

Banco de Dados

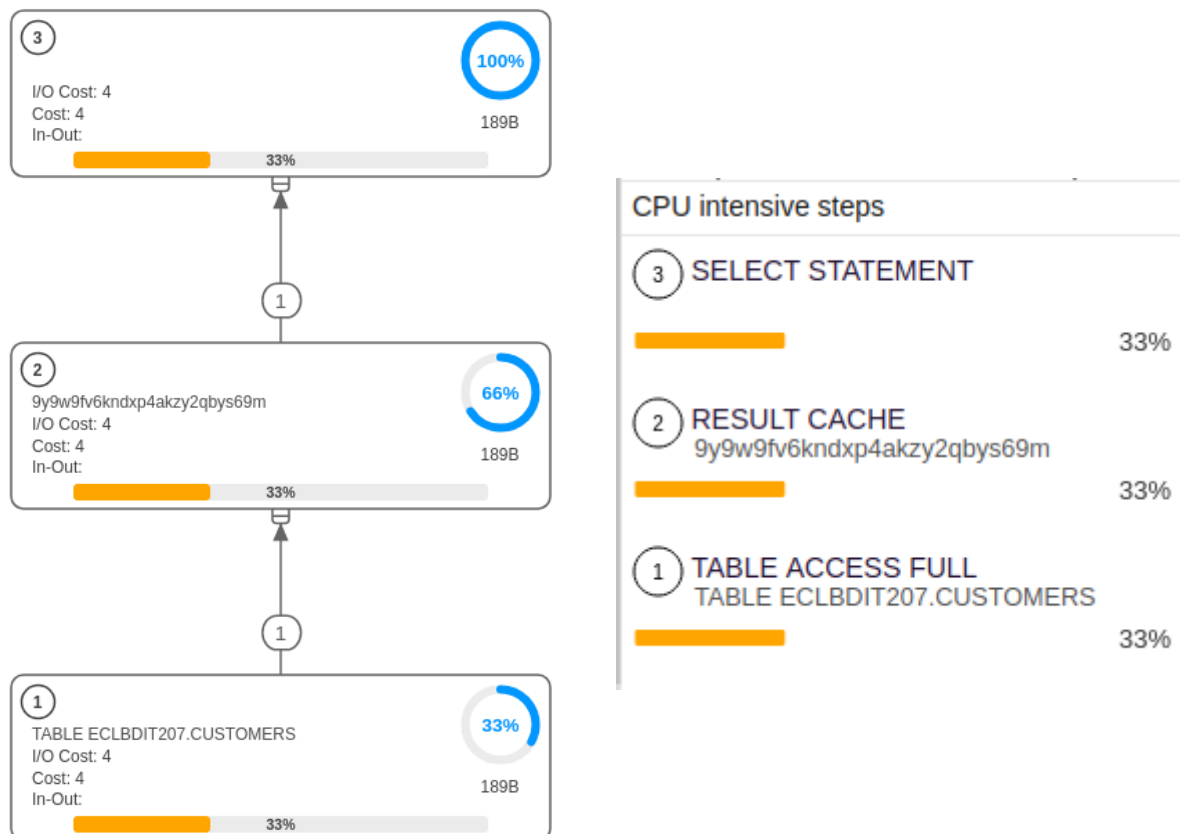
Exercício 05 - Processamento e Otimização de Consultas

Dupla:

Letícia de Oliveira Soares

Matheus Monteiro Huebra Perdigão

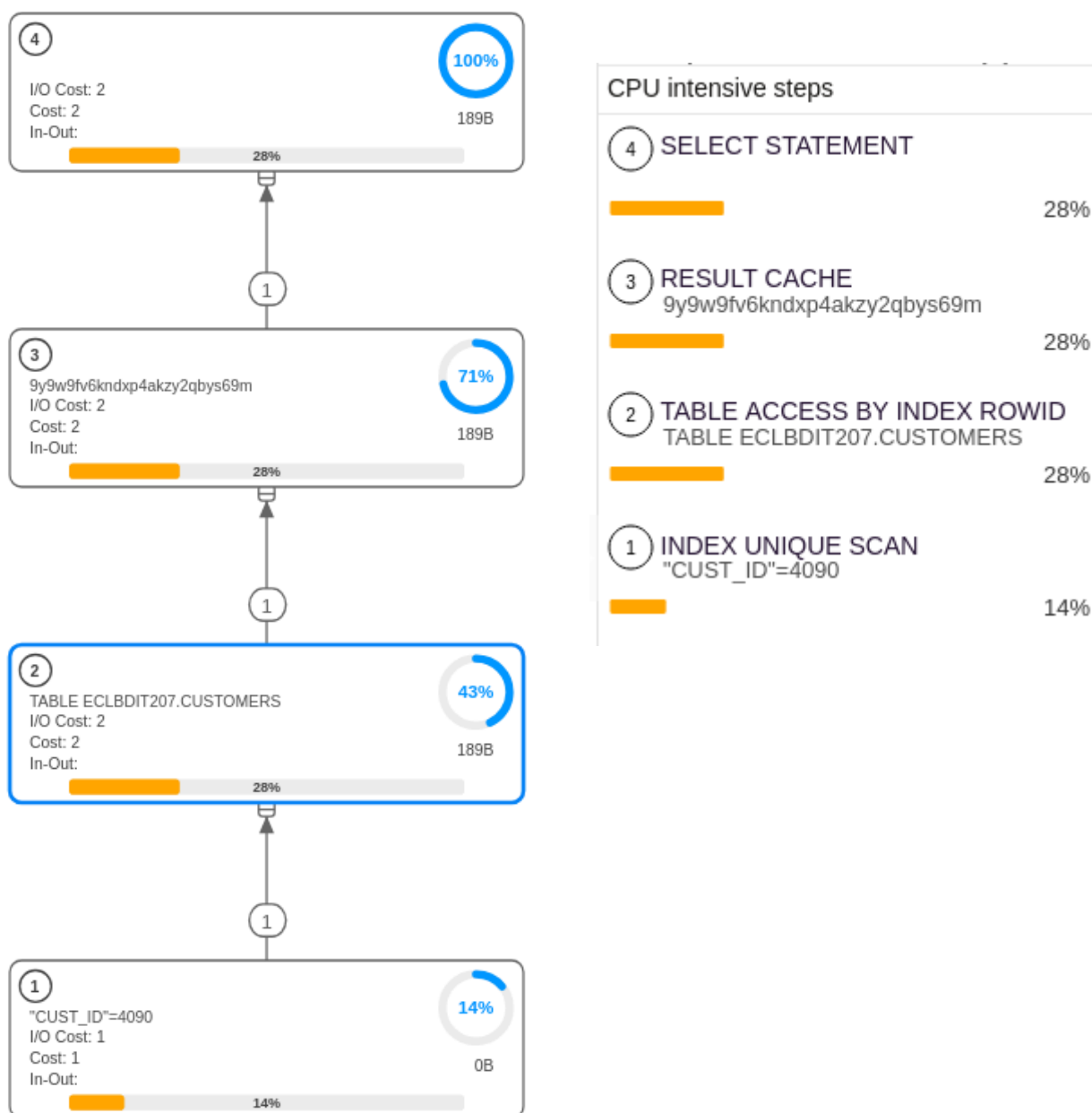
Consulta 01



A etapa Table Access Full indica que o banco de dados está realizando um scan completo na tabela 'customers' para encontrar a linha onde 'cust_id=4090'. Isso geralmente ocorre quando não há um índice adequado disponível para a coluna 'cust_id'. A etapa Result Cache indica que resultados da consulta estão sendo armazenados em cache e a última etapa representar a operação do 'SELECT'. O Cost indica .

A seletividade da consulta é de 1/55500.

Consulta 02

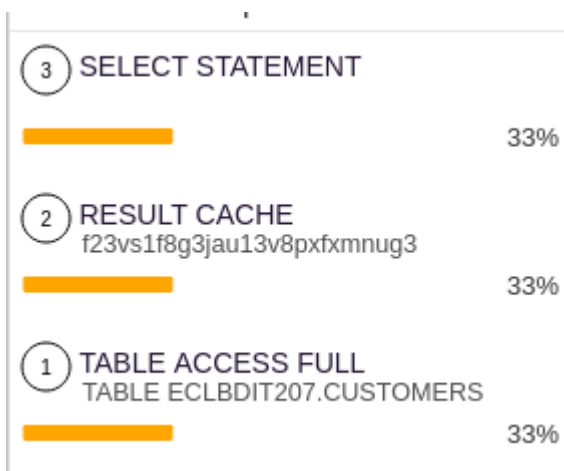
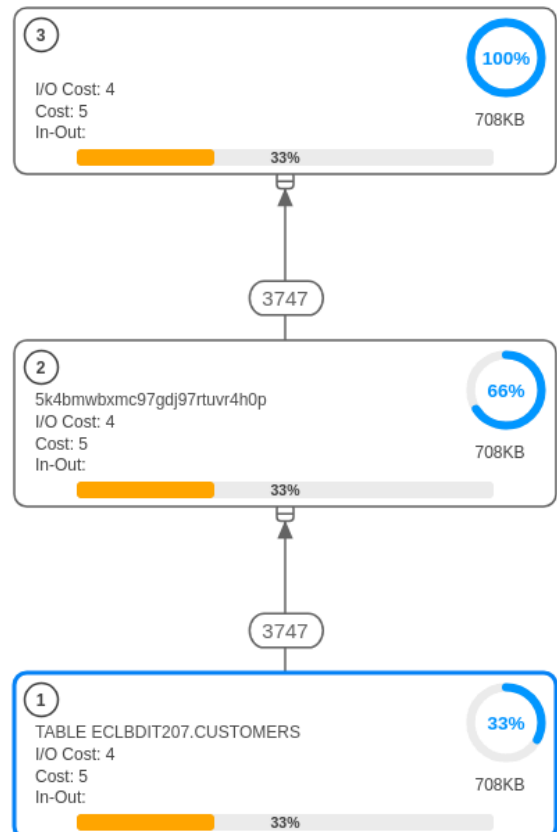
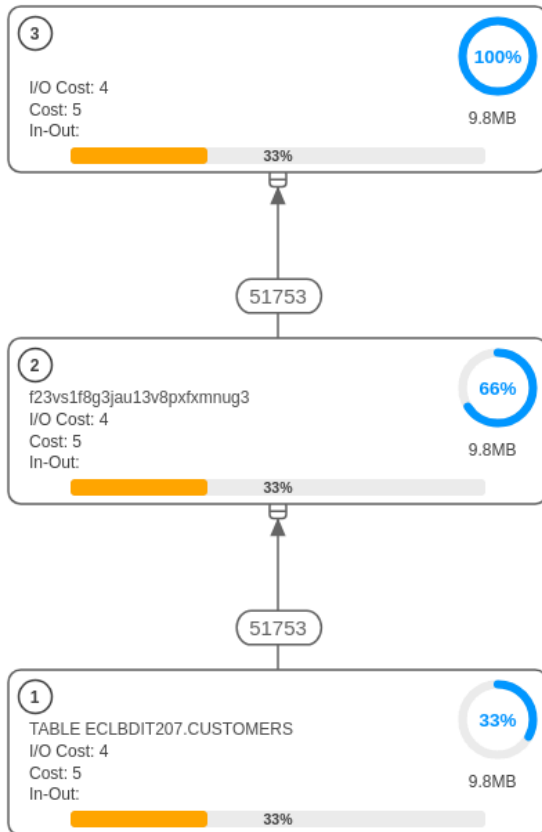


Com a transformação de 'cust_id' em chave primária, o plano de execução da consulta mudou para refletir o uso do índice na coluna. Index Unique Scan indica que o banco de dados está utilizando um índice único na coluna 'cust_id' para localizar a linha específica, e considerando que essa coluna é chave primária e a busca é eficiente, acessando diretamente a linha desejada.

A etapa Table Access By Index Row indica que após localizar a entrada no índice, o banco de dados usa o RowID (identificador de linha) para acessar diretamente a linha na tabela.

O Cost foi de 1 na primeira etapa e de 2 nas outras, mostrando que é mais eficiente do que uma leitura completa da tabela, pois permitem acessos diretos às linhas específicas através do índice. Isso reduz o custo e melhora o desempenho. A seletividade é de $\frac{1}{55500}$.

Consulta 03



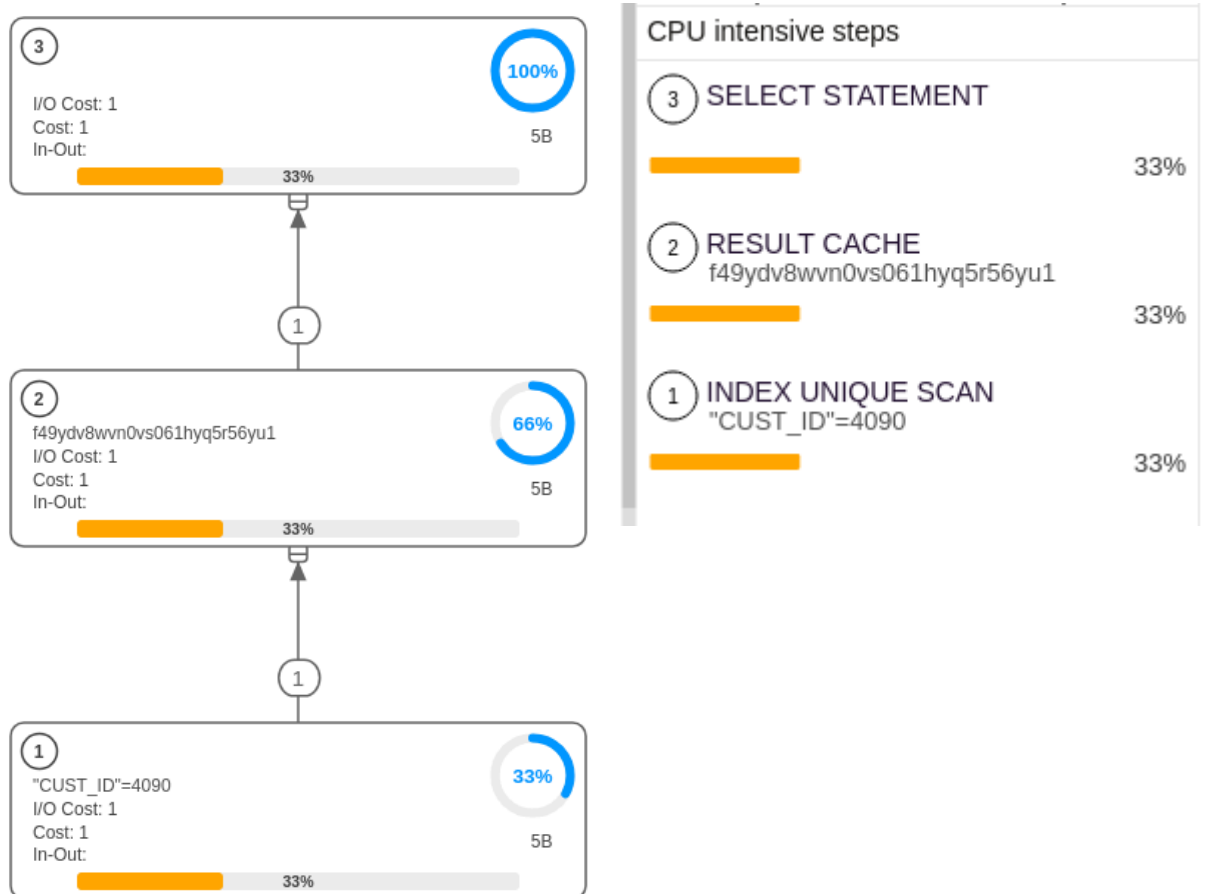
Tanto para a consulta de 'cust_id > 4090' quanto para a consulta de 'cust_id < 4090' as etapas do plano de execução foram as mesmas.

Para essas consultas, a condição envolve intervalos (cust_id > 4090 ou cust_id < 4090), o que potencialmente abrange uma grande quantidade de linhas. Mesmo que 'cust_id' seja uma chave primária, a consulta agora está procurando um grande conjunto de

valores, por isso uma varredura completa da tabela (TABLE ACCESS FULL) pode ser mais eficiente do que usar o índice e acessar muitas linhas individualmente.

A seletividade da consulta de 'cust_id > 4090' é de $\frac{51753}{55500}$. A seletividade da consulta de 'cust_id < 4090' é de $\frac{3747}{55500}$.

Consulta 04

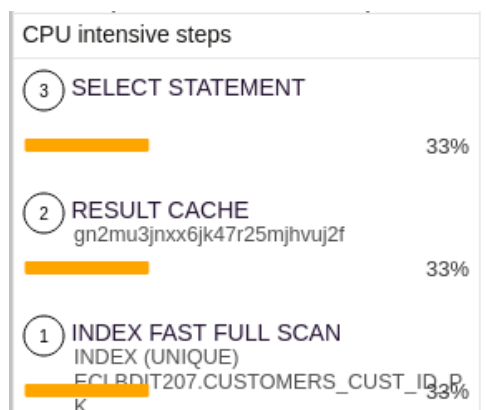


A etapa INDEX UNIQUE SCAN mostra que o banco de dados usa o índice único na coluna 'cust_id' para localizar a linha específica com cust_id = 4090. Como 'cust_id' é uma chave primária, a busca é muito eficiente, acessando diretamente o índice para encontrar a linha correspondente.

O passo TABLE ACCESS BY INDEX ROWID não foi necessário, pois a consulta está selecionando apenas a coluna 'cust_id', que já está no índice, dessa forma, não é necessário acessar a linha completa da tabela.

A seletividade é de $\frac{1}{55500}$.

Consulta 05



Tanto para a consulta de 'cust_id > 4090' quanto para a consulta de 'cust_id < 4090' as etapas do plano de execução foram as mesmas: 1. INDEX FAST FULL SCAN, 2. RESULT CACHE, 3. SELECT STATEMENT.



A varredura INDEX FAST FULL SCAN é semelhante a uma varredura completa da tabela, mas realizada no índice. O banco de dados lê todas as entradas no índice, sem usar os rowids para acessar as linhas na tabela, sendo mais eficiente.

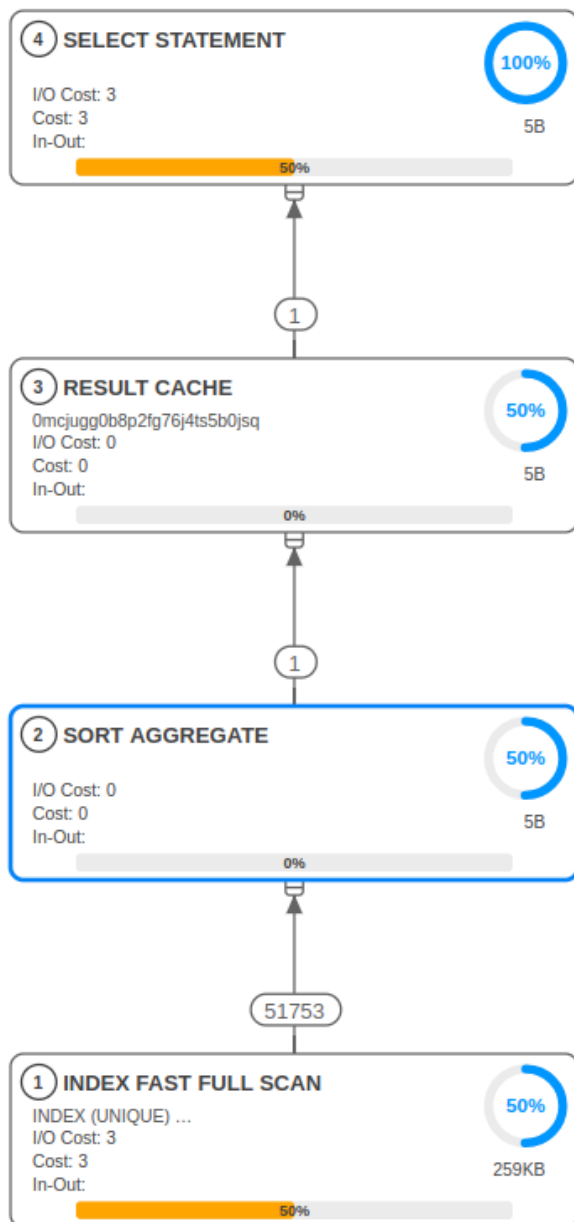
A seletividade da consulta de 'cust_id>4090' é de $\frac{51753}{55500}$. A seletividade da consulta de 'cust_id<4090' é de $\frac{3747}{55500}$.

Consulta 06

A varredura INDEX FAST FULL SCAN lê todas as entradas no índice sem usar os RowIDs para acessar as linhas na tabela. É uma varredura completa do índice que pode ser mais eficiente do que uma varredura completa da tabela.

A etapa de SORT AGGREGATE agrega os resultados da consulta. No caso de COUNT(*), ela simplesmente conta o número de linhas que atendem à condição cust_id > 4090. O custo de 0 indica que esta operação é muito barata.

A seletividade é de $\frac{51753}{55500}$.

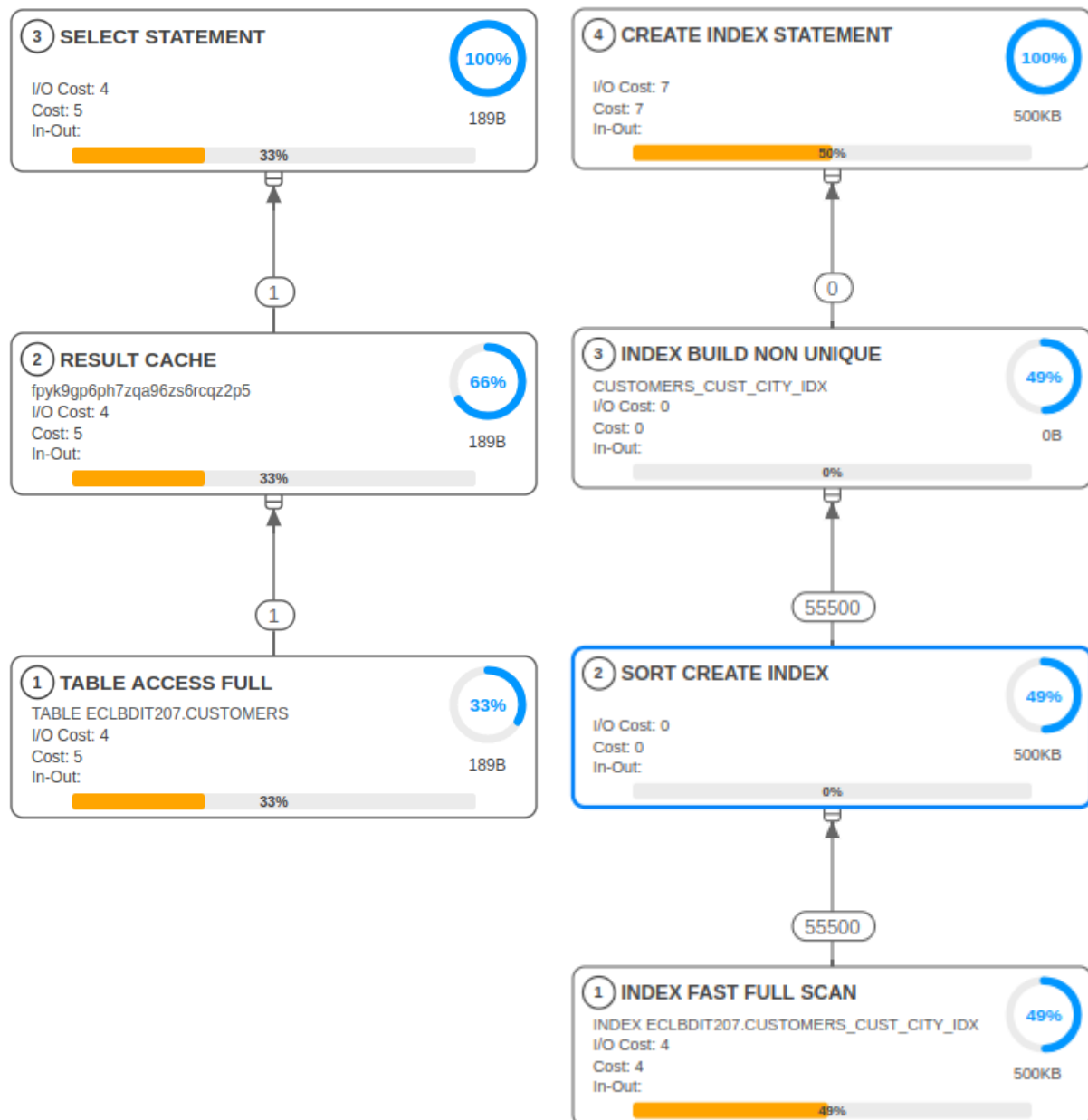


Consulta 11

Para a primeira consulta, o plano foi TABLE ACCESS FULL, no qual o banco de dados realiza uma varredura completa da tabela customers para localizar todas as linhas onde `cust_city = 'Tokyo'`, nesse caso a varredura completa da tabela é a única maneira de garantir que todas as linhas relevantes sejam encontradas. A seletividade é de $\frac{1}{55500}$.

Para a segunda consulta, o INDEX FAST FULL SCAN foi utilizado para ler rapidamente todas as entradas do índice existente (se houver) ou para ler os dados da tabela durante a criação do índice. Isso permite uma leitura eficiente e rápida dos dados necessários para criar o novo índice. Depois, na etapa do SORT CREATE INDEX os dados lidos são ordenados para a construção do índice. Esta operação organiza os valores de `'cust_city'` em ordem para otimizar a estrutura do índice. INDEX BUILD NON UNIQUE indica a criação

do índice não exclusivo na coluna cust_city. CREATE INDEX STATEMENT representa a operação de criação do índice em si, finalizando o processo de construção do índice.

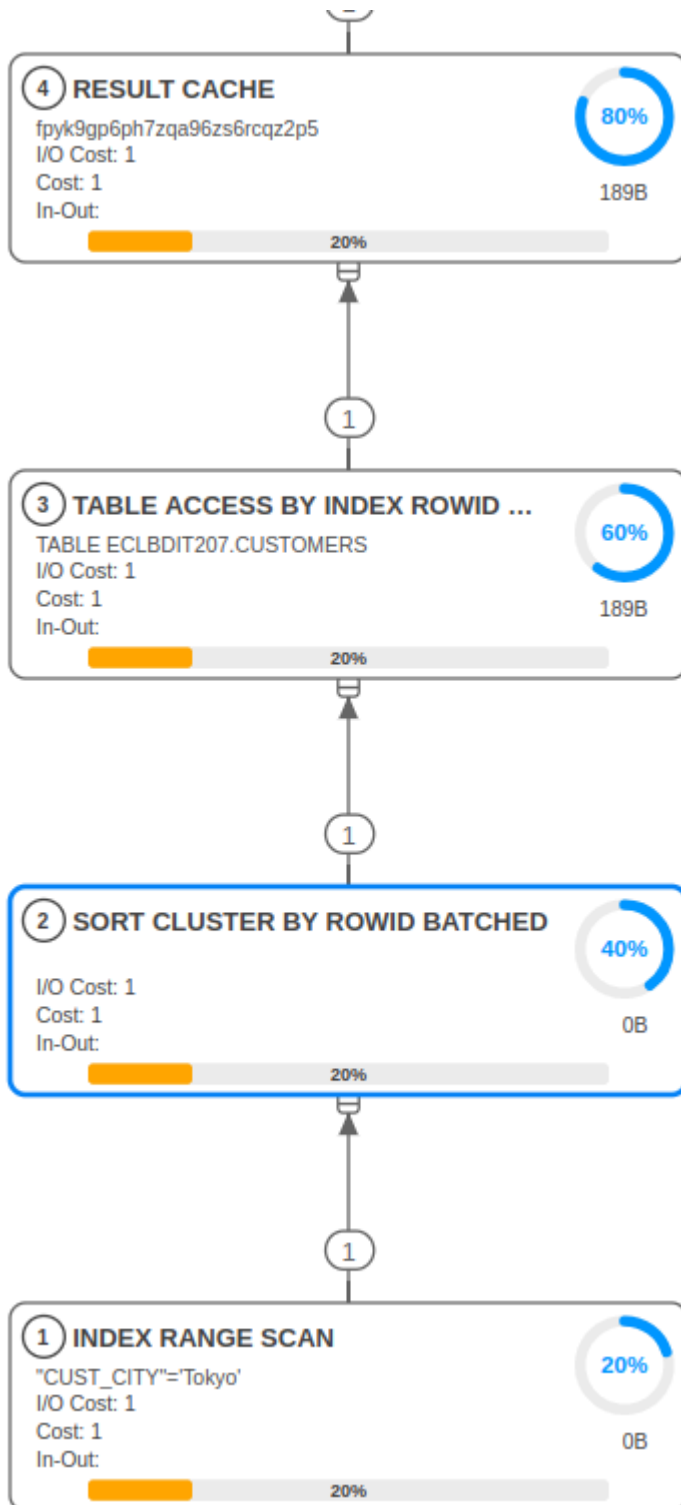


Consulta 13

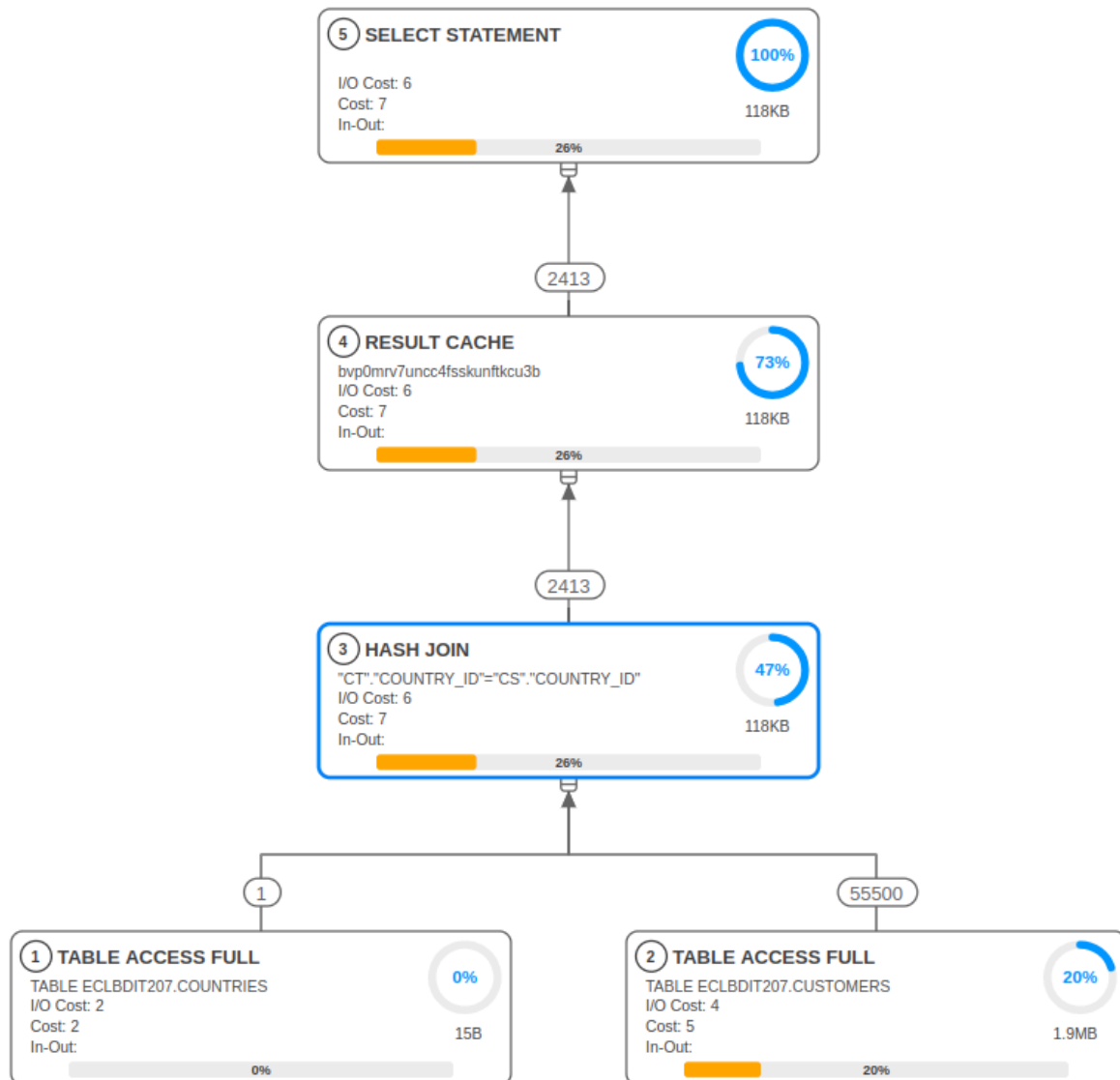
Na etapa do INDEX RANGE SCAN o banco de dados agora utiliza o índice customers_cust_city_idx para realizar uma varredura no índice. A operação Index Range Scan é eficiente para encontrar as linhas onde cust_city = 'Tokyo'.

SORT CLUSTER BY ROWID agrupa e ordena os RowIDs obtidos do índice para otimizar o acesso às linhas na tabela. O "Batched" indica que os RowIDs são processados em lotes, o que pode melhorar a eficiência.

TABLE ACCESS BY INDEX ROWID BATCHED usa os RowIDs obtidos do índice para acessar as linhas correspondentes na tabela.



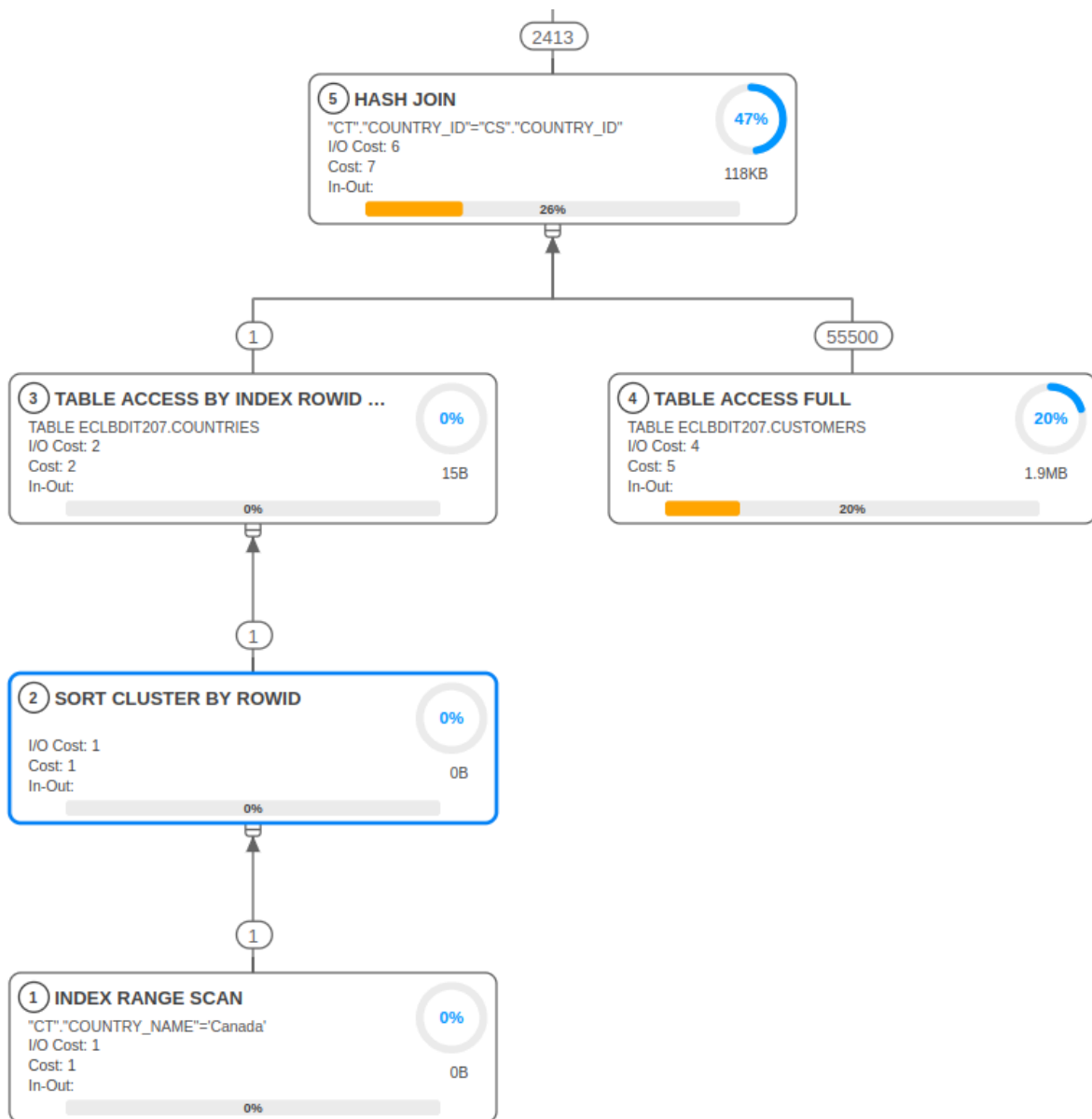
Consulta 15



As operações de TABLE ACCESS FULL realizam uma varredura completa na tabela customers para encontrar todas as linhas e uma varredura completa da tabela countries para localizar todas as linhas onde country_name = 'Canada'. O custo 5 indica que esta varredura é relativamente mais cara em comparação com a varredura na tabela customers.

HASH JOIN é utilizado para combinar as tabelas customers e countries com base na coluna country_id.

Consulta 16



O INDEX RANGE SCAN indica que o banco de dados usa um índice na coluna 'country_name' para encontrar as linhas na tabela countries onde country_name = 'Canada'.

O SORT CLUSTER BY ROWID é usado para ordenar esses RowIDs, otimizando a recuperação dos dados na tabela countries.

TABLE ACCESS BY INDEX ROWID usa os RowIDs obtidos do índice para acessar as linhas correspondentes na tabela countries. Assim, é possível recuperar as linhas completas que foram encontradas através do índice.

TABLE ACCESS FULL realiza uma varredura completa da tabela customers, pois a consulta requer a combinação dos resultados da tabela customers com os resultados da tabela countries usando a coluna country_id.

HASH JOIN é usado para combinar as tabelas customers e countries, associando os registros com base na coluna country_id.