



CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS

Departamento de Computação - DECOM

Laboratório de Banco de Dados I

Prova 02

Ana Clara Cunha Lopes - Engenharia da Computação
Thiago Ribeiro Corrêa - Engenharia da Computação

Professor: Evandrino Barros

Belo Horizonte

09 de agosto de 2024

Questão 1:

As tabelas a seguir foram criadas no BD do usuário Ana:

```
create table Cliente (  
    id_cliente NUMBER(3) PRIMARY KEY NOT NULL,  
    nome_cliente VARCHAR2(30) NOT NULL  
  
);
```

```
create table Pedido(  
    id_pedido NUMBER(3) PRIMARY KEY NOT NULL,  
    id_cliente NUMBER(3) NOT NULL,  
    item VARCHAR2(50) NOT NULL,  
    qtd_item NUMBER(5) NOT NULL,  
    valor_item NUMBER(5) NOT NULL  
  
);
```

```
Alter table Pedido add constraint fk_pedido FOREIGN KEY (id_cliente)  
REFERENCES Cliente(id_cliente);
```

```
Insert into Cliente VALUES(1, 'Thiago Ribeiro');  
Insert into Cliente VALUES(2, 'Ana Clara');
```

```
Insert into Pedido VALUES(11, 1, 'Arroz', 1, 40);  
Insert into Pedido VALUES(22, 2, 'Feijão', 1, 20);
```

```
commit;
```

```
select * from cliente;  
select * from pedido;
```

– Acesso liberado para o usuário Thiago fazer consultas na tabela do usuário Ana:
Grant select on ECLBDIT202.CLIENTE to ECLBDIT214;

. **Propriedade de Isolamento (I)**: vários usuários podem ler e gravar na mesma tabela ao mesmo tempo, mas as transações são isoladas para que as simultâneas não interfiram ou afetem umas às outras. Na verdade, cada solicitação é tratada como se estivesse ocorrendo de forma independente, mesmo que ocorram simultaneamente.

Exemplo:

Usuário Ana:

```
Insert into Cliente VALUES(3, 'João Silva');
select * from ECLBDIT202.cliente;
```

ID_CLIENTE	NOME_CLIENTE
1	1 Thiago Ribeiro
2	2 Ana Clara
3	3 João Silva

Usuário Thiago:

```
select * from eclbdit202.cliente;
```

ID_CLIENTE	NOME_CLIENTE
1	1 Thiago Ribeiro
2	2 Ana Clara

Explicação: O usuário Ana está realizando uma inserção na tabela Cliente, enquanto isso, o usuário Thiago realiza o comando select para consultar a tabela Cliente. A propriedade de isolamento se exemplifica nesse caso, pois a consulta da tabela cliente para o usuário Ana terá um resultado diferente da consulta feita pelo usuário Thiago. Lembrando que isso aconteceu pois o usuário Ana não executou o comando “commit” para finalizar a transação.

OBS: Ao final das transações do item anterior, o usuário Ana executou o comando “rollback”, fazendo com que a inserção não persistisse no banco de dados.

. **Propriedade de Durabilidade (D):** garante que as alterações de dados feitas por transações executadas com sucesso sejam preservadas, mesmo em caso de falha do sistema.

Exemplo:

Usuário Ana:

```
Insert into Cliente VALUES(4, 'Maria Pereira');
commit;
select * from ECLBDIT202.cliente;
```

ID_CLIENTE	NOME_CLIENTE
1	4 Maria Pereira
2	1 Thiago Ribeiro
3	2 Ana Clara

Explicação: O usuário Ana realizou uma inserção na tabela Cliente e confirmou a transação com o comando “commit”. Como a transação foi concluída, ela não pode ser desfeita, o que demonstra a propriedade de durabilidade. O usuário Thiago realiza a consulta no bd da Ana e consegue visualizar as mesmas informações apresentadas anteriormente.

Usuário Thiago:

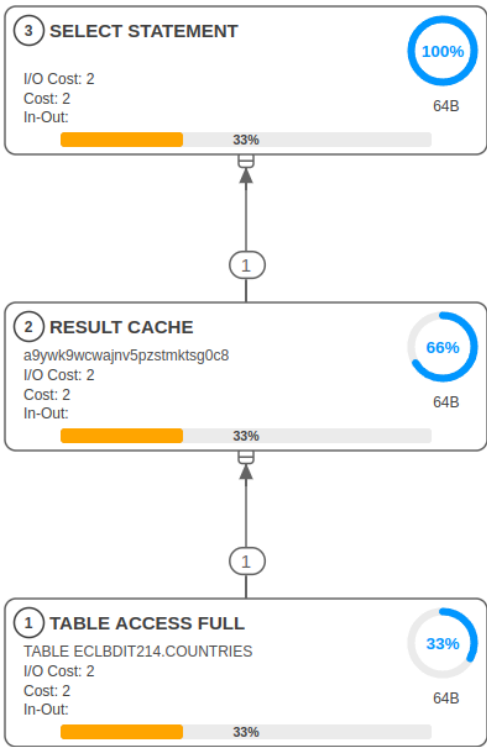
```
select * from eclbdt202.cliente;
```

	ID_CLIENTE	NOME_CLIENTE
1	4	Maria Pereira
2	1	Thiago Ribeiro
3	2	Ana Clara

Questão 2 - a)

```
select * from countries
where country_id = 52771;
```

	COUNTRY_ID	COUNTRY_ISO_COI	COUNTRY_NAME	COUNTRY_SUBREC	COUNTRY_SUBREC	COUNTRY_REGION	COUNTRY_REGION	COUNTRY_TOTAL	COUNTRY_TOTAL	COUNTRY_NAME_HIS
1	52771	CN	China	Asia	52793	Asia	52802	World total	52806	(nulo)



A consulta SQL busca todas as colunas da tabela countries para encontrar country_id = 52771. O plano de execução possui três etapas: TABLE ACCESS FULL, RESULT CACHE, e SELECT STATEMENT.

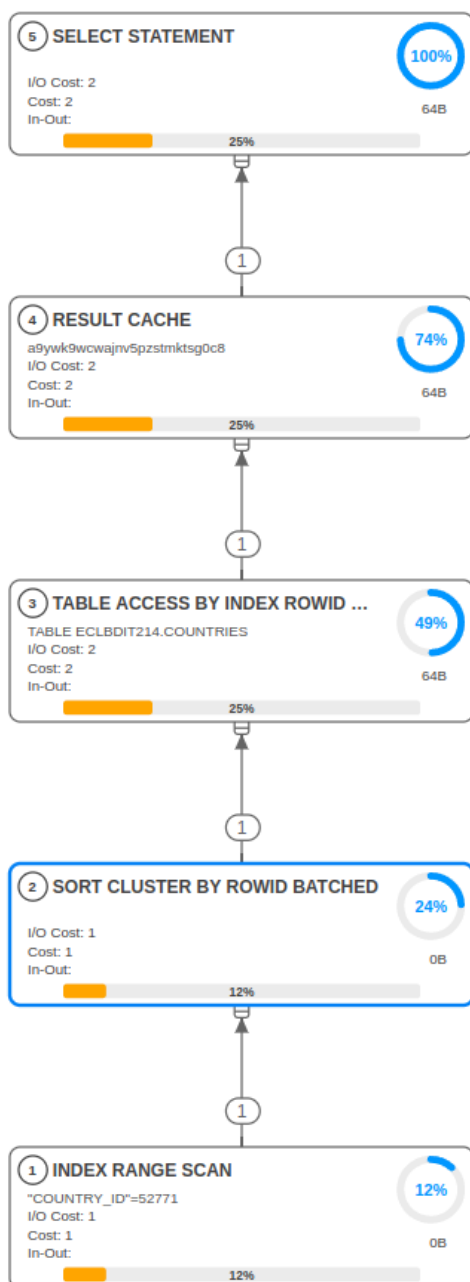
Na TABLE ACCESS FULL, a consulta realiza um acesso completo à tabela, lendo todas as linhas para encontrar a linha desejada. Este é um processo custoso em termos de I/O, representando 33% do custo total da consulta. O RESULT CACHE utiliza uma cache de resultados para armazenar dados intermediários (ou finais), acelerando execuções subsequentes da mesma consulta, representando 66% do

custo total. Já o SELECT STATEMENT engloba todas as operações anteriores, resultando em um custo total de 2.

A seletividade da consulta, calculada como a razão entre o número de linhas que satisfazem a condição de busca e o número total de linhas na tabela, é baixa (0,043). Isso indica que a consulta é altamente seletiva, retornando apenas uma linha entre 23.

Questão 2 - b)

```
create index country_id_index on countries(country_id);  
select * from countries  
where country_id = 52771;
```



A consulta utiliza um plano de execução que inicia com um INDEX RANGE SCAN para encontrar as entradas na tabela countries com country_id = 52771, com um custo baixo de 1. Em seguida, a consulta realiza um SORT CLUSTER BY ROWID para organizar os dados encontrados e um TABLE ACCESS BY INDEX ROWID para acessar diretamente as linhas da tabela countries associadas ao índice, com um custo de 2. O uso do RESULT CACHE representa 74% do custo total.

Questão 2 - c)

```
select  cs.cust_id,    cs.cust_first_name,    cs.cust_last_name,    cs.cust_city    ,
ct.country_name
from COUNTRIES ct join sh.customers cs on ct.country_id=cs.country_id
where ct.country_id=52771;
```

Resultado da consulta:

	CUST_ID	CUST_FIRST_NAME	CUST_LAST_NAME	CUST_CITY	COUNTRY_NAME
1	39670	Abel	Aaron	Pune	China
2	7671	Abel	Aaron	Hyderabad	China
3	11226	Abel	Aaron	Delhi	China
4	1449	Abel	Embrey	Hyderabad	China
5	33448	Abel	Embrey	Pune	China
6	1450	Abigail	Husky	Hyderabad	China
7	33449	Abigail	Husky	Pune	China
8	39671	Abigail	Kessel	Pune	China
9	25459	Abigail	Kessel	Secunderabad	China
10	7672	Abigail	Kessel	Hyderabad	China
11	11227	Abigail	Kessel	Delhi	China
12	13899	Adel	Peebles	Hyderabad	China
13	39676	Adel	Rhodes	Pune	China
14	7677	Adel	Rhodes	Hyderabad	China
15	13906	Antony	Chinn	Hyderabad	China
16	17461	Antony	Chinn	Alleppey	China
17	20128	Antony	Wolsey	Hyderabad	China

OBS: o resultado da consulta teve muito mais linha, o print acima mostra apenas as 17 primeiras linhas do resultado.

Plano de explicação



A consulta utiliza um plano de execução que inicia com um INDEX RANGE SCAN para encontrar as entradas na tabela countries com country_id=52771, com um custo baixo de 1 e sem leitura de dados (In-Out = 0B). Em seguida, a consulta

realiza um SORT CLUSTER BY ROWID para organizar os dados encontrados e um TABLE ACCESS BY INDEX ROWID para acessar diretamente as linhas da tabela customers associadas ao índice, com um custo de 2. O acesso completo à tabela customers ocorre em 24% do custo, para encontrar todas as linhas correspondentes. A etapa de HASH JOIN combina os resultados das duas tabelas, com um custo de 7 e leitura de 44KB. O uso do RESULT CACHE representa 74% do custo total, ajudando a armazenar e reutilizar os dados resultantes da junção.
