3η Εργασία: Εξερευνώντας τα MovieLens Δεδομένα

Προθεσμία: 15/4/2022

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 2η 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 **`min`**, **`max`**, **`avg`**, της λέξης κλειδί **`distinct`**, καθώς και χρησιμοποίηση των τελεστών σύγκρισης **`like`**, **`between`**.
- Κάθε πίνακας εκ των (i) Movie, (ii) Genre, (iii) Keywords, (iv) Movie_cast, (v) Movie_Genres, (vi) Movie_Keywords, (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`**.

Εργαλεία:

- Postgres Cloud SQL instance
- Postgres psql ή/και 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:

- 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 εξάγεται 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"*.
- Χρησιμοποιήστε την εντολή **`\i <full_path/filename>`** στην psql για να τρέξετε ένα SQL script. Για παράδειγμα **`\i /home/db_course/simple_queries.sql`**. Στο pgAdmin όπως έχουμε πει οι μετα-εντολές (δηλαδή οι εντολές που ξεκινούν με '\') δεν λειτουργούν. Το pgAdmin, βέβαια, φορτώνει και εκτελεί sql scripts κανονικά.
- Επιβεβαιώστε ότι ο χρήστης της βάσης του οποίου μας στέλνετε τα credentials όντως έχει πρόσβαση στη βάση σας.

Χρήσιμα links:

Εντολή select:
https://www.postgresql.org/docs/9.6/static/sql-select.html

Παραδοτέα:

1. Δημιουργήστε ένα .txt αρχείο στο οποίο θα αναγράφονται το endpoint του Azure instance σας (Server name στο Overview tab του Azure), το όνομα της βάσης σας και το username και το password ενός χρήστη με read-only δικαιώματα, ώστε να μπορούμε να δούμε τους πίνακες της βάσης σας. Το .txt αρχείο θα πρέπει να έχει την παρακάτω μορφή:

Endpoint: <name_of_the_endpoint>
Username: <username>

Password: <password>
Database: <name_of_the_database>

2. Βάλτε το αρχείο simple_queries.sql και το αρχείο .txt σε ένα φάκελο. Το όνομα του φακέλου πρέπει να αποτελείται από τους αριθμούς μητρώου σας χωρισμένους με παύλα, δηλαδή *αριθμός_μητρώου_1-αριθμός_μητρώου_2*. Δημιουργήστε ένα .zip αρχείο αυτού του φακέλου, το οποίο θα έχει το ίδιο όνομα με τον φάκελο.
3. Κάντε υποβολή το .zip αρχείο στο eclass στην ενότητα *Εργασίες / 3η Εργασία*.