

Indeksy klastrowe i nieklastrowe

Indeks nieklastrowy

Tworzenie tabeli notification

```
1  CREATE TABLE notification
2  (
3      id_notification BIGINT NOT NULL ,
4      type TINYINT NOT NULL DEFAULT 1 ,
5      name VARCHAR (200) NOT NULL ,
6      content CHAR (1000) NOT NULL ,
7      active TINYINT NOT NULL ,
8      created_at DATETIME NOT NULL ,
9      updated_at DATETIME
10 )
11 GO
12
13 ALTER TABLE notification
14     ADD CONSTRAINT notification_types
15     CHECK ( type IN (1, 2) )
16 GO
17
18 ALTER TABLE notification ADD CONSTRAINT notification_PK PRIMARY KEY
19     NONCLUSTERED (id_notification)
20     WITH (
21         ALLOW_PAGE_LOCKS = ON ,
22         ALLOW_ROW_LOCKS = ON )
23 GO
```

Tabela ma indeks **nieklastrowy** notification_PK na kolumnie **id_notification**.

Tak utworzoną tabelę zapełniamy danymi:

Funkcja dodająca dane do tabeli

```
1  CREATE SEQUENCE notification_seq START WITH 1 INCREMENT BY 1
2
3  DECLARE @i INTEGER
4  SET @i=1
5  WHILE @i<20001
6      BEGIN
7          INSERT INTO notification VALUES (NEXT VALUE FOR notification_seq,
```

```

8      FLOOR(RAND()*(2-1+1)+1),
9      'Notification' + cast(@i as varchar(100)),
10     CONVERT(CHAR(1000), CAST(CRYPT_GEN_RANDOM(500) AS UNIQUEIDENTIFIER),
11     2),
11     ROUND(RAND(), 0),
12     DATEADD(DAY, ABS(CHECKSUM(NEWID()) % 3653), '2014-01-01'),
13     null
14     )
15     SET @i=@i+1
16 END
17 GO

```

Następnie włączamy statystyki:

Włączenie statystyk

```
1 SET STATISTICS IO ON
```

I wykonujemy zapytanie:

Zapytanie pobierające rekord po id_notification

```
1 SELECT * FROM notification WHERE id_notification = 10000;
```

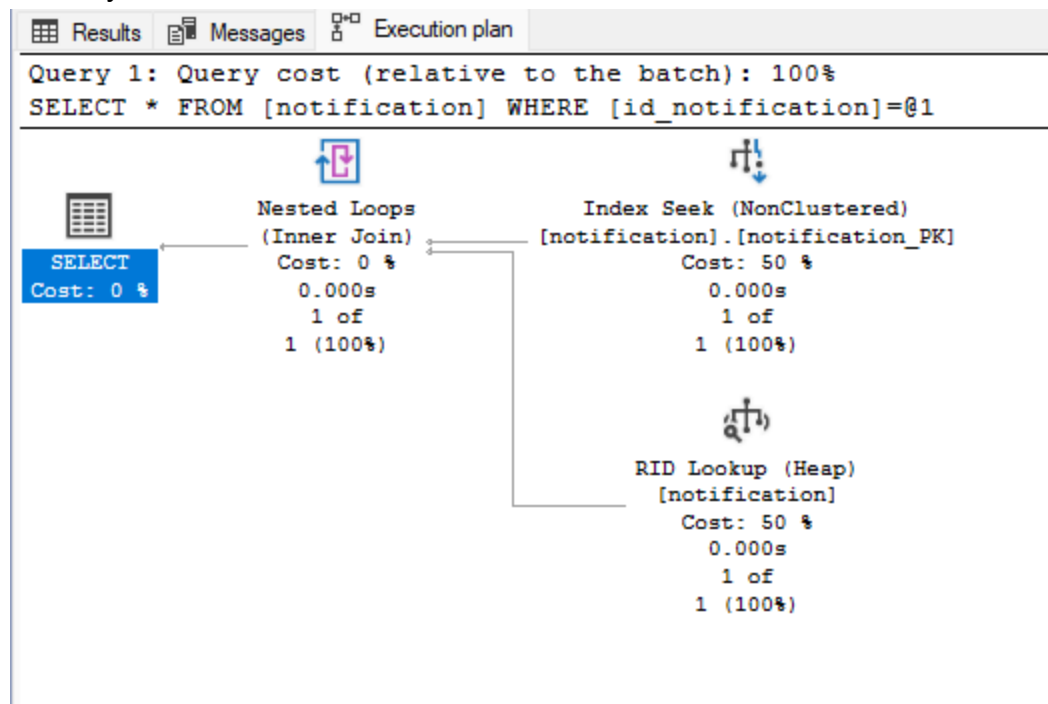
Wynikiem zapytania jest:

	id_notification	type	name	content	active	created_at	updated_at
1	10000	1	Notification10000	0AD99D25-AC4C-C28F-D6CA-374A55EB33DB	0	2020-10-20 00:00:00.000	NULL

Statystyki zapytania to:

Results	Messages	Execution plan
(1 row affected)		
Table 'notification'. Scan count 0, logical reads 3, physical reads 3, page server reads 0		
(1 row affected)		

Plan wykonania:



Indeks klastrowy

W pierwszym kroku usuwam tabele notification

Usuwanie tabeli notification

```
1 DROP TABLE notification;
```

Następnie tworzę ją ponownie, tak samo jak wcześniej, ale z indeksem klastrowym:

```
1 ALTER TABLE notification ADD CONSTRAINT notification_PK PRIMARY KEY
  CLUSTERED (id_notification)
2 WITH (
3     ALLOW_PAGE_LOCKS = ON ,
4     ALLOW_ROW_LOCKS = ON )
5 GO
```

i po zapelnieniu jej taką samą ilością danych jak wcześniej ponawiam zapytanie:

Zapytanie pobierające rekord po id_notification

```
1 SELECT * FROM notification WHERE id_notification = 10000;
```

Zapytanie oczywiście zwraca identyczny wynik:

	id_notification	type	name	content	active	created_at	updated_at
1	10000	1	Notification10000	D36FFC10-8296-F2AA-DD3E-49CB1E16409C	0	2022-08-16 00:00:00.000	NULL

Widać jednak różnicę w liczbie fizycznych odczytów:

	id_notification	type	name	content	active	created_at	updated_at
1	10000	1	Notification10000	D36FFC10-8296-F2AA-DD3E-49CB1E16409C	0	2022-08-16 00:00:00.000	NULL

(1 row affected)
Table 'notification'. Scan count 0, logical reads 3, physical reads 0, page server reads 0,
(1 row affected)

Plan wykonania:

