# 5η Εργασία: Βελτιστοποίηση Ερωτημάτων και Εναύσματα

## Δροσάτος Χαράλαμπος

## Πετρουλάκης Ιωσήφ

### Ερώτημα Α

Να δοθεί screenshot του πλάνου εκτέλεσης του ερωτήματος:



### Ερώτημα Β

1. Να φτιάξετε **2 ευρετήρια** τα οποία επιταχύνουν την εκτέλεση του ερωτήματος.
2. Να δοθεί το αντίστοιχο πλάνο εκτέλεσης του ερωτήματος αφού έχουν φτιαχτεί τα ευρετήρια.



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

### Ερώτημα C

Χρησιμοποιώντας τις εντολές GROUP BY και HAVING στον πίνακα Ratings:

1. Ποιό είναι το ερώτημα το οποίο επιστρέφει τον μέσο όρο βαθμολογίας ανά ταινία για ταινίες με *μέση βαθμολογία πάνω από 4;*
2. Να τρέξετε το ερώτημα και να επιστρέψετε το αντίστοιχο πλάνο εκτέλεσης.



1. Να δημιουργηθεί ένα *ευρετήριο επικάλυψης* το οποίο μπορεί να χρησιμοποιηθεί για την απάντηση στο συγκεκριμένο ερώτημα.
2. Καταγράψτε το πλάνο εκτέλεσης με το πλάνο επικάλυψης,



1. Τι παρατηρείτε σε αυτό ως προς το διάβασμα του πίνακα Ratings και γιατί συμβαίνει αυτό;

### Ερώτημα D

1. Να δοθεί η εντολή Alter Table για την δημιουργία του γνωρίσματος AVG\_Rating στον πίνακα Movie.

ALTER TABLE movie

ADD AVG\_Rating FLOAT (53);

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

with averages AS

(

    SELECT avg(r.rating) as average, m.id

    FROM movie m

    INNER JOIN ratings r ON m.id = r.movie\_id

    GROUP BY m.id

)

UPDATE movie

SET AVG\_Rating =

(

    SELECT average

    FROM averages

    WHERE averages.id = movie.id

);

1. 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 re\_calc\_avg\_rating ON ratings

AFTER INSERT

AS BEGIN

    UPDATE movie

    SET AVG\_Rating =

    (

        SELECT avg(r.rating)

        FROM ratings r

        WHERE r.movie\_id = inserted.movie\_id

    )

    FROM movie m

    INNER JOIN inserted ON m.id = inserted.movie\_id;

END

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

Requires previous parts

### 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.

Μπορούμε να προσθέσουμε στον πίνακα movie ένα ακόμα column όπου θα περιέχει το count των ratings του (με όνομα CountR), το οποίο για την Ariel, με id = 2, θα ήταν 107. Όποτε θα προσθέταμε ένα νέο rating, θα ενημερώναμε το AGV\_Rating με τον παρακάτω τύπο:

(CountR \* AVG\_Rating + inserted.rating) / (CountR + 1)

Με αυτόν τον τύπο, μπορούμε να ξαναβρούμε τον μέσο όρο χωρίς να χρειαστεί να απευθυνθούμε στις προηγούμενες εγγραφές του Rating. Στο τέλος αυξάνεται και το CountR κατά 1, για να μπορεί να χρησιμοποιηθεί στην επόμενη εισαγωγή. Με την χρήση του τροποποιημένου αυτού τύπου προσθέτουμε μόνο 1 παραπάνω column για αποθήκευση των απαιτούμενων δεδομένων, με μοναδικό έξτρα κόστος το κόστος του πολλαπλασιασμού.