

Report: SQL Queries for MovieLens Data Exploration

Stylianos Giagkos f3352410

Εισαγωγή

Το παρόν αρχείο περιλαμβάνει 12 SQL ερωτήματα που πληρούν τις ακόλουθες προϋποθέσεις:

Προϋποθέσεις

1. **Χρήση όλων των πινάκων:** Οι πίνακες `Movie`, `Genre`, `Keyword`, `Movie_cast`, `HasGenre`, `HasKeyword`, και `Ratings` χρησιμοποιούνται τουλάχιστον μία φορά.
2. **Τουλάχιστον 8 ερωτήματα περιέχουν JOIN :** Συμπεριλαμβάνονται `INNER JOIN` και `LEFT JOIN`.
3. **Τουλάχιστον 2 ερωτήματα περιέχουν OUTER JOIN :** Χρησιμοποιείται `LEFT JOIN`.
4. **Τουλάχιστον 2 ερωτήματα περιέχουν εμφωλευμένα ερωτήματα με IN , ALL , ANY ή EXISTS .**
5. **Κανένα ερώτημα δεν είναι απλή εμφάνιση δεδομένων:** Όλα τα ερωτήματα παρέχουν ουσιαστική πληροφορία.
6. **Τα αποτελέσματα είναι κατανοητά από έναν κινηματογραφόφιλο:** Επιστρέφουν δεδομένα όπως τίτλοι ταινιών, μέσοι όροι βαθμολογιών, ηθοποιοί, κ.λπ.

SQL Queries

```
-- Ερώτημα 1
/* Βρες τους τίτλους των ταινιών που έχουν μέσο όρο βαθμολογίας μεγαλύτερο
από 4,
και εμφανίζουν τουλάχιστον 3 διαφορετικά είδη (genre) */
SELECT "m"."title", AVG("r"."rating") AS avgRating
FROM "Movie" "m"
INNER JOIN "Ratings" "r" ON "m"."id" = "r"."movie_id"
INNER JOIN "HasGenre" "hg" ON "m"."id" = "hg"."movie_id"
GROUP BY "m"."id", "m"."title"
HAVING AVG("r"."rating") > 4 AND COUNT(DISTINCT "hg"."genre_id") >= 3;
```

```
-- Ερώτημα 2
/* Βρες τις ταινίες του 2000 που έχουν βαθμολογία μεταξύ 3 και 5
και έχουν περισσότερα από 2 λέξεις-κλειδιά (keywords) */
SELECT "m"."title", AVG("r"."rating") AS avgRating
FROM "Movie" "m"
```

```
INNER JOIN "Ratings" "r" ON "m"."id" = "r"."movie_id"
INNER JOIN "HasKeyword" "hk" ON "m"."id" = "hk"."movie_id"
WHERE "m"."release_date" BETWEEN '2000-01-01' AND '2000-12-31'
GROUP BY "m"."id", "m"."title"
HAVING AVG("r"."rating") BETWEEN 3 AND 5 AND COUNT(DISTINCT
"hk"."keyword_id") > 2;
```

-- Ερώτημα 3

```
/* Βρες τους 5 ηθοποιούς με τις περισσότερες συμμετοχές
σε ταινίες με μέσο όρο βαθμολογίας μεγαλύτερο από 4 */
SELECT "mc"."name" AS actor_name, COUNT("mc"."movie_id") AS movie_count
FROM "Movie_Cast" "mc"
INNER JOIN "Ratings" "r" ON "mc"."movie_id" = "r"."movie_id"
GROUP BY "mc"."name"
HAVING AVG("r"."rating") > 4
ORDER BY movie_count DESC
LIMIT 5;
```

-- Ερώτημα 4

```
/* Βρες τους τίτλους των ταινιών που έχουν περισσότερες από 10 βαθμολογίες
και μέσο όρο βαθμολογίας μεγαλύτερο από τη μέση βαθμολογία όλων των
ταινιών του είδους 'Action' */
SELECT "m"."title", AVG("r"."rating") AS avgRating
FROM "Movie" "m"
INNER JOIN "Ratings" "r" ON "m"."id" = "r"."movie_id"
INNER JOIN "HasGenre" "hg" ON "m"."id" = "hg"."movie_id"
INNER JOIN "Genre" "g" ON "hg"."genre_id" = "g"."id"
GROUP BY "m"."id", "m"."title"
HAVING COUNT("r"."user_id") > 10
AND AVG("r"."rating") > (
    SELECT AVG("r"."rating")
    FROM "Ratings" "r"
    INNER JOIN "Movie" "m" ON "r"."movie_id" = "m"."id"
    INNER JOIN "HasGenre" "hg" ON "m"."id" = "hg"."movie_id"
    INNER JOIN "Genre" "g" ON "hg"."genre_id" = "g"."id"
    WHERE "g"."name" = 'Action');
```

-- Ερώτημα 5

```
/* Βρες τους τίτλους των ταινιών και το πλήθος των διαφορετικών λέξεων-
κλειδιών που έχουν,
όπου ο μέσος όρος βαθμολογίας τους είναι πάνω από 4 και το πλήθος των
ηθοποιών που συμμετέχουν σε αυτές είναι πάνω από 3 */
SELECT "m"."title", COUNT(DISTINCT "hk"."keyword_id") AS keyword_count
FROM "Movie" "m"
INNER JOIN "Ratings" "r" ON "m"."id" = "r"."movie_id"
INNER JOIN "HasKeyword" "hk" ON "m"."id" = "hk"."movie_id"
INNER JOIN "Movie_Cast" "mc" ON "m"."id" = "mc"."movie_id"
GROUP BY "m"."id", "m"."title"
HAVING AVG("r"."rating") > 4 AND COUNT(DISTINCT "mc"."name") > 3;
```

```
-- Ερώτημα 6
/* Βρες τις ταινίες του 2000 με μέσο όρο βαθμολογίας πάνω από 3
και που εμφανίζουν πάνω από 2 διαφορετικά είδη (genre) */
SELECT "m"."title", AVG("r"."rating") AS avgRating
FROM "Movie" "m"
INNER JOIN "Ratings" "r" ON "m"."id" = "r"."movie_id"
INNER JOIN "HasGenre" "hg" ON "m"."id" = "hg"."movie_id"
WHERE "m"."release_date" BETWEEN '2000-01-01' AND '2000-12-31'
GROUP BY "m"."id", "m"."title"
HAVING AVG("r"."rating") > 3 AND COUNT(DISTINCT "hg"."genre_id") > 2;
```

```
-- Ερώτημα 7
/* Βρες τις 10 ταινίες με τις περισσότερες βαθμολογίες από χρήστες
που έχουν συνδυασμό τουλάχιστον 2 είδη και 1 λέξη-κλειδί */
SELECT "m"."title", COUNT("r"."user_id") AS ratings_count
FROM "Movie" "m"
INNER JOIN "Ratings" "r" ON "m"."id" = "r"."movie_id"
INNER JOIN "HasGenre" "hg" ON "m"."id" = "hg"."movie_id"
INNER JOIN "HasKeyword" "hk" ON "m"."id" = "hk"."movie_id"
GROUP BY "m"."id", "m"."title"
HAVING COUNT(DISTINCT "hg"."genre_id") >= 2 AND COUNT(DISTINCT
"hk"."keyword_id") >= 1
ORDER BY ratings_count DESC
LIMIT 10;
```

```
-- Ερώτημα 8
/* Βρες τις ταινίες που έχουν βαθμολογία πάνω από 4
και εμφανίζουν τις περισσότερες λέξεις-κλειδιά σε σύγκριση με τις ταινίες
του 2000 */
SELECT "m"."title", COUNT(DISTINCT "hk"."keyword_id") AS keyword_count
FROM "Movie" "m"
INNER JOIN "Ratings" "r" ON "m"."id" = "r"."movie_id"
INNER JOIN "HasKeyword" "hk" ON "m"."id" = "hk"."movie_id"
GROUP BY "m"."id", "m"."title"
HAVING AVG("r"."rating") > 4 AND COUNT(DISTINCT "hk"."keyword_id") > (
    SELECT AVG(keyword_count)
    FROM (
        SELECT "m"."id", COUNT(DISTINCT "hk"."keyword_id") AS
keyword_count
        FROM "Movie" "m"
        LEFT JOIN "HasKeyword" "hk" ON "m"."id" = "hk"."movie_id"
        WHERE "m"."release_date" BETWEEN '2000-01-01' AND '2000-12-31'
        GROUP BY "m"."id"
    ) AS avg_keywords
);
```

```
-- Ερώτημα 9
/* Βρες τους ηθοποιούς που έχουν παίξει σε ταινίες με βαθμολογία πάνω από
4
και έχουν συμμετοχή σε περισσότερες από 5 ταινίες του είδους 'Action' */
```

```
SELECT "mc"."name" AS actor_name
FROM "Movie_Cast" "mc"
INNER JOIN "Movie" "m" ON "mc"."movie_id" = "m"."id"
INNER JOIN "Ratings" "r" ON "m"."id" = "r"."movie_id"
INNER JOIN "HasGenre" "hg" ON "m"."id" = "hg"."movie_id"
INNER JOIN "Genre" "g" ON "hg"."genre_id" = "g"."id"
GROUP BY "mc"."name", "g"."name"
HAVING AVG("r"."rating") > 4 AND "g"."name" = 'Action'
AND COUNT(DISTINCT "m"."id") > 5;
```

```
-- Ερώτημα 10
/* Βρες τα 5 πιο δημοφιλή είδη ταινιών (genre) με τις περισσότερες ταινίες
και με μέσο όρο βαθμολογίας πάνω από 3 */
SELECT "g"."name", COUNT("hg"."movie_id") AS movie_count
FROM "Genre" "g"
LEFT JOIN "HasGenre" "hg" ON "g"."id" = "hg"."genre_id"
INNER JOIN "Ratings" "r" ON "hg"."movie_id" = "r"."movie_id"
GROUP BY "g"."name"
HAVING AVG("r"."rating") > 3
ORDER BY movie_count DESC
LIMIT 5;
```

```
-- Ερώτημα 11
/* Βρες τις ταινίες που έχουν βαθμολογία πάνω από 3 και έχουν περισσότερα
από 3 διαφορετικά είδη και λέξεις-κλειδιά */
SELECT "m"."title", COUNT(DISTINCT "hg"."genre_id") AS genre_count,
COUNT(DISTINCT "hk"."keyword_id") AS keyword_count
FROM "Movie" "m"
INNER JOIN "Ratings" "r" ON "m"."id" = "r"."movie_id"
INNER JOIN "HasGenre" "hg" ON "m"."id" = "hg"."movie_id"
INNER JOIN "HasKeyword" "hk" ON "m"."id" = "hk"."movie_id"
GROUP BY "m"."id", "m"."title"
HAVING AVG("r"."rating") > 3 AND COUNT(DISTINCT "hg"."genre_id") > 3 AND
COUNT(DISTINCT "hk"."keyword_id") > 3;
```

```
-- Ερώτημα 12
/* Βρες τις ταινίες που έχουν συμμετοχή σε περισσότερα από 2 διαφορετικά
είδη */
SELECT "m"."title", COUNT(DISTINCT "hg"."genre_id") AS genre_count
FROM "Movie" "m"
LEFT JOIN "HasGenre" "hg" ON "m"."id" = "hg"."movie_id"
GROUP BY "m"."id", "m"."title"
HAVING COUNT(DISTINCT "hg"."genre_id") > 2;
```

In []: