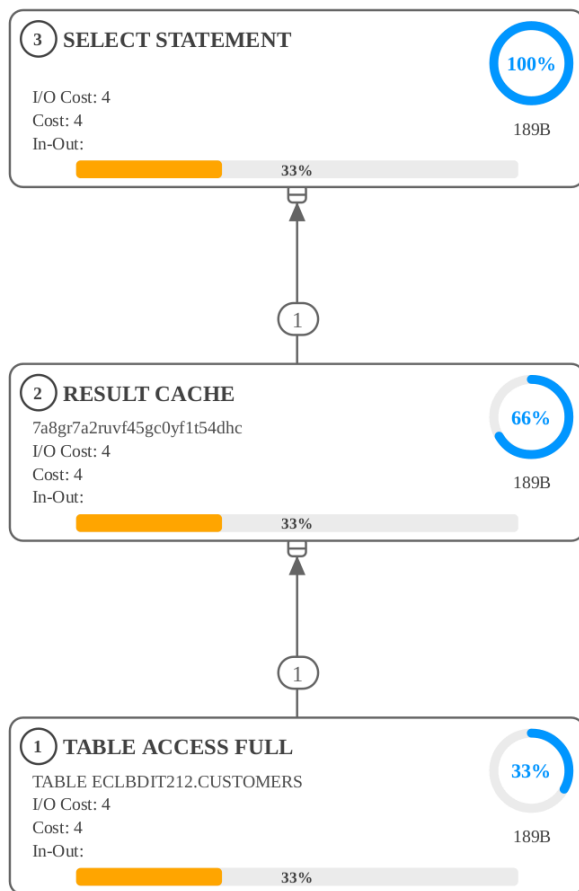


Dupla: Pedro Augusto de Portilho Ronzani - Ulisses Andrade Carvalho

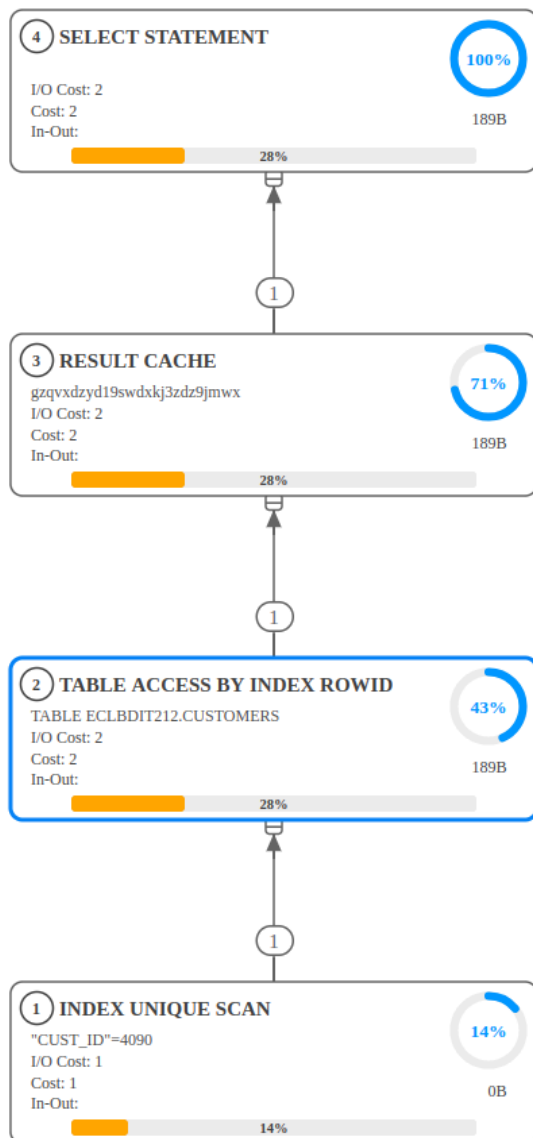
Consulta 01: `select * from customers where cust_id=4090`



Como essa consulta foi feita em uma coluna da tabela sem indexação, a procura da linha com `cust_id = 4090` requer uma busca linear por toda a tabela, gerando o "TABLE ACCESS FULL".

Seletividade = $1/55500 = 0,000018$.

Consulta 02: `select * from customers where cust_id=4090;`

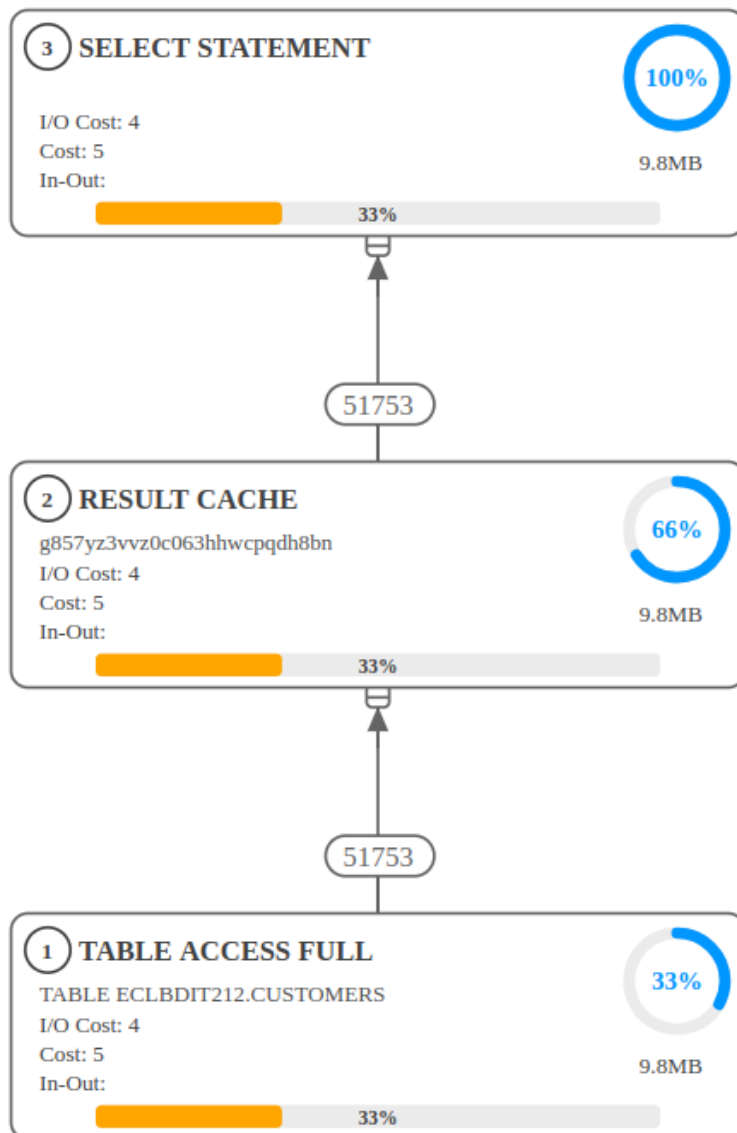


Como essa consulta foi feita na coluna `cust_id`, que agora é uma primary key, é feito um acesso único para localizar o rowid onde a entrada específica é 4090. Em seguida, o plano acessa a tabela 'CUSTOMERS' usando o rowid para recuperar as colunas necessárias.

Seletividade = $1/55500 = 0,000018$.

Consulta 03:

```
select * from customers where cust_id>4090;  
select * from customers where cust_id<4090;
```

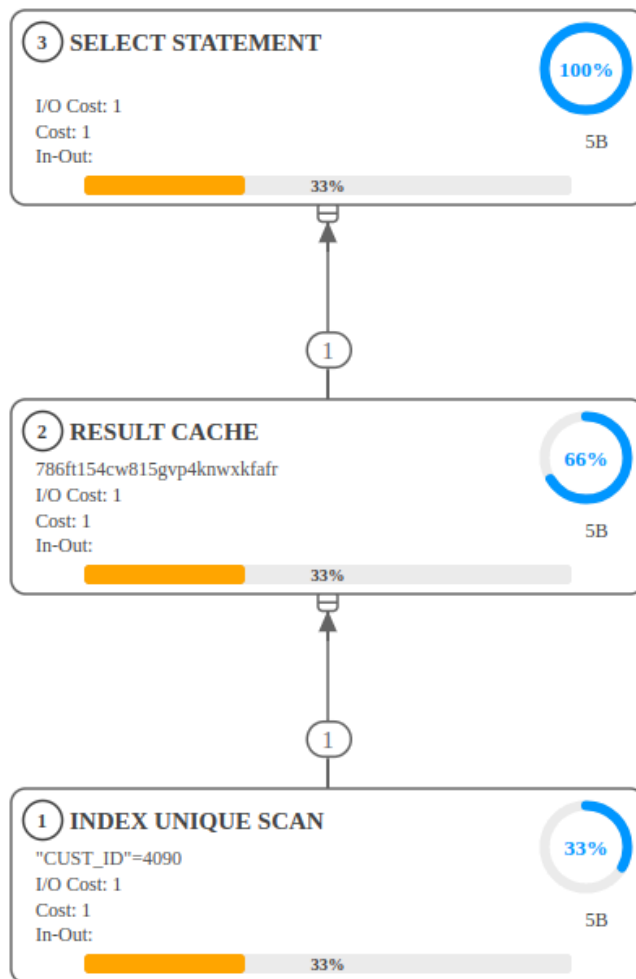


O plano realiza um acesso completo à tabela (TABLE ACCESS FULL), pois a consulta seleciona grande parte das linhas da tabela, dessa forma o índice não é eficiente para essa consulta.

Seletividade1: $51410/55500 = 0,9263$

Seletividade2: $4089/55500 = 0,0737$

Consulta 4: `select cust_id from customers where cust_id=4090;`

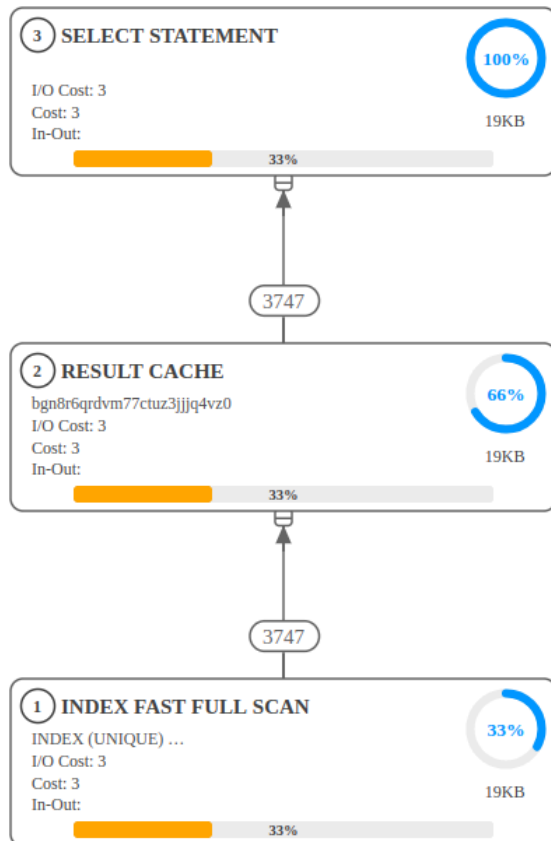


Explicação do Plano: O plano de execução utiliza o índice único CUSTOMERS_CUST_ID_PK para acessar a linha com `cust_id=4090` diretamente, sem varrer a tabela inteira, mas apenas o `cust_id` é retornado.

Seletividade = $1/55500 = 0,000018$.

Consulta 05:

```
select cust_id from customers where cust_id<4090;  
select cust_id from customers where cust_id>4090;
```

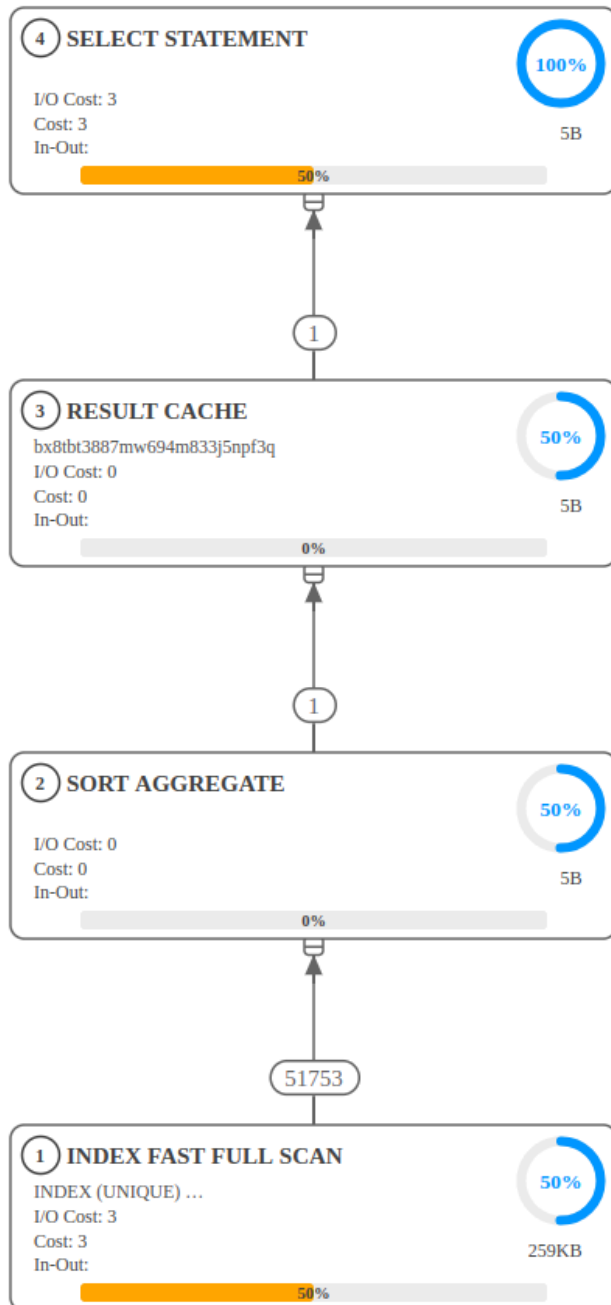


Explicação do Plano: O INDEX FAST FULL SCAN é usado para acessar rapidamente todas as entradas do índice CUSTOMERS_CUST_ID_PK que satisfazem a condição `cust_id<4090`. Esse tipo de varredura permite acessar todas as entradas do índice sem acessar a tabela base, o que é mais rápido quando apenas as colunas indexadas são necessárias.

Seletividade1: $51410/55500 = 0,9263$

Seletividade2: $4089/55500 = 0,0737$

Consulta 6: `select count(*) from customers where cust_id>4090;`



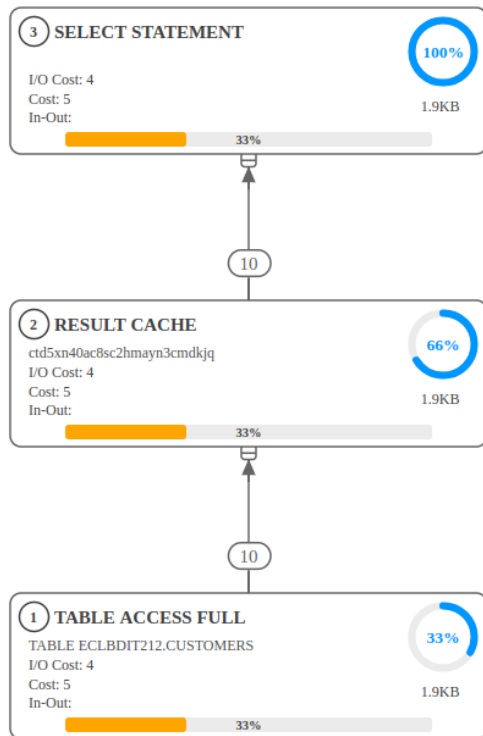
INDEX FAST FULL SCAN: A consulta começa com um INDEX FAST FULL SCAN no índice CUSTOMERS_CUST_ID_PK para recuperar rapidamente todas as entradas que satisfazem a condição `cust_id>4090`. Esse tipo de varredura é eficiente porque permite acessar o índice sem precisar acessar a tabela base.

SORT AGGREGATE: Em seguida, os resultados do índice são agregados para contar o número de linhas que satisfazem a condição.

Seletividade: $51410/55500 = 0,9263$

Consulta 11:

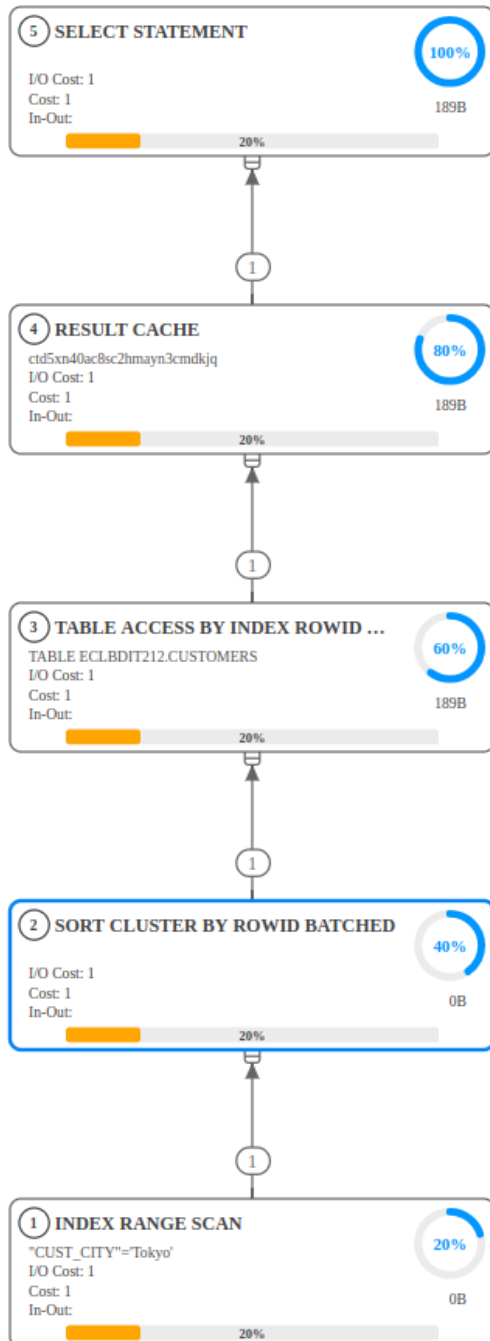
```
select * from CUSTOMERS where CUST_CITY='Tokyo';
```



Como essa consulta foi feita em uma coluna da tabela sem indexação, a procura da linha com `CUST_CITY = 'Tokyo'` requer uma busca linear por toda a tabela, gerando o “TABLE ACCESS FULL”.

Seletividade: $16/55500 = 0,000288$

Consulta 13: `select * from CUSTOMERS where CUST_CITY='Tokyo';`



INDEX RANGE SCAN:

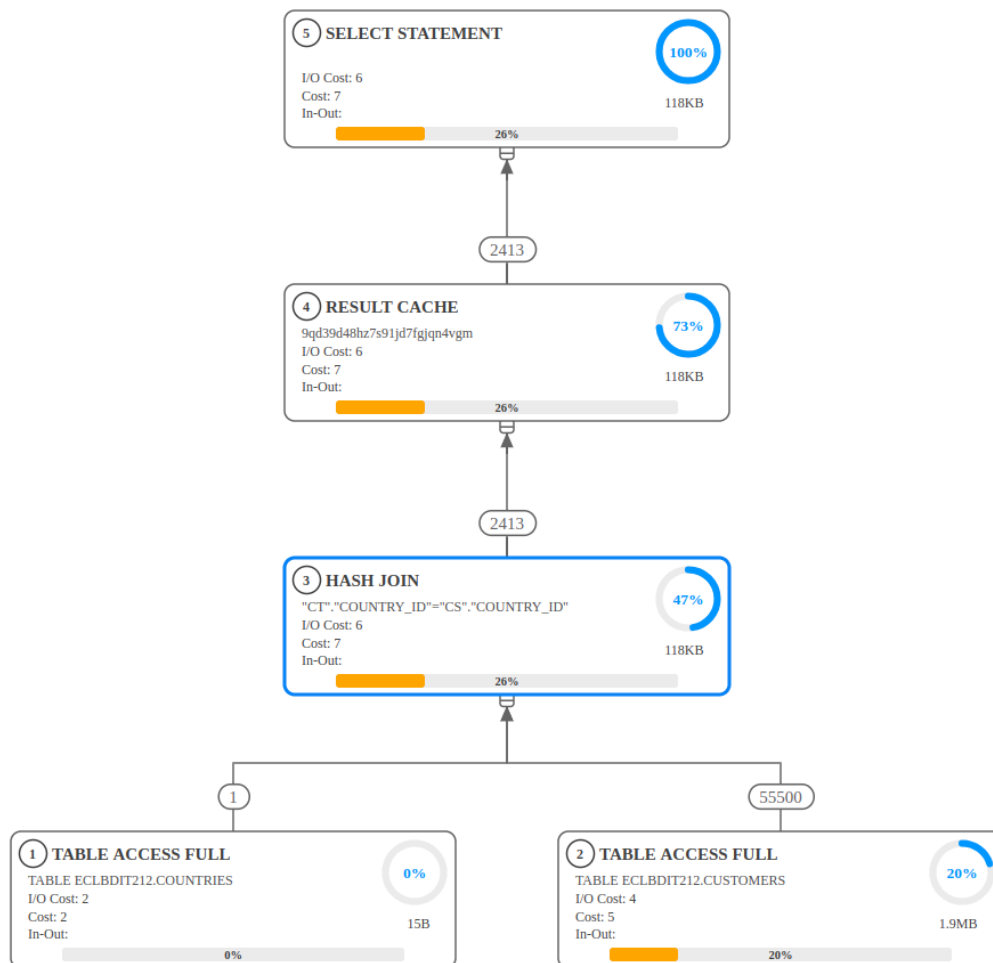
Operação: O plano de execução começa com uma varredura por intervalo no índice CUSTOMERS_CUST_CITY_IDX.

Detalhe: Esta operação utiliza o índice para localizar todas as entradas onde CUST_CITY é igual a 'Tokyo'.

Seletividade: $16/55500 = 0,000288$

Consulta 15:

```
select cs.cust_id, cs.cust_first_name, cs.cust_last_name,  
cs.cust_city , ct.country_name  
from customers cs join countries ct on ct.country_id=cs.country_id  
where ct.country_name='Canada';
```



Explicação:

TABLE ACCESS FULL (COUNTRIES):

Operação: A consulta começa com uma varredura completa na tabela COUNTRIES.
Detalhe: Esta operação varre todas as linhas da tabela COUNTRIES para encontrar todas as entradas.

TABLE ACCESS FULL (CUSTOMERS):

Operação: Em paralelo, ocorre uma varredura completa na tabela CUSTOMERS.

Detalhe: Esta operação varre todas as linhas da tabela CUSTOMERS para encontrar todas as entradas.

HASH JOIN:

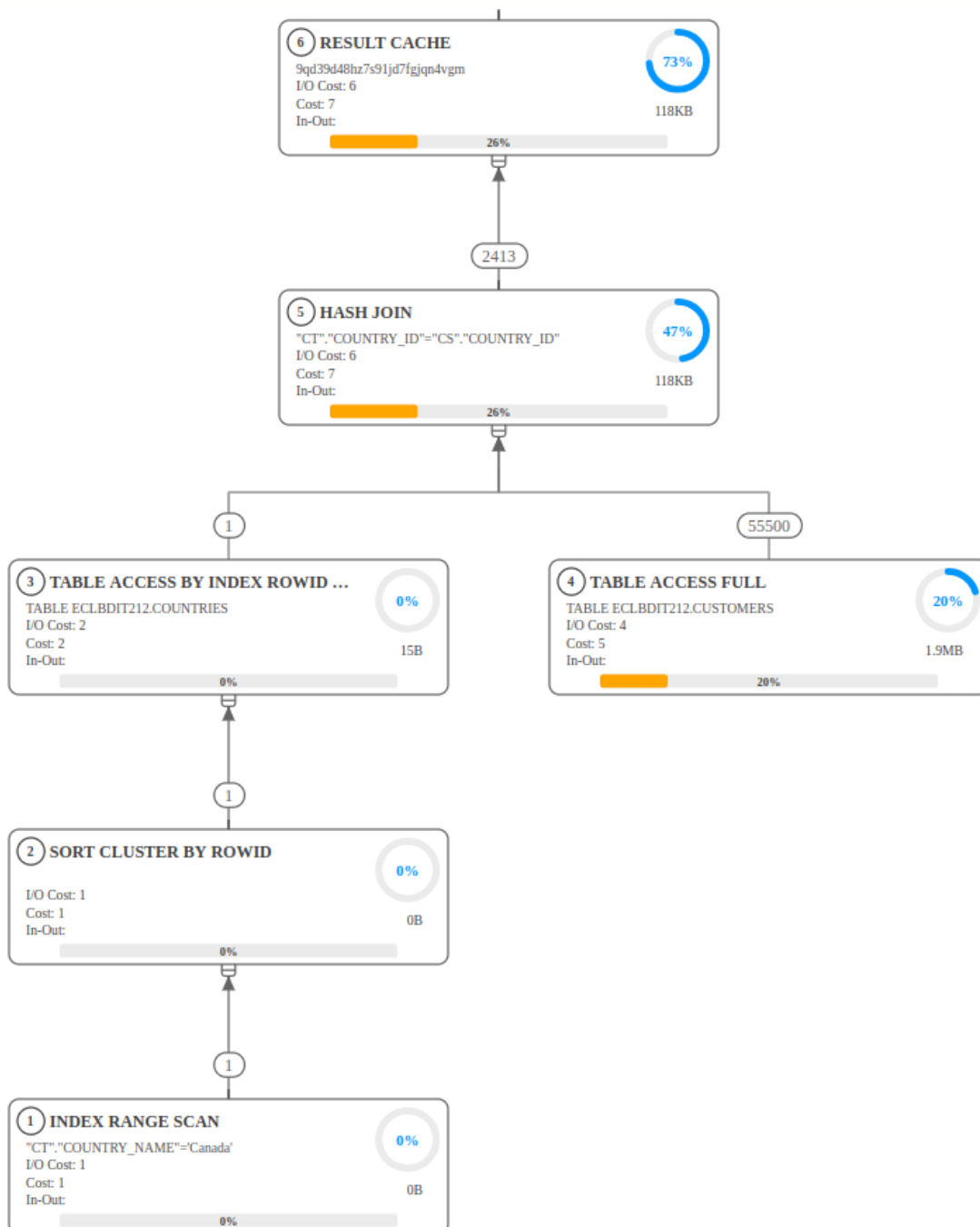
Operação: As tabelas COUNTRIES e CUSTOMERS são então unidas utilizando uma junção por hash baseada no campo country_id.

Detalhe: Esta operação é eficiente para junções de grandes conjuntos de dados onde ambos os lados da junção precisam ser completamente varridos.

Seletividade: $2010/55500 = 0,0362$

Consulta 16:

```
select cs.cust_id, cs.cust_first_name, cs.cust_last_name, cs.cust_city ,  
ct.country_name  
from customers cs join countries ct on ct.country_id=cs.country_id  
where ct.country_name='Canada';
```



Index Range Scan é eficiente para localizar rapidamente os registros relevantes na tabela countries onde country_name='Canada'.

Sort Cluster by ROWID e Table Access by Index ROWID permitem um acesso direto e eficiente às linhas específicas da tabela countries.

Table Access Full na tabela customers indica que toda a tabela está sendo lida, o que pode ser caro em termos de I/O, especialmente se a tabela for grande.

Hash Join é uma escolha eficiente para juntar grandes conjuntos de dados, especialmente quando há uma diferença significativa no tamanho das tabelas sendo juntadas.

Seletividade: $2010/55500 = 0,0362$