



Índices no PostgreSQL

Alunos:

Valdir Rugiski Jr.

Wesley Ricardo Lamb

Tipos de índices

O PostgreSQL fornece vários tipos de índice: B-tree, Hash, GiST, SP-GiST, GIN e BRIN. Cada tipo de índice usa um algoritmo diferente que é mais adequado para diferentes tipos de cláusulas indexáveis. Por padrão, o comando CREATE INDEX cria índices B-tree, que se encaixam nas situações mais comuns. Os outros tipos de índice são selecionados escrevendo a palavra-chave USING seguida pelo nome do tipo de índice. Por exemplo, para criar um índice Hash:

```
CREATE INDEX idx_nome_indice ON nome_tabela USING tipo_índice (coluna[s]);
```

B-Tree

As árvores B podem manipular consultas de igualdade e intervalo em dados que podem ser classificados em alguma ordem.

O planejador de consultas do PostgreSQL considerará o uso de um índice B-tree sempre que uma coluna indexada estiver envolvida em uma comparação usando um desses operadores:

```
< <= = >= > BETWEEN IN IS NULL IS NOT NULL
```

```
SELECT * FROM pessoas WHERE nome = 'João';
```

```
SELECT * FROM pessoas WHERE data_nascimento BETWEEN '2001-01-01' AND '2010-01-01';
```

Hash

Índices hash armazenam um código hash de 32 bits derivado do valor da coluna indexada.

Portanto, tais índices podem manipular apenas comparações de igualdade simples. O planejador de consultas considerará usar um índice hash sempre que uma coluna indexada estiver envolvida em uma comparação usando o operador igual ou similares (IN):

```
SELECT * from pessoas where nome = 'João';
```

GiST

Índices GiST não são um único tipo de índice, mas sim uma infraestrutura dentro da qual muitas estratégias de indexação diferentes podem ser implementadas. Consequentemente, os operadores particulares com os quais um índice GiST pode ser usado variam dependendo da estratégia de indexação (a classe do operador). Como exemplo, a distribuição padrão do PostgreSQL inclui classes de operadores GiST para vários tipos de dados geométricos bidimensionais, que suportam consultas indexadas usando estes operadores:

```
<< -- 0 primeiro objeto está estritamente à esquerda do segundo?
&< -- 0 primeiro objeto não se estende à direita do segundo?
&> -- 0 primeiro objeto não se estende à esquerda do segundo?
>> -- 0 primeiro objeto está estritamente à direita do segundo?
<<| -- 0 primeiro objeto está estritamente abaixo do segundo?
&<| -- 0 primeiro objeto não se estende abaixo do segundo?
|&> -- 0 primeiro objeto não se estende acima do segundo?
|>> -- 0 primeiro objeto está estritamente acima do segundo?
@> -- 0 primeiro objeto está contido no segundo?
<@ -- 0 primeiro objeto está contido no segundo ou sobre dele?
~= -- Esses objetos são iguais?
&& -- Esses objetos se sobrepõem?
```

SP-GiST

Índices SP-GiST, como índices GiST, oferecem uma infraestrutura que suporta vários tipos de pesquisas. O SP-GiST permite a implementação de uma ampla gama de diferentes estruturas de dados não balanceadas baseadas em disco, como quadrees, árvores k-d e árvores radix (tries). Como exemplo, a distribuição padrão do PostgreSQL inclui classes de operadores SP-GiST para pontos bidimensionais, que suportam consultas indexadas usando estes operadores:

```
<< >> ~= <@ <<| |>>
```

GIN

Índices GIN são “índices invertidos” que são apropriados para valores de dados que contêm múltiplos valores de componentes, como matrizes. Um índice invertido contém uma entrada separada para cada valor de componente e pode manipular eficientemente consultas que testam a presença de valores de componentes específicos.

Como GiST e SP-GiST, GIN pode suportar muitas estratégias de indexação definidas pelo usuário, e os operadores específicos com os quais um índice GIN pode ser usado variam dependendo da estratégia de indexação. Como exemplo, a distribuição padrão do PostgreSQL inclui uma classe de operadores GIN para matrizes, que suporta consultas indexadas usando estes operadores:

```
<@ -- A primeira matriz contém a segunda, ou seja, cada elemento que  
-- aparece na segunda matriz é igual a algum elemento da primeira matriz?  
@> -- O primeiro array está contido no segundo?  
= -- Os arrays são idênticos?  
&& -- os arrays se sobrepõem, ou seja, têm algum elemento em comum?
```

Índices BRIN (uma abreviação para Block Range INdexes) armazenam resumos sobre os valores armazenados em intervalos de blocos físicos consecutivos de uma tabela. Portanto, eles são mais eficazes para colunas cujos valores são bem correlacionados com a ordem física das linhas da tabela. Como GiST, SP-GiST e GIN, BRIN pode suportar muitas estratégias de indexação diferentes, e os operadores específicos com os quais um índice BRIN pode ser usado variam dependendo da estratégia de indexação. Para tipos de dados que têm uma ordem de classificação linear, os dados indexados correspondem aos valores mínimo e máximo dos valores na coluna para cada intervalo de blocos. Isso suporta consultas indexadas usando estes operadores:

```
< <= = >= >
```