ΚΑΡΡΑΣ ΚΩΝΣΤΑΝΤΙΝΟΣ 3180076

ΠΕΠΠΑ ΧΡΙΣΤΙΝΑ 3180154

ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ PROJECT

ΜΕΡΟΣ Α

Αρχικά δημιουργήσαμε τους πίνακες ορίζοντας μόνο τους τύπους των μεταβλητών χωρίς να θέσουμε σε κάποιον primary key(εκτός από τον ratings ο οποίος δεν χρειαζόταν κάποια επεξεργασία στα δεδομένα ώστε να οριστεί αργότερα primary key).

❖ ΠΙΝΑΚΕΣ

- Movies_metadata

Δημιουργήσαμε τον πίνακα χωρίς primary key. Ύστερα, κάναμε import το .csv αρχείο από την γραμμή εντολών.

```
project1=> \copy Movies_metadata FROM 'C:/Users/Kostas/Desktop/project1/movies_metadata.csv' DELIMITER ',' CSV HEADER ENCODING 'UTF8';
```

Στην συνέχεια, αντιγράψαμε τα μοναδικά στοιχεία σε έναν προσωρινό πίνακα (temp). Διαγράψαμε τα πάντα από τον Movies_metadata και τα ξανακάναμε insert(στον Movies_metadata) από τον προσωρινό. Ωστόσο, είχαν μείνει 13 διπλές εγγραφές που είχαν το ίδιο id και διαφορετικό popularity(όλα τα υπόλοιπα πεδία ήταν τις περισσότερες φορές ίδια (2 εγγραφές διέφεραν στο τελευταίο όρισμα που ήταν το vote_count)) με αποτέλεσμα να μην μπορούμε να βάλουμε primary key(μπορούσαμε να βάλουμε το id,popularity,vote_count αλλά αυτό θα μας εμπόδιζε ύστερα στα foreign keys). Αυθαίρετα αποφασίσαμε να κρατήσουμε αυτές που είχαν το μεγαλύτερο popularity θεωρώντας πως είναι πιο πρόσφατα ενημερωμένες. Αυτό το κάναμε πάλι χρησιμοποιώντας έναν προσωρινό πίνακα (temp) στον οποίο αυτή τη φορά αντιγράψαμε μόνο τις 13 αυτές εγγραφές. Τέλος, διαγράψαμε από τον Movies_metadata τις εγγραφές που είχαν το ίδιο id και μικρότερο popularity από τον temp, κάναμε drop τον temp και ορίσαμε primary key το id στον Movies_metadata.

- Credits

Δημιουργήσαμε τον πίνακα χωρίς primary key. Ύστερα, κάναμε import το .csv αρχείο από την γραμμή εντολών.

```
project1=> \copy Credits FROM 'C:/Users/Kostas/Desktop/project1/credits.csv' DELIMITER ',' CSV HEADER ENCODING 'UTF8';
```

Όπως και στον Movies_metadata, αντιγράψαμε τα μοναδικά στοιχεία σε έναν προσωρινό πίνακα (temp). Διαγράψαμε τα πάντα από τον Credits και τα ξανακάναμε insert(στον Credits) από τον προσωρινό(κάναμε drop τον temp). Ωστόσο, αυτή τη φορά είχαν μείνει 7 διπλές εγγραφές που είχαν το ίδιο id και διαφορετικό crew(το cast ήταν ίδιο) με αποτέλεσμα πάλι να μην μπορούμε να ορίσουμε primary key. Τώρα, όμως, δεν κρατήσαμε 1 από τις 2 εγγραφές αλλά τις συγχωνεύσαμε. Συγκεκριμένα, αντιγράψαμε τα 7 αυτά στοιχεία του Credits σε 2 πίνακες(temp,temp2). Κάναμε Update στον temp το crew, δηλαδή κάναμε concatenate το temp.crew,temp2.crew όπου τα id τους είναι ίδια. Εδώ θέλουμε να γίνει κατανοητό πως η κατάσταση είναι η εξής:

ΠΡΙΝ Temp	ΜΕΤΑ Temp
[X]	[X][Y]
[Y]	[Y][X]

Άρα πάλι το αποτέλεσμα δεν είναι το επιθυμητό(να δημιουργηθούν δηλαδή διπλότυπα ώστε να πάρουμε distinct). Ξανακάνουμε Update τον temp ώστε να μην υπάρχουν τα ενδιάμεσα][. Επομένως:

ΠΡΙΝ Temp	ΜΕΤΑ Temp
[X][Y]	[XY]
[Y][X]	[YX]

Μετά, αντιγράψαμε από τον temp σε έναν άλλο προσωρινό πίνακα (temp3) ότι περιέχει και ο temp. Κάναμε Update στον temp το crew = temp3.crew οπότε:

ΠΡΙΝ ΤΟ UPDATE		ΚΑΤΑ ΤΟ UPDATE		ΜΕΤΑ ΤΟ UPDATE	
TEMP	TEMP3	TEMP	TEMP3	TEMP	TEMP3
[XY]	[XY]	[XY]	[XY]	[YX]	[XY]
[YX]	[YX]	[XY]	[YX]	[YX]	[YX]

Αυτό έχει ως αποτέλεσμα ο temp να περιέχει μόνο ίδιες εγγραφές(και μάλιστα χωρίς να έχει <<πεταχτεί>> κάτι) για το ίδιο id. Αντιγράφουμε σε έναν πίνακα temp4 τα μοναδικά στοιχεία του temp, κάνουμε delete τα πάντα από τον temp

και μετά εισάγουμε ξανά στον temp τα πάντα από τον temp4(περιττό εν μέρει διότι είναι το ίδιο με το να χρησιμοποιούσαμε στην συνέχεια τον temp4).
Επιπλέον, κάνουμε Update τον Credits:
Credits.crew = temp.crew WHERE Credits.id = temp.id;
και κάνουμε drop τους πίνακες temp,temp2, temp3, temp4.
Τώρα ο Credits έχει για εκείνες τις 7 διπλές γραμμές το ίδιο crew. Αντιγράφουμε σε έναν προσωρινό πίνακα (temp) μόνο τις μοναδικές εγγραφές του Credits, σβήνουμε τα πάντα από τον Credits, κάνουμε ξανά insert στον Credits από τον temp τα πάντα και σβήνουμε τον temp. Τώρα, πλέον, ορίζουμε primary key το id στον Credits.

- Ratings

Δημιουργήσαμε τον πίνακα με primary key το userId,movieId. Ύστερα, κάναμε import το .csv αρχείο από την γραμμή εντολών.

```
project1=> \copy Ratings FROM 'C:/Users/Kostas/Desktop/project1/ratings.csv' DELIMITER ',' CSV HEADER ENCODING 'UTF8';
```

Δεν κάναμε κάτι παραπάνω αφού δεν έπρεπε να διαγράψουμε τα διπλότυπα.

- Links

Δημιουργήσαμε τον πίνακα χωρίς primary key. Ύστερα, κάναμε import το .csv αρχείο από την γραμμή εντολών.

```
project1=> \copy Links FROM 'C:/Users/Kostas/Desktop/project1/links.csv' DELIMITER ',' CSV HEADER ENCODING 'UTF8';
```

Στην συνέχεια, αντιγράψαμε τα μοναδικά στοιχεία σε έναν προσωρινό πίνακα temp, διαγράψαμε τα πάντα από τον Links, κάναμε ξανά insert τα πάντα από τον temp στον Links και κάναμε drop τον temp. Τέλος, ορίσαμε primary key το movieId.

- Keywords

Δημιουργήσαμε τον πίνακα χωρίς primary key. Ύστερα, κάναμε import το .csv αρχείο από την γραμμή εντολών.

```
project1=> \copy Keywords FROM 'C:/Users/Kostas/Desktop/project1/keywords.csv' DELIMITER ',' CSV HEADER ENCODING 'UTF8';
```

Στην συνέχεια, αντιγράψαμε τα μοναδικά στοιχεία σε έναν προσωρινό πίνακα temp, διαγράψαμε τα πάντα από τον Keywords, κάναμε ξανά insert τα πάντα από τον temp στον Keywords και κάναμε drop τον temp. Τέλος, ορίσαμε primary key το id.

❖ ΔΙΑΓΡΑΦΕΣ ΑΠΟ ΑΛΛΟΥΣ ΠΙΝΑΚΕΣ ΠΛΕΙΑΔΩΝ ΠΟΥ ΔΕΝ ΥΠΑΡΧΟΥΝ ΣΤΟΝ MOVIES_METADATA

Παρατηρούμε ότι

Movies_metadata.id = Credits.id = Keywords.id = Links.tmdbId = Ratings.movieId (1)

και για κάθε ένα πίνακα εκτελούμε από ένα query της μορφής:

Διέγραψε από τον εκάστοτε πίνακα τις πλειάδες όπου δεν περιέχονται στον Movies_metadata βάσει της σχέσης (1).

❖ FOREIGN KEYS

Ορίζουμε τα foreign keys βάσει της σχέσης (1) όπου όλοι οι πίνακες έχουν reference στον Movies_metadata.

----------ΤΕΛΟΣ ΜΕΡΟΣ Α----------

ΜΕΡΟΣ Β

Για το μέρος Β δημιουργήσαμε 2 βοηθητικούς πίνακες για τα 2 πρώτα ερωτήματα. Συγκεκριμένα, για το πρώτο ερώτημα φτιάξαμε έναν πίνακα ονόματι firstquery και για τον δεύτερο secondquery. Προκειμένου να εμφανιστούν τα σωστά αποτελέσματα παραθέτουμε τα 2 SFW(τα οποία εννοείται πως υπάρχουν και στο partB.sql).

- **Για το 1ο:** SELECT * FROM firstquery;
- **Για το 2ο:** SELECT COUNT(id),secondquery.reg FROM Movies_metadata,secondquery WHERE Movies_metadata.genres LIKE CONCAT('%',secondquery.reg,'%') GROUP BY(secondquery.reg) ORDER BY(secondquery.reg);

Για όλα τα υπόλοιπα ερωτήματα μπορείτε να ανατρέξετε στο partB.sql
Ο μόνος λόγος που το αναφέρουμε είναι διότι, ενώ δεν είχαν ζητηθεί επιπλέον πίνακες αλλά ούτε είχαν απαγορευτεί βάσει της εκφώνησης, θέλουμε να καταστήσουμε σαφές ότι δεν πρόκειται για απλά SFW queries αλλά θα πρέπει να δημιουργηθούν πρώτα οι πίνακες και μετά να τρέξουν τα συγκεκριμένα επερωτήματα.

Πόρισμα του View table

Το πόρισμα που λαμβάνουμε από το View table είναι ότι μπορούμε να καταλάβουμε πόσο αυστηρός είναι ο κάθε κριτής και πόσο η βαθμολογία έκαστου κριτή συμμετέχει στη μέση βαθμολογία της αντίστοιχης ταινίας(ανάλογα με τον αριθμό των κριτικών).

----------ΤΕΛΟΣ ΜΕΡΟΣ Β----------