

## 1<sup>η</sup> Εργασία Σχεδιασμός Βάσεων Δεδομένων:

Γεωργίου Αλέξιος-Λάζαρος 3180027

### Ζήτημα 1<sup>ο</sup> :

1:

Queries	Reads before Index	Reads after Index
Q1	1918	355
Q2	1918	355
Q3	2012	355

Έχουμε μετρήσει μόνο τα logical reads, τα υπόλοιπα είναι αμελητέα. Εφόσον έχουμε λιγότερα reads μετά το ευρετήριο, οι επερωτήσεις είναι πιο αποδοτικές.

Index:

```
CREATE INDEX idx  
ON movies(pyear ASC, title)
```

Εξήγηση:

Δημιουργώντας ένα ευρετήριο με τη χρονολογία και τον τίτλο, ταξινομούμε τις ταινίες με βάση την χρονολογία και τον τίτλο τους και έτσι μειώνουμε τους ελέγχους των cases που κάνουμε για να βρούμε τις ταινίες που συμβαδίζουν με τις συνθήκες “where”.

2:

Queries	Reads before Index (logical)	Reads after Index(logical)
Q1	2601	2232
Q2	2733	2346

Index:

```
CREATE INDEX idx  
ON user_movies(mid, rating)
```

Εξήγηση:

Ταξινομεί τον πίνακα με βάση το mid και είναι πιο αποδοτικό (πιο γρήγορο στον χρόνο εκτέλεσης).

## Ζήτημα 2° :

1:

Queries	Reads before Index (logical)	Reads after Index(logical)
Q1	>5500	-
Q2	2800	2500

Q1: επερώτηση εκφώνησης

Q2:

```
select distinct title
from movies, movies_genre
where movies.mid=movies_genre.mid and genre IN ('Adventure','Action')
```

Indexes:

```
CREATE INDEX idxA
ON movies(mid ASC, title)

CREATE INDEX idxA
ON movies_genre(mid ASC, genre)
```

Το Q1 ελέγχει δύο φορές κάθε στοιχείο, μια για κάθε genre και μετά ενώνει τα αποτελέσματα.

Βελτιώνουν την εύρεση των mid και την “σύνδεση” των δύο πινάκων στην επερώτηση.

2:

Q1

```
select DISTINCT title
from movies
join roles on roles.mid = movies.mid
where roles.mid IN(
    select roles.mid
    from roles
    EXCEPT
    select roles.mid
    from roles
    JOIN actors ON roles.aid=actors.aid
    WHERE actors.gender != 'M')
```

Q2

```
select title
from movies
JOIN roles ON movies.mid=roles.mid
JOIN actors ON roles.aid = actors.aid
group by title
having sum(case when gender != 'M' then 1 else 0 end) = 0
```

Queries	Reads before Index (logical)	Reads after Index(logical)
Q1	475000	475000
Q2	9500 (707ms)	4500 (650ms)

Το Q2 είναι πιο αποδοτικό αφού ψάχνουμε τον πίνακα μια φορά (με τα joins του) και ξέρουμε το αποτέλεσμα για κάθε ταινία, ενώ στο Q1 πρέπει να αναζητήσουμε για κάθε ρόλο αν βρίσκετε στους αποδεκτούς ρόλους (εμφωλευμένες επερωτήσεις)

Indexes:

```
CREATE INDEX idxmovies
ON movies(mid ASC, title)
```

```
CREATE INDEX idxroles
ON roles(aid, mid)
```

```
CREATE INDEX idxactors
ON actors(aid, gender)
```

Τα indexes είναι μια ταξινόμηση των πινάκων και στηλών χρησιμοποιούμε στο query.

### Ζήτημα 3° :

1:

Εμφάνιση όλων των χρηστών id, όνομα και πόσες βαθμολογίες έχουν δώσει σε ταινίες σε φθίνουσα σειρά.

-- USER NAMES WITH MOST VOTES

```
SELECT user_movies.userid, users.uname, count(user_movies.userid) as number_of_ratings
FROM user_movies
JOIN users ON user_movies.userid = users.userid
GROUP BY user_movies.userid, users.uname
ORDER BY number_of_ratings DESC
```

Sample results

	userid	uname	number_of_ratings
1	4169	Aryan Avery	2305
2	1680	Jerome Bentley	1826
3	4277	Stephany Hatfield	1733
4	1941	Amira Santos	1579
5	1181	Parker Gaines	1515
6	889	Jose McGuire	1503
7	3618	Melina Krause	1339
8	2063	Rose Lambert	1314
9	1150	Terrance Rose	1295
10	1015	Charles French	1281
11	5795	Olivia Bowers	1270
12	4344	Dereon Booker	1265
13	1980	Adrien Donaldson	1255
14	2909	Clarissa Haley	1251
15	1449	Wayne Bean	1239
16	4510	Adyson Walsh	1232
17	424	Carl Wilcox	1223
18	5831	Kailey Rivas	1217
19	3391	Cassandra Pitts	1212

Με αυτό το query μπορούμε να δούμε τα ονόματα των πιο ενεργών μελών.

-- Shows the id, title and rating of a movie in descending order. It removes movies with less than 1000 reviews to avoid showing unpopular movies with high rating

```
SELECT mid, title, mrank
FROM movies
WHERE mid IN(
    SELECT distinct user_movies.mid
    FROM user_movies
    GROUP BY user_movies.mid
    HAVING count(user_movies.mid) >= 1000)

ORDER BY mrank DESC
```

Sample result:

	mid	title	mrnk
1	227804	Narrow Trail, The	9.3
2	220347	Mountain Tops	8.5
3	188700	Lettres de mon moulin, Les	8.2
4	37232	Big Night	7.2
5	205969	Marion Bridge	6.7
6	141403	Heartbreak Kid, The	6.5
7	55370	Came	6.3
8	205584	Maria tis siopis, I	6.3
9	302235	Simple Twist of Fate, A	6.2
10	96148	Dmanty noci	6.2
11	224195	My Nightmare	6.0
12	296223	Sextrospective	5.9
13	358477	Waking Mele	5.8
14	83108	Derby	5.8
15	7395	Affair in Reno	5.8
16	92814	Dragon Seed	5.7

To query εμφανίζει τις καλύτερες ταινίες με βάση το rating που έχουν χωρίς να μετράει ταινίες με λίγες κριτικές (<1000).

2:

Queries	Reads before Index (logical)	Reads after Index(logical)
Q1	2800	1740
Q2	3200	2400

Indexes for Q1:

```
CREATE INDEX idxusers  
ON users(userid, uname)
```

```
CREATE INDEX idxusermovies  
ON user_movies(userid)
```

Indexes for Q2:

```
CREATE INDEX idxmovies  
ON movies(mid ASC, title, mrnk)
```

```
CREATE INDEX idxusermovies  
ON user_movies(mid)
```