

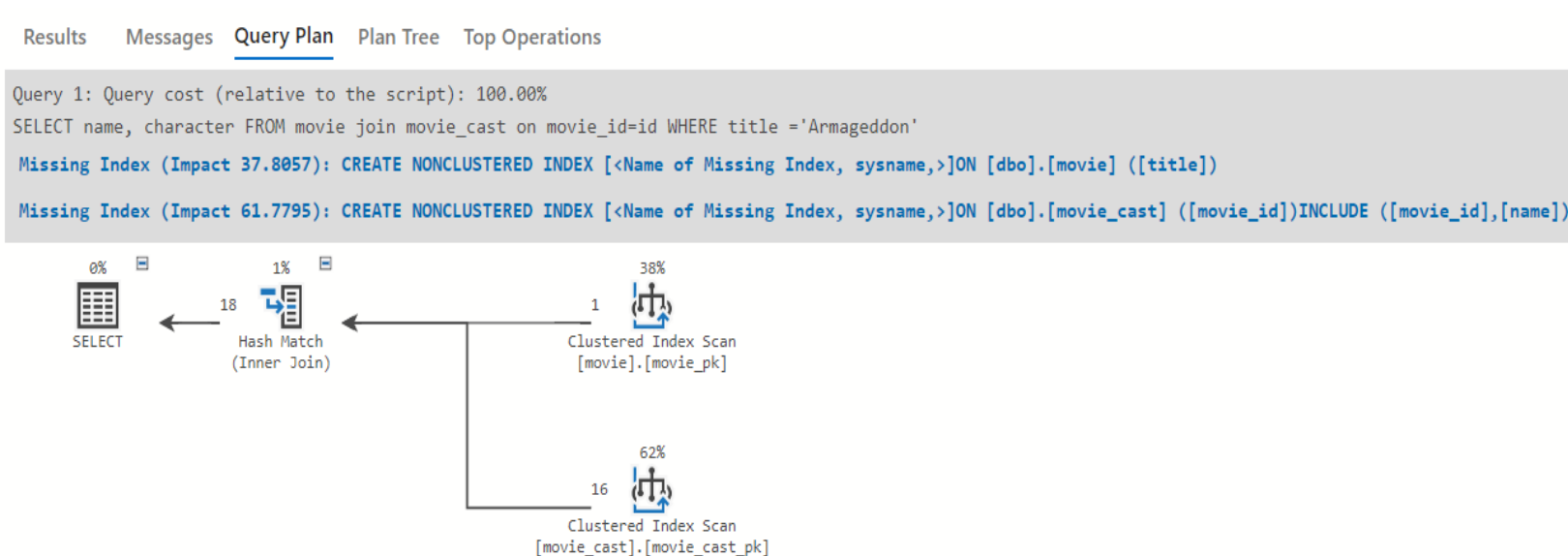
5η Εργασία: Βελτιστοποίηση Ερωτημάτων και Εναύσματα

Ονοματεπώνυμο

ΣΧΟΙΝΑΚΗ ΜΑΡΙΑ : 3210191
ΕΛΕΥΘΕΡΙΟΣ ΒΕΡΟΥΧΗΣ: 3200019

Ερώτημα Α

Να δοθεί screenshot του πλάνου εκτέλεσης του ερωτήματος:



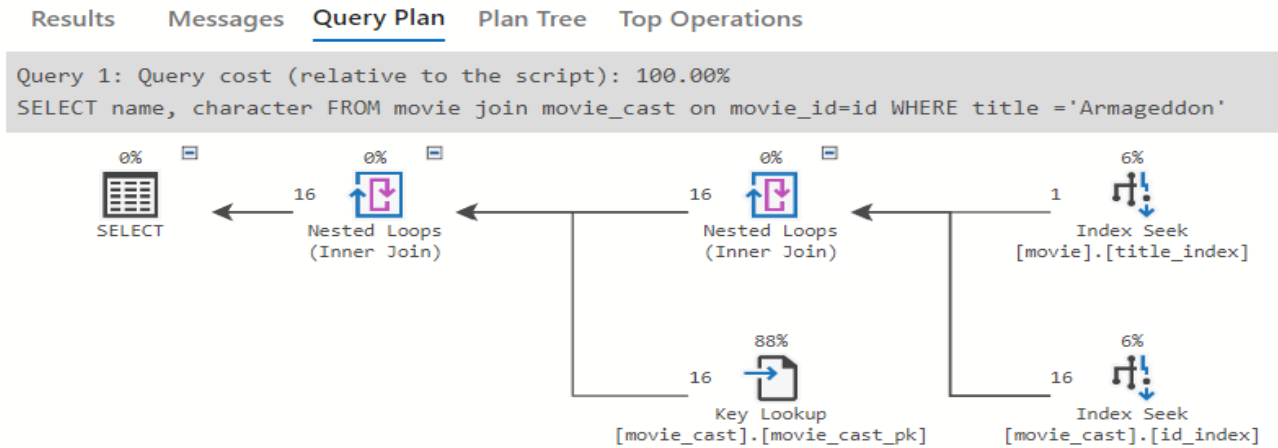
Ερώτημα Β

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

```
CREATE INDEX title_index ON movie(title)
```

```
CREATE INDEX movie_id_index 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;

Οι αλλαγές που έχουν γίνει στο πλάνο εκτέλεσης είναι η χρήση των 2 αυτών ευρετηρίων που δημιουργήσαμε για την επιτάχυνση της αναζήτησης. Το πρόγραμμα αντί να τρέξει γραμμικά (δηλαδή όλες της εγγραφές μία μία) τους πίνακες movie και movie_cast, χρησιμοποιεί τα ευρετήρια για να μειώσει τον αριθμό των υπό εξέταση εγγραφών, επιτυγχάνοντας έτσι μια πιο αποδοτική και αποτελεσματική εκτέλεση του ερωτήματος. Ο “κόπος” του συστήματος μειώνεται σημαντικά καθώς πλέον ξέρει που να ψάξει λόγω της χρήσης των ευρετηρίων και δεν ψάχνει όλο τον πίνακα. Έτσι, αντιστοιχίζει σε ραγδαία μικρότερο χρόνο και χώρο (πόροι αναζήτησης και ανάκτησης) τα κατάλληλα δεδομένα επεξεργασίας.

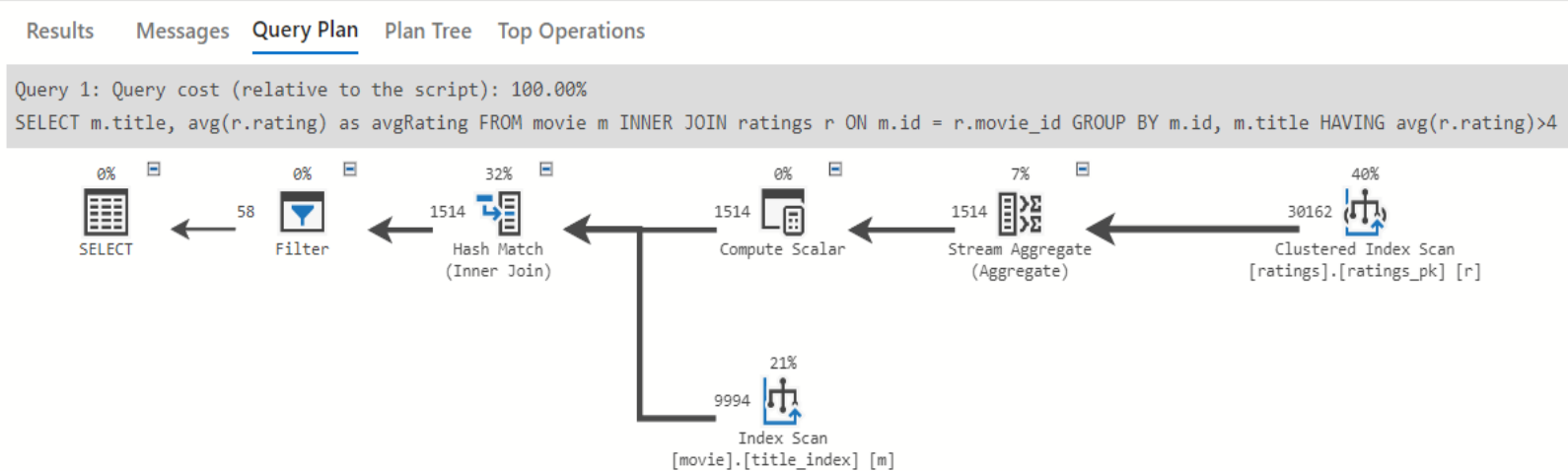
Ερώτημα C

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
```

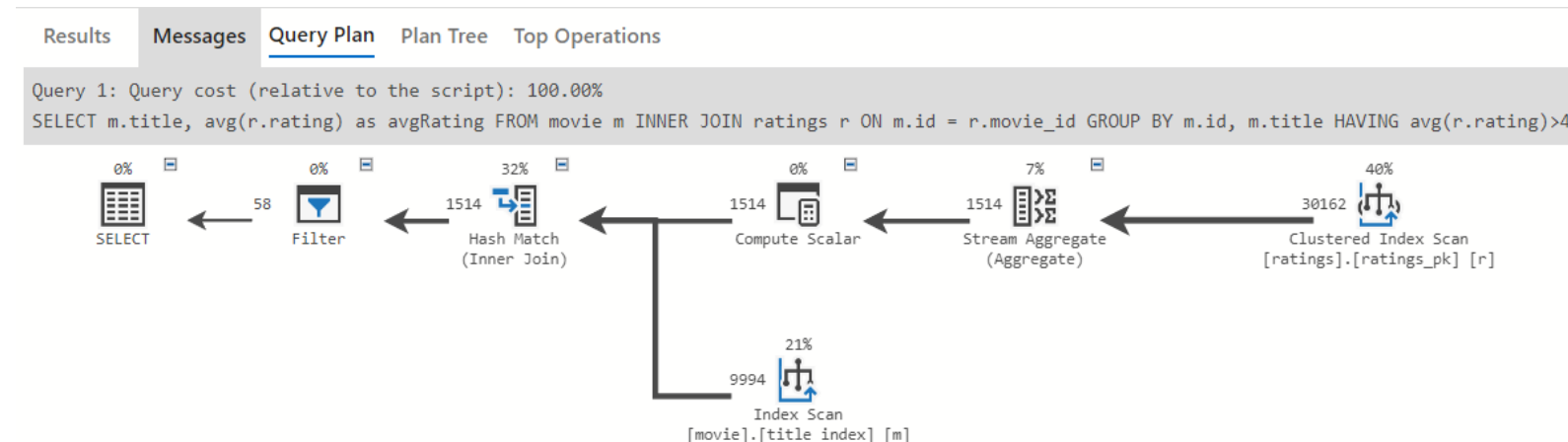
- b. Να τρέξετε το ερώτημα και να επιστρέψετε το αντίστοιχο πλάνο εκτέλεσης.



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

```
CREATE INDEX rating_and_movie_id_index 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;

Δεδομένου ότι στον πίνακα ratings έχουμε ως πρωτεύον κλειδί το (movie_id, user_id), η δημιουργία ενός ευρετηρίου επικάλυψης είναι περιττή. Με την δημιουργία του κλειδιού, δημιουργήθηκε ένα πρωτεύον ευρετήριο. Αυτό το ευρετήριο, διασφαλίζει την μοναδικότητα και ταξινόμηση των εγγραφών του πίνακα βάση των πεδίων movie_id, user_id. Με βάση λοιπόν αυτόν τον συνδυασμό πεδίων, η βάση μας έχει ήδη μια δομή για την ταχύτερη ανάκτηση πληροφορίας, καθώς αποφεύγει να διαβάσει γραμμικά όλο τον

πίνακα. Συνεπώς, το ευρετήριο επικάλυψης, που κάνει την ίδια δουλειά είναι περιττό, γιατί και αν δημιουργηθεί το ευρετήριο επικάλυψης, δεν θα αλλάξει τίποτα στο πλάνο εκτέλεσης.

Ερώτημα D

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

```
ALTER TABLE movie ADD AVG_Rating FLOAT
```

- 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(r.rating)
    FROM ratings r
    WHERE movie.id = r.movie_id
    GROUP BY r.movie_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 βαθμολογία μόνο των ταινιών των οποίων η βαθμολογία άλλαξε.

```
CREATE TRIGGER AVG_UPDATE
ON ratings
AFTER INSERT
AS
BEGIN
    UPDATE movie
    SET AVG_Rating = (
        SELECT avg(r.rating)
        FROM ratings r
        WHERE movie.id = r.movie_id
        GROUP BY r.movie_id
    )
    WHERE movie.id IN (SELECT movie_id FROM inserted)
END

insert into ratings values(17, 2, 4)
insert into ratings values(12, 2, 1)
insert into ratings values(17, 33, 4)
```

- d. Ποιό από τα ευρετήρια που φτιάξατε βοηθάει στην εκτέλεση του συγκεκριμένου trigger. Το πρωτεύον ευρετήριο του `ratings` βοηθάει, καθώς στο έναυσμα χρειάζεται η ανάκτηση κάποιων πληροφοριών του πίνακα `ratings` και με την χρήση του ευρετηρίου η ανάκτηση αυτή γίνεται ταχύτερα, αφού δεν διαβάζεται όλος ο πίνακας `ratings`, παρά μόνο οι εγγραφές που μας ενδιαφέρουν. Το ίδιο ισχύει και για το ευρετήριο επικάλυψης, το οποίο όπως προαναφέρθηκε είναι περιττό.

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.

Ένας εναλλακτικός τρόπος για την υλοποίηση του trigger, χωρίς την χρήση κάποιας συνάρτησης group by, είναι προσθέτοντας δύο γνωρίσματα στον πίνακα movie, έστω count_ratings, το οποίο αντιπροσωπεύει το πλήθος βαθμολογιών για κάθε ταινία και sum_ratings, το οποίο αντιπροσωπεύει το άθροισμα των βαθμολογιών για κάθε ταινία. Μπορούμε να υλοποιήσουμε έτσι το trigger, έτσι ώστε κάθε φορά να ανανεώνει 3 γνωρίσματα του πίνακα movie. Το count_ratings(+=1), το sum_ratings(+=new_rating) και φυσικά το AVG_Rating(sum_ratings/count_ratings). Η συνάρτηση AVG στην ουσία τρέχει όλο τον πίνακα για να βρει τις βαθμολογίες βάσει το id, τις προσθέτει και μετά διαιρεί με το πλήθος των βαθμολογιών. Με το να κρατάμε σε 2 γνωρίσματα το άθροισμα και το πλήθος των βαθμολογιών για κάθε ταινία μειώνουμε και τον χρόνο που κάνει η ανανέωση, καθώς πλέον η AVG δεν ψάχνει στον πίνακα ratings και το trigger δεν χρησιμοποιεί καν τον πίνακα ratings πια, παρά μόνο την καινούρια εγγραφή του. Μια μαθηματική μοντελοποίηση της παραπάνω ιδέας είναι :

$$\text{AVG_Rating} = (\text{old_sum} + \text{new rating}) / (\text{old_count} + 1)$$