

TIPOS DE INDICES EN POSTGRESQL

PostgreSQL proporciona varios tipos de índices: B-Tree, Hash, GiST, SP-GiST, GIN y BRIN. Cada tipo de índice utiliza un algoritmo diferente que se adapta mejor a diferentes tipos de consultas. Por defecto, el comando `CREATE INDEX` crea índices tipo B-TREE, que se ajustan a las situaciones más comunes.

Los índices tipo B-TREE pueden manejar consultas de igualdad y rango en datos que se pueden clasificar en algún orden. En particular, el planificador de consultas de PostgreSQL considerará el uso de un índice tipo B-TREE siempre que se trate de una columna indexada en una comparación utilizando uno de estos operadores:

`<`

`<=`

`=`

`>=`

`>`

Las construcciones equivalentes a combinaciones de estos operadores, como `BETWEEN` e `IN`, también pueden ser implementado con una búsqueda de índice tipo B-Tree. Además, una condición `IS NULL` o `IS NOT NULL` en una columna índice se puede utilizar con un índice tipo B-Tree.

El optimizador también puede usar un índice tipo B-Tree para consultas que involucren a los operadores de coincidencia de patrones `LIKE` y `~` si el patrón es una constante y está anclado al principio de la cadena, por ejemplo, `col LIKE 'foo%'` o `col ~ '^foo'`, pero

no col LIKE '% bar'. Sin embargo, si su base de datos no utiliza la configuración regional C, necesitará crear el índice con una clase de operador especial para admitir la indexación de consultas de coincidencia de patrones. También es posible utilizar índices tipo B-Tree para ILIKE y ~ *, pero solo si el patrón comienza con caracteres no alfabéticos, es decir, caracteres que no se ven afectados por conversiones mayúsculas/minúsculas.

Los índices tipo B-Tree también se pueden utilizar para recuperar datos en orden ordenado. Esto no siempre es más rápido que un simple escanear y ordenar, pero a menudo es útil.

Los índices tipo hash solo pueden manejar comparaciones de igualdad simples. El planificador de consultas considerará utilizar un índice hash siempre que una columna indexada esté involucrada en una comparación utilizando el operador =.

El siguiente comando se usa para crear un índice hash:

```
CREATE INDEX nombreIndice ON nombreTabla USING HASH (nombreColumna);
```

Los índices GiST no son un solo tipo de índice, sino una infraestructura dentro de la cual se pueden implementar muchas estrategias de indexación. En consecuencia, los operadores particulares con los que un índice GiST se pueden utilizar varían según la estrategia de indexación (la clase de operador). Como ejemplo, en el estándar de distribución de PostgreSQL incluye clases de operador GiST para varios datos de tipo

geométricos bidimensionales, que admiten consultas indexadas mediante estos operadores:

<<

&<

&>

>>

<<|

&<|

|&>

|>>

@>

<@

~=

&&

Los índices GiST también son capaces de optimizar las búsquedas del "vecino más cercano", como:

```
SELECT * FROM places ORDER BY location <-> point '(101,456)' LIMIT 10;
```

que encuentra los 10 lugares más cercanos a un punto objetivo determinado. La capacidad para hacer esto depende nuevamente de la clase de operador particular que se está utilizando.

Los índices SP-GiST, como los índices GiST, ofrecen una infraestructura que admite varios tipos de búsquedas. SPGiST permite la implementación de una amplia gama de diferentes estructuras de datos basadas en disco no balanceadas, como quadrees, k-d trees, y radix trees. Como ejemplo, la distribución estándar de PostgreSQL incluye clases de operador SP-GiST para puntos bidimensionales, que admiten consultas indexadas utilizando estos operadores:

`<<`

`>>`

`~=`

`<@`

`<^`

`>^`

Al igual que GiST, SP-GiST admite búsquedas de "vecino más cercano".

Los índices GIN son "índices invertidos" que son apropiados para valores de datos que contienen varios componentes valores, como matrices. Un índice invertido contiene una entrada separada para cada valor de componente y puede manejar de manera eficiente consultas que prueban la presencia de valores de componentes específicos.

Al igual que GiST y SP-GiST, GIN puede admitir muchas estrategias de indexación diferentes definidas por el usuario, y los operadores particulares con los que se puede utilizar un índice GIN varían según la estrategia de indexación. Como un ejemplo, la distribución estándar de PostgreSQL incluye una clase de operador GIN para matrices, que admite consultas indexadas utilizando estos operadores:

<@

@>

=

&&

Los índices BRIN (una abreviatura de Block Range INdexes) almacenan resúmenes sobre los valores almacenados en rangos de bloques físicos consecutivos de una tabla. Al igual que GiST, SP-GiST y GIN, BRIN puede admitir muchas diferentes estrategias de indexación y los operadores particulares con los que se puede utilizar un índice BRIN varían dependiendo de la estrategia de indexación.

Para los tipos de datos que tienen un orden de clasificación lineal, los datos indexados corresponde a los valores mínimo y máximo de los valores de la columna para cada rango de bloque. Esta admite consultas indexadas utilizando estos operadores:

<

<=

=

>=

>