New in v2.0：倒排索引通过帮助SQL在JSONB列中定位无schema数据来提升数据库性能。

有关使用倒排索引来提高JSONB列的查询性能的演示，请参阅[JSON教程](https://www.cockroachlabs.com/docs/stable/demo-json-support.html)。

倒排索引是如何工作的？

标准索引适用于基于排序数据前缀的搜索，但是，像JSONB这类无schema的数据查找则只能通过全表扫描，因为它不遵循普通值前缀比较运算符。JSONB需要比标准索引更多的细节来建立索引，这正是用到倒排索引的地方。

倒排索引在可标记数据组件上进行过滤，JSONB数据类型构建在两个可标记化的结构上：

* **Objects** - kv对的集合，其中每个kv对是一个标记。
* **Arrays** - 有序的值列表，其中数组中每个值都是一个标记。

例如，在person列中取以下JSONB值：

{

"firstName": "John",

"lastName": "Smith",

"age": 25,

"address": {

"state": "NY",

"postalCode": "10021"

},

"cars": [

"Subaru",

"Honda"

]

}

此对象的倒排索引将为每个组件创建一个条目，并将其映射回原始对象：

"firstName": "John"

"lastName": "Smith"

"age": 25

"address": "state": "NY"

"address": "postalCode": "10021"

"cars" : "Subaru"

"cars" : "Honda"

这使你可以基于子组件进行搜索。

创建

你可以使用倒排索引来提高查询JSONB列的性能。

你可以如此创建它们：

* 对于新表，CREATE TABLE建表时用INVERTED INDEX子句创建
* 对于已存在表，使用CREATE INVERTED INDEX创建
* 使用以下PostgreSQL兼容的语法：

> **CREATE** **INDEX** <optional name> **ON** <**table**> **USING** GIN (<**column**>);

选择

如果查询包含针对使用任何支持的运算符的索引“JSONB”列的过滤器，则将倒排索引添加到索引候选集。

因为每个查询只能使用一个索引，所以CockroachDB选择计算时将扫描最少行的索引（即最快的行）。 有关更多详细信息，请查看我们的博客文章 [Index Selection in CockroachDB](https://www.cockroachlabs.com/blog/index-selection-cockroachdb-2/)。

你还可以强制查询使用特定索引（也称为“index hinting”）来覆盖CockroachDB选择的索引。

存储

CockroachDB将索引直接存储在键值存储中。更多信息查阅：[Mapping Table Data to Key-Value Storage](https://www.cockroachlabs.com/blog/sql-in-cockroachdb-mapping-table-data-to-key-value-storage/).

加锁

CockroachDB创建索引时不会加锁，原因查阅：[schema change procedure](https://www.cockroachlabs.com/blog/how-online-schema-changes-are-possible-in-cockroachdb/).

性能

索引是一种权衡的选择：它们极大提高查询速度，但稍微降低写入速度（因为必须复制和对新值排序）。你创建的第一个索引影响最大，不过其他索引只会产生边际开销。

示例

在这个例子中，让我们创建一个带有JSONB列和倒排索引的表：

> **CREATE** **TABLE** users (

profile\_id **UUID** **PRIMARY** **KEY** **DEFAULT** gen\_random\_uuid(),

last\_updated **TIMESTAMP** **DEFAULT** **now**(),

user\_profile JSONB,

INVERTED **INDEX** user\_details (user\_profile)

);

然后，插入几行数据：

> **INSERT** **INTO** users (user\_profile) **VALUES**

('{"first\_name": "Lola", "last\_name": "Dog", "location": "NYC", "online" : true, "friends" : 547}'),

('{"first\_name": "Ernie", "status": "Looking for treats", "location" : "Brooklyn"}'),

('{"first\_name": "Carl", "last\_name": "Kimball", "location": "NYC", "breed": "Boston Terrier"}'

);

> **SELECT** \*, jsonb\_pretty(user\_profile) **FROM** users;

+--------------------------------------+----------------------------------+--------------------------------------------------------------------------+------------------------------------+

| profile\_id | last\_updated | user\_profile | jsonb\_pretty |

+--------------------------------------+----------------------------------+--------------------------------------------------------------------------+------------------------------------+

| 81330a51-80b2-44aa-b793-1b8d84ba69c9 | 2018-03-13 18:26:24.521541+00:00 | {"breed": "Boston Terrier", "first\_name": "Carl", "last\_name": | { |

| | | "Kimball", "location": "NYC"} | |

| | | | "breed": "Boston Terrier", |

| | | | "first\_name": "Carl", |

| | | | "last\_name": "Kimball", |

| | | | "location": "NYC" |

| | | | } |

| 81c87adc-a49c-4bed-a59c-3ac417756d09 | 2018-03-13 18:26:24.521541+00:00 | {"first\_name": "Ernie", "location": "Brooklyn", "status": "Looking for | { |

| | | treats"} | |

| | | | "first\_name": "Ernie", |

| | | | "location": "Brooklyn", |

| | | | "status": "Looking for treats" |

| | | | } |

| ec0a4942-b0aa-4a04-80ae-591b3f57721e | 2018-03-13 18:26:24.521541+00:00 | {"first\_name": "Lola", "friends": 547, "last\_name": "Dog", "location": | { |

| | | "NYC", "online": true} | |

| | | | "first\_name": "Lola", |

| | | | "friends": 547, |

| | | | "last\_name": "Dog", |

| | | | "location": "NYC", |

| | | | "online": true |

| | | | } |

+--------------------------------------+----------------------------------+--------------------------------------------------------------------------+------------------------------------+

现在，在 JSONB 列上进行过滤查询：

> **SELECT** \* **FROM** users **where** user\_profile @> '{"location":"NYC"}';

+--------------------------------------+----------------------------------+--------------------------------------------------------------------------+

| profile\_id | last\_updated | user\_profile |

+--------------------------------------+----------------------------------+--------------------------------------------------------------------------+

| 81330a51-80b2-44aa-b793-1b8d84ba69c9 | 2018-03-13 18:26:24.521541+00:00 | {*"breed"*: *"Boston Terrier"*, *"first\_name"*: *"Carl"*, *"last\_name"*: |

| | | *"Kimball"*, *"location"*: *"NYC"*} |

| ec0a4942-b0aa-4a04-80ae-591b3f57721e | 2018-03-13 18:26:24.521541+00:00 | {*"first\_name"*: *"Lola"*, *"friends"*: 547, *"last\_name"*: *"Dog"*, *"location"*: |

| | | *"NYC"*, *"online"*: **true**} |

+--------------------------------------+----------------------------------+--------------------------------------------------------------------------+

(2 rows)

See Also

* [JSONB](http://doc.cockroachchina.baidu.com/#develop/data-types/JSONB/)
* [JSON tutorial](http://doc.cockroachchina.baidu.com/#quick-start/explore-benefits/json-support/)
* [Computed Columns](http://doc.cockroachchina.baidu.com/#develop/sql-statements/CONPUTED-COLUMNS/)
* [CREATE INDEX](http://doc.cockroachchina.baidu.com/#develop/sql-statements/CREATE-INDEX/)
* [DROP INDEX](http://doc.cockroachchina.baidu.com/#develop/sql-statements/DROP-INDEX/)
* [RENAME INDEX](http://doc.cockroachchina.baidu.com/#develop/sql-statements/RENAME-INDEX/)
* [SHOW INDEX](http://doc.cockroachchina.baidu.com/#develop/sql-statements/SHOW-INDEX/)
* [Indexes](http://doc.cockroachchina.baidu.com/#develop/performance-optimization/indexes/)
* [SQL Statements](http://doc.cockroachchina.baidu.com/#develop/sql-statements/overview/)