



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ**  
**UNIVERSITY OF PIRAEUS**

**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**  
**DEPARTMENT OF INFORMATICS**

**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/ZEM86Fe6i4FeJcJ>.

- 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 αλλάζετε 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, εξηγήστε 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

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

## 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 used for analysis:

```
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:** Shows the results of the query in a table format:

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
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 00:00:00.084

## 2.3 Υποερώτημα iii

### 2.3.1 Κώδικας

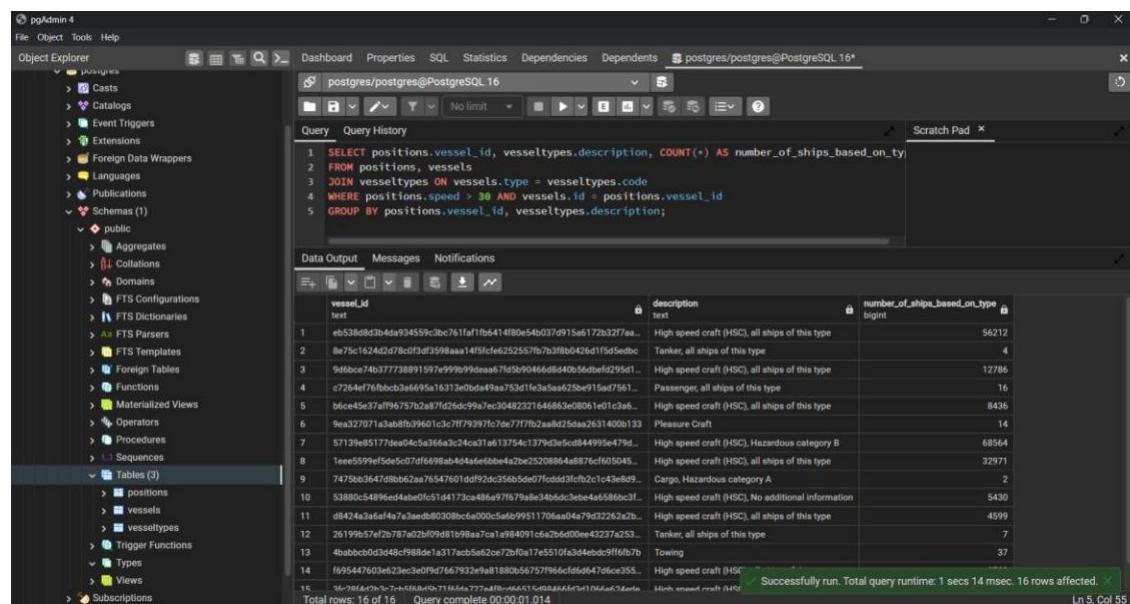
```
SELECT positions.vessel_id, vesseltypes.description, COUNT(*) AS number_of_ships_based_on_type
FROM positions, vessels
JOIN vesseltypes ON vessels.type = vesseltypes.code
WHERE positions.speed > 30 AND vessels.id = positions.vessel_id
GROUP BY positions.vessel_id, vesseltypes.description;
```

### 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:

- Object Explorer:** Shows the database structure with Schemas (1), Tables (3), and other objects like Casts, Catalogs, Event Triggers, Extensions, Foreign Data Wrappers, Languages, Publications, and Subscriptions.
- Query Editor:** Displays the SQL query used to generate the results.
- Data Output:** Shows the results of the query in a tabular format.
- Messages:** Shows the message "Successfully run. Total query runtime: 1 secs 14 msec. 16 rows affected."
- Notifications:** Shows the message "Ln 5, Col 55"

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

## 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
04c3373354007a7d811c13943e2593acceca75d3eeecd59c2f6fe5c284a5b...	2019-08-15 00:00:00	480
04c3373354007a7d811c13943e2593acceca75d3eeecd59c2f6fe5c284a5b...	2019-08-16 00:00:00	9500
04c3373354007a7d811c13943e2593acceca75d3eeecd59c2f6fe5c284a5b...	2019-08-17 00:00:00	140
05edaaf038e0b4952faf8ea6ee12f2e95066896dc0bc91fdaee37c4912468...	2019-08-16 00:00:00	4330
05edaaf038e0b4952faf8ea6ee12f2e95066896dc0bc91fdaee37c4912468...	2019-08-17 00:00:00	340
05edaaf038e0b4952faf8ea6ee12f2e95066896dc0bc91fdaee37c4912468...	2019-08-18 00:00:00	50
064f488f8edabd7beffcc207f05b7ec824c1c775b60a0fdbd82a3225285bc6c	2019-08-17 00:00:00	6150
064f488f8edabd7beffcc207f05b7ec824c1c775b60a0fdbd82a3225285bc6c	2019-08-18 00:00:00	4500
065809b408897ba8c70edf8ebd41aa8a52517815c144f952f25bc20eb79c6...	2019-08-16 00:00:00	14190
065809b408897ba8c70edf8ebd41aa8a52517815c144f952f25bc20eb79c6...	2019-08-17 00:00:00	8460
06f8e4d1a0a2052fa4031c3858930a597e039e228e4f9b91784e2e729c6...	2019-08-14 00:00:00	50
06f8e4d1a0a2052fa4031c3858930a597e039e228e4f9b91784e2e729c6...	2019-08-15 00:00:00	80
06f8e4d1a0a2052fa4031c3858930a597e039e228e4f9b91784e2e729c6...	2019-08-16 00:00:00	590

## 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
4 GROUP BY positions.vessel_id, positions.speed
5 ORDER BY positions.vessel_id;

```

Data Output Messages Notifications

vessel_id	speed
text	double precision
04b6c84e518f833b7206a114ada7660d5688de1af6abc7bbe2c46ea6a7d...	0
04c3373354007ad811c13943e2593accac75d3eecd59c2f65e284a5b...	0
05edas038e80b4952faf8e6ee12f2e95066896dc0bc91fdaea37c4912468...	0
065809b408897ba8c70df8ebd41aa85251781c144f952f25b20eb79c6...	0
06f8e4d1a0a2052afa4031c3858930597eb39e228baf9691784e2e729c6...	0
075124bee049acf9277844362b240a743c46c5e7e0315ef97661d4b1220...	0
08532cbd7297e77dca5c6d333376e2eb13e3475cac18b2c10d1aae19cb50...	0
0941beda1c9de6d2d4d26a27aeb7239d3f1cc13539f083d87266a6c4c0fa7...	0
09e2bead4dec076d56f86d7507ce2f2e1a7912470d44c9d71e4bd67d41...	0
0a5af5b90a5777ac625964445a3d3d89d9f704d87f5ade37fc2c03187579...	0
0acc5aceaf205b4b21fb5172fd65c0de3ce02eaa539d6831d56be7dc0f975bf...	0
0b5f91db5c56b72e5c37d877e83cc7d5e407179a3f9e6938fb355d42ba...	0
0bee2882311e4202052c5c9ad326b4bf4d69b82270fe7bcf4723e27cb79f6a...	0
0cad7dce927fb94a0619eb7c52fb47755e424d1ababf9537e22d35af...	0

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

(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;

```

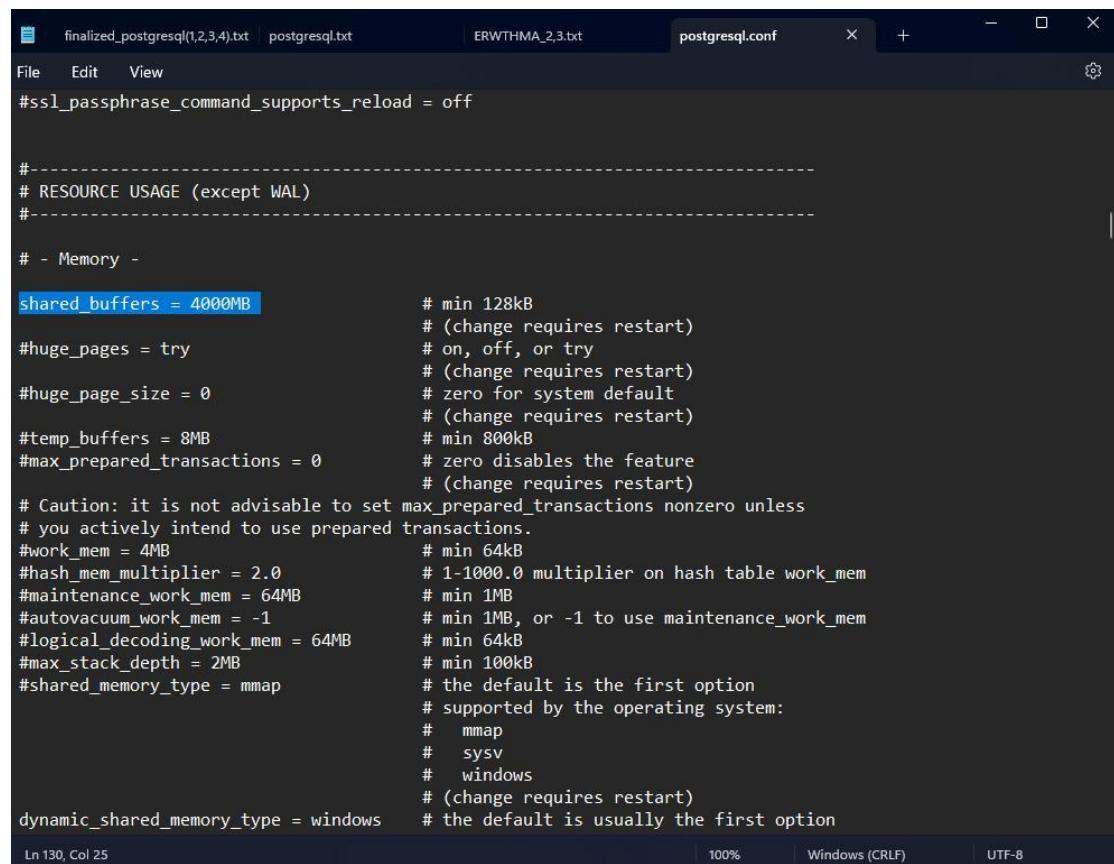
Data Output Messages Notifications

vessel_id	description	speed
text	double precision	
80bc912679dfb6ff8574a0876f2ec400a43a5f4fe141d76c4345d28ca27c91a	Cargo, Hazardous categor...	0
d664ca21a3677d0cc9d9f8f3bcae8138af5a54c52effd8f7935eb09e93cd3a...	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 ερωτήματων, 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

Ln 130, Col 25           100%           Windows (CRLF)           UTF-8
```

### 3.1 Υποερώτημα i

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

- Object Explorer:** Shows a tree view of the database structure under "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 the results of the query as a table:
 

day	count
2019-08-01 00:00:00	695757
2019-08-04 00:00:00	543654
2019-08-28 00:00:00	515392
2019-08-06 00:00:00	490819
2019-08-11 00:00:00	488012
2019-08-03 00:00:00	406690
2019-08-09 00:00:00	357939
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	30572
2019-08-01 00:00:00	303885
2019-08-12 00:00:00	284402
2019-08-08 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
2019-08-06 00:00:00	207113
2019-08-19 00:00:00	201904
2019-08-10 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
2019-08-28 00:00:00	3436
- Message Bar:** Shows "Successfully run. Total query runtime: 3 secs 56 msec. 24 rows affected." and "Ln 4, Col 21".

### 3.2 Υποερώτημα ii

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

- Object Explorer:** Shows a tree view of the database structure under "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 the results of the query as a table:
 

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
- Message Bar:** Shows "Successfully run. Total query runtime: 92 msec. 28 rows affected." and "Ln 8, Col 40".

### 3.3 Υποερώτημα iii

```

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

vessel_id	description	number_of_ships_based_on_type
1	High speed craft (HSC), all ships of this type	32171
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	Pasenger, 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

Total rows: 16 of 16    Query complete 00:00:02.317    Successfully 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, vesselytypes
WHERE
    vesselytypes.description LIKE '%Passenger%' AND positions.t BETWEEN '2019-08-14' AND '2019-08-16'
    /* an bawlo to orlo stis 18/08 palmer net ta stoixela ews kai tis 17/08 */
GROUP BY
    positions.vessel_id, day
ORDER BY
    positions.vessel_id, day;
  
```

day	number_of_positions
2019-08-14 00:00:00	2430
2019-08-16 00:00:00	5640
2019-08-15 00:00:00	4160
2019-08-16 00:00:00	4300
2019-08-17 00:00:00	10540
2019-08-18 00:00:00	6290
2019-08-15 00:00:00	480
2019-08-16 00:00:00	9500
2019-08-17 00:00:00	140
2019-08-16 00:00:00	4330
2019-08-17 00:00:00	340
2019-08-18 00:00:00	50
2019-08-17 00:00:00	6150
2019-08-18 00:00:00	4500
2019-08-16 00:00:00	14190
2019-08-17 00:00:00	8460
2019-08-14 00:00:00	50
2019-08-15 00:00:00	80
2019-08-16 00:00:00	590
2019-08-15 00:00:00	5170

Total rows: 893 of 893    Query complete 00:00:12.159    Successfully run. Total query runtime: 12 secs 159 msec. 893 rows affected.    Ln 11, Col 27

### 3.5 Υποερώτημα ν (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;

```

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

- Servers:** PostgreSQL 13, PostgreSQL 14, PostgreSQL 16.
- Object Explorer:** Vessels > Tables > positions.
- Query Editor:** Contains the SQL query above.
- Data Output:** Shows 241 rows of results, each with vessel\_id and speed.
- Status Bar:** "Successfully run. Total query runtime: 5 secs 785 msec. 241 rows affected." and "Ln 5, Col 30".

(β)

```

WITH cte_example AS (
  SELECT positions.vessel_id, vesseltypes.description, positions.speed
  FROM positions
  JOIN vessels ON positions.vessel_id = vessels.id
  JOIN vessels 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)) = 0
  ORDER BY positions.vessel_id
)
SELECT *
FROM cte_example
WHERE speed = 0;

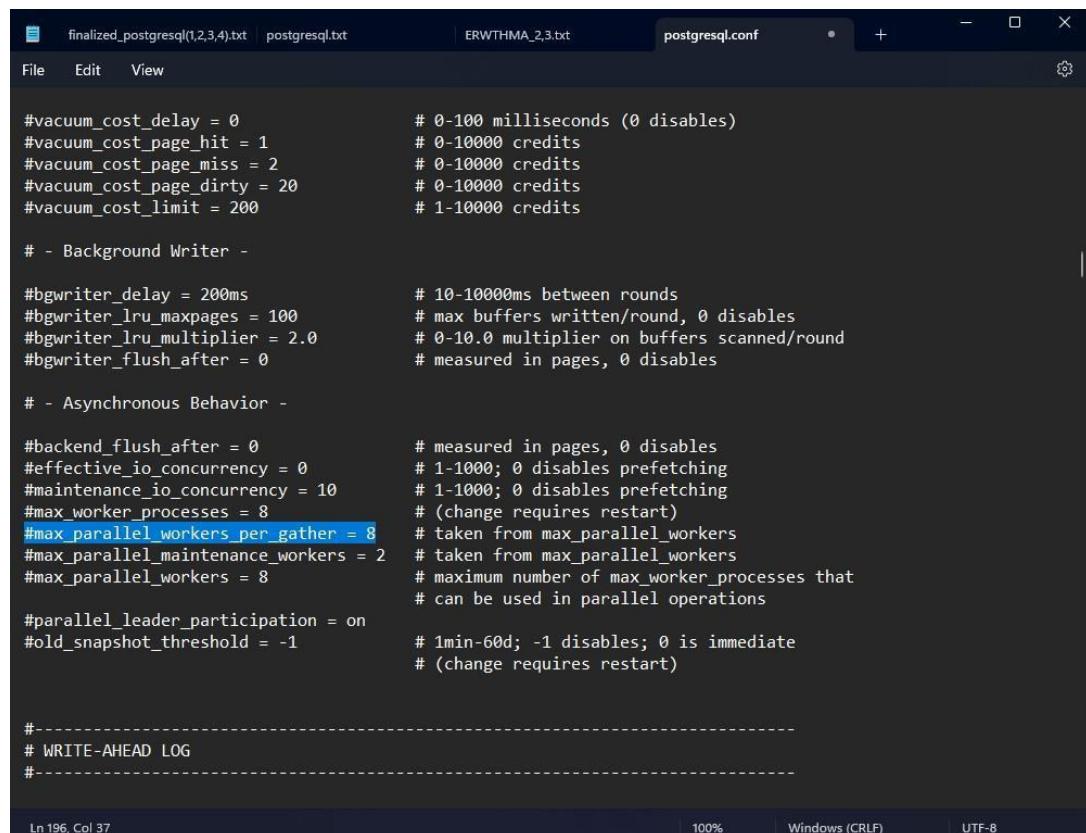
```

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

- Servers:** PostgreSQL 13, PostgreSQL 14, PostgreSQL 16.
- Object Explorer:** Vessels > Tables > positions.
- Query Editor:** Contains the complex SQL query above.
- Data Output:** Shows 2 rows of results, each with vessel\_id, description, and speed.
- Status Bar:** "Successfully run. Total query runtime: 5 secs 27 msec. 2 rows affected." and "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 # taken from max_parallel_workers
#max_parallel_maintenance_workers = 2 # taken from max_parallel_workers
#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. The left sidebar displays the Object Explorer with a tree view of servers, databases, and tables. The main area shows a SQL query in the Query tab:

```

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

The Data Output tab displays the results 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	515992
2019-08-06 00:00:00	499819
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	356823
2019-08-16 00:00:00	350299
2019-08-07 00:00:00	341773
2019-08-27 00:00:00	305782
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	125262
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

At the bottom right, a message indicates: "Successfully run. Total query runtime: 2 secs 967 msec. 24 rows affected. Ln 4, Col 21".

## 4.2 Υποερώτημα ii

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the Object Explorer with a tree view of servers, databases, and tables. The main area shows a SQL query in the Query tab:

```

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;
    
```

The Data Output tab displays the results of the query:

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
Sailing	29	Greece
Search and Rescue vessel	1	Greece
Tanker, all ships of this type	34	Greece
Tanker, Hazardous category A	10	Greece

At the bottom right, a message indicates: "Successfully run. Total query runtime: 63 msec. 28 rows affected. Ln 8, Col 40".

## 4.3 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;
  
```

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

- Object Explorer:** Shows a tree view of the database structure, including Servers, Databases, Vessels, and Tables.
- Query Editor:** Contains the SQL query above.
- Data Output:** Displays the results of the query, which lists various vessel types and their counts.
- Messages:** Shows a success message: "Successfully run. Total query runtime: 2 secs 161 msec. 16 rows affected."
- Scratch Pad:** An empty tab.

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 informati...	10859
4	High speed craft (HSC), Hazardous category A	37
5	High speed craft (HSC), No additional informati...	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

## 4.4 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 balw to orlo stis 18/08 pairnei ta stoixima eis kai tis 17/08 */
GROUP BY
    positions.vessel_id, day
ORDER BY
    positions.vessel_id, day;
  
```

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

- Object Explorer:** Shows a tree view of the database structure, including Servers, Databases, Vessels, and Tables.
- Query Editor:** Contains the SQL query above.
- Data Output:** Displays the results of the query, which lists the number of positions per day for passenger vessels.
- Messages:** Shows a success message: "Successfully run. Total query runtime: 12 secs 157 msec. 893 rows affected."
- Scratch Pad:** An empty tab.

vessel_id	day	number_of_positions
1	2019-08-14 00:00:00	2430
2	2019-08-16 00:00:00	5440
3	2019-08-15 00:00:00	4160
4	2019-08-16 00:00:00	4300
5	2019-08-17 00:00:00	10140
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	2019-08-17 00:00:00	5170

## 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;
    
```

Query complete 00:00:05.137

Successfully run. Total query runtime: 5 secs 137 msec. 241 rows affected.

(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
13 SELECT *
14 FROM cte_example
15 WHERE speed = 0;
    
```

Total rows: 2 of 2 Query complete 00:00:04.321

Successfully run. Total query runtime: 4 secs 321 msec. 2 rows affected.

## 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'. Under 'Tables (3)', there are three tables: 'positions', 'vesseltypes', and 'vessels'. The 'positions' table has 24 rows. The 'Data Output' tab displays a query result table:

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	49019
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	35625
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	284462
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	207119
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    In 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 informall..	9	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
Railtton	1	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);

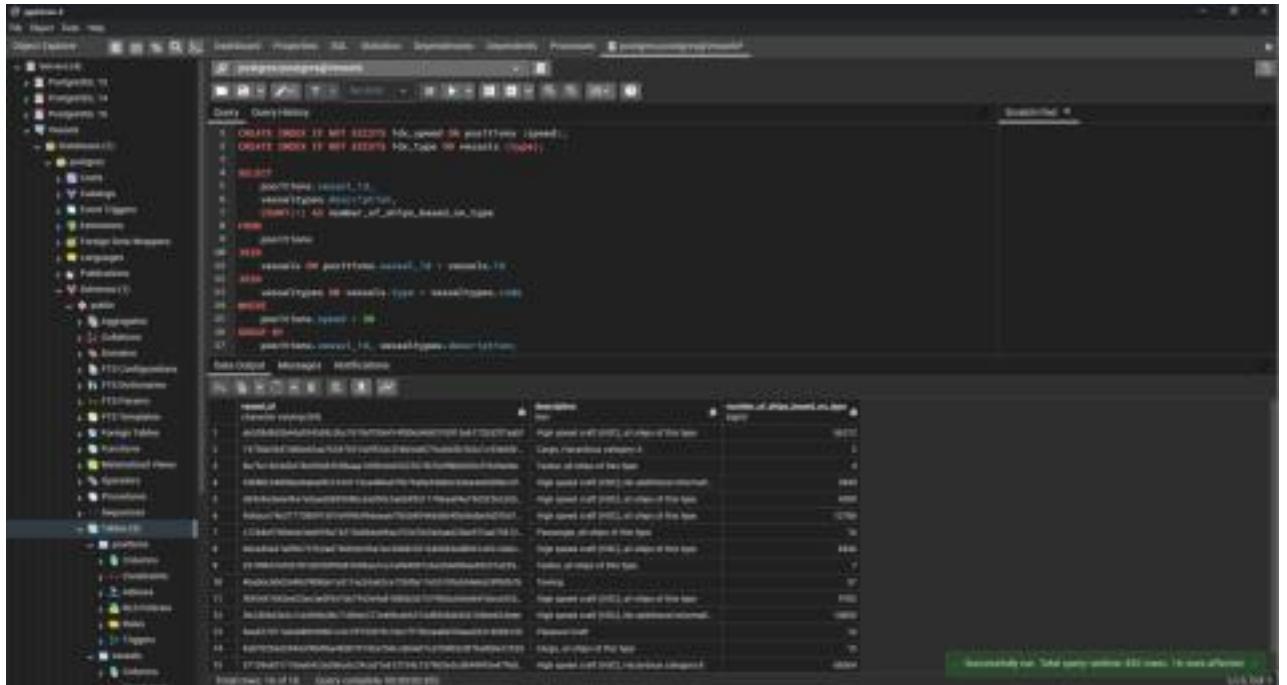
SELECT positions.vessel_id, vesseltypes.description, COUNT(*) AS number_of_ships_based_on_type
FROM positions, vessels
JOIN vesseltypes ON vessels.type = vesseltypes.code
WHERE positions.speed > 30 AND vessels.id = positions.vessel_id
GROUP BY positions.vessel_id, vesseltypes.description;
```

#### 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



```
CREATE INDEX IF NOT EXISTS idx_positions_1 ON positions (DATE_TRUNC('day', t));
CREATE INDEX IF NOT EXISTS idx_positions_2 ON positions (vessel_id);

SELECT
    vesseltypes.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 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;
  
```

Data Output:

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

Total rows: 893 Query complete on 2019-08-21 09:05:21

## 5.5 Υποερώτημα v

### 5.5.1 Κύδικας

(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 '2019-08-12' 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%'
GROUP BY positions.vessel_id, positions.speed
ORDER BY positions.vessel_id;

```

vessel_id	speed
048dc4a518f833b720a114a2676e05588d8e1af1c8ab7b2e2d46ea7d...	0
04c373534007a811b19394a2593acacc751cdeec0592695c834a5b...	0
05eda03f08e04052f0ffefefefefefefefefefefefefefefefefefefef...	0
06389f04089970ef0bed41aaed217815c4497027bc206e79c...	0
06f8a6f1e620529a9031389393597e079c726a47991784a2e79c...	0
075124ee049f9cf9278435252d40743c45c4e7313ef976816d01220...	0
083320d7d97977ba5a6d332176e2b13ef475ca18b2c10daee9fc50...	0
0941beda1c90ed2d5402d7a677e67293bf1fc1c139f0838726646cd0f7...	0
09c20baed4dec07c55fb6c7507c02fe1a79f12470644c9ff74e80b761...	0
0xa1a0fb9b05d77a7c25944453c3d9f9ff70487875ea937c2c03187579...	0
0accaceac20504b2f1517265c5d3d0e02ee539f8831d058e7d9575b8...	0
059910c8c5a0d2c374b77f8cc075e6b71179a3f29e6838fb3554d29a...	0
0be0292311e42030529c9ef25b4648682270f0f7c4729e7079f6a...	0
0cadaf0cbe92784a087fc507b4775c42d1ab0b9527e27d59f...	0
0d8ca28ae46c4a0a07cb5a5811722ffefefefefefefefefefefefef...	0
0f9566d42a1454ad02f14e78329c3e734472c42e998b6011aef725f82...	0
0f716f707004311ee9025e20920596761262588fbcc6e658c26205c5b659f6...	0
0f789f1ee232374608aa0d376a477165f48a006184b7c46c2b056600...	0
0f515baedc872f053d35d426470ff799272334e2e5547e0799992a09...	0
1079944a0e94ad74246878016575a2c4ff733a9499fb7a2c72d66d58...	0
11695aa0a8660362711b0f17386c9622232906a33716499793a88754...	0
12244858c3e50830832309ff50187890c4877994997a8730597...	0

(β)

The screenshot shows the pgAdmin 4 interface. In the Object Explorer, under the 'Vessels' database, there are tables like 'positions', 'vesseltypes', and 'vessels'. The 'positions' table has columns 'vessel\_id', 'description', and 'speed'. A SQL query is run in the Query tab:

```
1 CREATE INDEX IF NOT EXISTS idx_positions_t ON positions(t);
2 CREATE INDEX IF NOT EXISTS idx_vesselst_type ON vesselst(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
80dc9126-9d9f-0fffb55746bf762ec400a3a3347af41d76c4345d2bc2791a	Cargo, Hazardous categor...	0
d5b4ca21a3677d0cc9ef9ff3bcac8138af5a54c52eef0ff7935eb69e93c23a	Cargo, all ships of this type	0

Total rows: 2 of 2. Query complete 00:00:01.586.

## 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 ζεκινά 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';

```

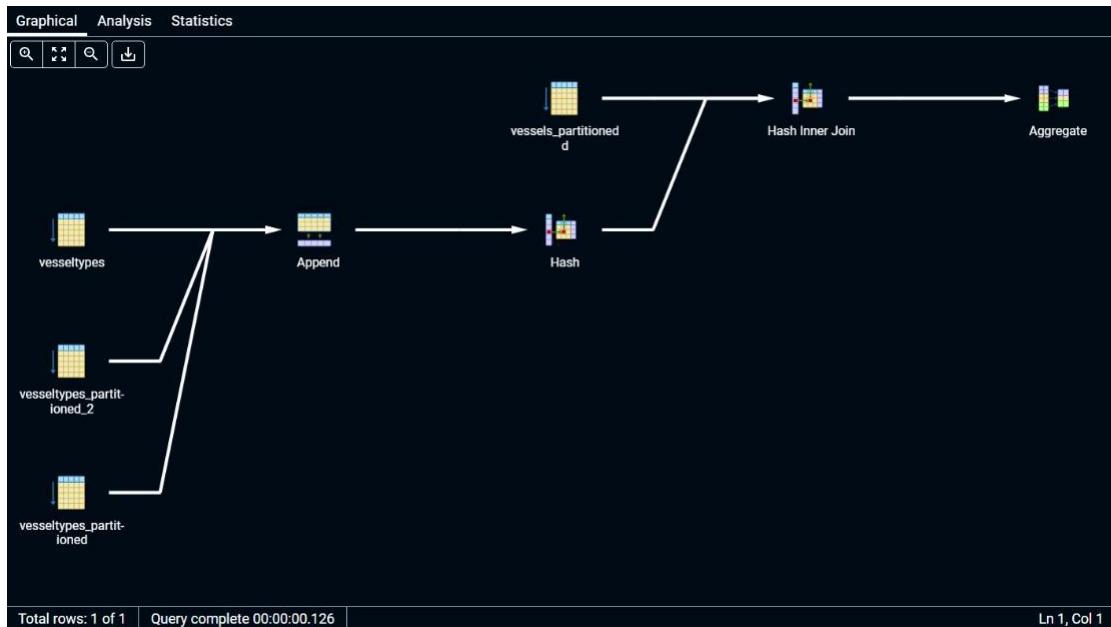
Εδώ βλέπουμε ότι η μεταφορά δεδομένων έχει γίνει με επιτυχία και ο partitioned πίνακας είναι γεμάτος με δεδομένα και το **flag** των πλοίων αρχίζει με **Greece**.

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

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

Tables (6)
> positions
> positions_partitioned_speed_high

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	23.62229	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	23.62228	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	23.62237	37.83283	178	181	30.6
6	2019-08-04 07:39:14	23.61028	37.93351	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	23.62235	37.83226	178	180.2	30.5

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
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;

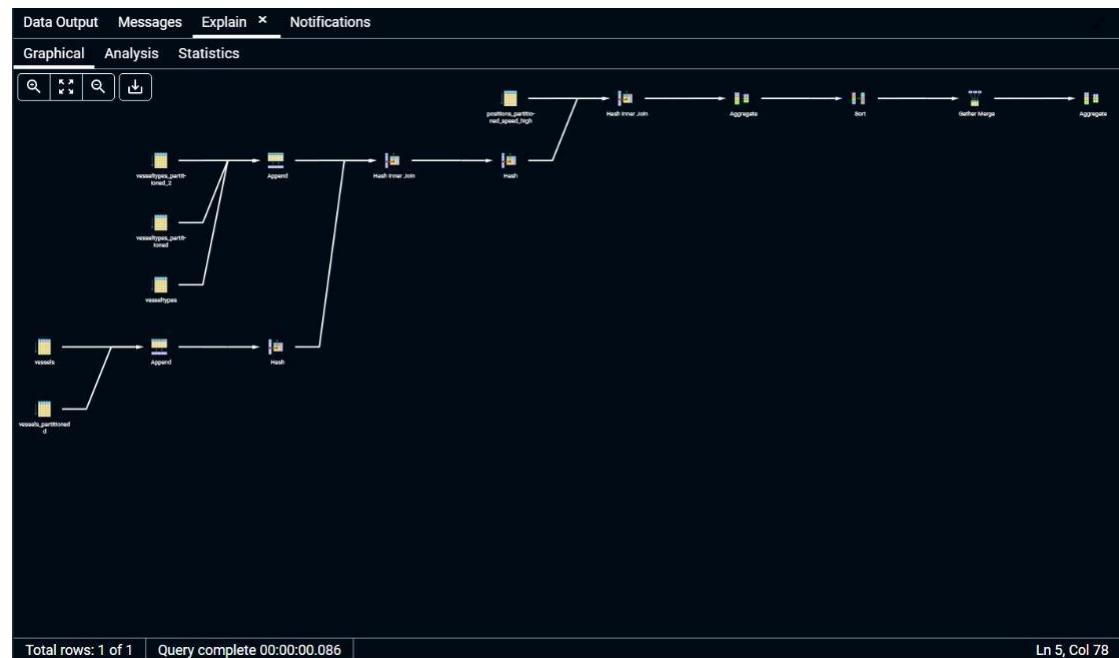
```

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

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);

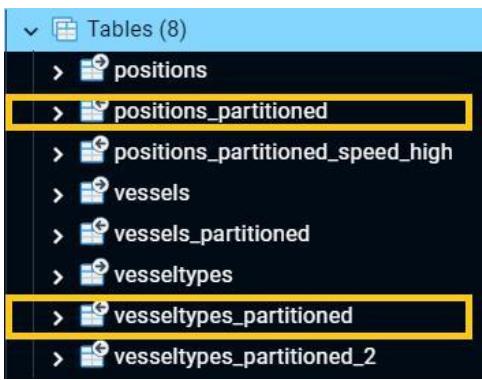
CREATE INDEX idx_vesseltypes_partitioned_description ON Vesseltypes_partitioned(description);
CREATE INDEX idx_positions_partitioned_t ON Positions_partitioned(t);

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 και '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.

```

1 SELECT
2   positions_partitioned.vessel_id, DATE_TRUNC('day', t) AS day, COUNT(*) AS number_of_positions
3   FROM
4     positions_partitioned, vesseltypes_partitioned
5   WHERE
6     vesseltypes_partitioned.description LIKE 'Passenger%' AND positions_partitioned.t BETWEEN '2019-08-14' AND '2019-08-19'
7   GROUP BY
8     positions_partitioned.vessel_id, day
9   ORDER BY
10    positions_partitioned.vessel_id, day

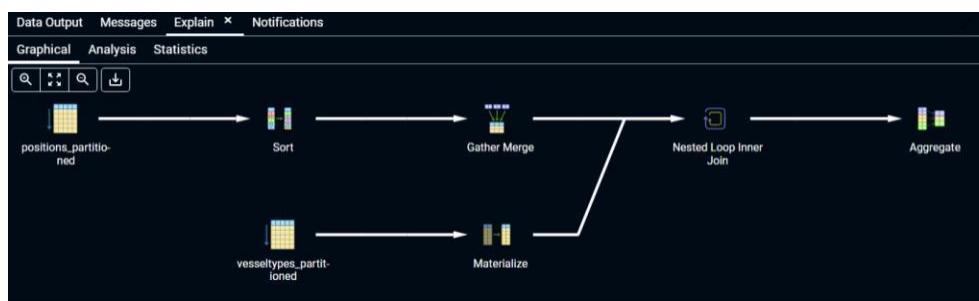
```

Total rows: 892 of 892    Query complete 00:00:06.699    Ln 10, Col 41

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

```

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%';

```

Total rows: 1 of 1 | Query complete 00:00:05.405. | Ln 2, Col 58

Εδώ βλέπουμε με επιτυχία ότι τα δεδομένα καταχωρήθηκαν στους partitioned πίνακες.

(α)

```

SELECT Positions_partitioned_2.vessel_id, Positions_partitioned_2.speed
FROM Positions_partitioned_2, Vesseltypes_partitioned_2
WHERE Positions_partitioned_2.t BETWEEN '2019-08-15' AND '2019-08-19' AND Positions_partitioned_2.speed = 0 AND Vesseltypes_partitioned_2.code = 'Cargo'
GROUP BY Positions_partitioned_2.vessel_id, Positions_partitioned_2.speed
ORDER BY Positions_partitioned_2.vessel_id;

```

vesselId	speed
0466c84a518ff33b7206a114ca7660d56889dr1fbccfb7bb2c46ea5a7d...	0
0463373354007a1d1c13945a2f5939acceca75d3eece59c2f6c52d4a45b...	0
05ecaa038e0b9452fa8ae01272e05056996d60b091deea374912468...	0
0658091408917bd78bc70edf8ca41aae552517815c1449322f5bc20d79e5...	0
069e41da2b052fa0313b58930a597ea9e278e49f91784a2e725e...	0
0751240ee49afcf9277844362b240a743a4e5c7e0315ef697661640722...	0
08532cd729fe7edaa5c633337baeb17e54759a1897c18d1aae19b50...	0
0941beda1c9e6d2d4d26a27eab7239c0f1cc13539f93687256a6c40f07...	0
09c2bbaad4ec07d5d4f86c7507ce2f2a1e791247044c5d71e4c8b7d4...	0

Total rows: 241 of 241 | Query complete 00:00:03.456. | Ln 3, Col 138

(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;

```

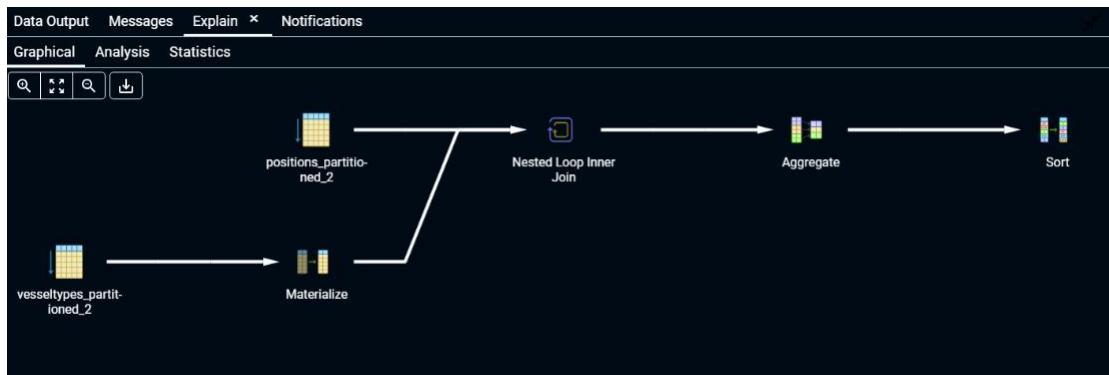
vessel_id	description	speed
00be912679db6f8574a08742ec402e43a5f5afe14d76c4545d28ca27c91a...	Cargo, Hazardous categor...	0
0fb4ca213677d0c9d9813bae138a554c52eef8f87935eb09e930d3a...	Cargo, all ships of this type	0

Total rows: 2 of 2 | Query complete 00:00:01.345. | Ln 1, Col 1

#### 6.5.4 Υποερώτημα ν: Διάγραμμα

Αν τρέξουμε την εντολή EXPLAIN ANALYZE έχουμε το εξής διάγραμμα:

(α)



(β)

