ΣΧΕΔΙΑΣΜΟΣ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

Project Part A΄

Όνομα: **Παναγιώτης Ντυμένος**

Περιεχόμενα

Ζήτημα Πρώτο 2
1. 2
2. 2
a. 2
b. 3
c. 3
Αιτιολόγηση 4
Ζήτημα Δεύτερο 5
1. 5
a. 5
b. 5
c. 5
2. 6
Ζήτημα Τρίτο 9
1. 9
2. 11
Ζήτημα Τέταρτο 12
1. 12
2. 12
Ζήτημα Πέμπτο 13

Ζήτημα Πρώτο

1.

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

Επεξήγηση:

Επέλεξα να χρησιμοποιήσω το title καθώς είναι αυτό που βρίσκεται στο condition

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:

Results Messages

```
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:

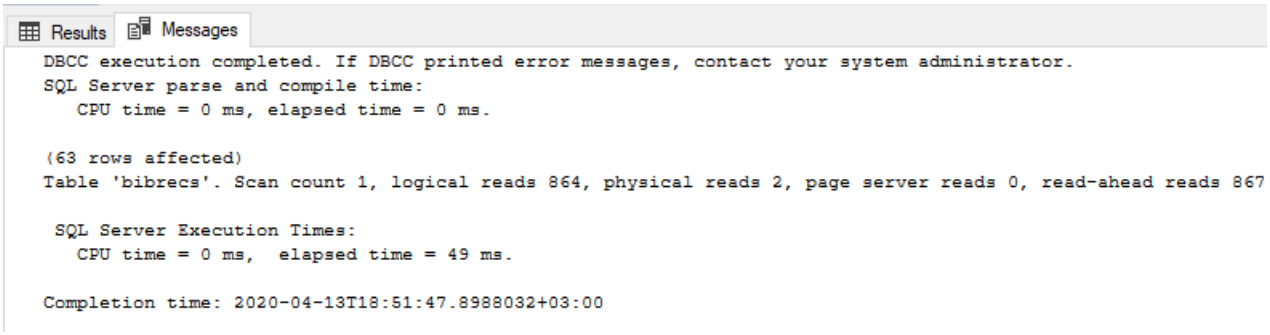
Results Messages

```
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
```

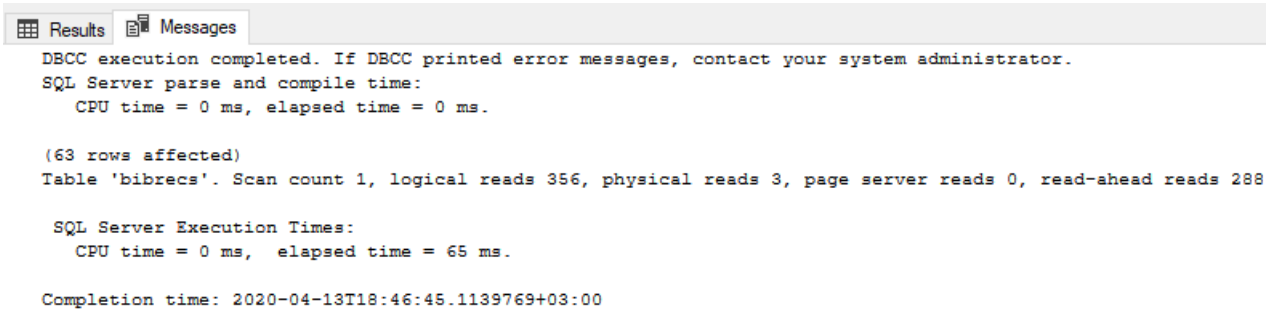
b.

```
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:



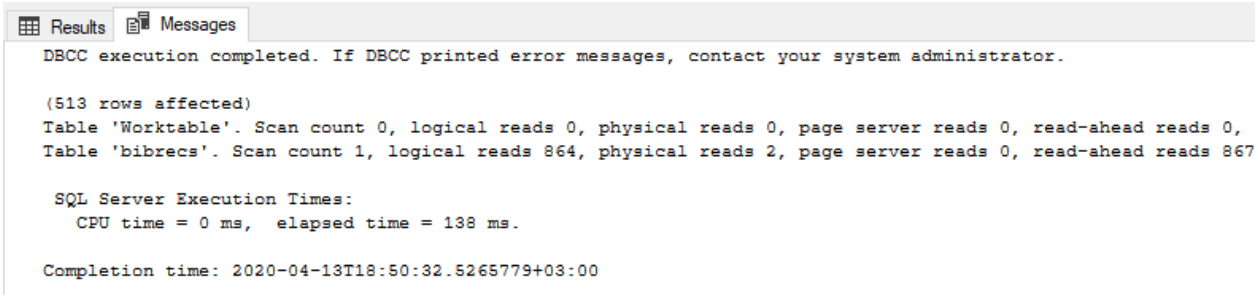
Με Index:



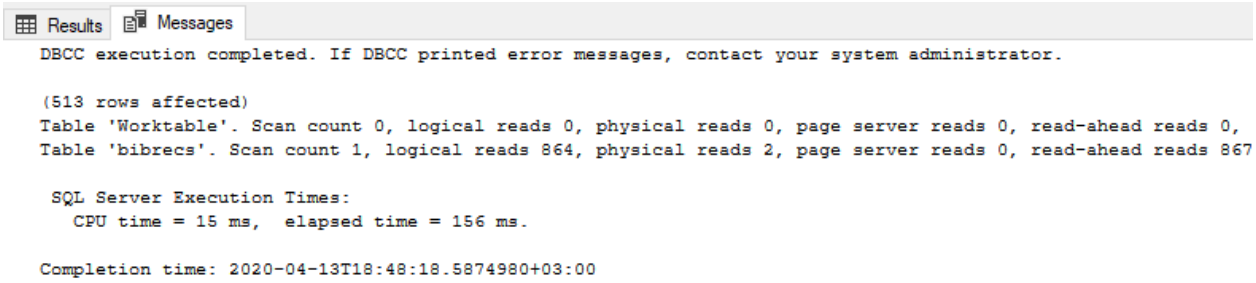
c.

```
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:



Με Index:



Αιτιολόγηση

Όπως βλέπουμε και στις παραπάνω φωτογραφίες, το ευρετήριο μας συμφέρει κυρίως όταν θέλουμε να ανακτήσουμε εγγραφές από το πεδίο **title** (φυσικό εφόσον το ευρετήριο μου είναι πάνω σε αυτή την στήλη). Στην περίπτωση όπου θέλουμε να ανακτήσουμε εγγραφές που δεν έχουμε στο ευρετήριο (όπως όλες τους τίτλους που αρχίζουν από την λέξη Economics), ο τελικός χρόνος προκύπτει από το πόσες εγγραφές θα αναγκαστούμε να προσπελάσουμε. Βλέπουμε πως το ευρετήριο μας συμφέρει στο ερώτημα α) και <u>όχι</u> στο ερώτημα γ) παρόλο που τα δυο Queries δεν διαφέρουν τόσο. Αυτό που ξέρουμε σίγουρα είναι πως όσο ψάχνουμε αυτούσιες εγγραφές του title, θα μας συμφέρει το ευρετήριο.

Ζήτημα 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 και βελτιστοποιεί τις προσπελάσεις στον πίνακα Publishers.

Χωρίς Index:

Results Messages

```
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:

Results Messages

```
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
```

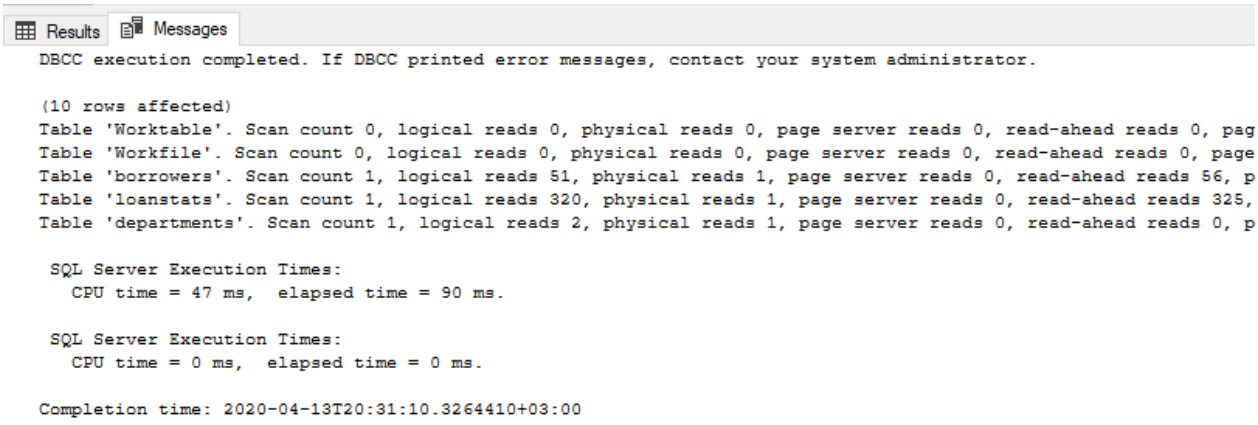
b.

```
CREATE INDEX IndxB ON Departments(depname)
```

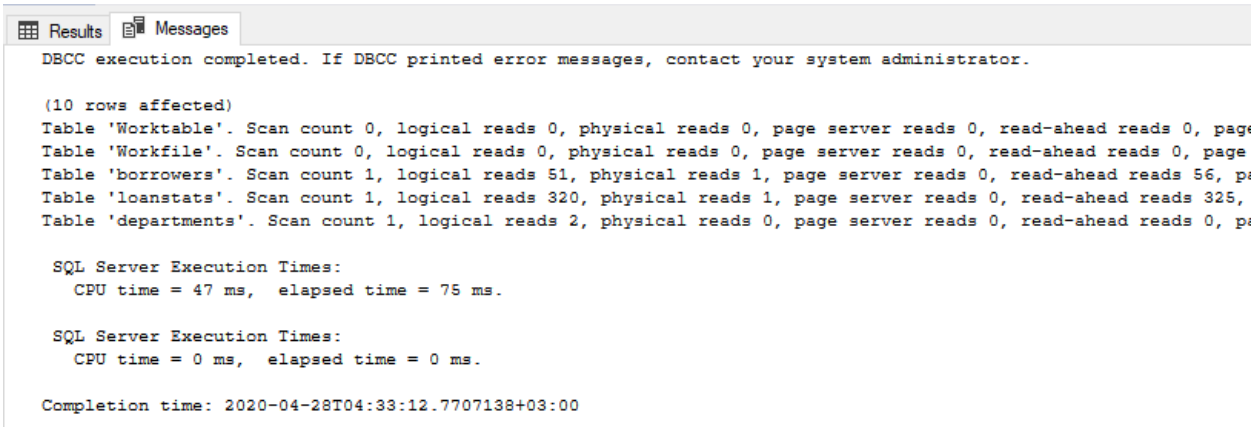
Επεξήγηση:

Επέλεξα να χρησιμοποιήσω το depname καθώς κάνει ένα λιγότερο physical read με index και δεν χρησιμοποίησα τα lid και loandate καθώς το πρώτο είναι primary key και στο άλλο θα αναγκαστώ να κάνω τις ίδιες προσπελάσεις εξαιτίας του 'LIKE'.

Χωρίς Index:



Με Index:



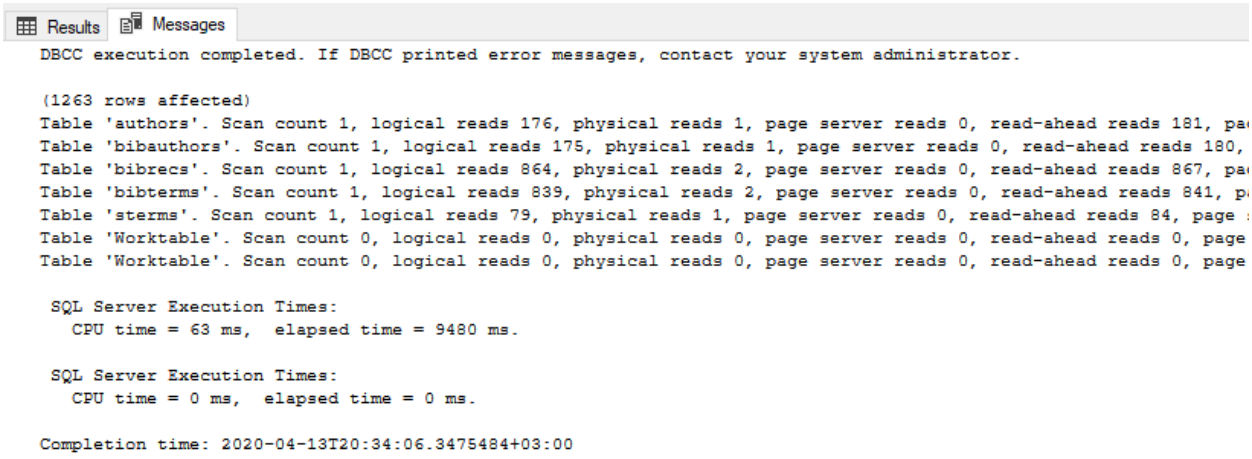
c.

```
CREATE INDEX IndxB ON Bibrecs(title, lang)

CREATE INDEX IndxS ON Sterms(term)

CREATE INDEX IndxA ON Authors(author)
```

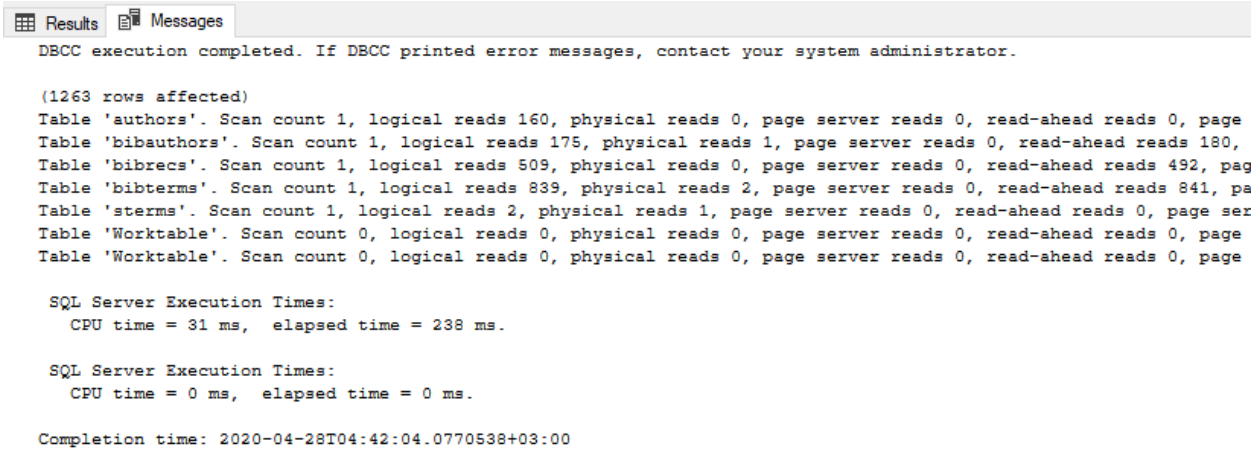
Επεξήγηση:

Επέλεξα να χρησιμοποιήσω το term καθώς βρίσκεται στο condition και τις εγγραφές που βρίσκονται στο SELECT. Παρακάτω φαίνεται πως βελτίωσαν τα Queries.

Χωρίς Index:



Με Index:



Ζήτημα Τρίτο

1.

1ος Τρόπος

```
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

```
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

```
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 ***3ο τρόπο***. Θεωρώ πιο σωστό το να επιλέξω *JOIN* από το να τσεκάρω κάθε φορά αν το *bibno* του ***Bibrecs*** είναι ίδιο με αυτό του ***Copies***, επομένως ο 1ος τρόπος απορρίπτεται. 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). Στον 3ο τρόπο κάνω απλά μια τομή δύο πινάκων τάξης μεγέθους ***Bibrecs***. Παρακάτω φαίνεται ξεκάθαρα γιατί επέλεξα τον 3ο και όχι τον 2ο τρόπο βάσει στατιστικών.

Χρόνος/Προσπελάσεις Query 1ου και 2ου τρόπου:

Results Messages

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

 SQL Server Execution Times:
   CPU time = 47 ms,  elapsed time = 169 ms.

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

Completion time: 2020-04-28T04:50:15.9208618+03:00
```

Χρόνος/Προσπελάσεις Query 3ου τρόπου:

Results Messages

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

(702 rows affected)
Table 'Workfile'. 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, pag
Table 'bibrecs'. Scan count 2, logical reads 1728, physical reads 2, page server reads 0, read-ahead reads 867,
Table 'copies'. Scan count 2, logical reads 512, physical reads 1, page server reads 0, read-ahead reads 261, pa

 SQL Server Execution Times:
   CPU time = 62 ms,  elapsed time = 233 ms.

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

Completion time: 2020-04-28T04:55:36.9581491+03:00
```

Παρατηρούμε λιγότερες προσπελάσεις στον πίνακα ***bibrecs***.

2.

Δημιουργία Index

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

Επεξήγηση:

Βλέπουμε πως βελτιώνεται κατά πολύ ο αριθμός προσπελάσεων στον πίνακα bibrecs. Δεν πείραξα το πεδίο bibno αφού είναι primary key και δεν βελτιώθηκε το Query όταν έφτιαξα ευρετήριο με το πεδίο copyloc του πίνακα copy.

Χωρίς Index:

Results Messages

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

(702 rows affected)
Table 'Workfile'. 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, pag
Table 'bibrecs'. Scan count 2, logical reads 1728, physical reads 2, page server reads 0, read-ahead reads 867,
Table 'copies'. Scan count 2, logical reads 512, physical reads 1, page server reads 0, read-ahead reads 261, pa

 SQL Server Execution Times:
   CPU time = 62 ms,  elapsed time = 233 ms.

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

Completion time: 2020-04-28T04:55:36.9581491+03:00
```

Με Index:

Results Messages

```
DBCC execution completed. If DBCC printed error messages, contact your system administrator.
SQL Server parse and compile time:
   CPU time = 16 ms, elapsed time = 62 ms.

(702 rows affected)
Table 'Workfile'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, pa
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, p
Table 'bibrecs'. Scan count 2, logical reads 970, physical reads 0, page server reads 0, read-ahead reads 17,
Table 'copies'. Scan count 2, logical reads 512, physical reads 1, page server reads 0, read-ahead reads 261,

 SQL Server Execution Times:
   CPU time = 78 ms,  elapsed time = 189 ms.

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

Completion time: 2020-04-28T05:00:31.6281300+03:00
```

Ζήτημα Τέταρτο

1.

```
CREATE TABLE words
( wid INT PRIMARY KEY,
  word VARCHAR(50) NOT NULL
);


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

2.

```
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.

Ζήτημα 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) NOT NULL
);


CREATE TABLE words
( wid INT PRIMARY KEY,
  word VARCHAR(50) NOT NULL,
  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)
);
```

```
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***.