

4η Εργασία: Επανασχεδιασμός της Βάσης μας

Προθεσμία: Παρασκευή 20 Μαΐου

Σκοπός:

Κάνουμε αλλαγές στον σχεδιασμό της βάσης μας ώστε σε επόμενο βήμα να κανονικοποιηθεί και να αποφευχθούν ανωμαλίες. Στα πλαίσια αυτά θα πρέπει να:

- Καθαρίσουμε τα δεδομένα μας από θόρυβο και αχρείαστη πληροφορία.
- Απεικονίσουμε το σχήμα της βάσης σε ένα διάγραμμα *οντοτήτων-συσχετίσεων* (ER).
- Να επεκτείνουμε το αρχικό μας σχήμα έτσι ώστε να περιέχει *πρωτογενή κλειδιά* για όλους τους πίνακες.

Προαπαιτούμενα:

Θα πρέπει να έχετε δημιουργήσει τη βάση δεδομένων που περιγράφεται στην 2η εργασία και να έχετε εισάγει στους πίνακες τα δεδομένα ταινιών.

Ζητούμενα Εργασίας:

Στην συγκεκριμένη Εργασία κάνουμε αλλαγές στην βάση μας ξεκινώντας από την κατάσταση που είχε όπως την ορίσαμε στην Εργασία 2. Στόχος μας είναι να δημιουργηθούν τα πρωτογενή κλειδιά για όλους τους πίνακες καθώς και το αντίστοιχο διάγραμμα οντοτήτων συσχετίσεων. Τα βήματα της εργασίας είναι τα ακόλουθα:

Καθάρισμα των Πινάκων

1. Να διαγραφούν οι διπλότυπες εγγραφές από τους πίνακες Movie_Cast, Movie_Collection, Movie_Crew, Movie_Genres, Movie_Keywords, Movie_ProductionCompanies. Προκειμένου να μην είναι πολύπλοκη η διαδικασία διαγραφής χρησιμοποιήστε την ακόλουθη εντολή βάζοντας στο πεδίο TableName το όνομα του αντίστοιχου πίνακα:

```
DELETE FROM TableName
WHERE ctid IN (
    SELECT t1.ctid
    FROM TableName t1
    NATURAL JOIN TableName t2
    WHERE t1.ctid>t2.ctid)
```

2. Χρησιμοποιώντας την εντολή UPDATE στον πίνακα Movie_Cast, να αλλάξουν οι τιμές του γνωρίσματος Character από NULL σε 'UNKNOWN'. Γιατί χρειάζεται η συγκεκριμένη αλλαγή ως προς το **Βήμα 6**;

Περιορισμοί Πινάκων

Να γραφούν τα SQL ερωτήματα τα οποία επιβεβαιώνουν τις ακόλουθες επιθυμητές ιδιότητες της βάσης μας:

3. Στον πίνακα Movie_Cast, το γνώρισμα person_id **καθορίζει συναρτησιακά** τα γνώρισμα name και gender. Δηλαδή, δεν μπορούμε να έχουμε δύο ηθοποιούς με το ίδιο person_id και διαφορετικό name ή gender.

Ιδέα: Δημιουργήστε ένα ερώτημα το οποίο να επιστρέφει όλα τα person_ids που παραβιάζουν αυτήν την συνθήκη. Σε περίπτωση που το συγκεκριμένο ερώτημα έχει κενή απάντηση, τότε ισχύει η ιδιότητά που ψάχνουμε.

Βοηθητικά: Η επίλυση να γίνει χρησιμοποιώντας τις εντολές GROUP BY και HAVING μαζί με την συνάρτηση συνάθροισης COUNT(DISTINCT ΓΝΩΡΙΣΜΑ). Η παράμετρος DISTINCT χρησιμοποιείτε προκειμένου να μην μετρώνται επαναλαμβανόμενα στοιχεία του ΓΝΩΡΙΣΜΑΤΟΣ πάνω από 1 φορά.

4. Στον πίνακα Movie_Crew, το γνώρισμα person_id **καθορίζει συναρτησιακά** τα γνώρισμα name και gender.

Δημιουργία ER Διαγράμματος

5. Να δημιουργηθεί το διάγραμμα *οντοτήτων-συσχετίσεων* για την βάση δεδομένων που monielens. Στο διάγραμμα οντοτήτων-συσχετίσεων να εμφανίζονται τα αντίστοιχα γνώρισμα κλειδιά υπογραμμισμένα, ενώ σε πίνακες με μεγάλο πλήθος γνωρισμάτων, να σχεδιαστούν ενδεικτικά κάποια από τα εναπομείναντα γνώρισμα.
- Για το διάγραμμα οντοτήτων συσχετίσεων, να θεωρήσετε επιπλέον την οντότητα user που περιγράφει χρήστες οι οποίοι βαθμολόγησαν ταινίες.

Δημιουργία Κλειδιών

6. Χρησιμοποιώντας την εντολή ALTER TABLE, να δημιουργηθούν πρωτογενή κλειδιά για τους ακόλουθους πίνακες της βάσης: Movie_Collection, Movie_Crew, Movie_Cast, Movie_Genres, Movie_Productioncompanies, Movie_Keywords.

Σε περίπτωση που τα προαναφερθέντα κλειδιά είχαν δημιουργηθεί στην δεύτερη εργασία, να δοθεί η αντίστοιχη εντολή CREATE που χρησιμοποιήθηκε για την δημιουργία τους.

Εργαλεία:

- Postgres Cloud SQL instance
- Postgres psql ή/και PgAdmin
- [Draw.io](https://draw.io)

Συμβουλές για την υλοποίηση:

- Επιβεβαιώστε ότι ο χρήστης της βάσης του οποίου μας στέλνετε τα credentials όντως έχει πρόσβαση στη βάση σας και μπορεί να τρέχει SELECT queries.
- Χρησιμοποιείτε ALTER TABLE ... ADD PRIMARY/FOREIGN KEY ...
- Για τον σχεδιασμό του ER προτείνεται η χρήση [Draw.io](https://draw.io).

Χρήσιμα links:

-DISTINCT: <https://www.postgresql.org/docs/9.6/static/sql-select.html#SQL-DISTINCT>

-ALTER TABLE: <https://www.postgresql.org/docs/9.6/static/sql-altertable.html>

- SELECT: <https://www.postgresql.org/docs/9.6/static/sql-select.html>

Παραδοτέα:

- A. 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>

- B. Να φτιαχτεί το αρχείο `functional_dependencies.sql` το οποίο να περιέχει και να εξηγεί την λειτουργία των ερωτημάτων που φτιάχτηκαν στο **3ο και 4ο βήμα** της άσκησης.
 - C. Να φτιαχτεί το *ER διάγραμμα* της βάσης δεδομένων όπως περιγράφεται στο **5ο βήμα** της άσκησης και να παραδοθεί σε μορφή pdf.
 - D. Για κάθε πίνακα από τους `Movie_Collection`, `Movie_Crew`, `Movie_Cast`, `Movie_Genres`, `Movie_Productioncompanies`, `Movie_Keywords` να δημιουργηθεί το αντίστοιχο sql αρχείο (π.χ. `movie_cast.sql`) το οποίο να περιέχει τις εντολές που άλλαξαν τον αντίστοιχο πίνακα προσθέτοντας πρωτογενή κλειδιά στο **6ο βήμα της άσκησης**. Μαζί με τον sql κώδικα να υπάρχει και η περιγραφή του.
- Βάλτε τα αρχεία sql, το αρχείο εικόνας και το αρχείο .txt σε έναν 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.
 - Κάντε υποβολή το .zip αρχείο στο eclass στην ενότητα *Εργασίες / 4η Εργασία*.