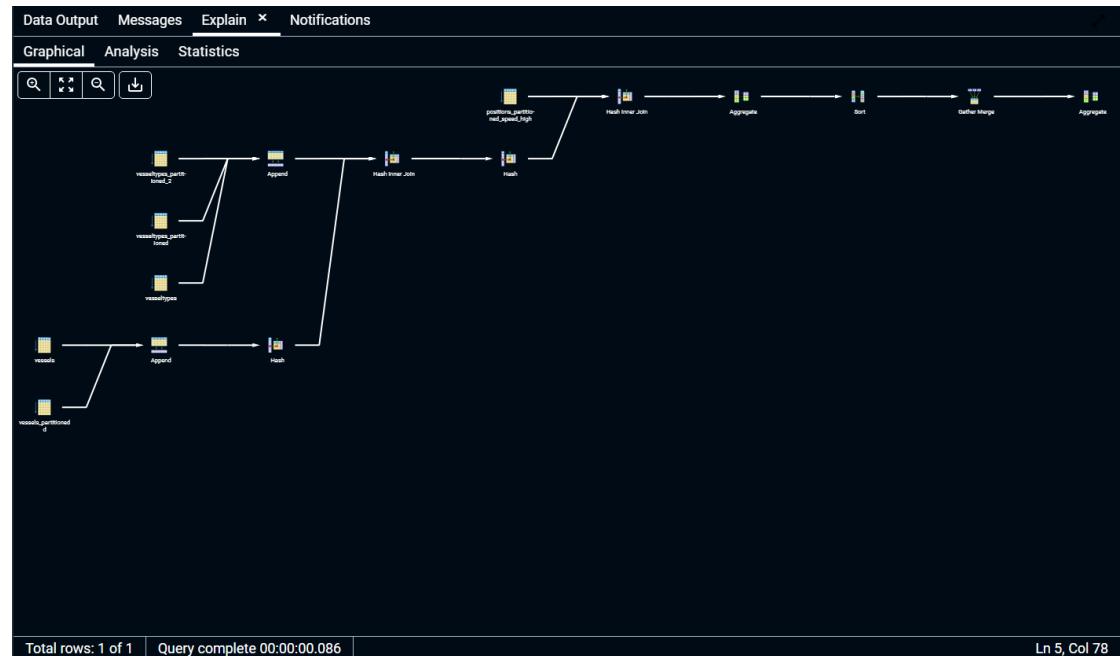
| vessel_id | description | number_of_ships_based_on_type |
|-----------|---------------------------------------------------|-------------------------------|
| 1 | High speed craft (HSC), all ships of this type | 32971 |
| 2 | Tanker, all ships of this type | 7 |
| 3 | High speed craft (HSC), No additional information | 10859 |
| 4 | Towing | 37 |
| 5 | High speed craft (HSC), No additional information | 5430 |
| 6 | High speed craft (HSC), Hazardous category B | 68564 |
| 7 | Cargo, Hazardous category A | 2 |
| 8 | Tanker all ships of this type | 1 |

Total rows: 16 of 16 | Query complete 00:00:00.154 | ✓ Successfully run. Total query runtime: 154 msec. 16 rows affected. | Ln 4, Col 83

Εδώ αντικαταστήσαμε τον πίνακα positions με τον partitioned πίνακα positions_partitioned_speed_high.

6.3.4 Υποερώτημα iii: Διάγραμμα

Αν τρέξουμε την εντολή EXPLAIN ANALYZE έχουμε το εξής διάγραμμα:



6.4 Υποερώτημα iv

6.4.1 Κώδικας

```
CREATE TABLE Vesseltypes_partitioned (
    CHECK (description LIKE 'Passenger%')
) INHERITS (vesseltypes);

CREATE TABLE Positions_partitioned (
    CHECK (t > '2019-08-14' AND t < '2019-08-19')
) INHERITS (positions);

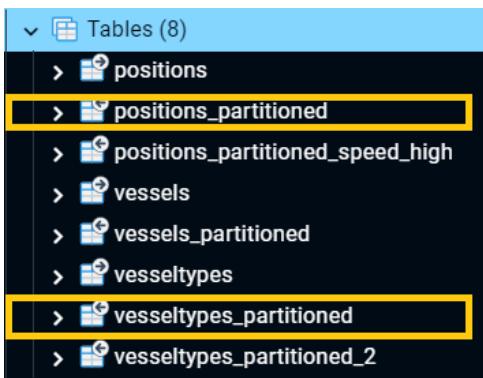
CREATE INDEX idx_vesseltypes_partitioned_description ON Vesseltypes_partitioned(description);
CREATE INDEX idx_positions_partitioned_t ON Positions_partitioned(t);

INSERT INTO vesseltypes_partitioned (code, description)
SELECT code, description
FROM vesseltypes
WHERE description LIKE 'Passenger%';

INSERT INTO positions_partitioned (id, vessel_id, t, lon, lat, heading, course, speed)
SELECT id, vessel_id, t, lon, lat, heading, course, speed
FROM positions
WHERE t > '2019-08-14' AND t < '2019-08-19';
```

6.4.2 Υποερώτημα iv: Ανάλυση Κώδικα

Εδώ δημιουργούνται 2 πίνακες, με την διαφορά ότι οι πίνακες κληρονομούν δεδομένα από διαφορετικούς γονείς, δηλαδή κάνουμε partition inheritance στους αρχικούς πίνακες **vesseltypes** και **positions**.



Για τον πίνακα **vesseltypes** δημιουργείται ο πίνακας **Vesseltypes_partitioned** που κληρονομεί από τον πίνακα **vesseltypes**. Έχει ένα CHECK constraint που ελέγχει αν το πεδίο **description** ξεκινά με τη φράση "Passenger%", δηλαδή ο τύπος του πλοίου είναι passenger. Αυτό θα χρησιμοποιηθεί για να κατανέμει τα δεδομένα σε υπο-πίνακες με βάσει της τιμής αυτού του πεδίου.

Για τον πίνακα **positions** δημιουργείται ο πίνακας **positions_partitioned** που κληρονομεί από τον πίνακα **positions**. Έχει ένα CHECK constraint που ελέγχει αν το πεδίο **t** (χρονική σήμανση) βρίσκεται μεταξύ των ημερομηνιών '2019-08-14' έως '2019-08-18'. Αυτό θα χρησιμοποιηθεί για να κατανέμει τα δεδομένα σε υπο-πίνακες βάσει της χρονικής σήμανσης.

Τέλος, δημιουργούμε 2 ευρετήρια στους πίνακες **vesseltypes_partitioned** και **positions_partitioned** για την βελτιστοποίηση των αναζητήσεων που βασίζονται στα πεδία **description** και **t** αντίστοιχα.

6.4.3 Υποερώτημα iv: Στιγμιότυπο από την εκτέλεση του κώδικα

```
1 INSERT INTO vesseltypes_partitioned (code, description)
2   SELECT code, description
3     FROM vesseltypes
4    WHERE description LIKE 'Passenger%';
5
6 INSERT INTO positions_partitioned (id, vessel_id, t, lon, lat, heading, course, speed)
7   SELECT id, vessel_id, t, lon, lat, heading, course, speed
8     FROM positions
9    WHERE t > '2019-08-14' AND t < '2019-08-19';
10
11
12
13
14
```

Εδώ βλέπουμε ότι η μεταφορά δεδομένων έχει γίνει με επιτυχία και στους 2 πίνακες.

```

SELECT
    positions_partitioned.vessel_id, DATE_TRUNC('day', t) AS day, COUNT(*) AS number_of_positions
FROM
    positions_partitioned, vesseltypes_partitioned
WHERE
    vesseltypes_partitioned.description LIKE 'Passenger%' AND positions_partitioned.t BETWEEN '2019-08-14' AND '2019-08-19'
GROUP BY
    positions_partitioned.vessel_id, day
ORDER BY
    positions_partitioned.vessel_id, day

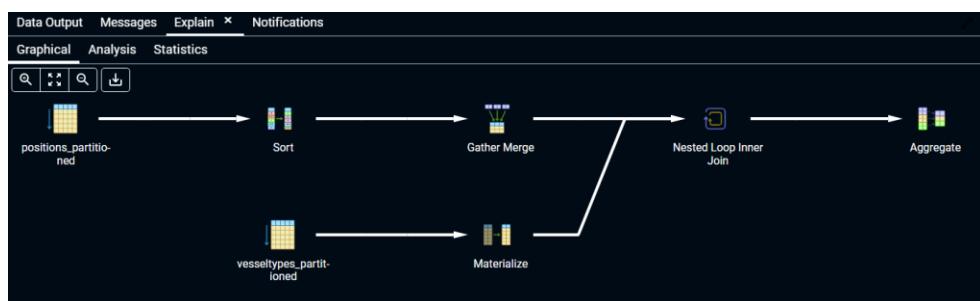
```

Total rows: 892 of 892 | Query complete 00:00:06.698 | Ln 10, Col 47

Εδώ αντικαταστήσαμε τους πίνακες positions και vesseltypes με τους partitioned πίνακες positions_partitioned και vesseltypes_partitioned.

6.4.4 Υποερώτημα iv: Διάγραμμα

Αν τρέξουμε την εντολή EXPLAIN ANALYZE έχουμε το εξής διάγραμμα:



6.5 Υποερώτημα v

6.5.1 Κύδικας

```

CREATE TABLE Positions_partitioned_2(
    CHECK (t > '2019-08-15' AND t < '2019-08-19'),
    CHECK (speed = 0)
) INHERITS (positions);

CREATE TABLE Vesseltypes_partitioned_2(
    CHECK (description LIKE 'Cargo%')
) INHERITS (vesseltypes);

CREATE INDEX idx_positions_partitioned_t_speed2 ON Positions_partitioned_2(t, speed);
CREATE INDEX idx_vesseltypes_partitioned_description2 ON Vesseltypes_partitioned_2(description);

```

```

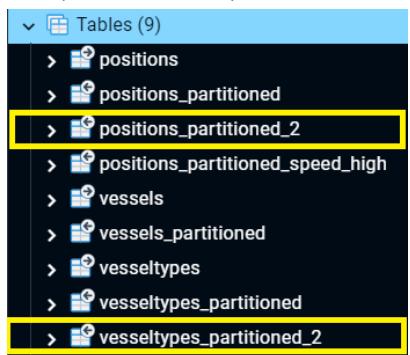
INSERT INTO Positions_partitioned_2 (id, vessel_id, t, lon, lat, heading, course, speed)
SELECT id, vessel_id, t, lon, lat, heading, course, speed
FROM positions
WHERE t > '2019-08-15' AND t < '2019-08-19' AND speed = 0;

INSERT INTO Vesseltypes_partitioned_2 (code, description)
SELECT code, description
FROM vesseltypes
WHERE description LIKE 'Cargo%';

```

6.5.2 Υποερώτημα ν: Ανάλυση Κώδικα

Όπως και στο προηγούμενο υποερώτημα, έτσι και εδώ δημιουργούνται 2 πίνακες με την διαφορά ότι οι πίνακες κληρονομούν δεδομένα από διαφορετικούς γονείς, δηλαδή κάνουμε partition inheritance στους αρχικούς πίνακες **vesseltypes** και **positions**.



Πρώτα, δημιουργείται ένας υπο-πίνακας ο **positions_partitioned_2** που κληρονομεί από τον πίνακα **positions**. Περιέχει 2 ελέγχους, ο ένας έλεγχος τσεκάρει ότι η τιμή του πεδίου **t** (χρονική σήμανση) βρίσκεται ανάμεσα στις ημερομηνίες '2019-08-15' έως και '2019-08-18'. Αυτό χρησιμοποιείται για να κατανείμει τα δεδομένα στον υπο-πίνακα βάσει της χρονικής σήμανσης. Ο άλλος έλεγχος τσεκάρει ότι η τιμή του πεδίου **speed** είναι 0. Αυτός ο έλεγχος χρησιμοποιείται για να κατανείμει τα δεδομένα στον υπο-πίνακα βάσει της ταχύτητας του πλοίου.

Ο άλλος υπο-πίνακας που δημιουργείται είναι ο **vesseltypes_partitioned_2** που περιέχει τον έλεγχο για να τσεκάρει ότι η τιμή του πεδίου **description** ξεκινά με τη φράση "Cargo%". Αυτό χρησιμοποιείται για να κατανείμει τα δεδομένα στον υπο-πίνακα βάσει της περιγραφής του τύπου του πλοίου.

Τέλος, δημιουργούνται 2 ευρετηρια στους υποπίνακες **positions_partitioned_2** και **vesseltypes_partitioned_2** για την βελτιστοποίηση των αναζητήσεων που βασίζονται στις στήλες **t**, **speed** και **description** αντίστοιχα.

6.5.3 Υποερώτημα v: Στιγμιότυπο από την εκτέλεση του κώδικα

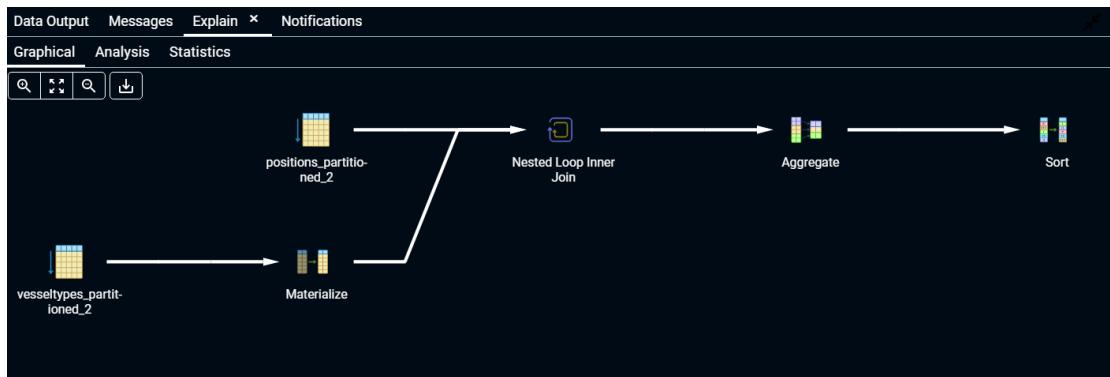
```

1 INSERT INTO Positions_partitioned_2 (id, vessel_id, t, lon, lat, heading, course, speed)
2 SELECT id, vessel_id, t, lon, lat, heading, course, speed
3 FROM positions
4 WHERE t > '2019-08-15' AND t < '2019-08-19' AND speed = 0;
5
6 INSERT INTO Vesseltypes_partitioned_2 (code, description)
7 SELECT code, description
8 FROM vesseltypes
9 WHERE description LIKE '%Cargo%';
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
289
290
291
292
293
294
295
296
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
989
989
990
991
992
993
994
995
996
997
997
998
999
999
1000
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1197
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1297
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1397
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1497
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1597
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1697
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1797
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1897
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2178
2179
2179
2180
2181
2182
2183
2184
2185
2186
```

6.5.4 Υποερώτημα ν: Διάγραμμα

Αν τρέξουμε την εντολή EXPLAIN ANALYZE έχουμε το εξής διάγραμμα:

(α)



(β)

