Adriano Barbieri	8921162
Douglas Mizuma	8920964
Laura Castro Vieira	8598822
Virgílio Fernandes Junior	7640870

Laboratório de Banco de Dados

Professor Luciano Araújo

São Paulo, SP

2017

Projeto de Laboratório de Bancos de Dados - Estudo e Desempenho de Consultas

O banco de dados instanciado conta com 11 tabelas e foi populado da seguinte forma:

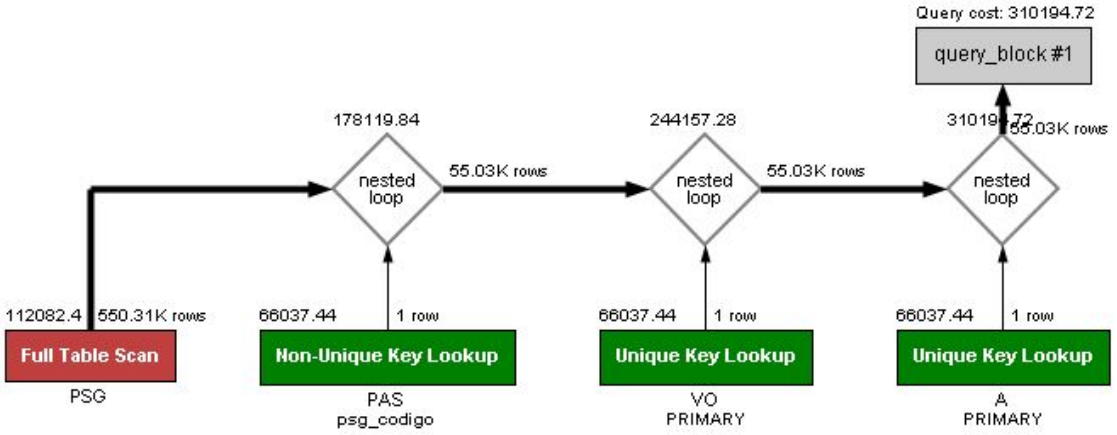
Tabela	**Número de Tuplas**
aeronave	99829
aeroporto	17928
assento	5938832
companhia	9999
historico	962260
passageiro	506916
passagem	535182
promocao	312201
trecho	17000
viagem	17478
voo	17254

Artefato 3 - A

Consulta 1:

SELECT VO.voo_codigo AS Voo, A.numero AS Assento
FROM Passagem PAS
JOIN Passageiro PSG ON PAS.psg_codigo = PSG.psg_codigo
JOIN Voo VO ON VO.voo_codigo=PAS.voo_codigo, Assento A
WHERE A.ass_codigo = PAS.ass_codigo
AND PSG.documento = '86554081456';

O plano de execução de consulta gerado pelo banco é:

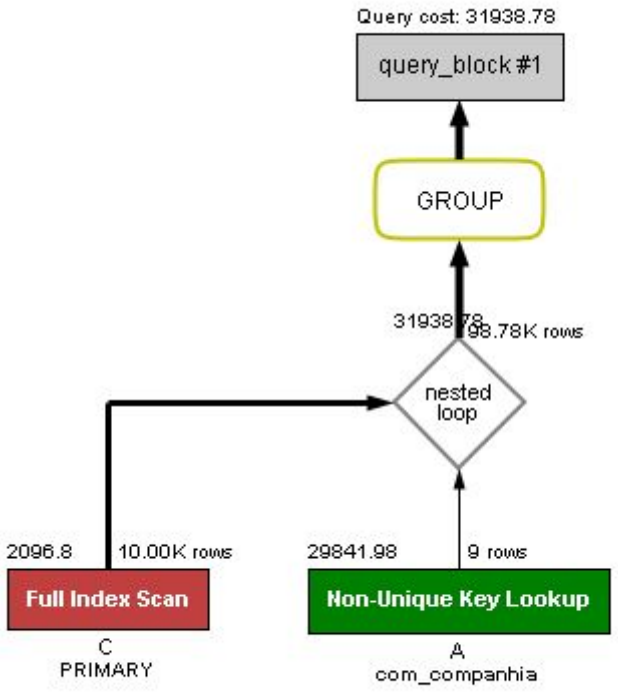


Observamos que o banco precisa fazer loops aninhados para executar a consulta, nesse caso o banco usa a tabela PSG, porque possui menos tuplas devido a operação de seleção para o loop mais externo.
Informações adicionais encontradas com ajuda do workbench:
PSG
- Por não usar indexação precisa checar todas as tuplas;
- Custo alto caso a tabela seja muito grande, se for pequena não há problemas;
PAS
- Custo baixo para médio, baixo se o número de tuplas correspondentes for pequeno, e vai aumentando o custo a medida que o número sobre;
VO e A
- Custo baixo, pois o otimizador é capaz de encontrar um índice que ele pode usar para recuperar registros necessários;
- Rápido porque a pesquisa de índice conduz diretamente a página com todos os dados da linha;

Consulta 2:

SELECT C.nome, COUNT(A.aer_codigo) AS Quantidade_Aeronaves
FROM companhia C, aeronave A
WHERE C.com_codigo = A.com_companhia
GROUP BY C.com_codigo;

O plano de execução gerado pelo banco é:

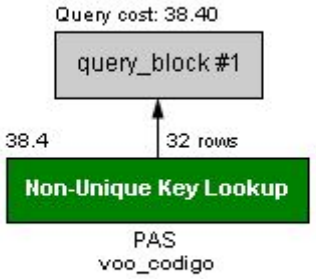


Novamente observamos loops aninhados, e da mesma forma da consulta anterior, a tabela com menor número de tuplas, nesse caso companhia, foi escolhida para ser o loop mais externo.
Informações adicionais encontradas com ajuda do workbench:
C
- Custo alto, especialmente quando o índice é grande;
A
- Custo baixo para médio, baixo se o número de tuplas correspondentes for pequeno, e vai aumentando o custo a medida que o número sobre;

Consulta 3:

SELECT COUNT(PAS.psg_codigo) AS num_pas
FROM passagem PAS
WHERE PAS.voo_codigo = 60;

O plano de execução da consulta gerado pelo banco foi:

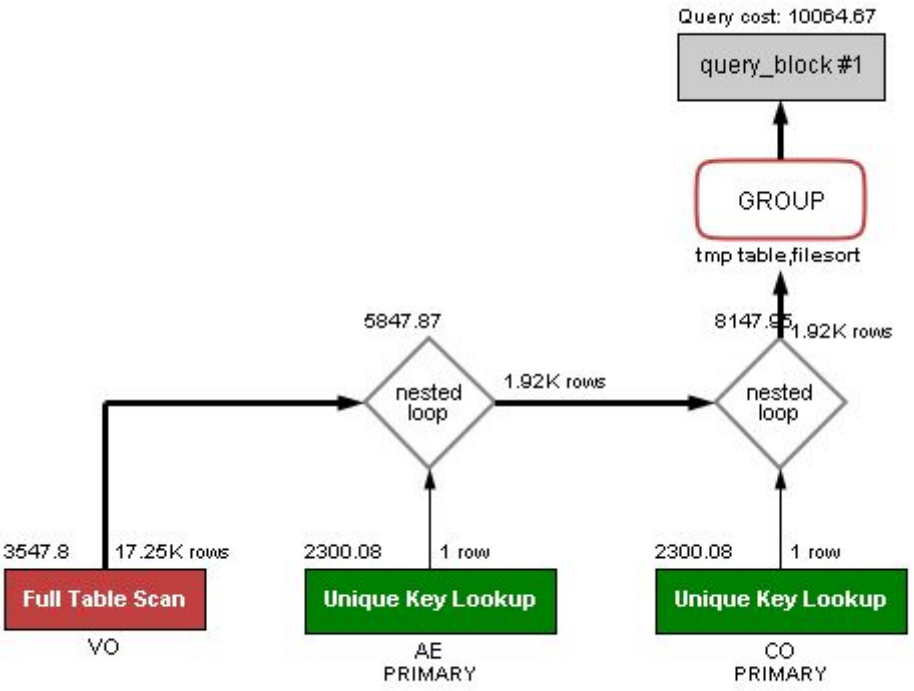


Por ser uma consulta mais simples e sem o uso de índices, o banco deve fazer uma busca linear pela tabela.
Informações adicionais encontradas com ajuda do workbench:
PAS
- Custo baixo para médio, baixo se o número de tuplas correspondentes for pequeno, e vai aumentando o custo a medida que o número sobre;

Consulta 4:

```
SELECT CO.nome, COUNT(VO.voo_codigo) AS QTD_VOOS
FROM Voo AS VO, Aeronave AE, Companhia CO
WHERE CO.com_codigo = AE.com_companhia
AND AE.aer_codigo = VO.aer_codigo
AND VO.data_hora_ini >='2000-01-01'
AND VO.data_hora_ini < '2000-12-31'
GROUP BY CO.com_codigo;
```

O plano de execução de consulta gerado foi:



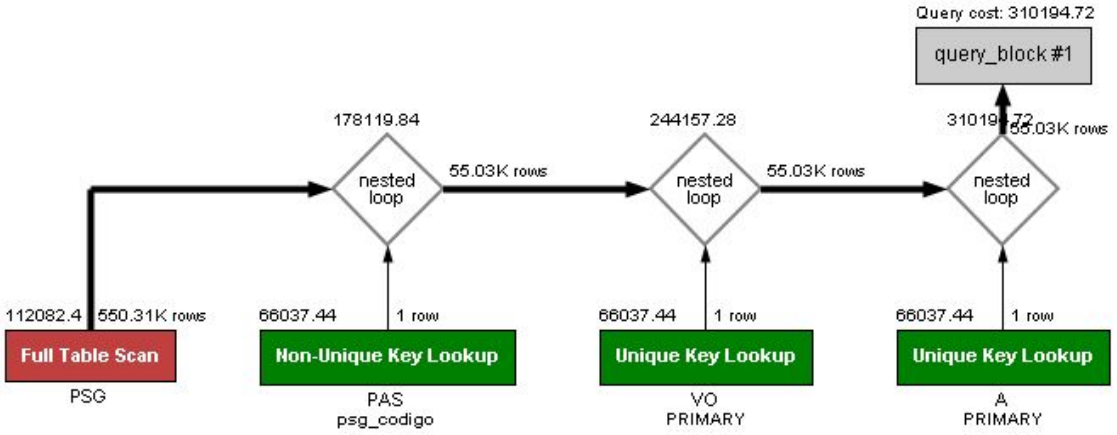
Apesar da tabela companhia ter menos tuplas que a tabela voo, devido às operações de seleção sobre a tabela voo, o número de tuplas que essa tabela retorna é menor do que o número de tuplas da tabela companhia, por este motivo a tabela voo é usada no loop mais externo da consulta.
Informações adicionais encontradas com ajuda do workbench:
VO
- Por não usar indexação precisa checar todas as tuplas;
- Custo alto caso a tabela seja muito grande, se for pequena não há problemas;
- Isso também pode significar que o alcance da pesquisa é tão amplo que o índice seria inútil
AE e CO
- Custo baixo, pois o otimizador é capaz de encontrar um índice que ele pode usar para recuperar registros necessários;

Artefato 3 - B

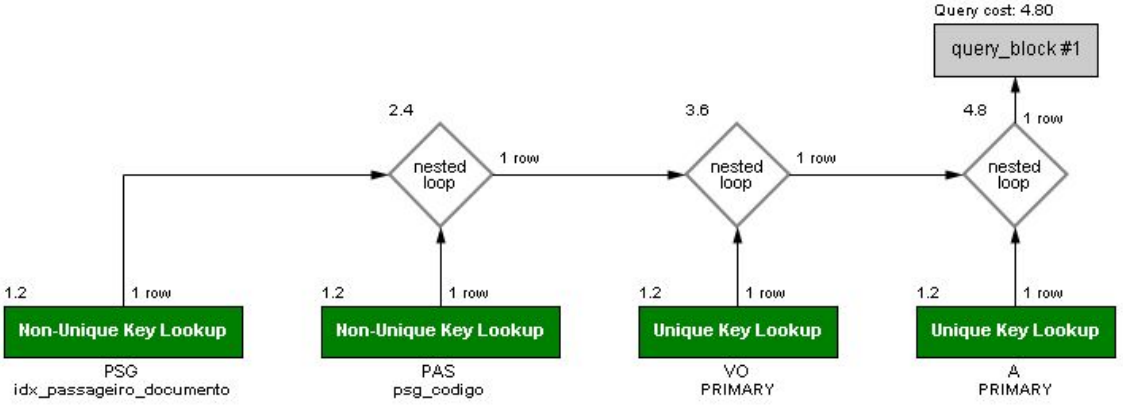
Consulta 1:

Alterando levemente a ordem dos JOINS, o banco produziu o mesmo plano de consulta.

select VO.voo_codigo as Voo, A.numero AS assento
from Voo VO
JOIN passagem PAS ON VO.voo_codigo = PAS.voo_codigo
JOIN passageiro PSG ON PSG.psg_codigo = PAS.psg_codigo, Assento A
where A.ass_codigo = PAS.ass_codigo AND PSG.documento = '86554081456';



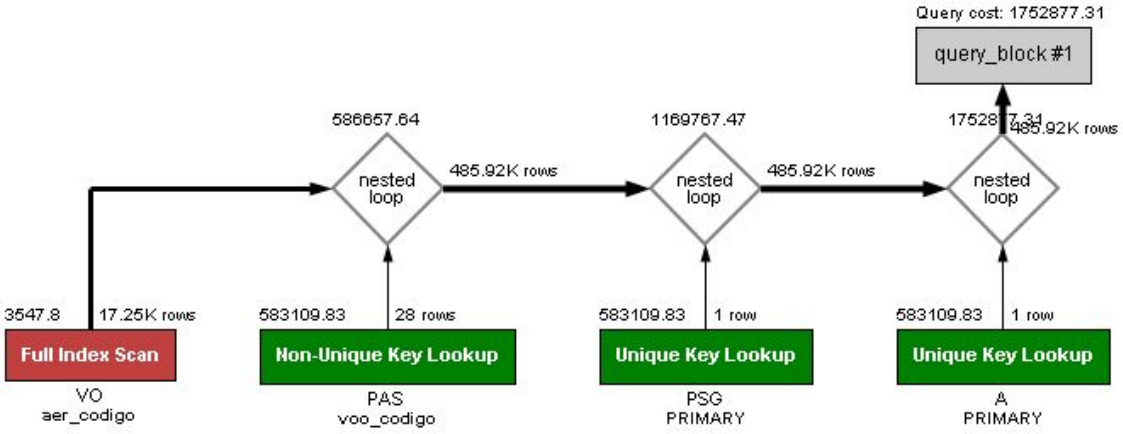
Para a mesma consulta acima, vamos criar um índice sobre a coluna documento.



Observamos que o banco agora usa o índice de documento para o loop mais externo, e isso faz com que ele não precise percorrer a tabela de passageiros, apenas o índice, aumentando a velocidade da consulta consideravelmente.

Outra modificação dessa consulta que podemos fazer é retirar a condição do documento do passageiro.

SELECT VO.voo_codigo AS Voo, A.numero AS Assento
FROM Passagem PAS
JOIN Passageiro PSG ON PAS.psg_codigo = PSG.psg_codigo
JOIN Voo VO ON VO.voo_codigo=PAS.voo_codigo, Assento A
WHERE A.ass_codigo = PAS.ass_codigo;



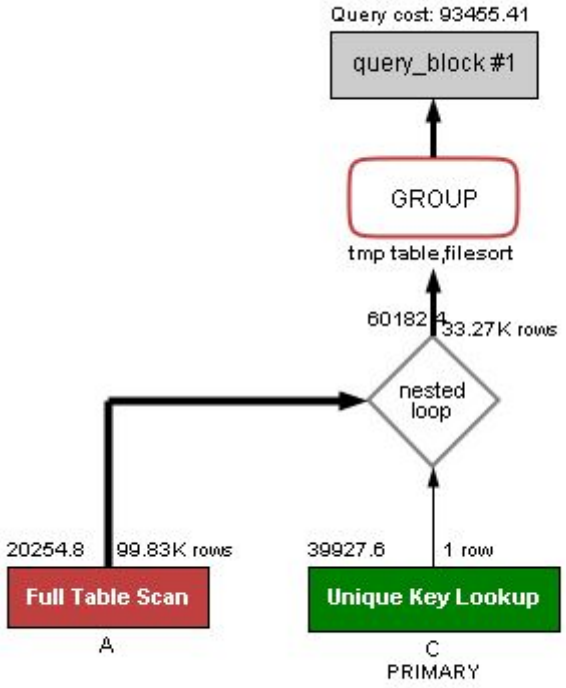
Observamos que sem a operação de seleção na tabela passageiro, a tabela com menor número de tuplas é a tabela voo, que foi a usada para ser o loop mais externo do plano de consulta.

Consulta 2:

Nessa consulta, podemos adicionar mais uma condição de busca, sobre a tabela aeronave:

SELECT C.nome, COUNT(A.aer_codigo) AS Quantidade_Aeronaves
FROM aeronave A, companhia C
WHERE C.com_codigo = A.com_companhia AND A.qtd_assentos > 60
GROUP BY C.com_codigo;

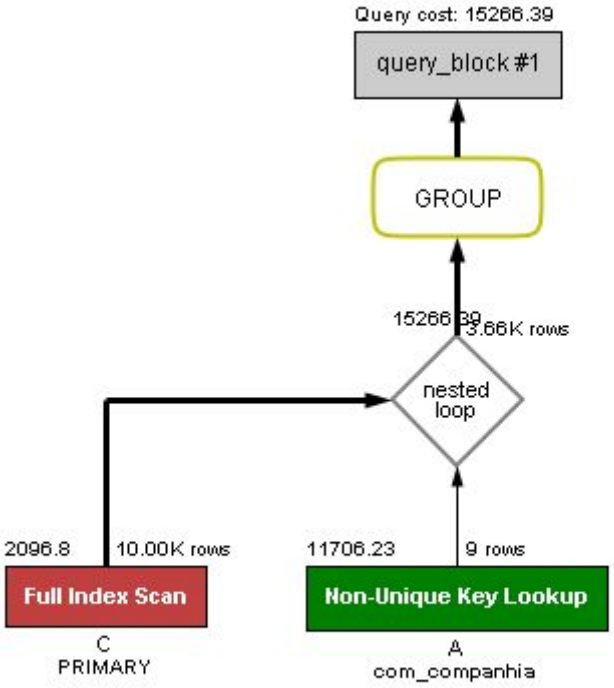
Com essa mudança, o plano de execução também é alterado:



Observamos que devido a seletividade na tabela aeronave, o loop externo mudou.

Alterando novamente as condições do WHERE:

SELECT C.nome, COUNT(A.aer_codigo) AS Quantidade_Aeronaves
FROM aeronave A, companhia C
WHERE C.com_codigo = A.com_companhia AND A.qtd_assentos > 60 AND C.nome LIKE '%x%'
GROUP BY C.com_codigo;

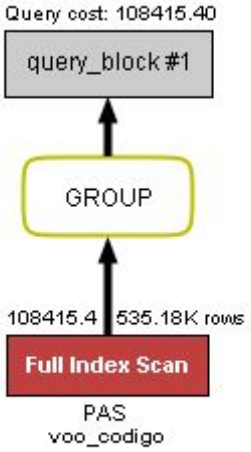


Novamente a tabela usada para o loop externo mudou, o número de tuplas da tabela companhia é muito menor do que a tabela de aeronaves.

Consulta 3:

Nessa query podemos trocar a cláusula WHERE por uma cláusula GROUP BY para mudar o comportamento da consulta:

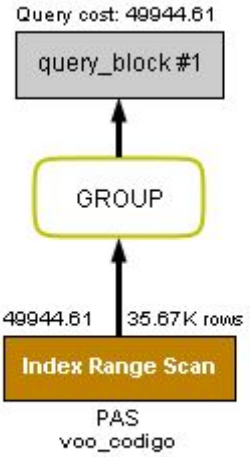
select count(PAS.psg_codigo) as num_pas
from passagem PAS
group by PAS.voo_codigo;



Nesse caso, o plano de consulta foi alterado, já que agora precisamos percorrer todas as tuplas da tabela passagem para completar a consulta.

Alterando novamente esta consulta temos:

select count(PAS.psg_codigo) as num_pas, PAS.voo_codigo
from passagem PAS
where PAS.voo_codigo > 17000
group by PAS.voo_codigo;

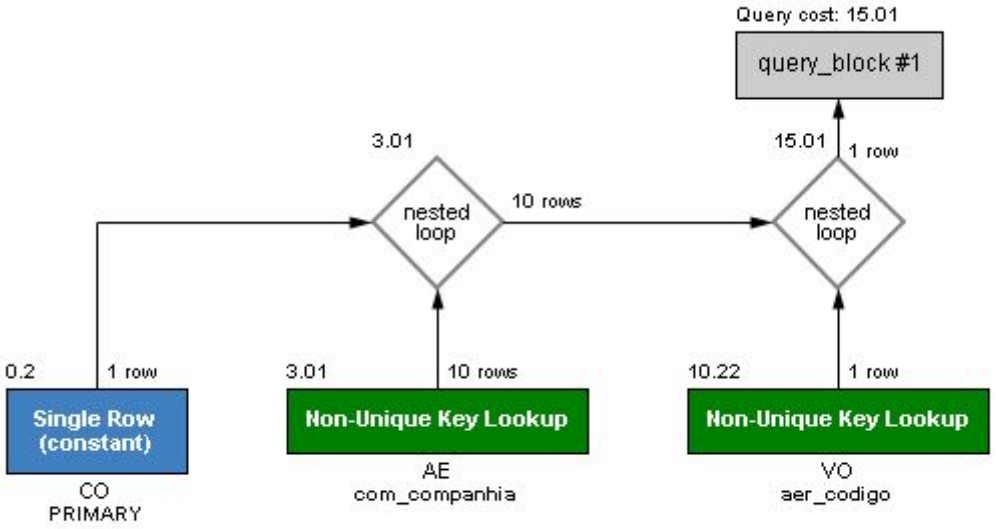


Novamente o comportamento da execução da consulta mudou, já que o banco pode usar um algoritmo diferente para realizar a busca, nesse caso, apenas uma parte do índice é usada.

Consulta 4:

Nessa consulta podemos alterar a clausula GROUP BY para uma condição extra na cláusula WHERE.

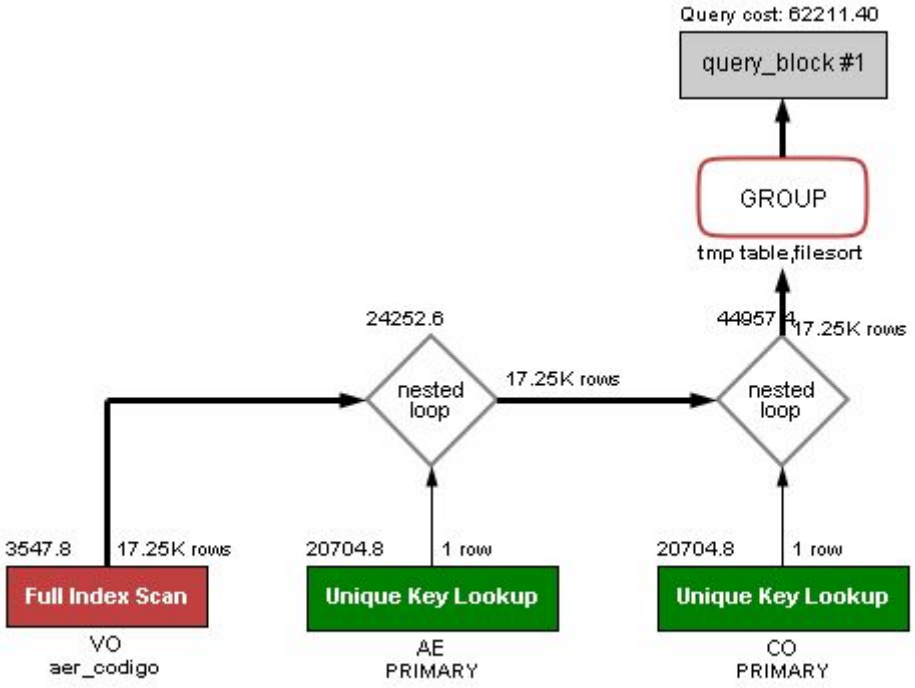
```
SELECT CO.nome, COUNT(VO.voo_codigo) AS QTD_VOOS
FROM Voo AS VO, Aeronave AE, Companhia CO
WHERE CO.com_codigo = AE.com_companhia
AND AE.aer_codigo = VO.aer_codigo
AND VO.data_hora_ini >='2000-01-01'
AND VO.data_hora_ini < '2000-12-31'
AND CO.com_codigo = 7;
```



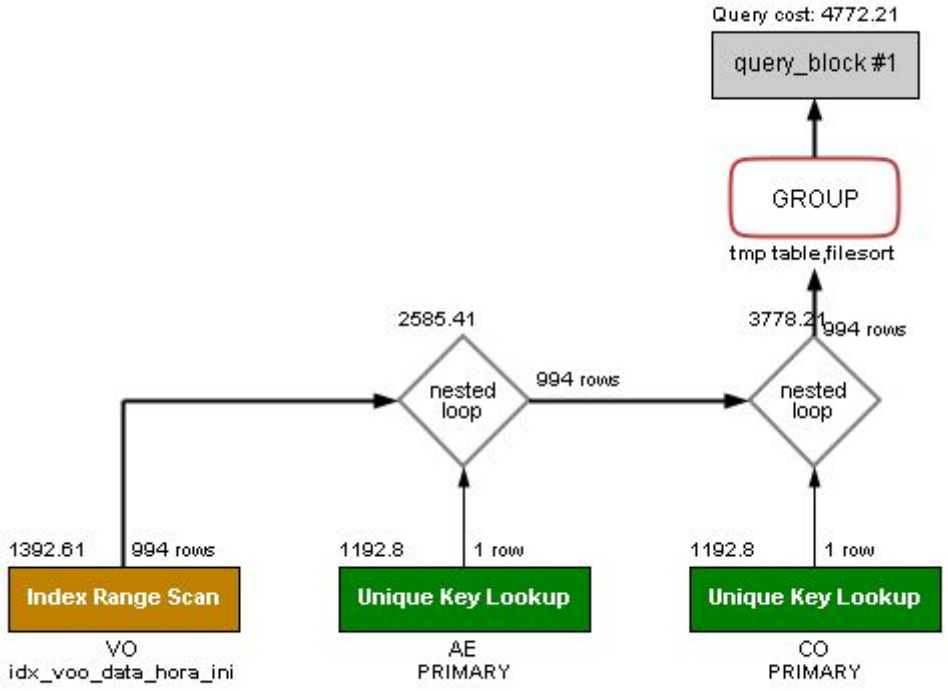
Nesse caso observamos que não é mais necessário percorrer toda a tabela companhia, e por este motivo, a tabela companhia é usada no loop mais externo no plano de consulta.

Outra modificação que podemos fazer é a remoção das condições de datas gerando a seguinte consulta:

```
SELECT CO.nome, CO.com_codigo, COUNT(VO.voo_codigo) AS QTD_VOOS
FROM Voo AS VO, Aeronave AE, Companhia CO
WHERE CO.com_codigo = AE.com_companhia
AND AE.aer_codigo = VO.aer_codigo
GROUP BY CO.com_codigo;
```



Observamos que para essa consulta, o plano de execução é igual ao da original.
Podemos também criar um índice na colunas data_hora_ini na tabela voo e utilizaremos a consulta original para testar os resultados:



O banco usou o índice para realizar a consulta e o custo da busca ficou menor.

Artefato 3 - C

Reescrita de consulta:

A consulta escolhida para ser reescrita foi a seguinte:

SELECT VO.voo_codigo AS Voo, A.numero AS Assento
FROM Passagem PAS
JOIN Passageiro PSG ON PAS.psg_codigo = PSG.psg_codigo
JOIN Voo VO ON VO.voo_codigo=PAS.voo_codigo, Assento A
WHERE A.ass_codigo = PAS.ass_codigo
AND PSG.documento = '86554081456';

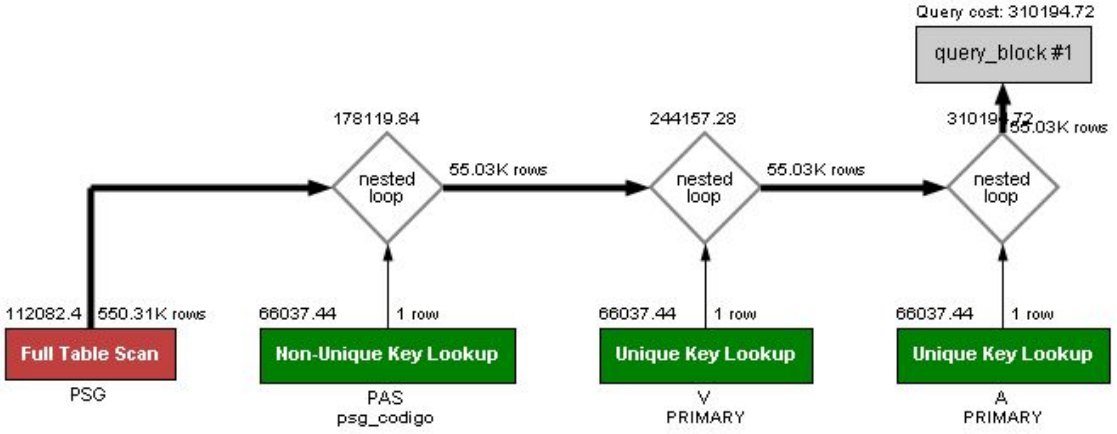
Podemos reescrevê-la para:

SELECT V.voo_codigo AS voo, A.numero AS Assento
FROM passageiro PSG, assento A, voo V, passagem PAS
WHERE PAS.psg_codigo = PSG.psg_codigo AND A.ass_codigo = PAS.ass_codigo AND V.voo_codigo = PAS.voo_codigo AND PSG.documento = '86554081456';

Ou para:

select V.voo_codigo AS voo, A.numero AS assento
from (passagem PAS JOIN passageiro PSG ON PAS.psg_codigo = PSG.psg_codigo), voo V, assento A
where A.ass_codigo = PAS.ass_codigo AND V.voo_codigo = PAS.voo_codigo AND PSG.documento = '86554081456';

Todas as três consultas geram o mesmo plano de consulta abaixo:



Artefato 3 - D

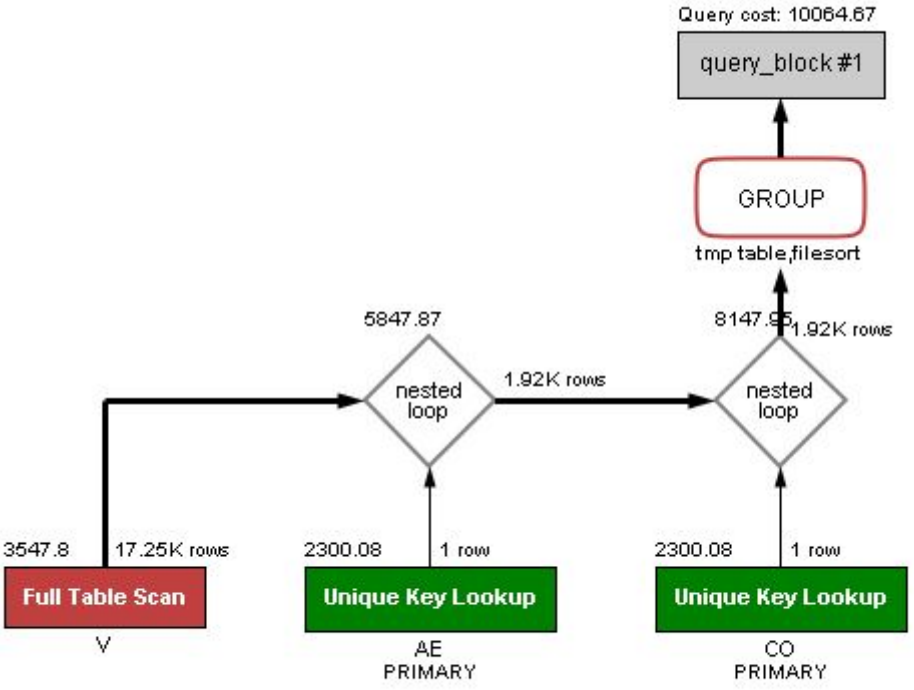
Reescrita de consulta:

A consulta escolhida para ser reescrita é a seguinte:

SELECT CO.nome, COUNT(VO.voo_codigo) AS QTD_VOOS
FROM Voo AS VO, Aeronave AE, Companhia CO
WHERE CO.com_codigo = AE.com_companhia
AND AE.aer_codigo = VO.aer_codigo
AND VO.data_hora_ini >='2000-01-01'
AND VO.data_hora_ini < '2000-12-31'
GROUP BY CO.com_codigo

Podemos reescrever a consulta utilizando subconsultas, o que gera a seguinte consulta:

select CO.nome, COUNT(V.voo_codigo) AS QTD_VOOS
FROM Aeronave AE, Companhia CO, (select V.aer_codigo, V.voo_codigo from voo V where V.data_hora_ini >= '2000-01-01' AND V.data_hora_ini < '2000-12-31') V
WHERE AE.aer_codigo = V.voo_codigo AND CO.com_codigo = AE.com_companhia
GROUP BY CO.com_codigo;



Apesar da consulta utilizar uma subconsulta, o plano de execução gerado pelo banco é igual ao da consulta original.