

**Banco de Dados**  
**Administração de Banco de Dados**  
**AC4 – Planos de Execução / Indexação**

Nome do Aluno: **Daniela Alexandra da Silva RA: 2100282**

Turma: **BD4A** Série: **2º** Data: \_\_\_/\_\_\_/\_\_\_ Professor(a) Responsável: **Gustavo Bianchi Maia**

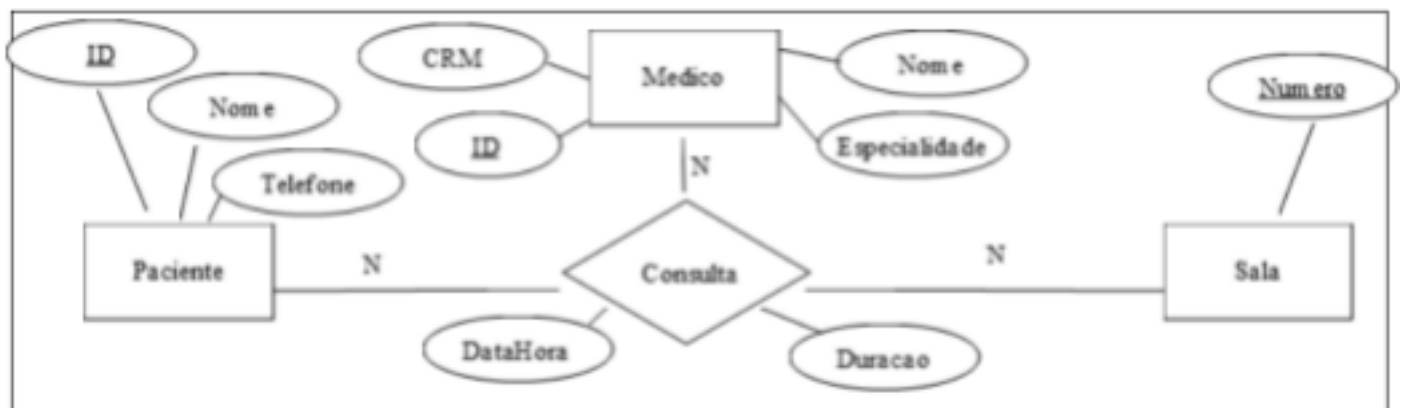
Nota: ( ) Visto do Prof.

**Regras:** Entrega individual.

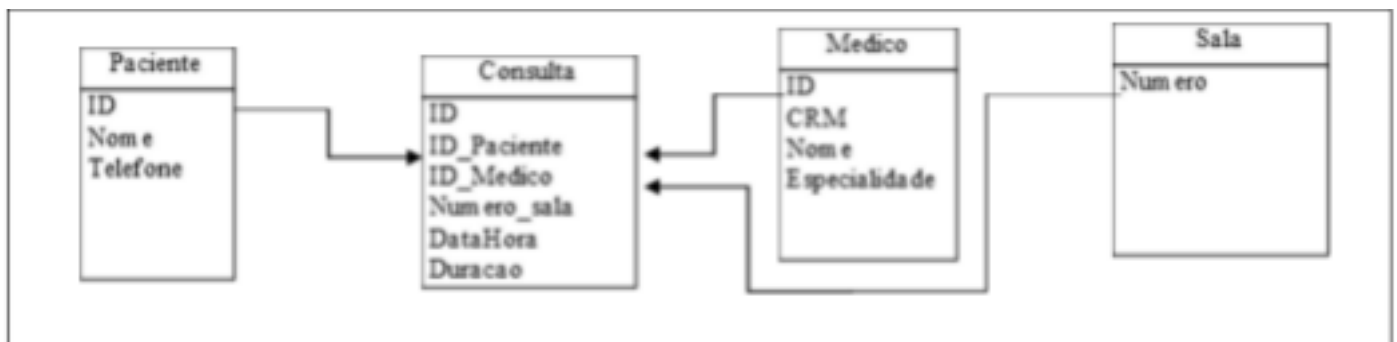
Imagine que um sistema de consultas para um pequeno consultório médio, evoluiu à partir da seguinte tabela em excel, utilizada por muito tempo pelo 'cliente', sejam os seguintes dados de exemplo:

Paciente Contato Consulta (data/hora) Duração Médico	Especialidade Sala
Almir dos Santos 99923232 21/10/2017 15:00 30 Creusa	Dentista 9
Almir dos Santos 99923232 23/10/2017 15:00 15 Juvenal	Psicólogo 9
Adamastor Silva 32324414 26/10/2017 15:00 60 Creusa	Dentista 12

Foi lhe sugerido o seguinte Diagrama Entidade Relacionamento:



A partir do DER, foi feita a seguinte sugestão de um diagrama Lógico Relacional



Ele deseja realizar obter os seguintes relatórios, para atendê-lo crie as seguintes estruturas:

- (1) Visão que devolva: Lista com Nome do Paciente, número do telefone, Data e hora da consulta, duração da consulta, nome do Médico e Sala em que foi feito o atendimento. ( consulta idêntica ao exemplo original ).

```
CREATE OR ALTER VIEW vw_medicoSala AS
SELECT P.Nome as Paciente, P.Telefone, C.DataHora, C.Duracao,
      M.Nome as Medico, S.Numero
FROM Paciente P
      Join Consulta C on C.ID_Paciente = P.ID
      Join Medico M on M.ID = C.ID_Medico
      Join Sala S on S.Numero = C.Numero_Sala
GO
select*from vw_medicoSala
```

## Avaliação Inicial:

SQL Server Execution Times:

CPU time = 578 ms, elapsed time = 21132 ms.

Table 'Consulta' : logical reads 4257

Table 'Medico' : logical reads 4

Table 'Paciente' : logical reads 10

Table 'Sala' : logical reads 4

Custo Subárvore: 9.15459

133 %

Results Messages

(1000000 rows affected)

Table 'Consulta'. Scan count 3, logical reads 4257, physical reads 1, page server reads 0, read-ahead reads 421

Table 'Medico'. Scan count 3, logical reads 4, physical reads 1, page server reads 0, read-ahead reads 0, page

Table 'Paciente'. Scan count 3, logical reads 10, physical reads 1, page server reads 0, read-ahead reads 10, p

Table 'Sala'. Scan count 3, logical reads 4, physical reads 1, 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, pa

SQL Server Execution Times:

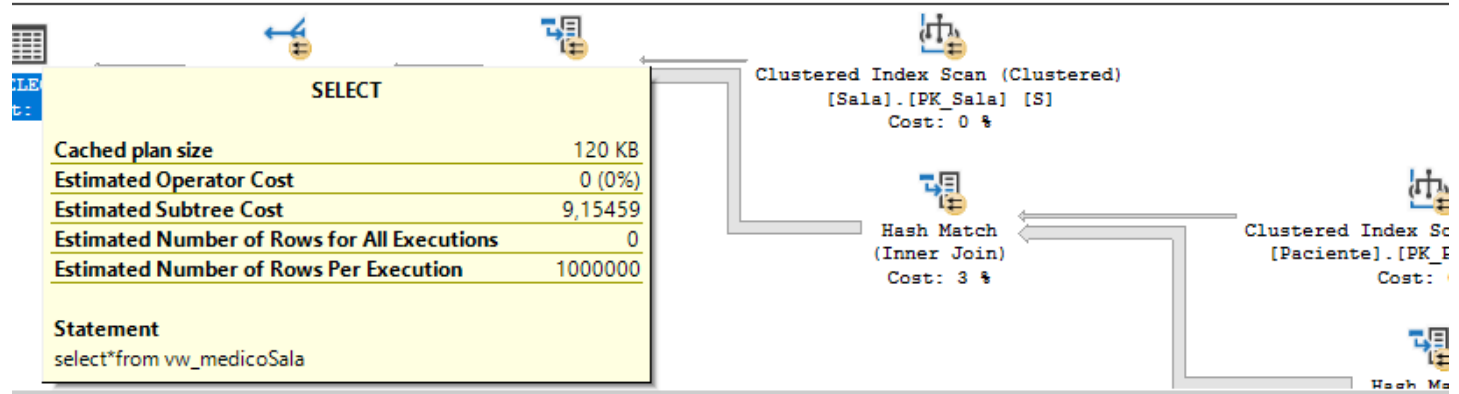
CPU time = 578 ms, elapsed time = 21132 ms.

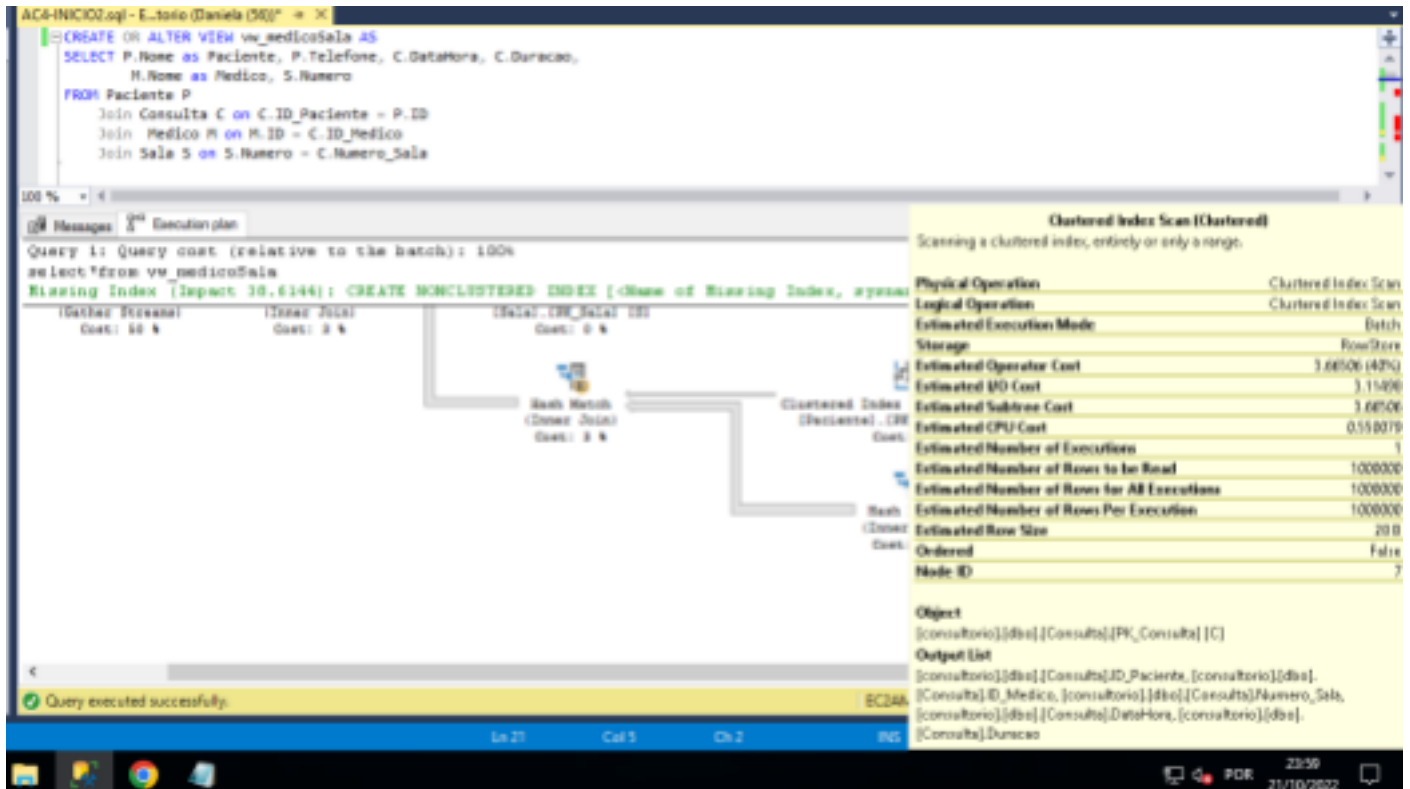
2

:

select\*from vw\_medicoSala

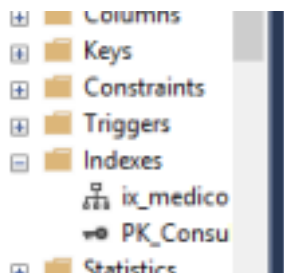
using Index (Impact 38.6144): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [dbo]





## INTERVENÇÃO DBA:

CRIAÇÃO DE INDEX - ix\_medico on consulta (ID\_Medico) include (ID\_Paciente, Numero\_Sala,Duracao,DataHora)



## AVALIAÇÃO FINAL:

Table 'Consulta' : logical reads 3877

Table 'Medico' : logical reads 4

Table 'Paciente' : logical reads 10

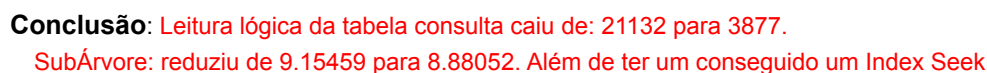
Table 'Sala' : logical reads 4

Tempo: CPU time = CPU time = 609 ms, elapsed time = 23249 ms.

Custo da SubÁrvore : 8.88052

```
(1000000 rows affected)
Table 'Consulta'. Scan count 3, logical reads 3877, physical reads 0, page server reads 0, read-ahead reads 0,
Table 'Medico'. Scan count 3, logical reads 4, physical reads 1, page server reads 0, read-ahead reads 0,
Table 'Paciente'. Scan count 3, logical reads 10, physical reads 1, page server reads 0, read-ahead reads 0,
Table 'Sala'. Scan count 3, logical reads 4, physical reads 1, page server reads 0, read-ahead reads 0,
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0,
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0,
```

CPU time = 609 ms, elapsed time = 23249 ms.



- (2) Função que, dado o nome do paciente, devolva seu telefone.

```
CREATE OR ALTER FUNCTION FN_PACIENTE_TELEFONE ( @PACIENTE VARCHAR(50))
RETURNS INT
AS
BEGIN
DECLARE @TELEFONE INT

SELECT @TELEFONE = TELEFONE
FROM PACIENTE
WHERE NOME = @PACIENTE

RETURN @TELEFONE
END
```

```
select dbo.FN_PACIENTE_TELEFONE('Alan Lazari')
```

## AVALIAÇÃO INICIAL:

Table 'Paciente' : logical reads 5

Tempo: CPU time = CPU time = 0 ms, elapsed time = 2 ms.

Custo da SubÁrvore : 0.005463

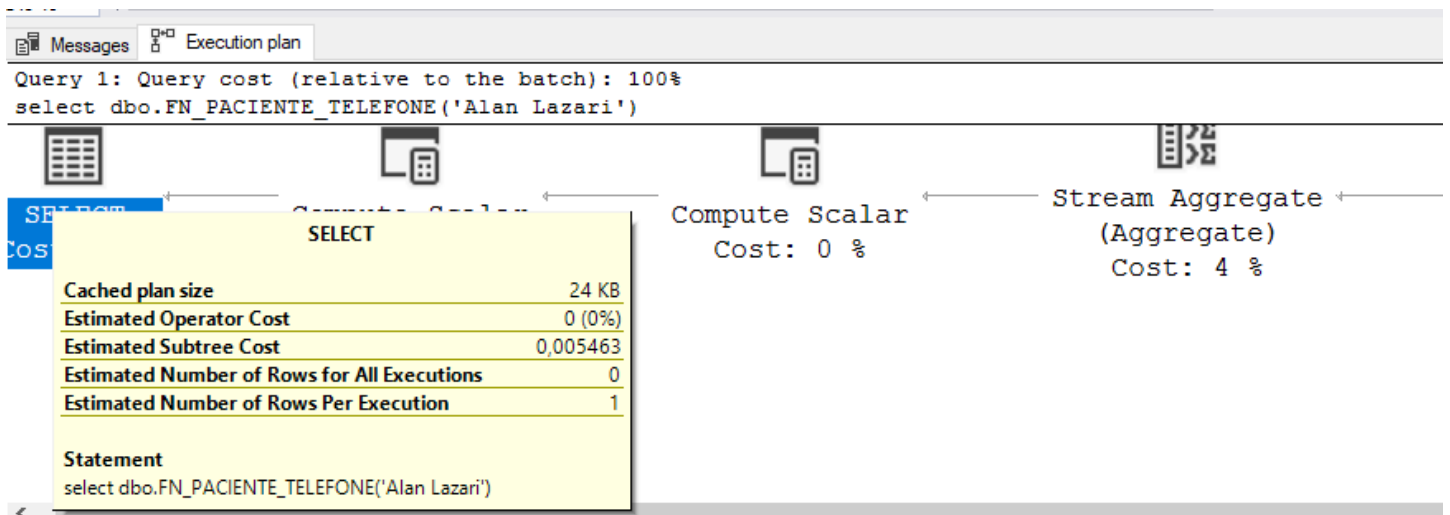
Results Messages

(1 row affected)

Table 'Paciente'. Scan count 1, logical reads 5, physical reads 1, page server reads 0, rea

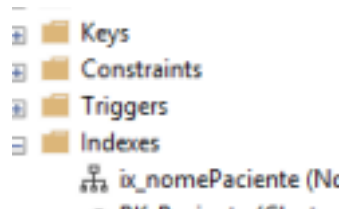
SQL Server Execution Times:

CPU time = 0 ms, elapsed time = 2 ms.



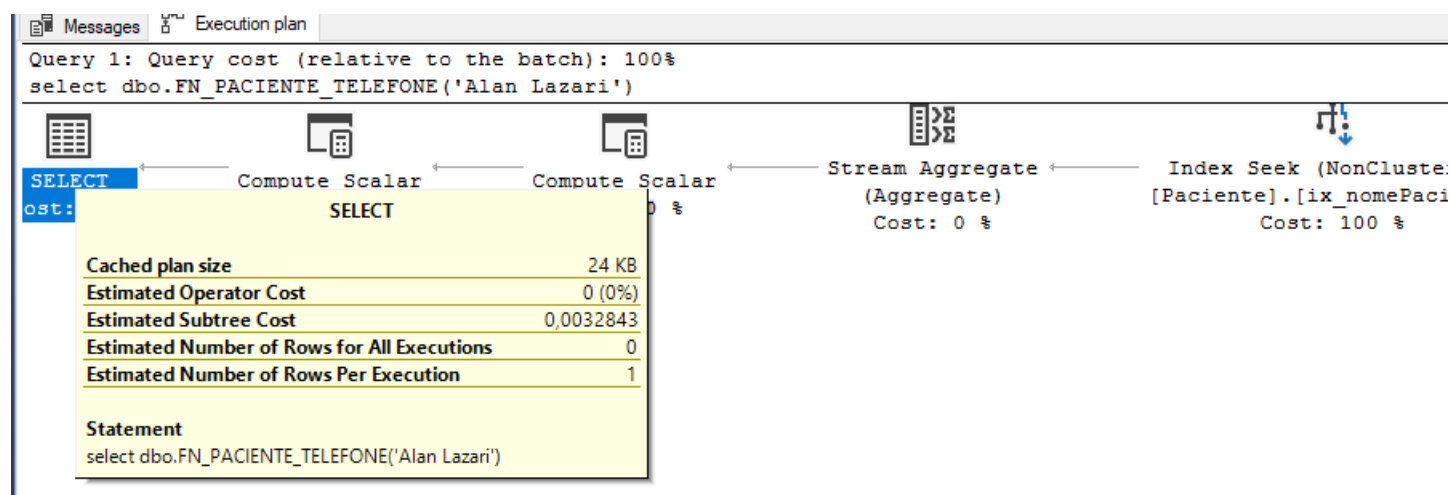
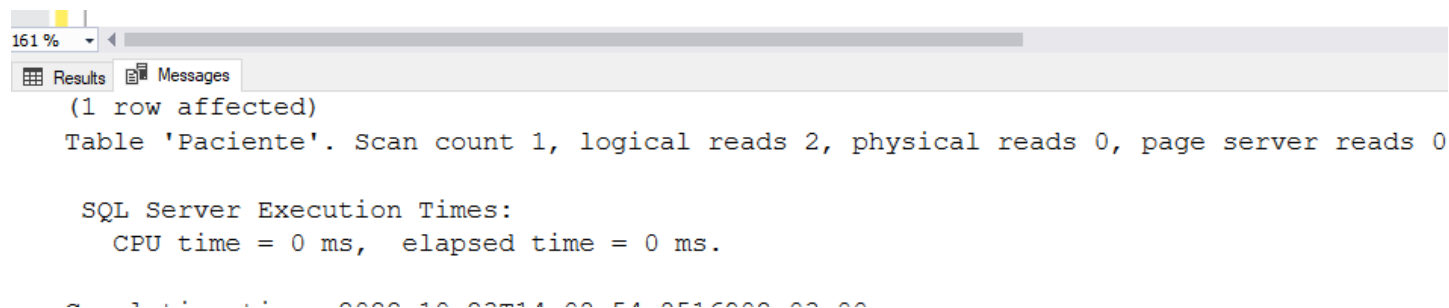
## INTERVENÇÃO DBA:

Criação do Índice - create index ix\_nomePaciente on Paciente ( nome ) include (telefone)



## AVALIAÇÃO FINAL :

Leituras: Table 'Paciente': logical reads 2



**CONCLUSÃO:** O número de Leituras lógicas caíram de 5 para 2 Custo SubÁrvore caiu de .0.005463 para 0.0032843. Além de ter um conseguido um Index Seek

- (3) Função para, dado o telefone, devolver o nome do paciente.

```
CREATE OR ALTER FUNCTION FN_TELEFONE_PACIENTE ( @TELEFONE INT)
RETURNS VARCHAR
AS
BEGIN
DECLARE @PACIENTE VARCHAR(50)

SELECT @PACIENTE = NOME
FROM PACIENTE
WHERE TELEFONE = @TELEFONE

RETURN @PACIENTE
END
```

--- TESTANDO A FUNC

```
select * FROM [dbo].fnc_TelefonePaciente (91270637)
```

## AVALIAÇÃO INICIAL:

Leituras: Table 'Paciente'. logical reads 5  
 Tempo : CPU time = 0 ms, elapsed time = 0 ms.  
 Custo SubÁrvore : 0.0052497

100 %

Results Messages

SQL Server parse and compile time:  
 CPU time = 0 ms, elapsed time = 0 ms.

(1 row affected)  
 Table 'Paciente'. Scan count 1, logical reads 5, physical reads 0, page server reads 0, read-ahead reads 0, page ser

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

Completion time: 2022-10-22T04:03:31.4560111+00:00

Messages Execution plan

Query 1: Query cost (relative to the batch): 100%  
 select \* FROM [dbo].fnc\_TelefonePaciente (91270637)

Index Scan (NonClustered)

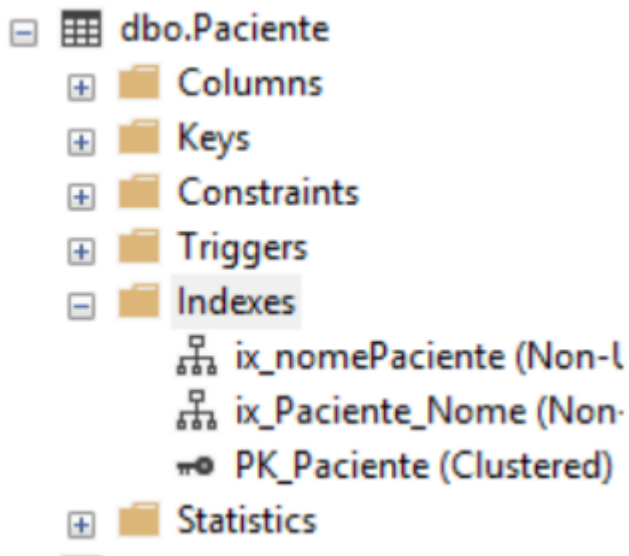
SELECT  
 Cost: 0

SELECT	
Cached plan size	16 KB
Estimated Operator Cost	0 (0%)
Estimated Subtree Cost	0.0052497
Estimated Number of Rows for All Executions	0
Estimated Number of Rows Per Execution	1

Statement  
 select \* FROM [dbo].fnc\_TelefonePaciente (91270637)

## INTERVENÇÃO DBA:

create index ix\_Paciente\_Nome on Paciente (telefone) include (nome)



## AVALIAÇÃO FINAL:

Leituras: Table 'Paciente'. logical reads 2  
Tempo : CPU time = 0 ms, elapsed time = 0 ms.  
Custo SubÁrvore : 0.0052497

```
SQL Server parse and compile time:
  CPU time = 0 ms, elapsed time = 0 ms.

(1 row affected)
Table 'Paciente'. Scan count 1, logical reads 2, physical reads 0, page server reads 0, :

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

Completion time: 2022-10-22T20:30:59.4004905+00:00
|
```



100 %

Messages Execution plan

Query 1: Query cost (relative to the batch): 100%

select \* FROM [dbo].fnc\_TelefonePaciente (91270637)

Index Seek (NonClustered)

SELECT

Cost: 0

SELECT	
Cached plan size	16 KB
Estimated Operator Cost	0 (0%)
Estimated Subtree Cost	0.0032831
Estimated Number of Rows for All Executions	0
Estimated Number of Rows Per Execution	1

Statement

select \* FROM [dbo].fnc\_TelefonePaciente (91270637)

**CONCLUSÃO:** O número de Leituras lógicas caíram de 5 para 2 Custo SubÁrvore caiu de 0.0052497 para 0.00.32831. Além de ter um conseguido um Index Seek

(4) Procedure que, dado o numero de uma sala e um dia, devolva todas as consultas naquele dia, naquela sala. Devolva o nome do paciente, nome do médico, horário e duração da consulta

```
CREATE or Alter PROCEDURE proc_consulta
    @Numero_sala int, @Data date
AS
BEGIN

    SELECT P.Nome as Paciente, M.Nome as Medico, C.Numero_Sala, Convert(date,C.DataHora) as
        Data, C.Duracao FROM Consulta C
        Join Paciente P on C.ID_Paciente = P.ID
        Join Medico M on M.ID = C.ID_Medico
        Join Sala S on S.Numero = C.Numero_Sala
    WHERE C.Numero_Sala = @Numero_sala and Convert(date,C.DataHora) = @Data;
END;

exec proc_consulta 502, '2020-06-06'
```

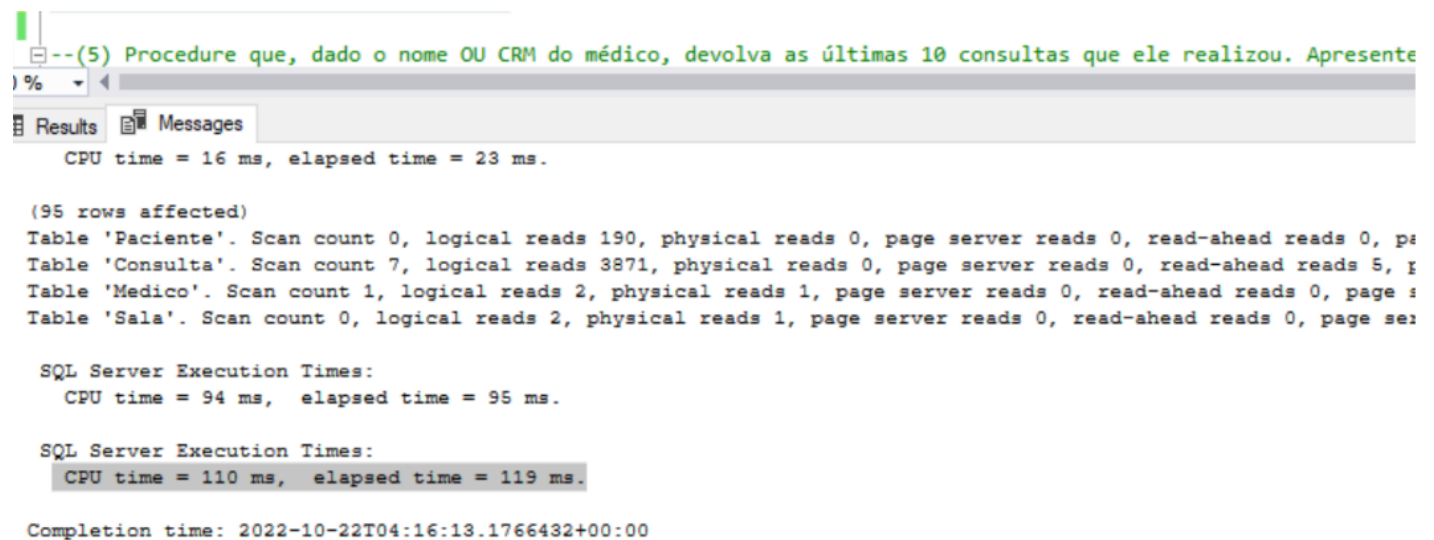
6

### AVALIAÇÃO INICIAL:

Table 'Consulta' : logical reads 3871  
Table 'Medico ': logical reads 2  
Table 'Paciente' : logical reads 190  
Table 'Sala' : logical reads 2

Tempo: CPU time = 110 ms, elapsed time = 119 ms.

Custo da SubÁrvore : 4.54669



```
--(5) Procedure que, dado o nome OU CRM do médico, devolva as últimas 10 consultas que ele realizou. Apresente
) %
Results Messages
CPU time = 16 ms, elapsed time = 23 ms.

(95 rows affected)
Table 'Paciente'. Scan count 0, logical reads 190, physical reads 0, page server reads 0, read-ahead reads 0, page
Table 'Consulta'. Scan count 7, logical reads 3871, physical reads 0, page server reads 0, read-ahead reads 5, p
Table 'Medico'. Scan count 1, logical reads 2, physical reads 1, page server reads 0, read-ahead reads 0, page s
Table 'Sala'. Scan count 0, logical reads 2, physical reads 1, page server reads 0, read-ahead reads 0, page ser

SQL Server Execution Times:
    CPU time = 94 ms,  elapsed time = 95 ms.

SQL Server Execution Times:
    CPU time = 110 ms,  elapsed time = 119 ms.

Completion time: 2022-10-22T04:16:13.1766432+00:00
```

100 %

Messages Execution plan

EXECUTE PROC  
Cost: 0 %

Query 2: Query cost (1)  
proc\_consulta

T-SQL

Stored Procedure  
Cost: 0 %

Statement

CREATE PROCEDURE proc\_consulta  
@Numero\_sala int, @Data date  
AS  
BEGIN  
  
SELECT P.Nome as Paciente, M.Nome as Medico,  
C.Numero\_Sala, Convert(date,C.DataHora) as Data,  
C.Duracao  
FROM Consulta C  
Join Paciente P on C.ID\_Paciente = P.ID  
Join Medico M on M.ID = C.ID\_Medico  
Join Sala S on S.Numero = C.Numero\_Sala  
WHERE C.Numero\_Sala = @Numero\_sala and Convert

SELECT

Cached plan size 48 KB

Estimated Operator Cost 0 (0%)

Estimated Subtree Cost 4,546,69

Estimated Number of Rows for All Executions 0

Estimated Number of Rows Per Execution 9,73047

Scalar  
Cost: 0 %

Nested Loops  
(Inner Join)  
Cost: 22 %

Nested Loops  
(Inner Join)  
Cost: 0 %

Seek (Clustered)  
[Paciente] [P]  
Cost: 0 %

Index Seek (NonClustered)  
[Consulta].[ix\_medico] [C]  
Cost: 78 %

EC2AMAZ-NTN190M (15.0 RTM) | Daniela (51) | consultorio | 00:00:00 | 0 rows

Custo da SubÁrvore : 0.0372258

```

--(5) Procedure que, dado o nome OU CRM do médico, devolva as últimas 10 consultas que ele realizou. Apresente
--o nome do paciente, data, hora e duração da consulta e numero da sala
CREATE PROCEDURE proc_consulta
    @Numero_sala int, @Data date
AS
BEGIN
    SELECT P.Nome as Paciente, M.Nome as Medico,
           C.Numero_Sala, Convert(date,C.DataHora) as Data, C.Duracao
    FROM Consulta C
    JOIN Paciente P ON C.ID_Paciente = P.ID
    JOIN Medico M ON M.ID = C.ID_Medico
    JOIN Sala S ON S.Numero = C.Numero_Sala
    WHERE C.Numero_Sala = @Numero_sala and Convert(date,C.DataHora) = @Data;
END;

```

100 %

Results Messages

Table 'Paciente'. Scan count 0, logical reads 190, physical reads 3, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, Table 'Consulta'. Scan count 7, logical reads 28, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, Table 'Medico'. Scan count 1, logical reads 2, physical reads 1, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, Table 'Sala'. Scan count 0, logical reads 2, physical reads 1, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob :

SQL Server Execution Times:  
CPU time = 0 ms, elapsed time = 8 ms.

SQL Server Execution Times:  
CPU time = 0 ms, elapsed time = 33 ms.

Completion time: 2022-10-22T04:36:04.7577505+00:00

```

create index ix_medico2 on consulta (ID_Medico,DataHora) include (Numero_Sala,Duracao,Id_Paciente)

```

--(5) Procedure que, dado o nome OU CRM do médico, devolva as últimas 10 consultas que ele realizou. Apresente  
--o nome do paciente, data, hora e duração da consulta e numero da sala

100 %

Messages Execution plan

Query 1: Query cost (r  
exec proc\_consulta 502

T-SQL

EXECUTE PROC  
Cost: 0 %

Query 2: Query cost (r  
proc\_consulta

T-SQL

Stored Procedure  
Cost: 0 %

Query executed successfully.

SELECT

Cached plan size	56 KB
Estimated Operator Cost	0 (0%)
Estimated Subtree Cost	0.0372258
Estimated Number of Rows for All Executions	0
Estimated Number of Rows Per Execution	9.73047

Statement

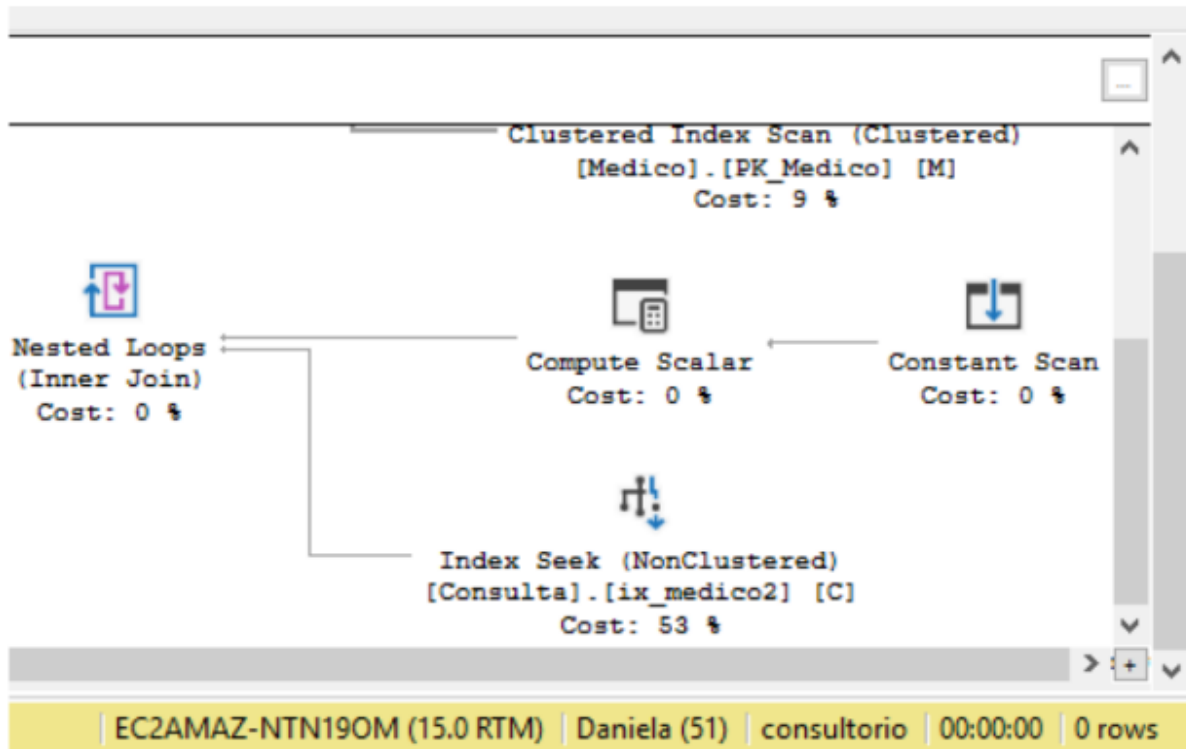
```

CREATE PROCEDURE proc_consulta
    @Numero_sala int, @Data date
AS
BEGIN
    SELECT P.Nome as Paciente, M.Nome as Medico,
           C.Numero_Sala, Convert(date,C.DataHora) as Data, C.Duracao
    FROM Consulta C
    JOIN Paciente P ON C.ID_Paciente = P.ID
    JOIN Medico M ON M.ID = C.ID_Medico
    JOIN Sala S ON S.Numero = C.Numero_Sala
    WHERE C.Numero_Sala = @Numero_sala and Convert

```

Execution plan diagram showing Nested Loops (Inner Join) and Nested Loops (Inner Join) with a Cost of 0 %.

EC2AMAZ-NTN19OM (15.0 RTM) Daniela (51) consultorio 00:00:00 0 rows



## CONCLUSÃO:

O número de Leituras lógicas caíram na tabela consulta de 3871 para 28. O tempo foi reduzido para 33 ms. de 119 ms.

O custo da Subárvore caiu de 4.54669 para 0.0372258. Além de ter um conseguido um Index Seek

(5) Procedure que, dado o nome OU CRM do médico, devolva as últimas 10 consultas que ele realizou. Apresente o nome do paciente, data, hora e duração da consulta e número da sala.

```
CREATE or Alter PROCEDURE proc_medico
    @Nome varchar(150)
AS
BEGIN

    SELECT TOP (10) P.Nome as Paciente, M.CRM, M.Nome as Medico, C.Numero_Sala,
    Convert(date,C.DataHora) as Data, C.Duracao, Convert(varchar(11),C.DataHora,114) as Hora
    FROM Consulta C
    Join Paciente P on C.ID_Paciente = P.ID
    Join Medico M on M.ID = C.ID_Medico
    Join Sala S on S.Numero = C.Numero_Sala

    WHERE M.Nome = @Nome or M.CRM = @Nome
    Order by 5 desc, 7 desc
END;
```

7

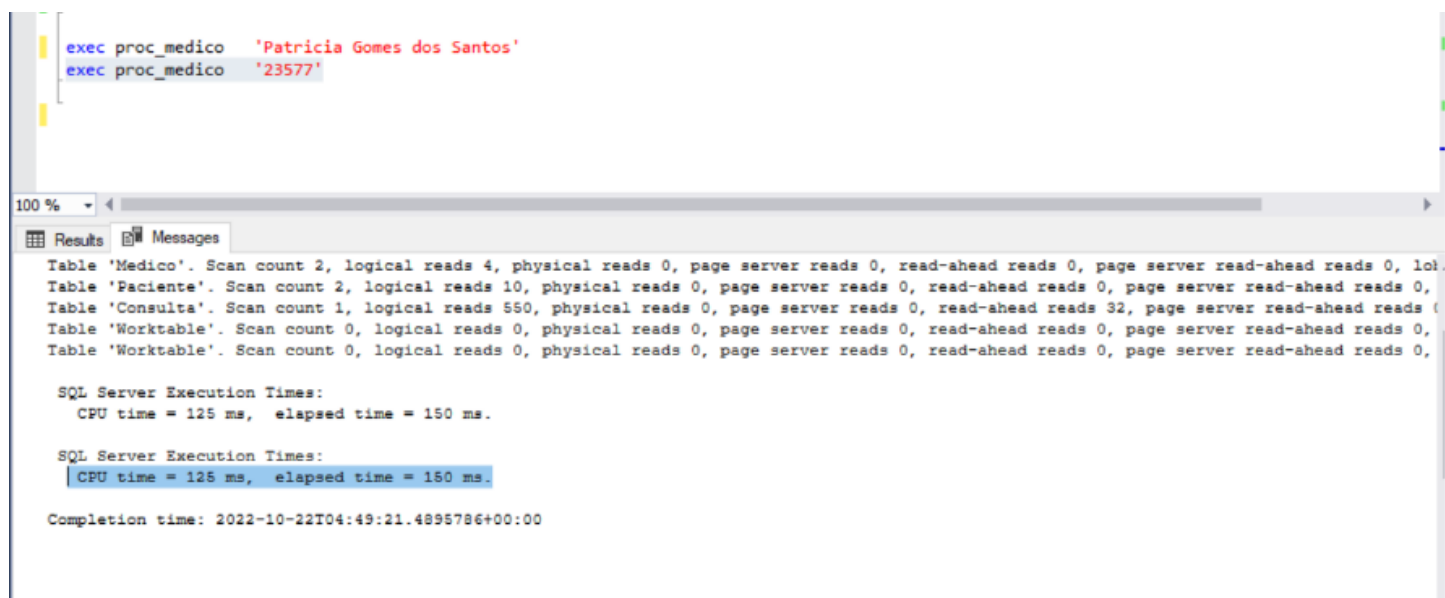
```
exec proc_medico 'Nicolas Cesar Alves'
exec proc_medico '23577'
```

### AValiação Inicial:

Table 'Consulta' : logical reads 550  
Table 'Medico' : logical reads 4  
Table 'Paciente' : logical reads 10

Tempo: CPU time = CPU time = 125 ms, elapsed time = 150 ms.

Custo da SubÁrvore : 1.61748



```
exec proc_medico 'Patricia Gomes dos Santos'
exec proc_medico '23577'
```

100 %

Results Messages

Table 'Medico'. Scan count 2, logical reads 4, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, logical reads 4.

Table 'Paciente'. Scan count 2, logical reads 10, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, logical reads 10.

Table 'Consulta'. Scan count 1, logical reads 550, physical reads 0, page server reads 0, read-ahead reads 32, page server read-ahead reads 0, logical reads 550.

Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, logical reads 0.

Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, logical reads 0.

SQL Server Execution Times:

CPU time = 125 ms, elapsed time = 150 ms.

SQL Server Execution Times:

CPU time = 125 ms, elapsed time = 150 ms.

Completion time: 2022-10-22T04:49:21.4895786+00:00

```

dbcc freeproccache
dbcc dropcleanbuffers

--set statistics time on
--set statistics io on

exec proc_medico 'Patricia Gomes dos Santos'
exec proc_medico '2357'

```

SELECT	
Cached plan size	96 KB
Estimated Operator Cost	0 (0%)
Estimated Subtree Cost	1.61748
Estimated Number of Rows for All Executions	0
Estimated Number of Rows Per Execution	10

**Statement**

```

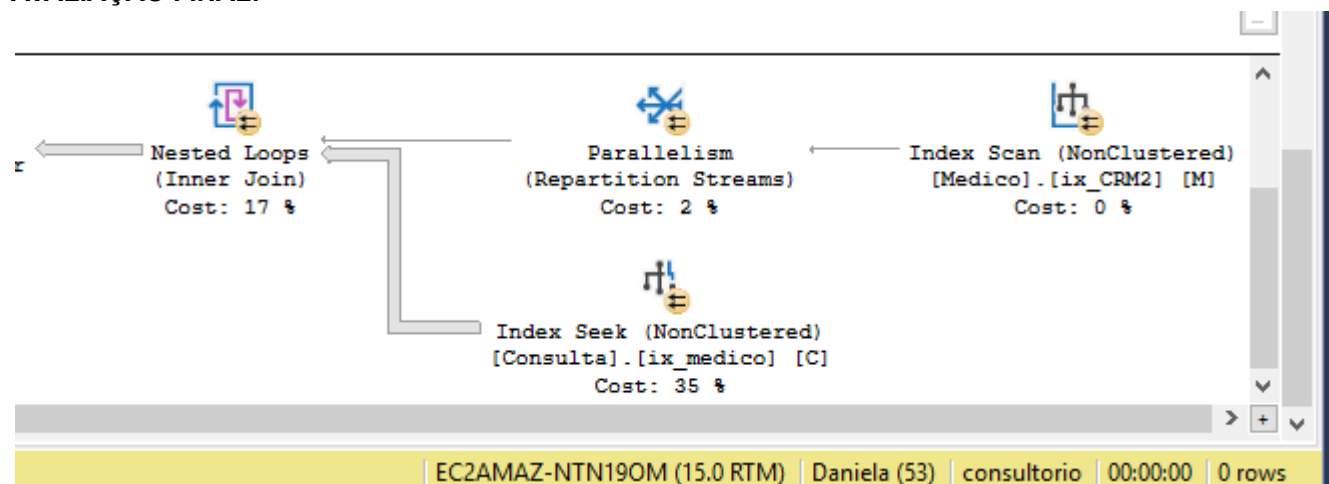
CREATE PROCEDURE proc_medico
    @Nome varchar(150)
AS
BEGIN
    SELECT TOP (10) P.Nome as Paciente, M.CRM, M.Nome as Medico, C.Numero_Sala,
        Convert(date,C.DataHora) as Data, C.Duracao, Convert
        (varchar(11),C.DataHora,114) as Hora
    FROM Consulta C
    Join Paciente P on C.ID_Paciente = P.ID
    Join Medico M on M.ID = C.ID_Medico
    Join Sala S on S.Numero = C.Numero_Sala
    WHERE M.Nome = @Nome or M.CRM = @Nome
    Order by 5 desc, 7 desc

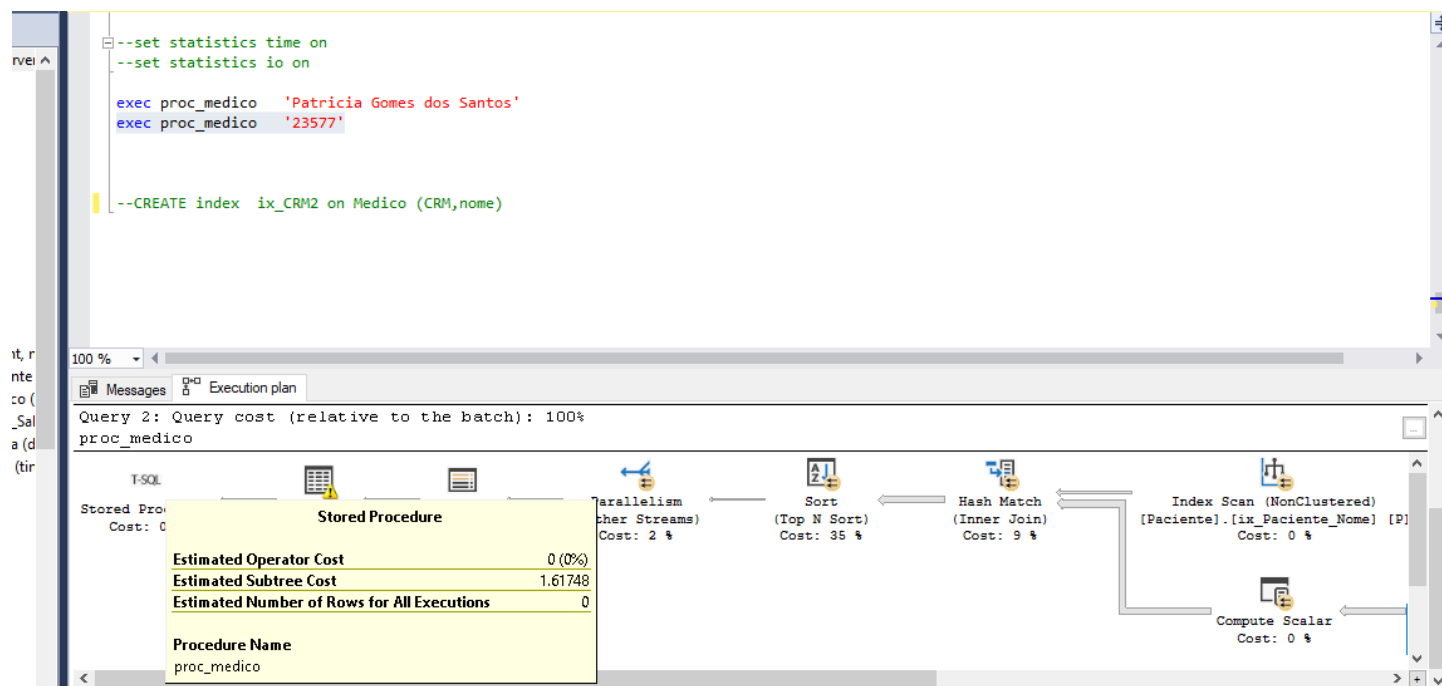
```

Execution plan diagram showing: Sort (Top N Sort) Cost: 35 %, Hash Match (Inner Join) Cost: 9 %, Index Scan (NonClustered) [Paciente].[ix\_Pacien] Cost: 0 %, Compute Scalar Cost: 0 %.

**INTERVENÇÃO DBA:** A intervenção continuou sendo o index ix\_medico criada no exercício anterior.

## **AValiação Final:**





## CONCLUSÃO:

Como o mesmo utiliza o index\_medico anterior não conseguiu uma melhor intervenção. E conseguiu um Index Seek

## Critérios de sucesso / Pontuação:

### Prévia ( ou parte 1 )

2pts – Montou os 5 selects já nas respectivas visões, funções e procedures.

+4pts – Você detalhou o cenário inicial de cada condição( planos de execução, estatísticas de execução

### ) Conclusão ( ou parte 2 )

+2pts – Você implementou índices necessários para otimizar as consultas, reduzindo ( ou eliminando se possível ) os Table e Clustered Index Scan, assim como os Keys e Heaps Lookups.

+2pts – Você coletou estatísticas após a indexação, demonstrando o plano de execução pós otimização 8

Observações:

Boa Avaliação!