**Questionário 3 - Tipos de índices para situações distintas**

Nome: Gabriel Stankevix Soares

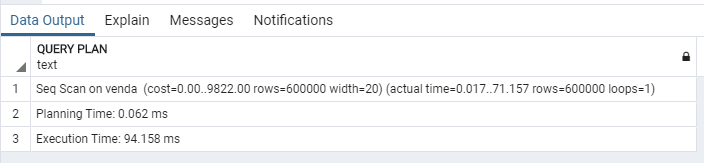
1. Criar duas tabelas, sendo que em qualquer uma delas possua: um campo texto, um campo discreto (categórico) e uma chave estrangeira em relação à outra tabela.

|  |
| --- |
| CREATE TABLE produto (  cod\_produto serial primary key,  produto varchar,  origem varchar(1) check (origem in ('I','N'))  );  CREATE TABLE venda (  cod\_venda serial primary key,  quantidade integer,  preço\_unitario float,  cod\_produto serial references produto(cod\_produto)  ); |

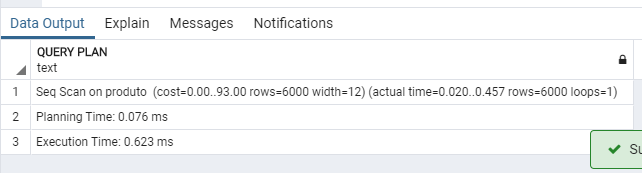
2. Inserir dados nas duas tabelas de maneira que possamos testar os índices. Faça um analyze em cada tabela.

|  |
| --- |
| CREATE OR REPLACE FUNCTION get\_random\_number(INTEGER, INTEGER) RETURNS  INTEGER AS $$  DECLARE  start\_int ALIAS FOR $1;  end\_int ALIAS FOR $2;  BEGIN  RETURN trunc(random() \* (end\_int-start\_int) + start\_int);  END;  $$ LANGUAGE 'plpgsql' STRICT;  -- IMPORTADO  do $$  begin  for i IN 1..3000 LOOP  insert into produto values (i,  (CASE get\_random\_number(1,5)  WHEN 1 THEN 'Cerveja'  WHEN 2 THEN 'Vinho'  WHEN 3 THEN 'Agua'  WHEN 4 THEN 'Pinga'  WHEN 5 THEN 'Vodka' END),  'I');  end loop;  end $$;  -- NACIONAL  do $$  begin  for i IN 3001..6000 LOOP  insert into produto values (i,  (CASE get\_random\_number(1,5)  WHEN 1 THEN 'Cerveja'  WHEN 2 THEN 'Vinho'  WHEN 3 THEN 'Agua'  WHEN 4 THEN 'Pinga'  WHEN 5 THEN 'Vodka' END),  'N');  end loop;  end $$;  do $$  begin  for i IN 1..600000 LOOP  insert into venda values (i,  cast(random()\*10 + 1 as int),  random() \* 10 + 1,  get\_random\_number(1,6000));  end loop;  end $$;  analyze public.venda; -- Query returned successfully in 238 msec.  analyze public.produto; -- Query returned successfully in 186 msec. |

**explain analyze select \* from public.venda;**



**explain analyze select \* from public.produto;**

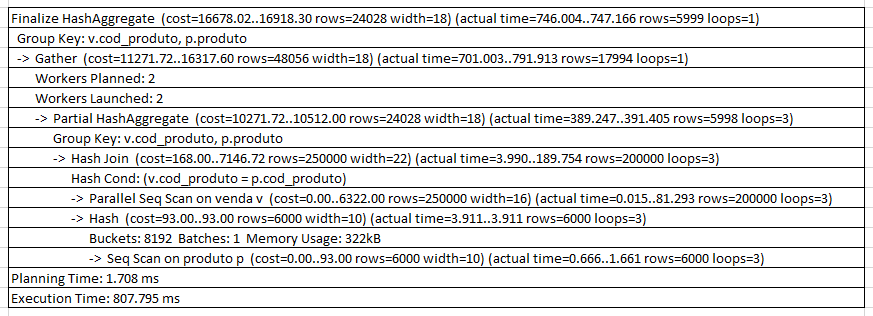


3. Criar três índices, um para o campo textual, um para o campo discreto (bitmap), e para a chave estrangeira.

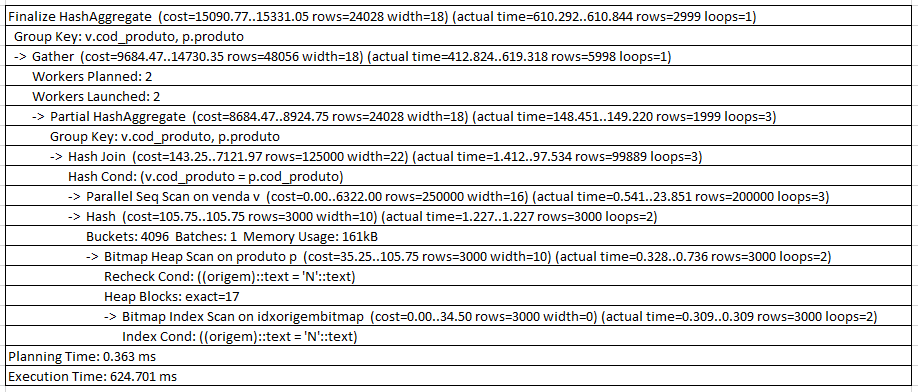
|  |
| --- |
| create index idxproduto on produto(produto);  --create extension btree\_gin;  create index idxOrigemBitmap on produto using gin (origem);  create index idxcod\_produto on venda(cod\_produto); |

4. Formule e mostre o plano das consultas SQL que utilizem cada os índices criados no item anterior.

|  |
| --- |
| explain analyze  select v.cod\_produto,p.produto,sum((v.quantidade \* v.preço\_unitario)) as total  from venda v  inner join produto p  on v.cod\_produto = p.cod\_produto  group by v.cod\_produto,p.produto; |



|  |
| --- |
| explain analyze  select v.cod\_produto,p.produto,sum((v.quantidade \* v.preço\_unitario)) as total  from venda v  inner join produto p  on v.cod\_produto = p.cod\_produto  where p.origem = 'N'  group by v.cod\_produto,p.produto; |



|  |
| --- |
| explain analyze  select v.cod\_produto,p.produto,sum((v.quantidade \* v.preço\_unitario)) as total  from venda v  inner join produto p  on v.cod\_produto = p.cod\_produto  where p.origem = 'N' and p.produto LIKE '%n%'  group by v.cod\_produto,p.produto; |

