

1 Exercícios sobre índices

Para os exercícios seguintes, assuma que estão definidas as seguintes relações.

```
CREATE TABLE Aluno(  
    Nome VARCHAR(50) NOT NULL,  
    RA DECIMAL(8) NOT NULL,  
    DataNasc DATE NOT NULL,  
    Idade DECIMAL(3),  
    NomeMae VARCHAR(50) NOT NULL,  
    Cidade VARCHAR(30),  
    Estado CHAR(2),  
    Curso VARCHAR(50),  
    periodo integer  
);  
  
CREATE TABLE Discip(  
    Sigla CHAR(7) NOT NULL,  
    Nome VARCHAR(25) NOT NULL,  
    SiglaPreReq CHAR(7),  
    NNCred DECIMAL(2) NOT NULL,  
    Monitor DECIMAL(8),  
    Depto CHAR(8)  
);  
  
CREATE TABLE Matricula(  
    RA DECIMAL(8) NOT NULL,  
    Sigla CHAR(7) NOT NULL,  
    Ano CHAR(4) NOT NULL,  
    Semestre CHAR(1) NOT NULL,  
    CodTurma DECIMAL(4) NOT NULL,  
    NotaP1 NUMERIC(3,1),  
    NotaP2 NUMERIC(3,1),  
    NotaTrab NUMERIC(3,1),  
    NotaFIM NUMERIC(3,1),  
    Frequencia DECIMAL(3)  
);
```

Exercício 1) Crie as constraints que julgar necessário após as tabelas forem criadas. Popule as tabelas com várias tuplas, de preferência acima de mil tuplas, até que os índices sejam usados em suas consultas. Não se esqueça de atualizar as métricas das tabelas após a inserção (analyze). Pode-se utilizar algum gerador de dados (<https://www.mockaroo.com/> , www.generatedata.com), ou use as funções direto em SQL (final deste documento).

Exercício 2) Suponha que o seguinte índice foi criado para a relação Aluno.

- CREATE UNIQUE INDEX IdxAlunoNNI ON Aluno (Nome, NomeMae, Idade);
 - Escreva uma consulta que utilize esse índice.
 - Mostre um exemplo onde o índice não é usado mesmo utilizando algum campo indexado na cláusula where, e explique por quê.

Exercício 3) Crie Índices e mostre exemplos de consultas (resultados e explain) que usam os seguintes tipos de acessos:

- a) Sequential Scan
- b) Bitmap Index Scan
- c) Index Scan
- d) Index-Only Scan
- e) Multi-Index Scan

Exercício 4) Faça consultas com junções entre as tabelas e mostre o desempenho criando-se índices para cada chave estrangeira.

Exercício 5) Utilize um índice bitmap para período e mostre-o em uso nas consultas.

Exercício 6) Compare na prática o custo de executar uma consulta com e sem um índice clusterizado na tabela aluno. Ou seja, faça uma consulta sobre algum dado indexado, clusterize a tabela naquele índice e refaça a consulta. Mostre os comandos e os resultados do explain analyze.

Exercício 7) Acrescente um campo adicional na tabela de Aluno, chamado de informacoesExtras, do tipo JSON. Insira dados diferentes telefônicos e de times de futebol que o aluno torce para cada aluno neste JSON. Crie índices para o JSON e mostre consultas que o utiliza (explain analyze). Exemplo: retorne os alunos que torcem para o Internacional.

Funções de dados aleatórios em SQL

```
--Numero aleatorio (Quantidade max de digitos por parametro)
create or replace function numero(digitos integer) returns integer as
$$
begin
    return trunc(random()*power(10,digitos));
end;
$$language plpgsql;
--select numero(3);
```

```
--Data aleatoria (indicar periodo na funcao se quiser alterar)
create or replace function data() returns date as
$$
begin
    return date(timestamp '1980-01-01 00:00:00' +
        random() * (timestamp '2017-01-30 00:00:00' -
            timestamp '1990-01-01 00:00:00'));
end;
$$language plpgsql;
--select data()
```

```
--Texto aleatorio
Create or replace function texto(tamanho integer) returns text as
$$
declare
    chars text[] := '{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z,
```

```

a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z}';
result text := '';
i integer := 0;
begin
  if tamanho < 0 then
    raise exception 'Tamanho dado nao pode ser menor que zero';
  end if;
  for i in 1..tamanho loop
    result := result || chars[1+random()*(array_length(chars, 1)-1)];
  end loop;
  return result;
end;
$$ language plpgsql;

--select texto(5)

```