

## 2η Εργασία: Εξερευνώντας τα MovieLens Δεδομένα

**Προθεσμία: 25/11/2024**

### Σκοπός:

Σε αυτή την εργασία θα εξερευνήσουμε τα δεδομένα του MovieLens με ερωτήματα σε SQL.

### Προαπαιτούμενα:

Θα πρέπει να έχετε δημιουργήσει τη βάση δεδομένων που περιγράφεται στην 1η εργασία και να έχετε εισάγει στους πίνακες τα MovieLens δεδομένα.

### Τι θα φτιάξουμε:

- 12 SQL ερωτήματα που θα περιλαμβάνουν `inner join`, `outer join`, `where`, `order by`, `group by`, `limit` καθώς και χρησιμοποίηση των συναρτήσεων συνάθροισης `min`, `max`, `avg`, της λέξης κλειδί `distinct`, καθώς και χρησιμοποίηση των τελεστών σύγκρισης `like`, `between`.
- Κάθε πίνακας εκ των (i) Movie, (ii) Genre, (iii) Keyword, (iv) Movie\_cast, (v) hasGenre, (vi) hasKeyword, (vii) Ratings θα πρέπει να χρησιμοποιηθεί σε τουλάχιστον ένα ερώτημα.
- Κάθε ερώτημα θα πρέπει να συνοδεύεται από μια μικρή περιγραφή που θα εξηγεί ποιος είναι ο σκοπός του δηλαδή τι ζητάμε. Επίσης θα συνοδεύεται και από το πλήθος των εγγραφών που επεστράφησαν ως αποτέλεσμα.
- Τουλάχιστον 8 ερωτήματα θα πρέπει να περιέχουν ένα τουλάχιστον `join`.
- Τουλάχιστον 2 ερωτήματα θα πρέπει να περιέχουν ένα τουλάχιστον `outer join`.
- Τουλάχιστον 2 ερωτήματα τα οποία θα περιέχουν Εμφωλευμένα Ερωτήματα και συνθήκες που χρησιμοποιούν τα `IN`, `ALL`, `ANY`, `EXISTS`.

### Εργαλεία:

- PGAdmin

### Οδηγίες:

- Τοποθετήστε όλα τα SQL ερωτήματα σε ένα αρχείο με όνομα `simple_queries.sql`
- Προσθέστε τις σύντομες περιγραφές των ερωτημάτων και τα πλήθη των αποτελεσμάτων στο ίδιο αρχείο με τη μορφή σχολίων. Π.χ:  

```
/* "Βρες μου τους τίτλους των ταινιών με μέσο όρο βαθμολογίας από  
χρήστες μεγαλύτερο του 4, μαζί με τον μέσο όρο βαθμολογίας τους"  
Output: 205 rows  
*/  
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
```

## Συμβουλές για την υλοποίηση:

- Σε περίπτωση που μια στήλη δεν έχει τον επιθυμητό τύπο δεδομένων, θα πρέπει να αλλάξουμε τον τύπο δεδομένων της χρησιμοποιώντας την εντολή ALTER TABLE/ALTER COLUMN. Π.χ. για να μετατρέψω το πεδίο rating του πίνακα ratings σε float (από TEXT), χρησιμοποιώ την εντολή:

```
ALTER TABLE ratings
ALTER COLUMN rating float ;
```

- Τρέξτε και ελέγξτε κάθε ερώτημα στην MovieLens βάση σας.
- Επιβεβαιώστε ότι κανένα ερώτημα δεν είναι άνευ ουσίας όσον αφορά την εξερεύνηση των δεδομένων με την έννοια ότι δεν είναι απλή εμφάνιση κάποιου πίνακα. Το ζητούμενο είναι να υπάρχει συνδυασμός κριτηρίων ώστε να εξαχεται κάποια γνώση. αντι-π.χ: **select \* from movie;** ή **select \* from movie where id="123";**.
- Επίσης, οι απαντήσεις στα ερωτήματα θα πρέπει να είναι σε μορφή κατανοητή από έναν κινηματογραφόφιλο, π.χ. Ένα ερώτημα που επιστρέφει *“το id των ταινιών με μέσο όρο βαθμολογίας μεγαλύτερο του 4”* δεν θα θεωρηθεί σωστό. Αντίθετα, θεωρείται σωστό το ερώτημα το οποίο μας επιστρέφει *“τους τίτλους των ταινιών με μέσο όρο βαθμολογίας μεγαλύτερο του 4”*.

## Παραδοτέα:

1. Βάλτε το αρχείο simple\_queries.sql και το αρχείο .txt σε ένα φάκελο. Δημιουργήστε ένα .zip αρχείο αυτού του φακέλου, το οποίο θα έχει το ίδιο όνομα με τον φάκελο.
2. Κάντε υποβολή το .zip αρχείο στο eclass στην ενότητα *Εργασίες /2η Εργασία*.