

GUIÃO 8

1.

#	Query	Rows	Cost	Pags. Read	Time (ms)	Index used	Index Op.
1	SELECT * FROM Production.WorkOrder	72591	0,488	540	1383	PK	Clustered Index Scan
2	SELECT * FROM Production.WorkOrder WHERE WorkOrderID=1234	1	0,003	14	40	PK	Clustered Index Seek
3.1	SELECT * FROM Production.WorkOrder WHERE WorkOrderID between 10000 and 10010	11	0,003	14	116	PK	Clustered Index Seek
3.2	SELECT * FROM Production.WorkOrder WHERE WorkOrderID between 1 and 72591	72591	0,488	542	1349	PK	Clustered Index Seek
4	SELECT * FROM Production.WorkOrder WHERE StartDate = '2007-06-25'	55	0,523	542	265	PK	Clustered Index Scan
5	SELECT * FROM Production.WorkOrder WHERE ProductID = 757	9	0,003	34	112	ProductID	Non Clustered Index Seek / Clustered Key Lookup
6.1	SELECT WorkOrderID, StartDate FROM Production.WorkOrder WHERE ProductID = 757	9	0,003	16	5	ProductID covered (Start Date)	Non Clustered Index Seek
6.2	SELECT WorkOrderID, StartDate FROM Production.WorkOrder WHERE ProductID = 945	1105	0,006	20	111	ProductID covered (Start Date)	Non Clustered Index Seek
6.3	SELECT WorkOrderID FROM Production.WorkOrder WHERE ProductID = 945 AND StartDate = '2006-01-04'	1	0,006	22	7	ProductID covered (Start Date)	Non Clustered Index Seek
7	SELECT WorkOrderID, StartDate FROM Production.WorkOrder WHERE ProductID = 945 AND StartDate = '2006-01-04'	1	0,017	23	70	ProductID and StartDate	Non Clustered Index Seek (twice)
8	SELECT WorkOrderID, StartDate FROM Production.WorkOrder WHERE ProductID = 945 AND StartDate = '2006-01-04'	1	0,003	18	6	Composite (ProductID, StartDate)	Non Clustered Index Seek

2.

a. CREATE UNIQUE CLUSTERED INDEX rid_index ON dbo.mytemp(rid)

Como queremos que *rid_index* seja chave primária, tem que ser um index *unique* e *clustered*.

b.

	database_id	object_id	index_id	partition_number	index_type_desc	alloc_unit_type_desc
1	13	597577167	1	1	CLUSTERED INDEX	IN_ROW_DATA
2	13	597577167	1	1	CLUSTERED INDEX	IN_ROW_DATA
3	13	597577167	1	1	CLUSTERED INDEX	IN_ROW_DATA
4	13	1977058079	1	1	CLUSTERED INDEX	IN_ROW_DATA

index_depth	index_level	avg_fragmentation_in_percent	page_count	avg_page_space_used_in_percent
3	0	98.7849331713244	823	69.7808376575241
3	1	0	2	86.4035087719298
3	2	0	1	0.395354583642204
0	0	0	0	0

c.

Fillfactor 65

Inserted 50000 total records

Milliseconds used: 140080

Fillfactor 80

Inserted 50000 total records

Milliseconds used: 138410

Fillfactor 90

Inserted 50000 total records

Milliseconds used: 163433

d. Eliminar tabela mytemp, executar a query e criar o Clustered Index

```
CREATE TABLE mytemp (  
    rid BIGINT IDENTITY (1, 1) NOT NULL,  
    at1 INT NULL,  
    at2 INT NULL,  
    at3 INT NULL,  
    lixo varchar(100) NULL  
);
```

SET IDENTITY_INSERT mytemp ON;

Inserted 50000 total records

Milliseconds used: 180714

- e. CREATE NONCLUSTERED INDEX at1_index ON mytemp(at1);
CREATE NONCLUSTERED INDEX at2_index ON mytemp(at2);
CREATE NONCLUSTERED INDEX at3_index ON mytemp(at3);
CREATE NONCLUSTERED INDEX lixo_index ON mytemp(lixo);

Sem indexes

Inserted 50000 total records

Milliseconds used: 249760

Com indexes

Inserted 50000 total records

Milliseconds used: 304643

A performance é pior com todos os indexes porque não há eficiência na inserção dos tuplos na tabela

3.

a.

- i. **CREATE UNIQUE CLUSTERED INDEX index_ssn ON
Company.Employee(Ssn);**

Utilizando o SSN como índice e sendo este já a chave primária, tem que ser "*unique clustered*".

- ii. **CREATE NONCLUSTERED INDEX index_full_name ON
Company.Employee(Fname, Lname);**

Tendo os dados já todos, conseguimos de forma posterior aceder ao empregado pelos nomes daí utilizarmos um índice "*nonclustered*".

À semelhança de uma pesquisa de um índice no fim de um livro.

- iii. **CREATE NONCLUSTERED INDEX index_emp_dep ON
Company.Employee(Dno);
CREATE UNIQUE CLUSTERED INDEX index_dep ON
Company.Department(Dnumber);**

Identificando o número de departamento associado a um empregado e verificando a lista de números de departamento, temos com os dois índices criados acima um acesso rápido ao que procuramos.

iv. **CREATE UNIQUE CLUSTERED INDEX index_works ON
Company.Works_on(Essn, Pno);
CREATE UNIQUE CLUSTERED INDEX index_ssn ON
Company.Employee(Ssn);
CREATE NONCLUSTERED INDEX index_proj_no ON
Company.Project(Pnumber);**

Os funcionários que trabalham para determinado projeto serão pesquisados pelo seu número de segurança social associado a determinado projeto (Essn, Pno). Pesquisando depois pelos vários números de projeto (Pnumber) e números de segurança social (Ssn) conseguimos mais eficazmente aceder à identificação desses funcionários, utilizando apenas as chaves primárias de cada tabela, já existentes.