ΣΧΕΔΙΑΣΜΟΣ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

Project Part A΄

Όνομα: **Παναγιώτης Ντυμένος**

Περιεχόμενα

Ζητούμενο Πρώτο

1.

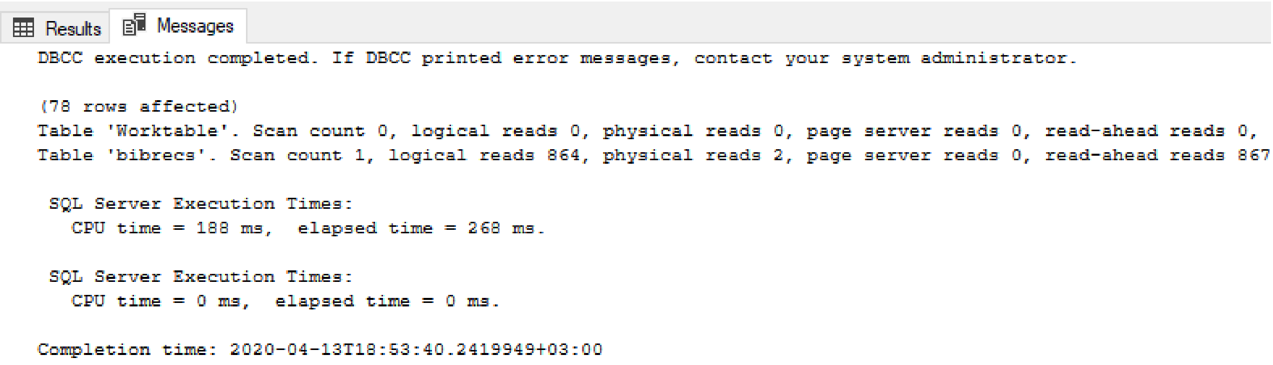
```
CREATE INDEX Indx ON Bibrecs(title);

SELECT title FROM Bibrecs WHERE title LIKE 'Οικ%'
ORDER BY title
```

2.

α.

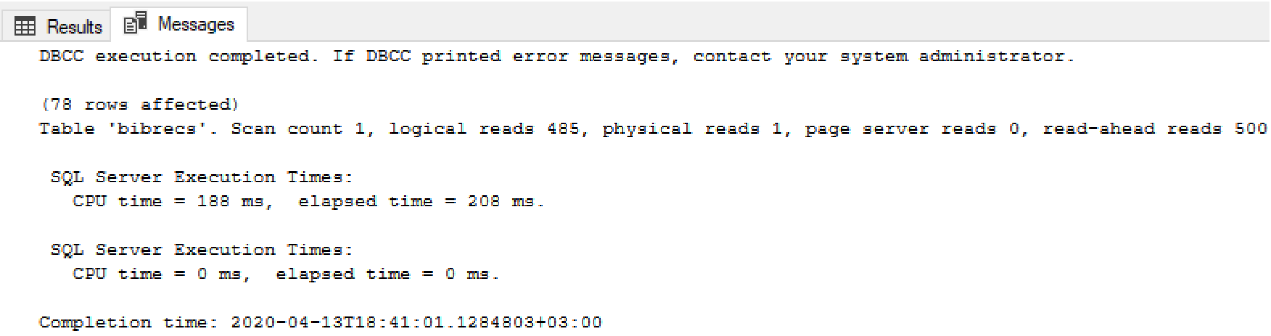
```
DBCC DROPCLEANBUFFERS
SET STATISTICS IO ON
SET STATISTICS TIME ON

SELECT title FROM Bibrecs WHERE title LIKE '%Πληροφορική%'
ORDER BY title

SET STATISTICS TIME OFF
SET STATISTICS IO OFF
```

Χωρίς Index:



```
DBCC execution completed. If DBCC printed error messages, contact your system administrator.

(78 rows affected)
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0,
Table 'bibrecs'. Scan count 1, logical reads 864, physical reads 2, page server reads 0, read-ahead reads 867

 SQL Server Execution Times:
   CPU time = 188 ms,  elapsed time = 268 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.

Completion time: 2020-04-13T18:53:40.2419949+03:00
```

Με Index:

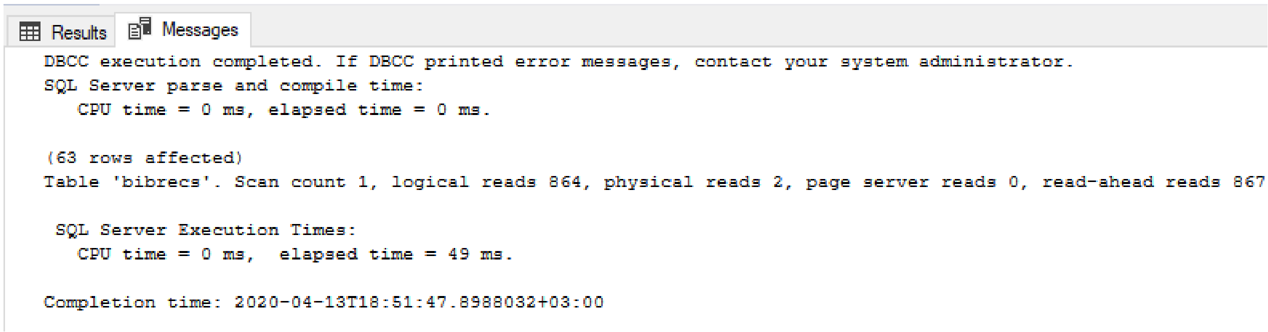


```
DBCC execution completed. If DBCC printed error messages, contact your system administrator.

(78 rows affected)
Table 'bibrecs'. Scan count 1, logical reads 485, physical reads 1, page server reads 0, read-ahead reads 500

 SQL Server Execution Times:
   CPU time = 188 ms,  elapsed time = 208 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.

Completion time: 2020-04-13T18:41:01.1284803+03:00
```

b.

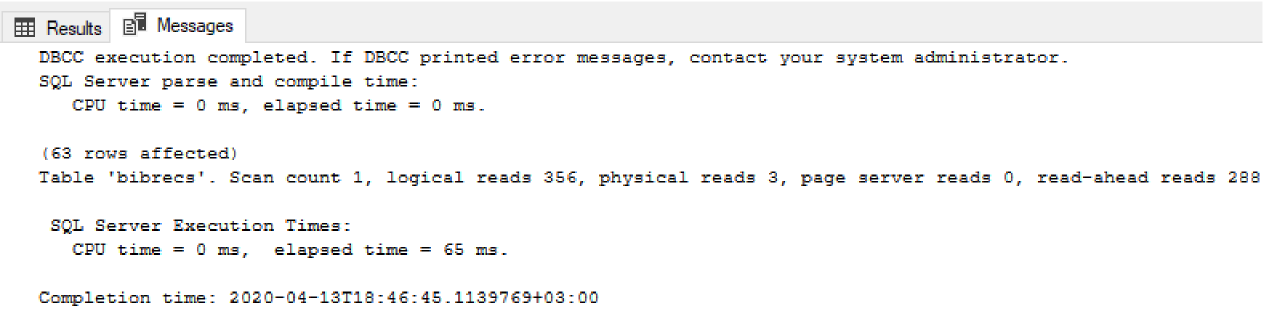
```
DBCC DROPCLEANBUFFERS
SET STATISTICS IO ON
SET STATISTICS TIME ON

SELECT title, material FROM Bibrecs WHERE title = 'Economics'
ORDER BY title

SET STATISTICS TIME OFF
SET STATISTICS IO OFF
```

Χωρίς Index:



```
DBCC execution completed. If DBCC printed error messages, contact your system administrator.
SQL Server parse and compile time:
   CPU time = 0 ms, elapsed time = 0 ms.

(63 rows affected)
Table 'bibrecs'. Scan count 1, logical reads 864, physical reads 2, page server reads 0, read-ahead reads 867

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 49 ms.

Completion time: 2020-04-13T18:51:47.8988032+03:00
```

Με Index:

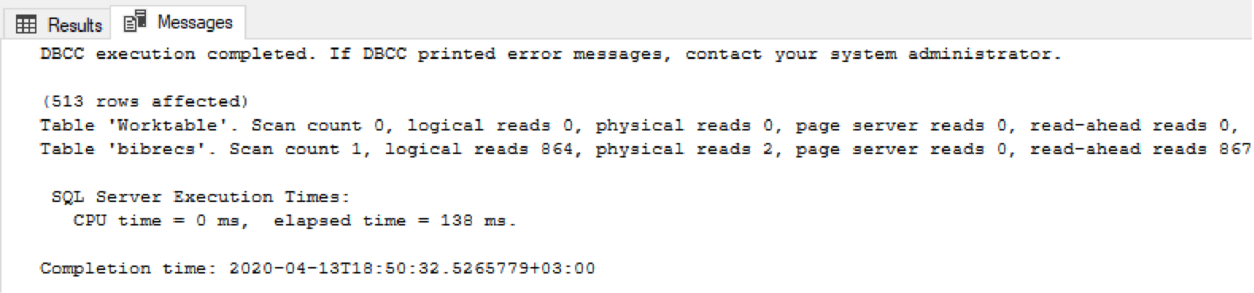


```
DBCC execution completed. If DBCC printed error messages, contact your system administrator.
SQL Server parse and compile time:
   CPU time = 0 ms, elapsed time = 0 ms.

(63 rows affected)
Table 'bibrecs'. Scan count 1, logical reads 356, physical reads 3, page server reads 0, read-ahead reads 288

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 65 ms.

Completion time: 2020-04-13T18:46:45.1139769+03:00
```

c.

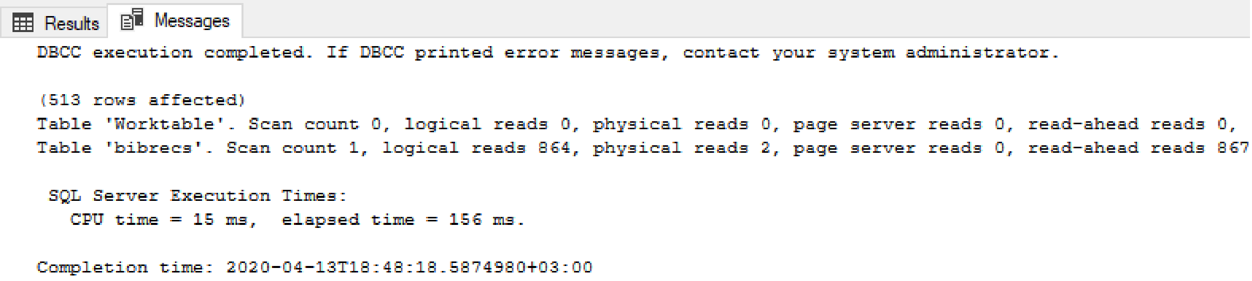
```
DBCC DROPCLEANBUFFERS
SET STATISTICS IO ON
SET STATISTICS TIME ON

SELECT title, material FROM Bibrecs WHERE title LIKE 'Economics%'
ORDER BY title

SET STATISTICS TIME OFF
SET STATISTICS IO OFF
```

Χωρίς Index:



```
DBCC execution completed. If DBCC printed error messages, contact your system administrator.

(513 rows affected)
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0,
Table 'bibrecs'. Scan count 1, logical reads 864, physical reads 2, page server reads 0, read-ahead reads 867

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 138 ms.

Completion time: 2020-04-13T18:50:32.5265779+03:00
```

Με Index:



```
DBCC execution completed. If DBCC printed error messages, contact your system administrator.

(513 rows affected)
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0,
Table 'bibrecs'. Scan count 1, logical reads 864, physical reads 2, page server reads 0, read-ahead reads 867

 SQL Server Execution Times:
   CPU time = 15 ms,  elapsed time = 156 ms.

Completion time: 2020-04-13T18:48:18.5874980+03:00
```

Αιτιολόγηση

Όπως βλέπουμε και στις παραπάνω φωτογραφίες, το ευρετήριο μας συμφέρει μόνο όταν θέλουμε να ανακτήσουμε εγγραφές από το πεδίο **title** (φυσικό εφόσον το ευρετήριο μου είναι πάνω σε αυτή την στήλη). Στα *b* και *c* ο χρόνος ήταν ελαφρώς καλύτερος *χωρίς* index. Ειδικά αν σκεφτούμε πως ο χρόνος που δείχνεται παραπάνω είναι χωρίς να έχω λάβει υπόψιν τον χρόνο της δημιουργίας του δείκτη. Αυτό που μου κάνει εντύπωση εμένα είναι το γεγονός ότι στο *b* γίνονται λιγότερες προσπελάσεις εγγραφών με index μα και πάλι έκανε παραπάνω χρόνο. Αν δεν έβλεπα τα στατιστικά του IO θα έλεγα πως σε αυτήν την περίπτωση δεν θα ήταν κακό να χρησιμοποιήσουμε ευρετήριο.

Ζητούμενο Δεύτερο

1.

a.

```
SELECT title, lang FROM Bibrecs
JOIN Publishers
ON Publishers.pubid = Bibrecs.pubid
WHERE Publishers.pubname = 'Κλειδάριθμος'
ORDER BY title
```

b.

```
SELECT depname, COUNT(Loanstats.lid) AS loans FROM Departments
LEFT JOIN Borrowers
ON Departments.depcode = Borrowers.depcode
JOIN Loanstats
ON Borrowers.bid = Loanstats.bid
WHERE Loanstats.loandate LIKE '2000%'
GROUP BY depname
```

c.

```
SELECT title, lang, author FROM Bibrecs
JOIN Bibauthors
ON Bibrecs.bibno = Bibauthors.bibno
JOIN Authors
ON Bibauthors.aid = Authors.aid
JOIN Bibterms
ON Bibrecs.bibno = Bibterms.bibno
JOIN Sterms
ON Bibterms.tid = Sterms.tid
WHERE term = 'Databases'
ORDER BY title
```

(*ΣΗΜΕΙΩΣΗ*: Εάν ένα βιβλίο έχει πολλαπλούς συγγραφείς, τότε αυτό το βιβλίο θα εμφανιστεί παραπάνω φορές. Δεν προσπάθησα να ενώσω τα rows με ίδιο bibno εφόσον είπατε να μην φτιάξουμε όψεις ή προσωρινούς πίνακες.)

2.

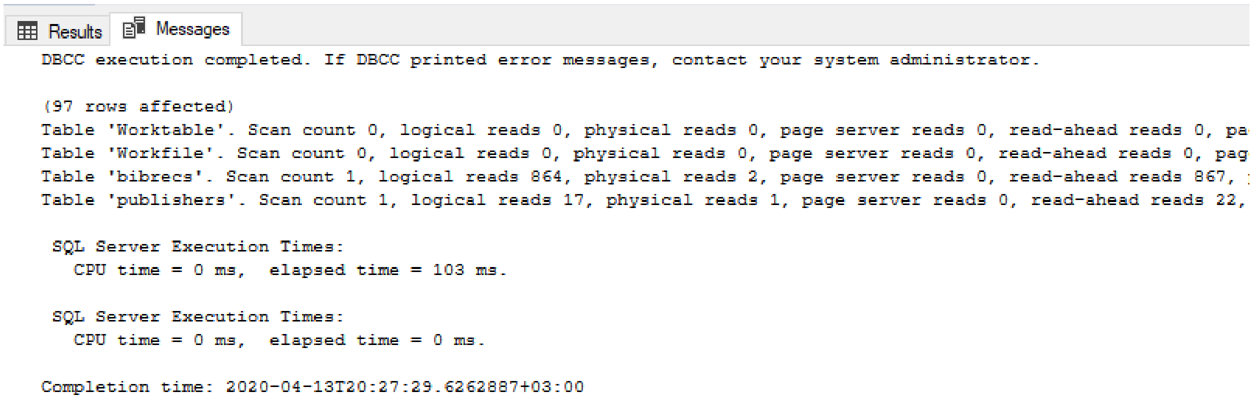
a.

```
CREATE INDEX IndxA ON Publishers(pubname)
```

Επεξήγηση:

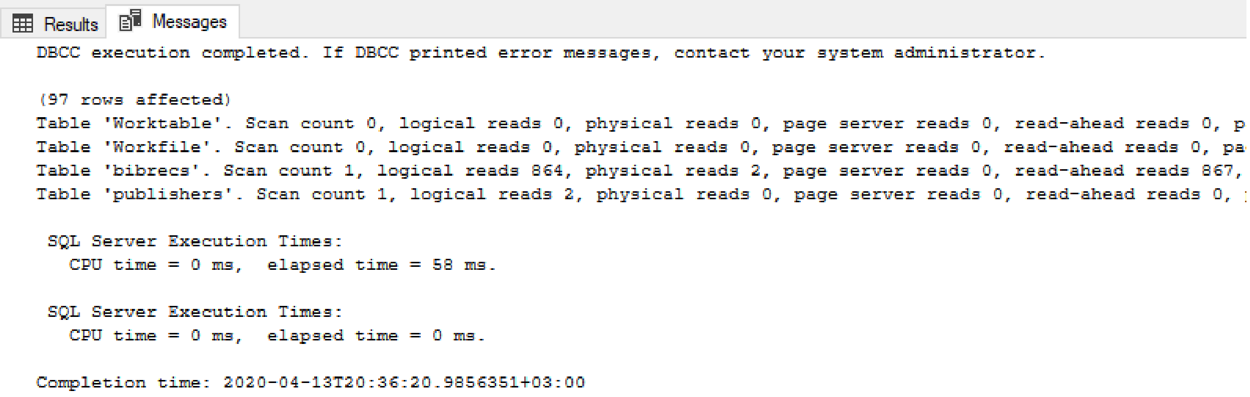
Επέλεξα να χρησιμοποιήσω το pubname καθώς είναι αυτό που βρίσκεται στο condition.

Χωρίς Index:



```
DBCC execution completed. If DBCC printed error messages, contact your system administrator.

(97 rows affected)
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, pa
Table 'Workfile'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, pag
Table 'bibrecs'. Scan count 1, logical reads 864, physical reads 2, page server reads 0, read-ahead reads 867,
Table 'publishers'. Scan count 1, logical reads 17, physical reads 1, page server reads 0, read-ahead reads 22,

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 103 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.

Completion time: 2020-04-13T20:27:29.6262887+03:00
```

Με Index:



```
DBCC execution completed. If DBCC printed error messages, contact your system administrator.

(97 rows affected)
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, p
Table 'Workfile'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, pa
Table 'bibrecs'. Scan count 1, logical reads 864, physical reads 2, page server reads 0, read-ahead reads 867,
Table 'publishers'. Scan count 1, logical reads 2, physical reads 0, page server reads 0, read-ahead reads 0,

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 58 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.

Completion time: 2020-04-13T20:36:20.9856351+03:00
```

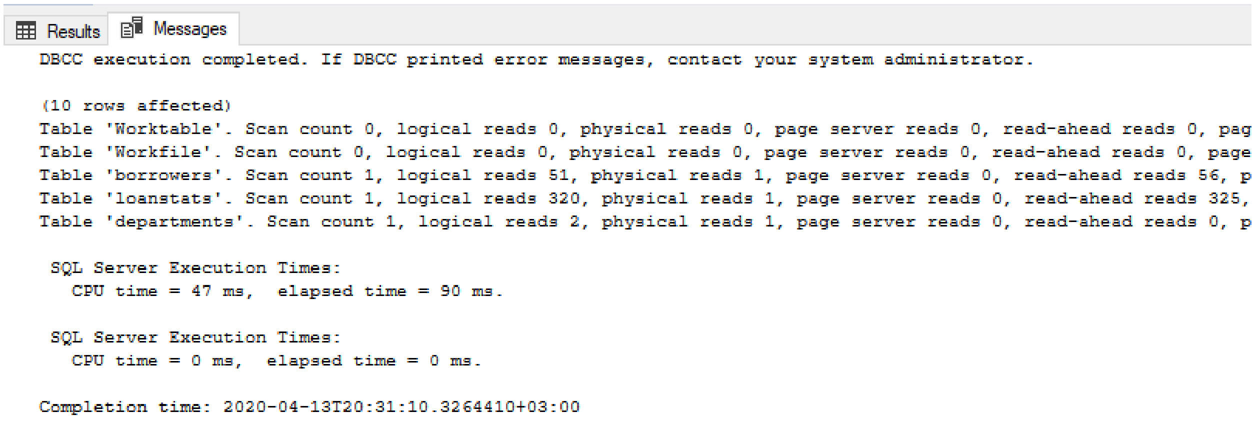
b.

```
CREATE INDEX IndxB ON Loanstats(lid, loandate)
```

Επεξήγηση:

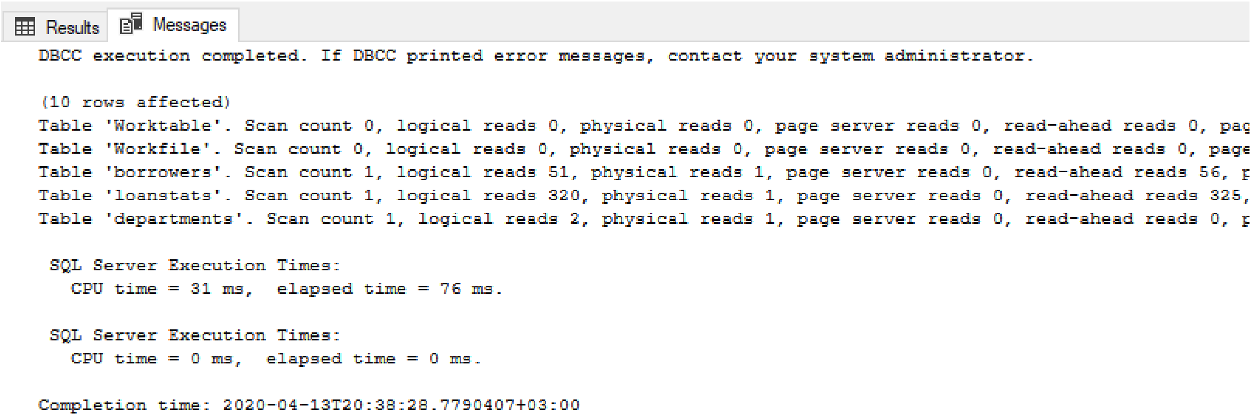
Επέλεξα να χρησιμοποιήσω τα lid και loandate καθώς το loandate βρίσκεται στο condition και το lid το χρησιμοποιώ στην COUNT().

Χωρίς Index:



```
DBCC execution completed. If DBCC printed error messages, contact your system administrator.

(10 rows affected)
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, pag
Table 'Workfile'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, page
Table 'borrowers'. Scan count 1, logical reads 51, physical reads 1, page server reads 0, read-ahead reads 56, p
Table 'loanstats'. Scan count 1, logical reads 320, physical reads 1, page server reads 0, read-ahead reads 325,
Table 'departments'. Scan count 1, logical reads 2, physical reads 1, page server reads 0, read-ahead reads 0, p

 SQL Server Execution Times:
   CPU time = 47 ms,  elapsed time = 90 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.

Completion time: 2020-04-13T20:31:10.3264410+03:00
```

Με Index:



```
DBCC execution completed. If DBCC printed error messages, contact your system administrator.

(10 rows affected)
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, pag
Table 'Workfile'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, page
Table 'borrowers'. Scan count 1, logical reads 51, physical reads 1, page server reads 0, read-ahead reads 56, p
Table 'loanstats'. Scan count 1, logical reads 320, physical reads 1, page server reads 0, read-ahead reads 325,
Table 'departments'. Scan count 1, logical reads 2, physical reads 1, page server reads 0, read-ahead reads 0, p

 SQL Server Execution Times:
   CPU time = 31 ms,  elapsed time = 76 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.

Completion time: 2020-04-13T20:38:28.7790407+03:00
```

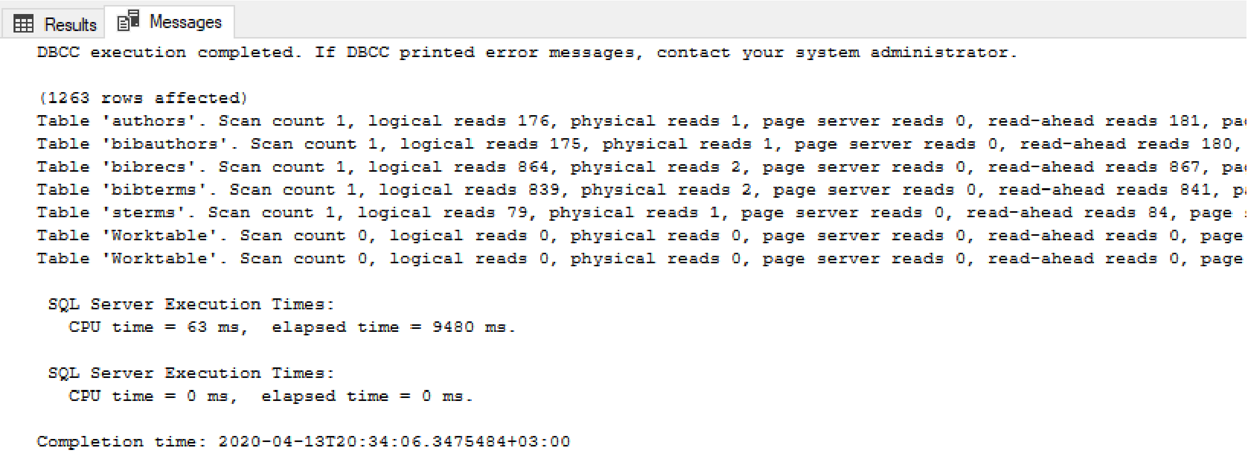
c.

```
CREATE INDEX IndxC ON Sterms(term)
```

Επεξήγηση:

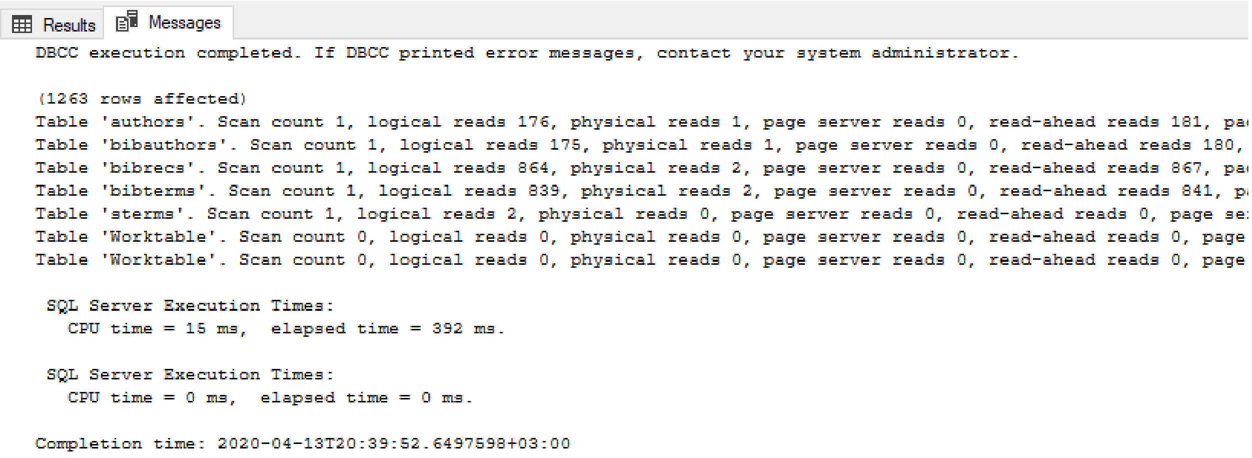
Επέλεξα να χρησιμοποιήσω το term καθώς βρίσκεται στο condition.

Χωρίς Index:



```
DBCC execution completed. If DBCC printed error messages, contact your system administrator.

(1263 rows affected)
Table 'authors'. Scan count 1, logical reads 176, physical reads 1, page server reads 0, read-ahead reads 181, pa
Table 'bibauthors'. Scan count 1, logical reads 175, physical reads 1, page server reads 0, read-ahead reads 180,
Table 'bibrecs'. Scan count 1, logical reads 864, physical reads 2, page server reads 0, read-ahead reads 867, pa
Table 'bibterms'. Scan count 1, logical reads 839, physical reads 2, page server reads 0, read-ahead reads 841, p
Table 'sterms'. Scan count 1, logical reads 79, physical reads 1, page server reads 0, read-ahead reads 84, page
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, page
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, page

 SQL Server Execution Times:
   CPU time = 63 ms,  elapsed time = 9480 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.

Completion time: 2020-04-13T20:34:06.3475484+03:00
```

Με Index:



```
DBCC execution completed. If DBCC printed error messages, contact your system administrator.

(1263 rows affected)
Table 'authors'. Scan count 1, logical reads 176, physical reads 1, page server reads 0, read-ahead reads 181, pa
Table 'bibauthors'. Scan count 1, logical reads 175, physical reads 1, page server reads 0, read-ahead reads 180,
Table 'bibrecs'. Scan count 1, logical reads 864, physical reads 2, page server reads 0, read-ahead reads 867, pa
Table 'bibterms'. Scan count 1, logical reads 839, physical reads 2, page server reads 0, read-ahead reads 841, p
Table 'sterms'. Scan count 1, logical reads 2, physical reads 0, page server reads 0, read-ahead reads 0, page se
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, page
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, page

 SQL Server Execution Times:
   CPU time = 15 ms,  elapsed time = 392 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.

Completion time: 2020-04-13T20:39:52.6497598+03:00
```

Ζητούμενο Τρίτο

1.

1ος Τρόπος

```sql
SELECT DISTINCT Bibrecs.bibno, Bibrecs.title FROM Bibrecs
        JOIN
        (SELECT Bibrecs.bibno FROM Bibrecs, Copies
                WHERE Bibrecs.bibno = Copies.bibno AND copyloc = 'OPA') AS OPA
        ON Bibrecs.bibno = OPA.bibno
        JOIN
        (SELECT Bibrecs.bibno FROM Bibrecs, Copies
                WHERE Bibrecs.bibno = Copies.bibno AND copyloc = 'ANA') AS ANA
        ON OPA.bibno = ANA.bibno
```

2ος Τρόπος

```sql
SELECT DISTINCT Bibrecs.bibno, Bibrecs.title FROM Bibrecs
        JOIN
        (SELECT Bibrecs.bibno FROM Bibrecs
                JOIN Copies
                ON Bibrecs.bibno = Copies.bibno
                WHERE copyloc = 'OPA') AS OPA
        ON Bibrecs.bibno = OPA.bibno
        JOIN
        (SELECT Bibrecs.bibno FROM Bibrecs
                JOIN Copies
                ON Bibrecs.bibno = Copies.bibno
                WHERE copyloc = 'ANA') AS ANA
        ON OPA.bibno = ANA.bibno
```

3ος Τρόπος

```sql
SELECT Bibrecs.bibno, Bibrecs.title FROM Bibrecs
JOIN Copies
        ON Bibrecs.bibno = Copies.bibno
        WHERE copyloc = 'OPA'
INTERSECT
SELECT Bibrecs.bibno, Bibrecs.title FROM Bibrecs
JOIN Copies
        ON Bibrecs.bibno = Copies.bibno
        WHERE copyloc = 'ANA'
```

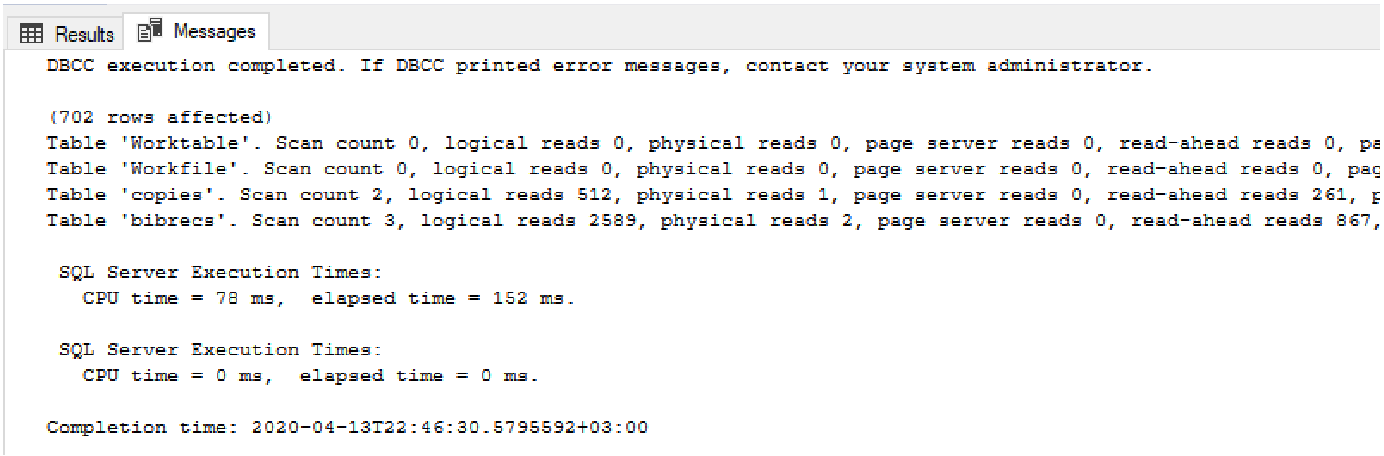
Θα χρησιμοποιούσα τον 2ο τρόπο. Είναι προφανές πως θα επιλέξω JOIN από το να τσεκάρω κάθε φορά αν το bibno του Bibrecs είναι ίδιο με αυτό του Copies. Στον 2ο τρόπο κάνω JOIN όλο τον πίνακα Bibrecs με έναν πίνακα με λιγότερες εγγραφές (Όσες έγγραφες είναι στο ΟΠΑ). Ο μικρότερος πίνακας κάνει JOIN με έναν μικρό πίνακα ακόμα (Όσες έγγραφες είναι στο ΑΝΑ). Επομένως πιστεύω είναι καλύτερο από το να κάνω INTERSECT δύο πίνακες Bibrecs. Αν είχα μικρότερους πίνακες και περισσότερες εγγραφές με ‘OPA’ και ‘ANA’ ίσως να σύμφερε το INTERSECT αφού εν τέλη θα έκανα τρία JOIN σε πίνακες τάξης μεγέθους Bibrecs. Άρα στην προκειμένη περίπτωση επιλέγω τον 2ο τρόπο.

2.

Δημιουργία Index

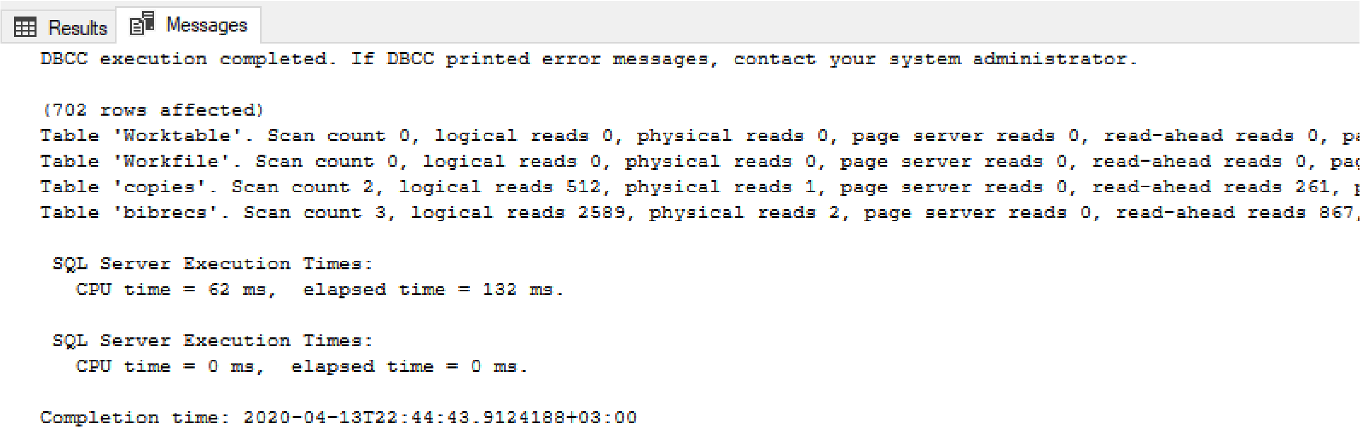
```
CREATE INDEX Indx ON Copies(copyloc)
```

Χωρίς Index:



```
DBCC execution completed. If DBCC printed error messages, contact your system administrator.

(702 rows affected)
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, pa
Table 'Workfile'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, pag
Table 'copies'. Scan count 2, logical reads 512, physical reads 1, page server reads 0, read-ahead reads 261, p
Table 'bibrecs'. Scan count 3, logical reads 2589, physical reads 2, page server reads 0, read-ahead reads 867,

 SQL Server Execution Times:
   CPU time = 78 ms,  elapsed time = 152 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.

Completion time: 2020-04-13T22:46:30.5795592+03:00
```

Με Index:



```
DBCC execution completed. If DBCC printed error messages, contact your system administrator.

(702 rows affected)
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, pa
Table 'Workfile'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, pag
Table 'copies'. Scan count 2, logical reads 512, physical reads 1, page server reads 0, read-ahead reads 261, p
Table 'bibrecs'. Scan count 3, logical reads 2589, physical reads 2, page server reads 0, read-ahead reads 867,

 SQL Server Execution Times:
   CPU time = 62 ms,  elapsed time = 132 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.

Completion time: 2020-04-13T22:44:43.9124188+03:00
```

Ζητούμενο Τέταρτο

1.

```sql
CREATE TABLE words
( wid INT PRIMARY KEY,
  word VARCHAR(50)
);


CREATE TABLE bibwords
( wid INT FOREIGN KEY REFERENCES words(wid),
  bibno INT FOREIGN KEY REFERENCES bibrecs(bibno),
  PRIMARY KEY(wid)
);
```

2.

```sql
CREATE INDEX IndxB ON Bibrecs(title, series)

CREATE INDEX IndxP ON Publishers(pubname)

CREATE INDEX IndxA ON Authors(author)

CREATE INDEX IndxS ON Sterms(term)

CREATE INDEX IndxW ON Words(word)
```

Θεώρησα πως οι στήλες που χρειάζονται index είναι τα free text fields καθώς θα υποστούν τις αναζητήσεις και φυσικά η στήλη word του πίνακα Words αφού πάνω σε αυτόν θα ψάχνουμε.

Ζητούμενο Πέμπτο

Εφόσον έχουμε αυτά τα 5 πεδία ως free text fields θα μπορούσαμε να τα χρησιμοποιήσουμε ως identifiers των λέξεων. Θα κρατούσα τους πίνακες ***wid*** και ***bibwords*** και θα μετέτρεπα τον ***wid*** ως εξής:

wid	word	typeid
1	Σχεδιασμός	1
2	Βάσεων	1
3	Δεδομένων	1
4	Σμπίλιας	3
5	Γιαννακουδάκης	4
6	Εμμανουήλ	4
7	Βάσεις	5
8	Δεδομένων	5

wid	bibno
1	100
2	100
3	100
4	100
5	100
6	100
7	100
8	100

typeid	typename
1	title
2	series
3	pubname
4	author
5	term

Όπως βλέπετε έφτιαξα έναν πίνακα **entity_type** με primary key το *typeid*. Το *typeid* είναι foreign key στον πίνακα **wid** και θεωρώ είναι ένας καλός τρόπος να ξεχωρίσουμε τα πεδία. Ακολουθεί και SQL για καλύτερη επεξήγηση.

SQL

```
CREATE TABLE entity_type
( typeid INT PRIMARY KEY,
  typename VARCHAR (10)
);


CREATE TABLE words
( wid INT PRIMARY KEY,
  word VARCHAR(50),
  typeid INT FOREIGN KEY REFERENCES entity_type(typeid)
);


CREATE TABLE bibwords
( wid INT FOREIGN KEY REFERENCES words(wid),
  bibno INT FOREIGN KEY REFERENCES bibrecs(bibno),
  PRIMARY KEY(wid)
);
```

Πρόταση Εκφώνησης σε SQL

```sql
SELECT Bibrecs.bibno, Bibrecs.title FROM Bibrecs
JOIN
(SELECT Bibwords.bibno FROM Bibwords
        JOIN Words
        ON Bibwords.wid = Words.wid
        JOIN Entity_type
        ON Words.typeid = Entity_type.typeid
        WHERE Words.typeid = 1 AND Words.word = 'Οικονομία') AS TITL
ON Bibrecs.bibno = TITL.bibno
JOIN
(SELECT Bibwords.bibno FROM Bibwords
        JOIN Words
        ON Bibwords.wid = Words.wid
        JOIN Entity_type
        ON Words.typeid = Entity_type.typeid
        WHERE Words.typeid = 2 AND Words.word = 'Ελληνική') AS SER
ON TITL.bibno = SER.bibno
JOIN
(SELECT Bibwords.bibno FROM Bibwords
        JOIN Words
        ON Bibwords.wid = Words.wid
        JOIN Entity_type
        ON Words.typeid = Entity_type.typeid
        WHERE Words.typeid = 4 AND Words.word = 'Οικονόμου') AS AUTH
ON SER.bibno = AUTH.bibno
ORDER BY Bibrecs.bibno
```

Όσα διαφορετικά πεδία θέλουμε να ψάξουμε , τόσα JOIN θα κάνουμε! Καταλαβαίνω πως ίσως να μην είναι βέλτιστος ο τρόπος μου αλλά είναι πλήρως λειτουργικός καθώς τον δοκίμασα σε τοπική βάση μου. Επίσης ο βασικός μου ενδοιασμός είναι το ότι θα πρέπει το πρόγραμμα που τρέχει κατά την εκχώρηση των δεδομένων, να κάνει και την απαραίτητη εκχώρηση στην στήλη *typeid* του πίνακα ***wid*** με εγγραφές του πίνακα ***Entity_Type***.