

Συστήματα Διαχείρισης και Ανάλυσης Δεδομένων

Διδάσκων: Ιωάννης Κωτίδης

Εαρινό εξάμηνο 2024-2025

Ενδεικτικές Λύσεις Πρώτης Εργασίας

Λύση Πρώτου Ζητήματος

Δημιουργούμε ευρετήριο στο κριτήριο αναζήτησης.

```
create index ix_user_movies ON user_movies (rating)
```

Λύση Δεύτερου Ζητήματος

Δημιουργούμε ευρετήριο στο ζεύγος των πεδίων (pyear, title). Το ευρετήριο στο ζεύγος των πεδίων (pyear, title) επιταχύνει την αναζήτηση (μέσω φίλτρου στο pyear) και αποτρέπει την ανάγκη για επιπλέον ταξινόμηση στο τρίτο ερώτημα, καθώς καλύπτει πλήρως το ORDER BY pyear, title.

```
create index pyear_idx on movies(pyear,title)
```

Λύση Τρίτου Ζητήματος

Δημιουργούμε ευρετήριο στο ζεύγος των πεδίων (title, pyear). Το ευρετήριο επιταχύνει την αναζήτηση με τον τίτλο και υποστηρίζει την ταξινόμηση με το pyear.

```
CREATE INDEX ix_title ON movies(title,pyear);
```

Το ακόλουθο ευρετήριο επιταχύνει την αναζήτηση με φίλτρο στο πεδίο lastname και υποστηρίζει την ταξινόμηση βάσει του πεδίου firstname.

```
CREATE INDEX ix_director ON directors(lastname, firstname);
```

Το ευρετήριο στο ζεύγος των πεδίων (did,mid) διευκολύνει την αποδοτική σύζευξη (JOIN) μεταξύ των movie_directors και directors, επιτρέποντας γρήγορη εύρεση όλων των ταινιών για κάθε σκηνοθέτη.

```
CREATE INDEX ix_movie_director ON movie_directors (did,mid);
```

Λύση Τέταρτου Ζητήματος

A.

Το ευρετήριο στον πίνακα actors καλύπτει την αναζήτηση και την ταξινόμηση για το επερώτημα 1.

```
CREATE INDEX ix_actors_lastname_firstname ON actors(lastname, firstname)
```

Το index στον πίνακα roles βοηθά την σύζευξη(JOIN) και στα δύο επερωτήματα. Ειδικά για το δεύτερο επερώτημα αποφεύγεται η λειτουργία lookup στον πίνακα roles, καθώς το ευρετήριο καλύπτει όλα τα απαιτούμενα πεδία μέσω του INCLUDE.

```
CREATE INDEX ix_roles_aid ON roles (aid) include (a_role)
```

B.

```
SELECT actors.aid, firstname, lastname  
FROM actors  
WHERE NOT EXISTS (SELECT 1 FROM roles WHERE roles.aid = actors.aid);
```

Λύση Πέμπτου Ζητήματος

Λύση

A.

Το παρακάτω ευρετήριο επιταχύνει το join στο δεύτερο επερώτημα που έχει συχνότητα εκτέλεσης 80% διότι στο πρωτεύον κλειδί ο SQL SERVER δημιουργεί cluster index.

```
ALTER TABLE movies_genre  
ADD CONSTRAINT pk_mid_genre PRIMARY KEY CLUSTERED (mid,genre);
```

B.

Το ακόλουθο ευρετήριο επιταχύνει την αναζήτηση με κατηγορία καθώς και το JOIN στο πρώτο επερώτημα.

```
CREATE INDEX ix_movies_genre ON movies_genre(genre) include(mid)
```

Το ευρετήριο ix_title επιταχύνει την αναζήτηση με βάση τον τίτλο στο δεύτερο επερώτημα.

```
CREATE INDEX ix_title ON movies(title)
```