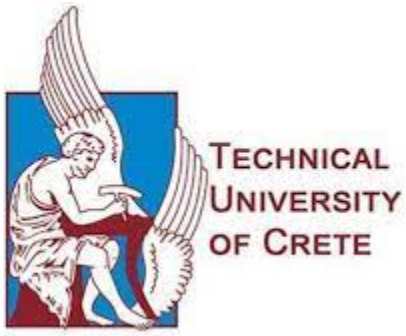


Βάσεις Δεδομένων

Αναφορά 2ης φάσης

Ιούνιος 21, 2024

Δήμας Χρήστος, 2021030183
Ατματζίδου Ευσταθία, 2018030028

Contents

1 Εισαγωγή

Για την εκπλήρωση της φάσης Β της εργαστηριακής εργασίας που μας ανατέθηκε στα πλαίσια του μαθήματος, χρειάστηκε να υλοποιήσουμε μια εντολή σε PostgreSQL και έπειτα να προσπαθήσουμε να μελετίσουμε την απόδοση της ερώτησης και να την βελτιστοποιήσουμε.

Η ζητούμενη εντολή που υλοποιήσαμε ειναι η εξής :

Τύπωσε τα επίπεδα εκπαίδευσης (eduLevel) και το πλήθος, ανά επίπεδο εκπαίδευσης, μελών που έχουν σπουδάσει στον Καναδά (Canada) και τα οποία έχουν αναρτήσει τους τελευταίους 6 μήνες τουλάχιστον 2 αγγελίες προσφοράς εργασίας για άτομα ηλικίας (fromAge) μεγαλύτερης των 21 ετών και μικρότερης των 30 ετών και έχοντας δεχτεί μηνύματα (msg) τους τελευταίους 6 μήνες.

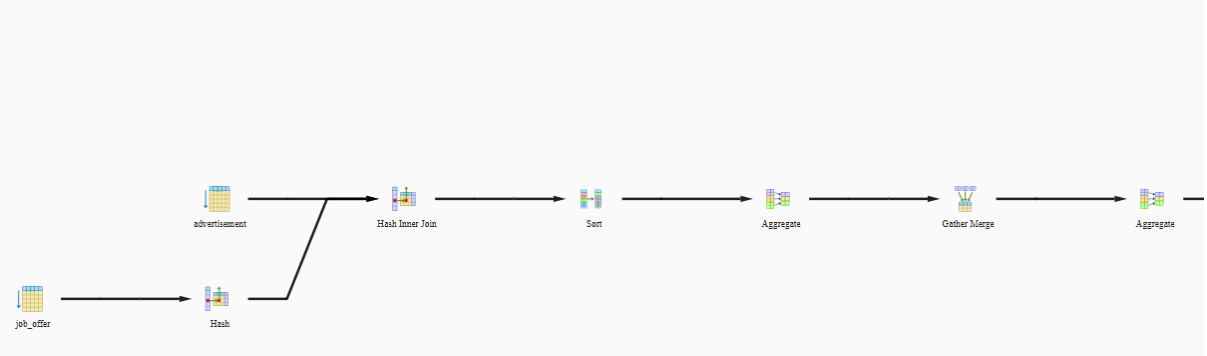
2 Επίλυση και αποτέλεσματα πρίν τη βελτιστοποίηση

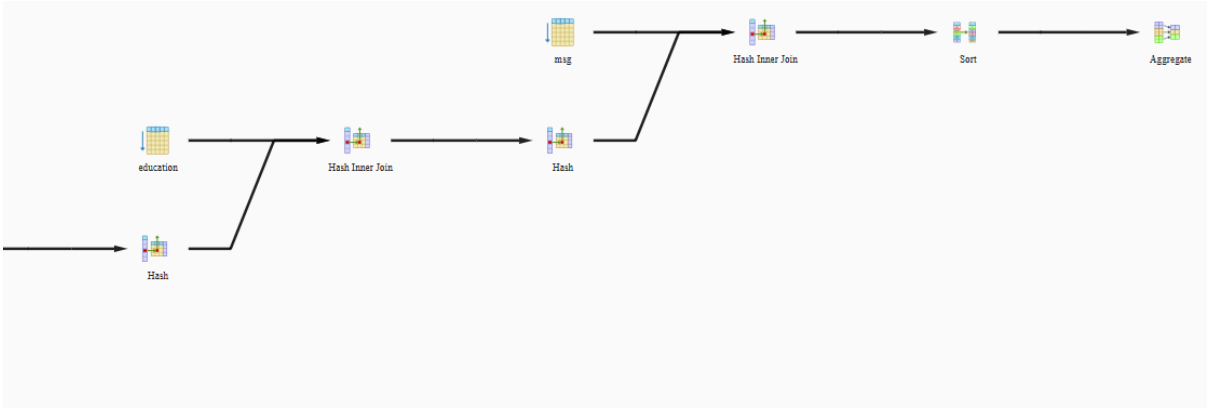
Ο τρόπος με τον οποίο επιλέχθηκε να επιλυθεί το ζητούμενο ο ακόλουθος :

```
1  SELECT e.edu_level, COUNT(*) AS members
2  FROM education e
3  JOIN (
4      SELECT a.email
5      FROM advertisement a
6      JOIN job_offer j ON a.advertisement_id = j.
           advertisement_id
7      WHERE a.date_posted >= CURRENT_DATE - INTERVAL '6 months'
8        AND j.from_age > 21
9        AND j.from_age < 30
10     GROUP BY a.email
11     HAVING COUNT(j.advertisement_id) >= 2
12 ) ads ON e.email = ads.email
13 JOIN msg msg ON e.email = msg.receiver_email
14 WHERE e.country = 'Canada'
15 AND msg.date_sent >= CURRENT_DATE - INTERVAL '6 months'
16 GROUP BY e.edu_level
```

Listing 1: SQL Query

Έπειτα από την εκτέλεση του προαναφερθέντος query, με τη χρήση της EXPLAIN ANALYZE, λάβαμε αποτελέσματα σχετικά με τον μέσο χρόνο περάτωσης του αιτήματος ο οποίος υπολογίστηκε πως είναι **2.384,540 ms**. Η δομή του query αποφασίστηκε, έπειτα από δοκιμές στη σειρά των συνδέσεων (JOINS) και η υπάρχουσα κρίθηκε η πιο αποδότικη χρονικά. Επιπλέον παρατηρήσαμε και το πλάνο εκτέλεσης που επιλέγει ο βελτιστοποιητής και είναι το εξής :





3 Βελτιστοποίηση

Εφόσον στο προηγούμενο στάδιο κατανοήσαμε το πρόβλημα και εντοπίσαμε τα σημεία στα οποία δημιουργείται καθυστέριση, οδηγηθήκαμε στις παρακάτω αποφάσεις σχετικά με τα ευρετήρια:

1. Δημιουργήσαμε ευρετήριο στον πίνακα education με την χρήση btree στη στήλη country.

2. Δημιουργήσαμε ευρετήριο στον πίνακα education με την χρήση hash στη στήλη email.

3. Δημιουργήσαμε ευρετήριο στον πίνακα msg, στη στήλη received_email.

4. Δημιουργήσαμε ευρετήριο στον πίνακα advertisement, στη στήλη email.

Μετά απο τις αλλαγές αυτές ο χρόνος εκτέλεσης μειώθηκε στα : **1.177,75 ms**, και αποτέλεσε τον καλύτερο χρόνο που σημειώθηκε έπειτα απο εκτενείς δοκιμές πάνω σε αρκετές στήλες πινάκων, καθώς είτε υπήρχε συνδυασμός είτε μεμονωμένο ευρετήριο κανένα αποτέλεσμα δεν ήταν καλύτερο. Ακόμα παρατηρήθηκε πως στη περίπτωση μας οι συνδυασμοί ευρετηρίων κατά κύριο λόγο έδιναν καλύτερο αποτέλεσμα.

Στη συνέχεια, δοκιμάσαμε να ομαδοποιήσουμε (CLUSTER) πίνακες βάση των ευρετηρίων που δημιουργήσαμε με τα αποτελέσματα να μην δικαιώνουν την επιλογή μας και να μην παρατηρείται κάποια ιδιαίτερη βελτίωση στο πρόβλημα.

Έπειτα, απενεργοποιήσαμε επιλεκτικά αλγόριθμους συνδέσεων και παρατηρούμε πως άνα περίπτωση έχουμε τα εξής αποτελέσματα :

```
SET enable_nestloop = ON;
SET enable_hashjoin = ON;
SET enable_mergejoin = OFF;
```

Listing 2: **execution_time = 791 ms**

```
SET enable_nestloop = ON;
SET enable_hashjoin = OFF;
SET enable_mergejoin = ON;
```

Listing 3: **execution_time = 995 ms**

```
SET enable_nestloop = OFF;
SET enable_hashjoin = ON;
SET enable_mergejoin = ON;
```

Listing 4: **execution_time > 1000 ms**

```
SET enable_nestloop = OFF;
SET enable_hashjoin = OFF;
SET enable_mergejoin = ON;
```

Listing 5: **execution_time > 2000 ms**

```
SET enable_nestloop = OFF;
SET enable_hashjoin = ON;
SET enable_mergejoin = OFF;
```

Listing 6: **execution_time > 1000 ms**

```
SET enable_nestloop = ON;
SET enable_hashjoin = OFF;
SET enable_mergejoin = OFF;
```

Listing 7: **execution_time = 722 ms**

```
SET enable_nestloop = OFF;
SET enable_hashjoin = OFF;
SET enable_mergejoin = OFF;
```

Listing 8: **execution_time > 1000 ms**

```
SET enable_nestloop = ON;
SET enable_hashjoin = ON;
SET enable_mergejoin = ON;
```

Listing 9: **execution_time = 684 ms**

Με αυτόν τον τρόπο καταλήξαμε στο γεγονός πως οι αλγόριθμοι συνδέσεων είναι πιο αποδοτικό να είναι ενεργοποιημένοι καθώς παρατηρήθηκε σημαντική βελτίωση του χρόνου εκτέλεσης του αιτήματος ο οποίος απο **2.384,540 ms** μειώθηκε εν τέλη σε **684 ms**. Συμπληρωματικά παρατηρήθηκε πως η απόδοση των αλγόριθμων συνδέσεων επηρεάστηκε από τα ευρετήρια, καθώς στις μετρήσεις που έγιναν πρίν τη δημιουργία ευρετηρίων λάβαμε πολύ διαφορετικά αποτελέσματα.

Το τελευταίο ερώτημα δεν μπορέσαμε να το ολοκληρώσουμε λόγω περιορισμένου χρονικού περιθωρίου.

4 Συμπεράσματα

Συμπερασματικά, καταλήξαμε πως με τα ζητούμενα της εκφώνησης τα οποία υλοποιήσαμε και μειώθηκε αρκετά ο χρόνος εκτέλεσης του ζητούμενου αιτήματος, κατανοήσαμε καλύτερα και διά της πράξης την διαδικασία βελτιστοποίησης και επεξεργασιας των αιτημάτων σε SQL.

Ακόμα παρακολουθόντας τα αποτελέσματα της EXPLAIN ANALYSE πρίν και μετά απο κάθε αλλαγή μας κατανοήσαμε τη λειτουργία των αλγόριθμων σύνδεσης και την επιρροή των ευρετηρίων στη ταχύτητα της εκτέλεσης του αιτήματος και στην επιλογή των αλγορίθμων.