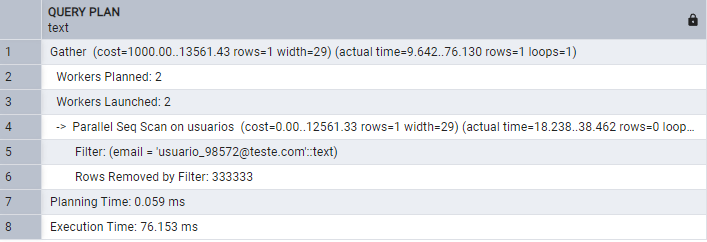
1 - Realize uma consulta sem índice

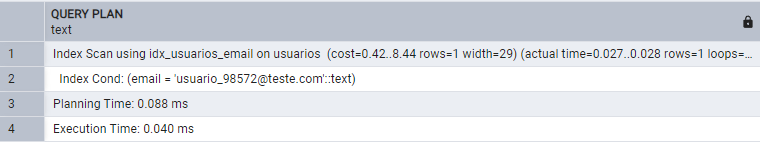
EXPLAIN ANALYZE

select\* from usuarios where email ='usuario\_98572@teste.com';



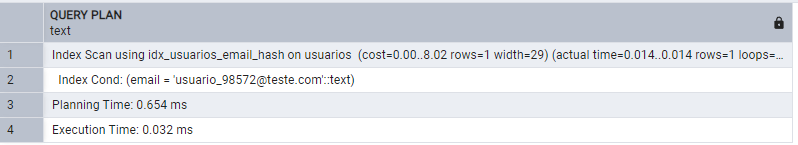
2 - Crie índice B-Tree

create index idx\_usuarios\_email on usuarios(email);



3 - Crie índice HASH

CREATE INDEX idx\_usuarios\_email\_hash ON usuarios USING hash (email);

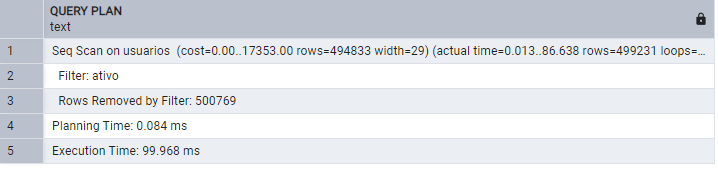


4 -Execute as três versões (sem índice, com B-Tree, com Hash) e compare: ● Planning Time ● Execution Time ● Se o plano usou Seq Scan ou Index Scan ● Se o Hash realmente foi mais rápido que o B-Tree.

|  |  |  |  |
| --- | --- | --- | --- |
|  | sem índice | com B-Tree | com Hash |
| Execution Time | 76,153 | 0,040 | 0,032 |
| Planning Time | 0,059 | 0,088 | 0,654 |

5 - Para a coluna booleana EXPLAIN ANALYZE SELECT \* FROM usuarios WHERE ativo = true; a) Será que o índice HASH (ou até B-Tree) ajuda? Ou o planner decide ignorar o índice e fazer Seq Scan?

Ele ignora e faz o seg scan



6 - Valores repetidos

1. O índice hash ajuda mesmo se a seletividade é baixa (muitos registros com o mesmo valor)?

**Não necessariamente pois** quando tem poucas opções de registros , o **índice hash pode não ser usado** ou **não trazer um ganho** significativo e em muitos casos, o **postgreSQL o ignora.**

**Porem esse não foi o casso ele utilizou o hash**

