Συστήματα Διαχείρισης Βάσεων Δεδομένων

Μυρτώ-Χριστίνα Ελευθέρου, 3170046

Project 1

Ζήτημα 1ο

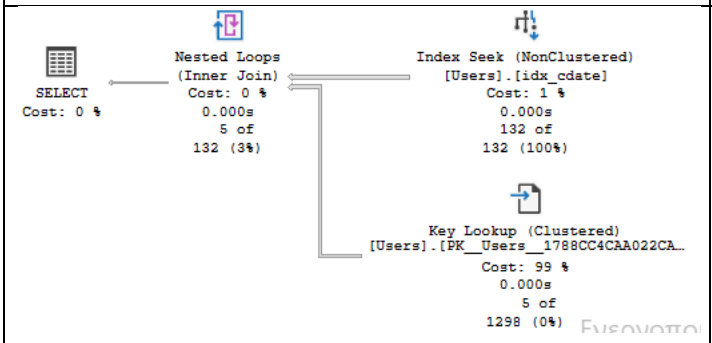
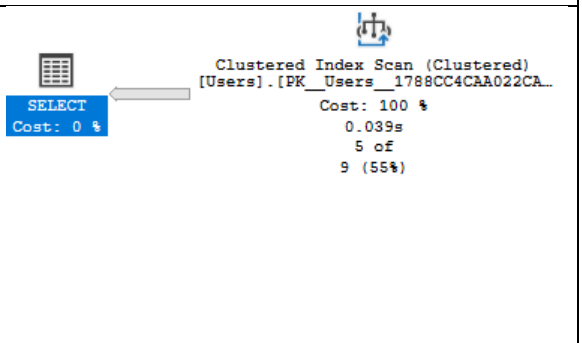
1) Θα επέλεγα το δεύτερο ευρετήριο, δηλαδή αυτό που αναφέρεται στην τοποθεσία του χρήστη. Στο συγκεκριμένο ερώτημα η τοποθεσία «Athens, Greece» θεωρώ ότι είναι πολύ πιο σπάνια από ότι η ημερομηνία «2008-09-15» καθώς μιλάμε για έναν παγκόσμιο ιστότοπο στον οποίο κάνουν εγγραφή καθημερινά πάρα πολλοί άνθρωποι από όλον τον κόσμο. Επομένως οι εγγραφές που αφορούν την τοποθεσία θα είναι κατά πάσα πιθανότητα πολύ λιγότερες από εκείνες της ημερομηνίας.

2) Χρησιμοποιώντας τα εργαλεία του Microsoft SQL Server Management Studio παρατηρούμε ότι το index βοηθάει σημαντικά στην ελαχιστοποίηση το χρόνου I/O εφόσον μειώνονται οι σελίδες που γίνονται load.

Σελίδες που κάνει load:

	With index	Without index
Logical reads	418	5716
Physical reads	0	2

Execution plans:

With index	Without Index
	

3) Ένα ευρετήριο που επιταχύνει ακόμα περισσότερο την εκτέλεση του ερωτήματος E1 είναι το εξής:

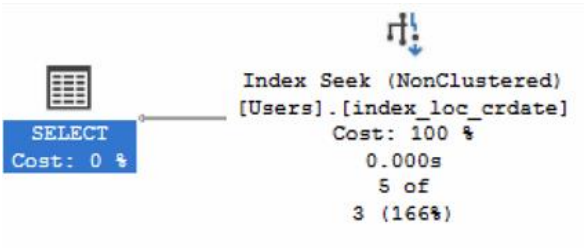
```
create index index_loc_crdate on users(userlocation, creationdate) include(displayname);
```

Συγκεκριμένα μέσω του ευρετηρίου αυτού μειώνεται κατά μεγάλο βαθμό ο χρόνος I/O. Επίσης αξίζει να αναφερθεί ότι το γνώρισμα userlocation βρίσκεται πρώτο έτσι ώστε να λειτουργεί με όσο λιγότερα outputs γίνεται.

Σελίδες που κάνει Load:

Logical reads	3
Physical reads	2

Execution Plan:



Ζήτημα 2ο

1) Τα indexes που χρησιμοποίησα για την βελτιστοποίηση του συγκεκριμένου ερωτήματος είναι:

```
create index index_tagname on Tags(tagname);

create index index_tagid on PostTags(tagid);
```

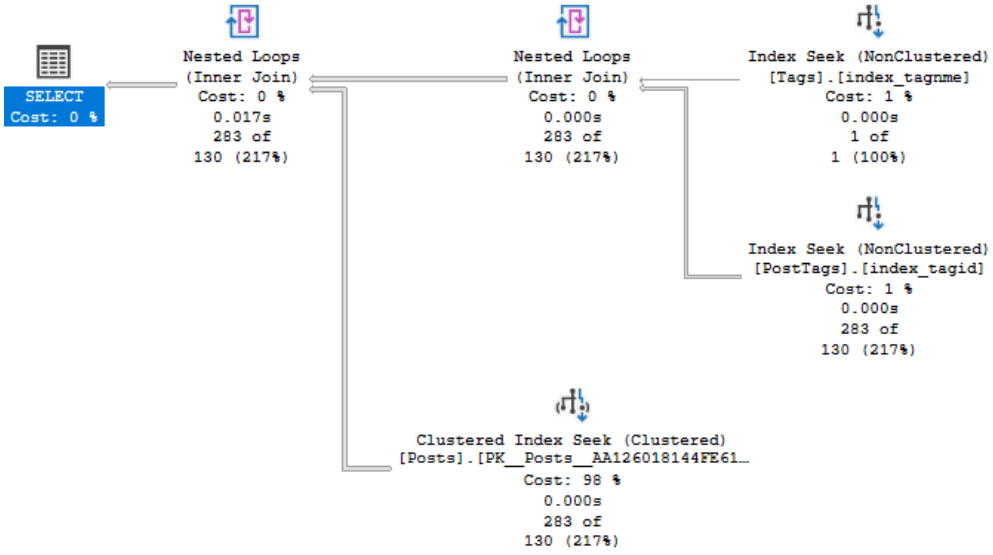
Φαίνεται ξεκάθαρα ότι με τα παραπάνω indexes φορτώνονται λιγότερες σελίδες . Επίσης μπορούμε να δούμε και από το πλάνο εκτέλεσης ότι το DBMS όντως χρησιμοποίησε τα index που δημιούργησα το οποίο δηλώνει ότι θεωρήθηκαν χρήσιμα κατά την διαδικασία εύρεσης καλύτερου πλάνου.

Σελίδες που κάνει load (Για κάθε πίνακα):

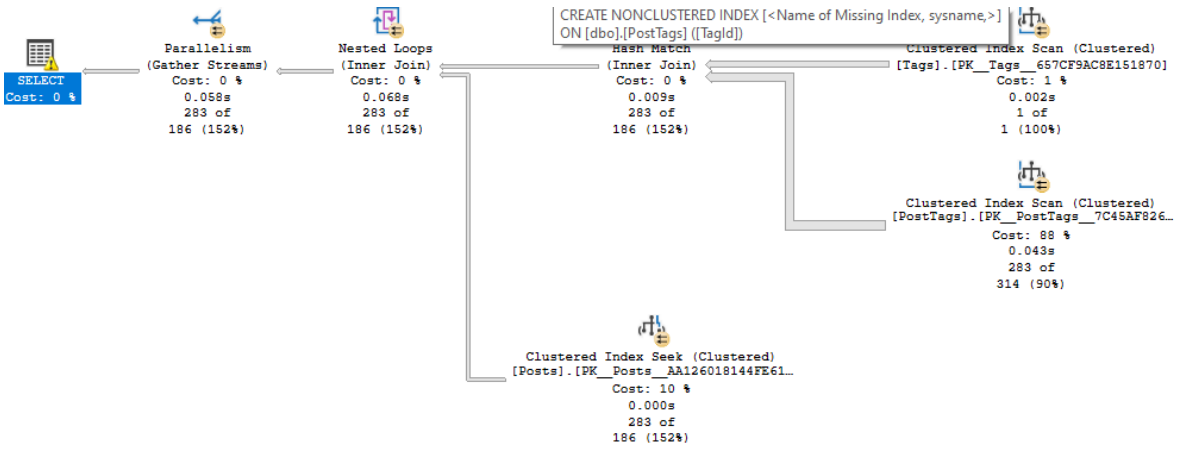
	With index	Without index
Logical reads	2163+4+2 = 2169	6740+349+2120 = 9209
Physical reads	1+0+2 =3	1+1+1 =3

Execution Plans:

With index



Without Index



2) Για το ερώτημα Ε3:

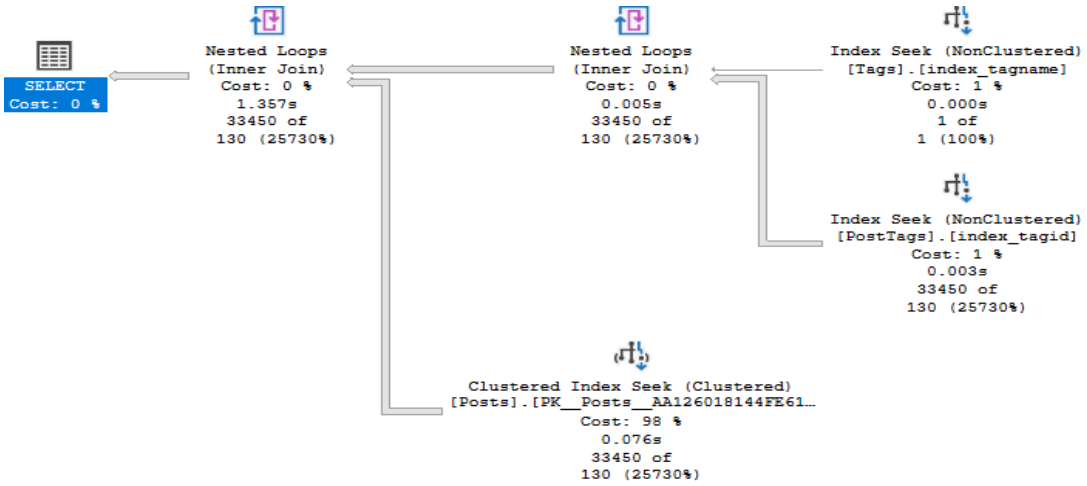
Το nested loop κάνει παραπάνω χρόνο να εκτελεστεί αφού είναι πάρα πολλές οι εγγραφές που αναφέρονται στην Sql σε σχέση με εκείνες του Informix.

Σελίδες που κάνει load (Για κάθε πίνακα):

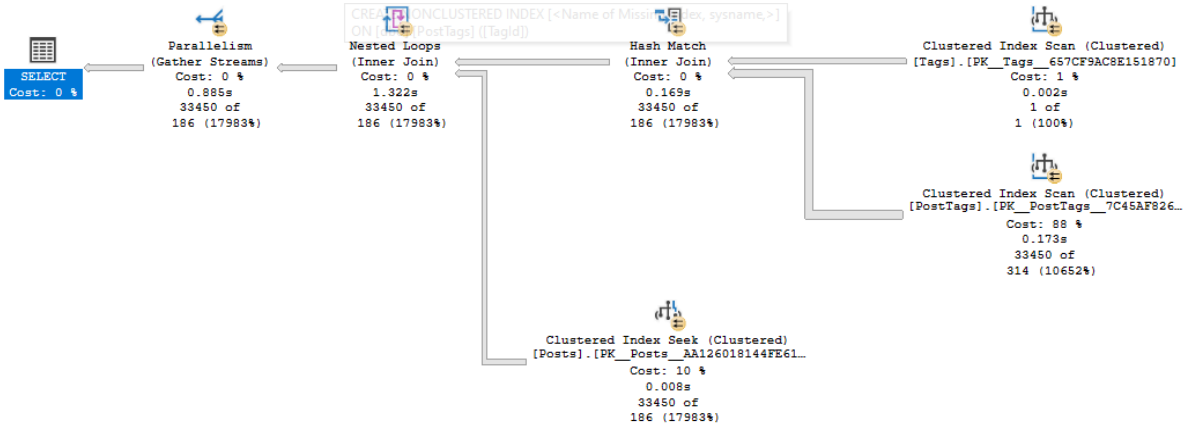
	With index	Without index
Logical reads	219572+62+2 = 219.636	6740+349+219031 = 226.120
Physical reads	1+0+3 = 4	1+1+1 = 3

Execution plans:

With index



Without index



Ζήτημα 3ο

"Εμφανίστε τον κωδικό (userid) και το όνομα (displayName) των χρηστών των οποίων όλες οι αναρτήσεις έχουν αρνητικό σκορ (score)."

1ος τρόπος

```
select distinct userId, displayName
from users, posts
where userId = ownerUserId and 0 > ALL(select score
from posts
where userId = ownerUserId)
```

Το ευρετήριο που θα χρησιμοποιούσα για τη βελτίωση του ερωτήματος είναι το εξής:

```
create index idx_score on posts(score) include(owneruserid);
```

Σελίδες που κάνει load (Για κάθε πίνακα):

	With index	Without index
Logical reads	16726+6001 = 22.727	747119+6001 = 753.120
Physical reads	0+2 = 2	2+2 = 4

Φαίνεται ότι το ευρετήριο βοηθάει πολύ στη μείωση του χρόνου καθώς τα reads μειώνονται κατά πολύ. Επίσης με το ευρετήριο το elapsed time είναι 238ms ενώ χωρίς το ευρετήριο είναι 2243ms.

2ος τρόπος

```
select userid, displayname
from users
inner join posts on posts.owneruserid = users.userid
where  0 > all (
select posts.score
from users u
inner join posts on u.userid = posts.owneruserid and users.userid = u.userid);
```

Τα ευρετήρια που θα χρησιμοποιούσα για τη βελτίωση του ερωτήματος είναι τα εξής:

```
create index idx on posts(score) include (owneruserid);
create index idx on users(userid) include(displayname);
```

Σελίδες που κάνει load (Για κάθε πίνακα):

	With indexes	Without index
Logical reads	16.726+2.396 = 19.122	747.079+12.002 = 759.081
Physical reads	0+1 = 1	2+2 = 4

Elapsed time:

With indexes: 227ms

Without indexes: 2236ms

Θα επέλεγα το πρώτο ερώτημα γιατί με τη χρήση ευρετηρίων τα στατιστικά και οι χρόνοι cpu είναι σχεδόν ίδιοι, ωστόσο στο πρώτο ερώτημα χρειάζεται μόνο ένα ευρετήριο ενώ στο δεύτερο ερώτημα χρειάζονται δύο, που σημαίνει μεγαλύτερο κόστος για τη δημιουργία τους.

Ζήτημα 4ο

Ερώτημα 1: Να εμφανιστεί το όνομα του χρήστη, του οποίου το σχόλιο έχει τις περισσότερες θετικές ψήφους. (index στο score του comments)

```
select users.displayname
from users, comments, posts
where Comments.score = (select max(comments.score) from comments)
and comments.postid = posts.postid and
posts.OwnerUserId = users.userid
group by comments.score, displayname
```

Στη συνέχεια δημιούργησα το παρακάτω ευρετήριο για την βελτιστοποίηση του ερωτήματος:

```
create index idx_comscore on Comments(score);
```

Σελίδες που κάνει load (Για κάθε πίνακα):

	With index	Without index
Logical reads	3+4+9 = 16	163380+3+4 = 163387
Physical reads	3+4+6 = 13	1+3+4 = 8

Είναι προφανές ότι στο συγκεκριμένο ερώτημα το ευρετήριο παίζει πολύ σημαντικό ρόλο, καθώς οι σελίδες που γίνονται Load σχεδόν μηδενίζονται.

Ερώτημα 2: Να εμφανίζονται όλες οι ερωτήσεις που έχει κάνει ένας χρήστης, καθώς και ο αριθμός των απαντήσεων που έχει λάβει η κάθε ερώτηση. (Πχ για τον χρήστη με κωδικό userid=’1’)

```
select posts.Title, AnswerCount
from posts, users
where AcceptedAnswerId is not null and ownerUserId = userid and userid = '1';
```

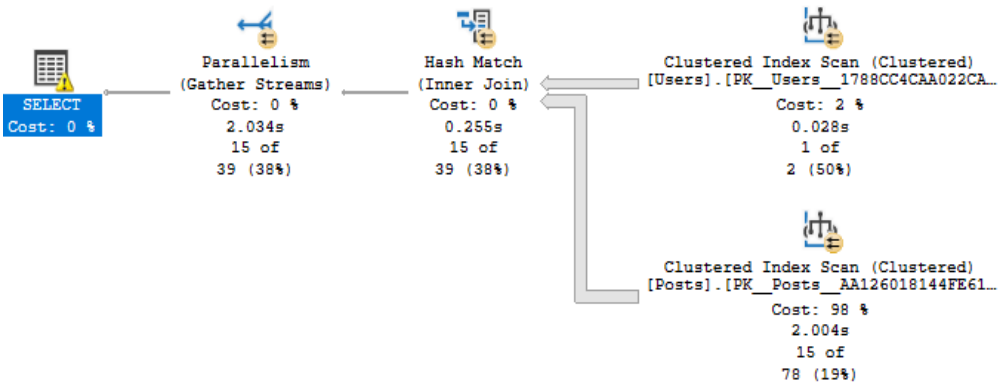
Στη συνέχεια δημιούργησα το παρακάτω ευρετήριο για την βελτιστοποίηση του ερωτήματος:

```
create index idx_owner on Posts(owneruserid);
```

Σελίδες που κάνει load (Για κάθε πίνακα):

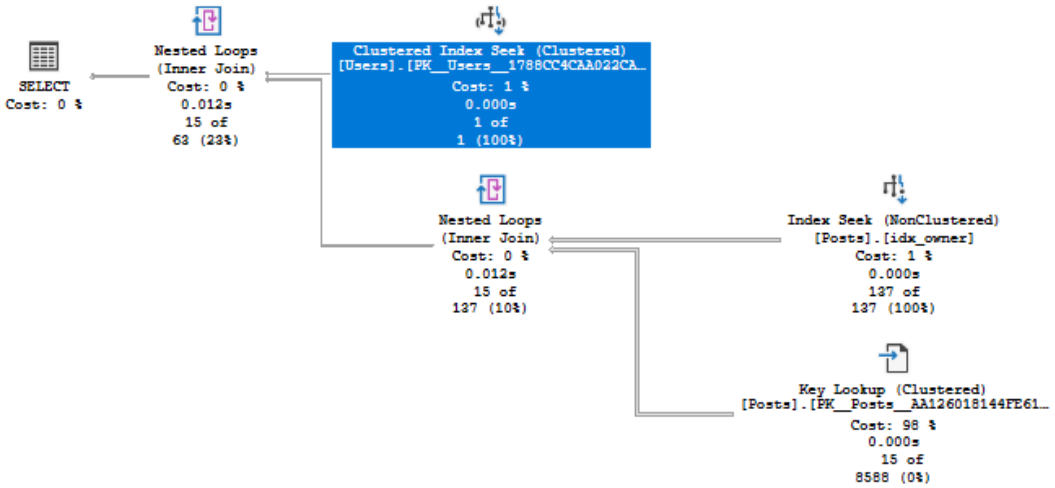
	With index	Without index
Logical reads	568+3 = 571	371927+3 = 371930
Physical reads	3+3 = 6	2+3=5

Without index (Elapsed time = 2026ms)



With index

Elapsed time = 34ms



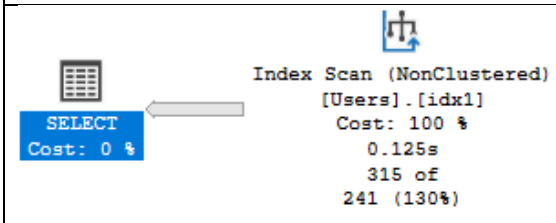
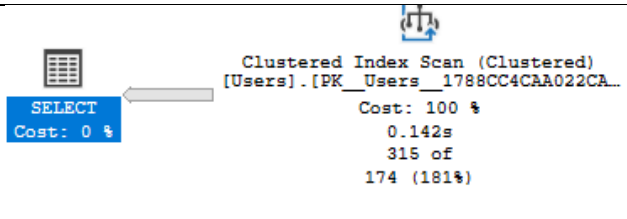
Ζήτημα 5ο

1)

Σελίδες που κάνει load:

	With index	Without index
Logical reads	1633	5716
Physical reads	0	2

Execution Plan

With index	Without Index
	

Όπως παρατηρούμε από την μείωση σελίδων φόρτωσης το ευρετήριο βελτιώνει το ερώτημα. Ωστόσο, πιθανώς να μην είναι η βέλτιστη λύση καθώς παρατηρούμε ότι γίνεται scan όλου του ευρετηρίου, κάτι το οποίο θα μπορούσε να αποφευχθεί ώστε να μειωθεί το κόστος.

2) Η ιδέα για την βελτιστοποίηση του ερωτήματος είναι να φτιάξουμε μία στήλη μόνο με τις χώρες, και στη συνέχεια να φτιάξουμε ευρετήριο για αυτή την στήλη.