

5η Εργασία: Βελτιστοποίηση Ερωτημάτων και Εναύσματα

Ονοματεπώνυμο:

Ελένη Κεχριώτη

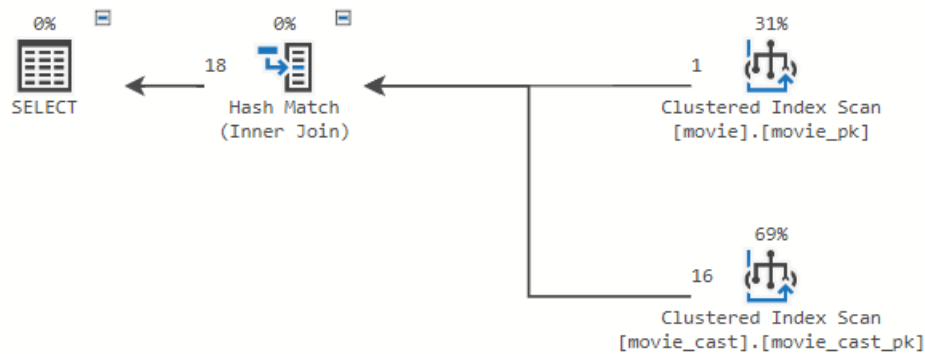
ΑΜ: 3210078

Τριαντάφυλλος Εφραίμ Κιοσσές

ΑΜ: 3210079

Ερώτημα Α

Να δοθεί screenshot του πλάνου εκτέλεσης του ερωτήματος:



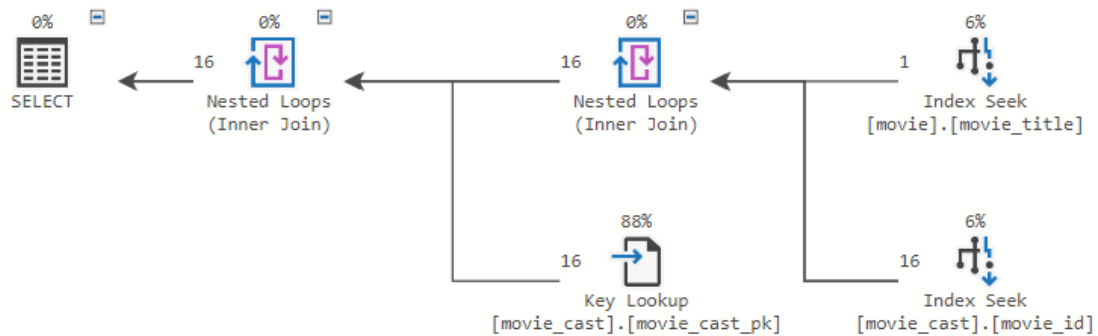
Ερώτημα Β

- a. Να φτιάξετε **2 ευρετήρια** τα οποία επιταχύνουν την εκτέλεση του ερωτήματος.

```
CREATE INDEX movie_title  
on movie(title);
```

```
CREATE INDEX movie_id  
on movie_cast(movie_id);
```

- b. Να δοθεί το αντίστοιχο πλάνο εκτέλεσης του ερωτήματος αφού έχουν φτιαχτεί τα ευρετήρια.



c. Ποιές είναι οι αλλαγές που έχουν γίνει στο πλάνο εκτέλεσης;

Στο δεύτερο πλάνο εκτέλεσης παρατηρούμε ότι χρησιμοποιούνται τα 2 ευρετήρια που φτιάξαμε `movie_id` και `movie_title` για τους πίνακες `movie_cast` και `movie` αντίστοιχα. Στο προηγούμενο πλάνο εκτέλεσης σαρώνονται τα ευρετήρια `movie_pk` και `movie_cast_pk` και επιστρέφουν τις εγγραφές για το επόμενο στάδιο Hash Match (inner join), που σημαίνει ότι δεν υπάρχει κατάλληλο ευρετήριο για να γίνει γρηγορότερα η διαδικασία. Αντίθετα, στο νέο πλάνο σαρώνεται το `movie_title` και επιστρέφει την εγγραφή με τίτλο 'Armageddon'. Ακόμη, σαρώνεται το ευρετήριο `movie_id` και επιστρέφει τις εγγραφές που ταιριάζουν (στο `movie_id`) με τη εγγραφή που επέστρεψε η σάρωση του προηγούμενου ευρετηρίου. Τέλος, γίνεται σύζευξη των εγγραφών που προκύπτουν από το προηγούμενο στάδιο με τις εγγραφές που προκύπτουν από το ψάξιμο μέσω του πρωτεύοντος κλειδιού του πίνακα `movie_cast`.

Εν συντομία, η κυριότερη αλλαγή που βλέπουμε είναι ότι στο δεύτερο πλάνο εκτέλεσης χρησιμοποιούνται τα ευρετήρια για την απάντηση του ερωτήματος, ενώ στο πρώτο πλάνο εκτέλεσης χρησιμοποιούνται τα πρωτεύον ευρετήρια.

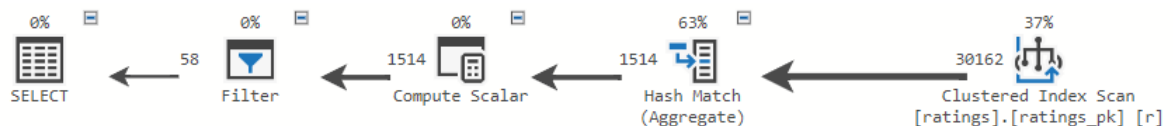
Ερώτημα C

Χρησιμοποιώντας τις εντολές GROUP BY και HAVING στον πίνακα Ratings:

- a. Ποιό είναι το ερώτημα το οποίο επιστρέφει τον μέσο όρο βαθμολογίας ανά ταινία για ταινίες με μέση βαθμολογία πάνω από 4;

```
SELECT m.title, avg(r.rating) as avgRating
FROM movie m
INNER JOIN ratings r
ON m.id = r.movie_id
GROUP BY m.id, m.title
HAVING avg(r.rating)>4
```

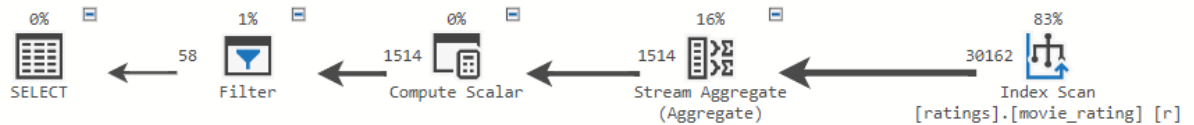
- b. Να τρέξετε το ερώτημα και να επιστρέψετε το αντίστοιχο πλάνο εκτέλεσης.



- c. Να δημιουργηθεί ένα *ευρετήριο επικάλυψης* το οποίο μπορεί να χρησιμοποιηθεί για την απάντηση στο συγκεκριμένο ερώτημα.

```
CREATE INDEX movie_rating
ON ratings (movie_id, rating);
```

- d. Καταγράψτε το πλάνο εκτέλεσης με το πλάνο επικάλυψης,



- e. τι παρατηρείτε σε αυτό ως προς το διάβασμα του πίνακα `Ratings` και γιατί συμβαίνει αυτό;

Στο ευρετήριο `movie_rating` τοποθετούνται τα γνωρίσματα του `movie_id` και `rating` του πίνακα `ratings` με αύξουσα σειρά πρώτα με βάση το `movie_id` και ύστερα με βάση το `rating`. Με αυτή τη ταξινόμηση η βαθμολογία για κάθε ταινία βρίσκεται συγκεντρωμένη και όχι διάσπαρτη σε διάφορα σημεία του πίνακα `rating`, γεγονός που επιταχύνει το υπολογισμό του μέσου όρου βαθμολόγησης κάθε ταινίας και άρα επιταχύνει την προβολή των αποτελεσμάτων. Ακόμα, αγνοείται η πληροφορία για το `user_id` του πίνακα `rating`, γεγονός που βοηθάει στην επιτάχυνση.

Ερώτημα D

- a. Να δοθεί η εντολή `Alter Table` για την δημιουργία του γνωρίσματος `AVG_Rating` στον πίνακα `Movie`.

```
ALTER TABLE movie ADD AVG_Rating FLOAT NULL;
```

- b. Χρησιμοποιώντας την εντολή `UPDATE`, αρχικοποιήστε το συγκεκριμένο γνώρισμα με την μέση βαθμολογία ανά ταινία σε περίπτωση που αυτή υπάρχει και `Null` διαφορετικά.

```
UPDATE movie
SET AVG_Rating = (
  SELECT avg(rating)
  FROM ratings
  WHERE movie_id = id
);
```

- c. Γράψτε ένα *έναυσμα* (trigger) το οποίο για κάθε εισαγωγή στον πίνακα `Ratings` ενημερώνει το γνώρισμα `AVG_Rating` του πίνακα `Movie` με την μέση βαθμολογίαμόνο των ταινιών των οποίων η βαθμολογία άλλαξε.

```

CREATE TRIGGER avg_Rating_Trigger
ON ratings
AFTER INSERT
AS
BEGIN
    Update movie
    SET AVG_Rating = (
        SELECT avg(rating)
        FROM ratings
        WHERE movie_id = id
        GROUP BY ratings.movie_id
    )
    WHERE movie.id IN (SELECT movie_id FROM inserted)
END;

```

```

insert into ratings values (15, 8420, 5);
insert into ratings values (9, 9051, 2);

```

- d. Ποιό από τα ευρετήρια που φτιάξατε βοηθάει στην εκτέλεση του συγκεκριμένου trigger.

Κατά την εκτέλεση του συγκεκριμένου trigger χρησιμοποιείται το πρωτεύον ευρετήριο movie_pk για τη γρήγορη εύρεση της ταινίας που έγινε εισαγωγή βαθμολογίας στο πίνακα ratings από τον πίνακα inserted.

Ερώτημα Ε

Περιγράψτε κάποιον εναλλακτικό τρόπο με τον οποίο θα μπορούσατε να υλοποιήσετε το trigger σας, χωρίς να χρειαστεί να τρέχετε κάποια group by συνάρτηση στα υπάρχοντα δεδομένα.

Αρχικά θα μπορούσαμε να προσθέσουμε δύο στήλες στον πίνακα movie με ονόματα π.χ sum_rating και sum_users, που θα δηλώνουν το άθροισμα όλων των βαθμολογιών που έχουν βάλει οι χρήστες καθώς και τον αριθμό όλων των χρηστών που έχουν βαθμολογήσει την κάθε ταινία. Έτσι το avg_rating μπορεί πολύ εύκολα να βρεθεί με μια απλή διαίρεση sum_rating/sum_users.

Αντίστοιχα το trigger μπορεί να υλοποιηθεί θέτοντας στο κομμάτι του update το avg_rating να είναι ίσο με το (sum_ratings + SUM(rating))/((sum_users + COUNT(user_id))