CREATE INDEX语句为表创建索引。 索引通过帮助SQL无需遍历表快速定位数据，从而提高数据库的性能。

New in v2.0: 要在JSONB列中的无schema数据上创建索引，请使用倒排索引。

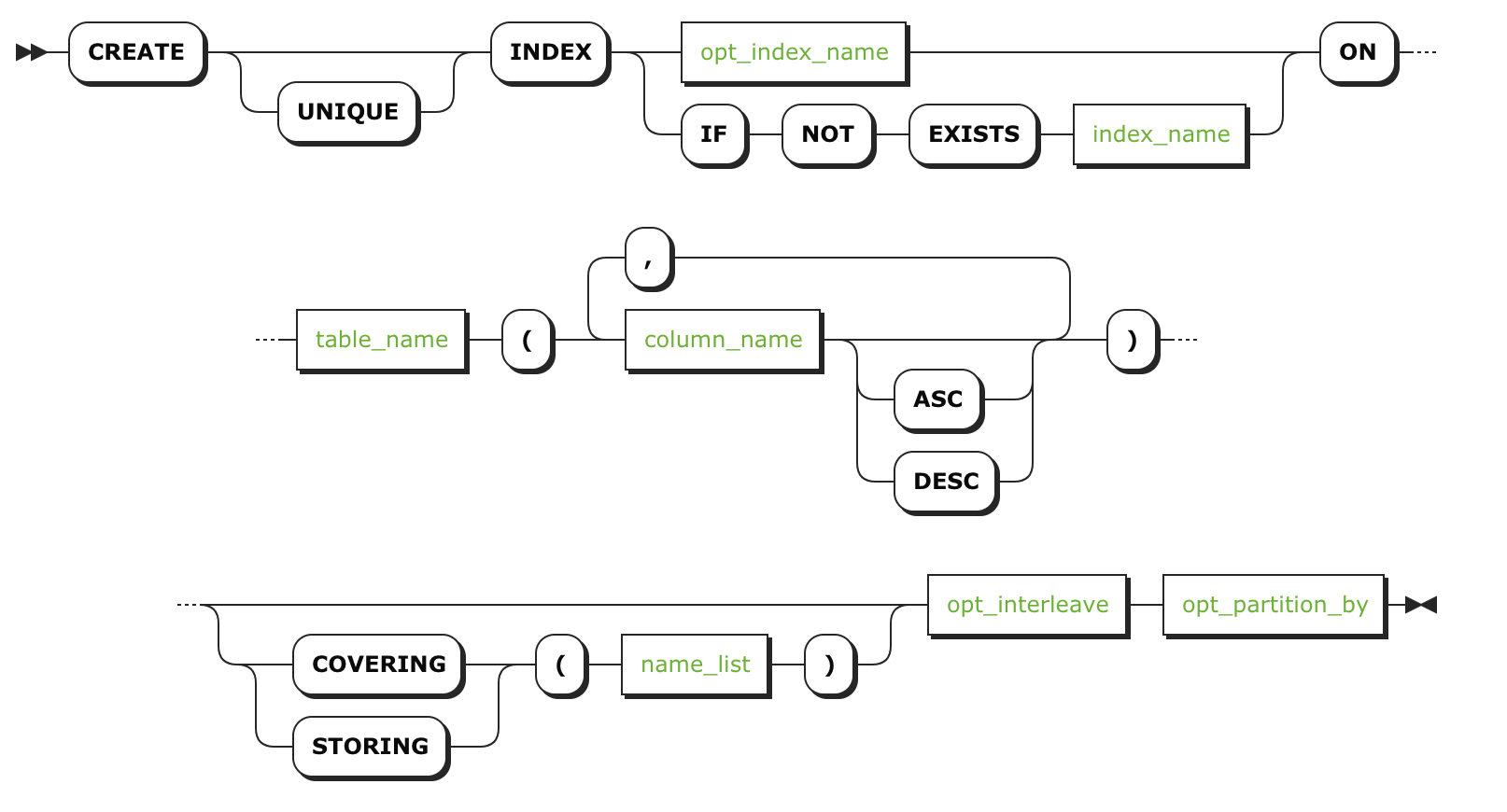
表的PRIMARY KEY和UNIQUE列自动创建索引。  
查询表时，CockroachDB使用最快的索引。 有关该过程的更多信息，请参阅CockroachDB中的索引选择。

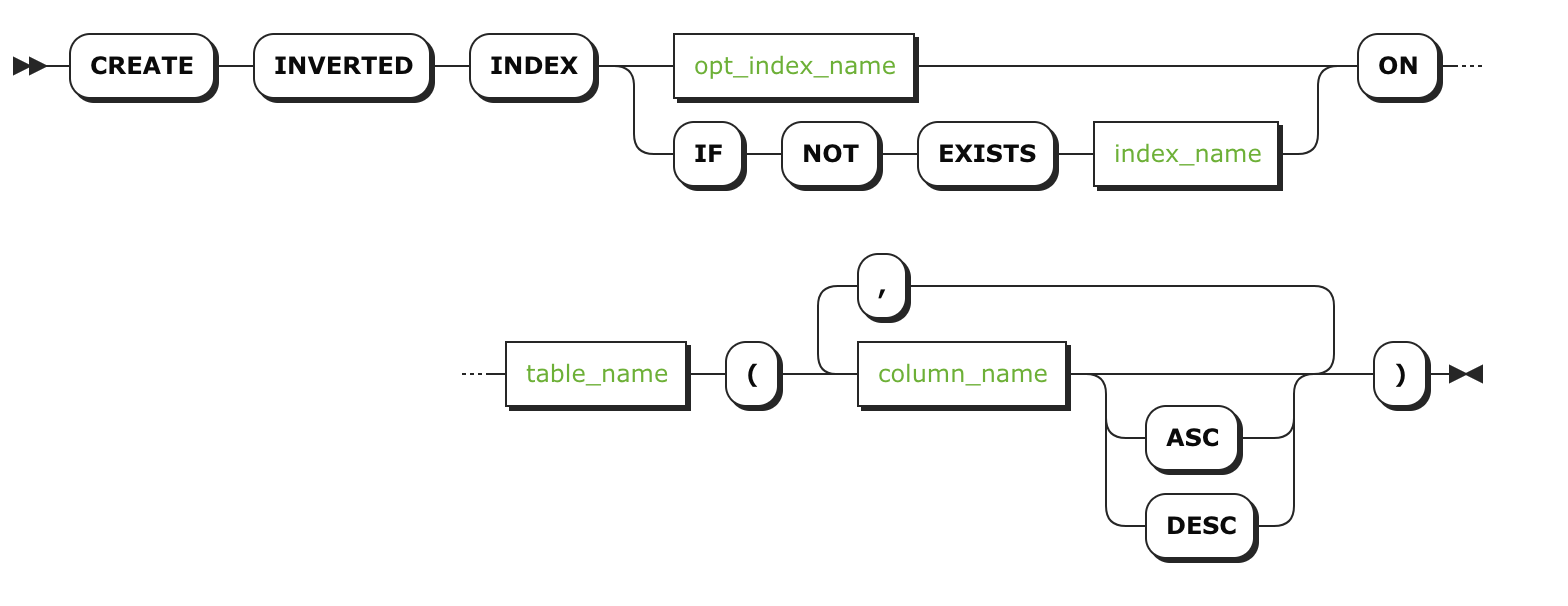
所需权限

用户必须具有表的CREATE权限。

概要

**标准索引:**



**倒排索引:** 

参数

| **Parameter** | **Description** |
| --- | --- |
| UNIQUE | 将Unique约束应用于索引列。  系统检查索引创建时的现有重复值。 它还在表级应用Unique约束，因此系统在插入或更新数据时会检查重复值。 |
| INVERTED | New in v2.0:在指定的JSONB列中的无schema数据上创建倒排索引。  你还可以使用PostgreSQL兼容的语法USING GIN。 有关更多详细信息，请参阅倒排索引文档。 |
| IF NOT EXISTS | 仅当不存在同名索引时才创建新索引; 如果确实存在，不返回错误。 |
| opt\_index\_name index\_name | 要创建的索引的名称，该索引的名称必须是唯一的，并遵循[identifier rules](http://doc.cockroachchina.baidu.com/" \l "develop/sql-syntax/keywords-and-identifiers/#identifiers).  如果你没有指定名称，CockroachDB则使用<table> \_ <columns> \_key/idx。 key表示索引应用Unique约束，idx则表示没有。 示例：accounts\_balance\_idx |
| table\_name | 要创建索引的表的名称。 |
| column\_name | 要索引的列的名称。 |
| ASC or DESC | 在索引中按升序（ASC）或降序（DESC）顺序对列进行排序。 列的排序方式会影响查询结果，尤其是在使用LIMIT时.  **默认:** ASC |
| STORING ... | 存储（但不要排序）包含其名称的每个列。  更多信息，查看 [Store Columns](http://doc.cockroachchina.baidu.com/#develop/sql-statements/CREATE-INDEX/#store-columns).  COVERING别名STORING，工作方式相同。 |
| opt\_interleave | 您可以通过交错索引( interleaving indexes)来优化查询性能，这会改变CockroachDB存储数据的方式。 |
| opt\_partition\_by | 文档即将推出。 |

示例

创建标准索引

要创建最有效的索引，我们建议你查看：

* [索引：最佳时间](http://doc.cockroachchina.baidu.com/#develop/performance-optimization/indexes/#best-practices)
* [CockroachDB中的索引选择](https://www.cockroachlabs.com/blog/index-selection-cockroachdb-2)

单列索引

单列索引对单个列的值进行排序。

> **CREATE** **INDEX** **ON** products (price);

由于每个查询只能使用一个索引，因此单列索引通常不如多列索引有用。

多列索引

多列索引按列出的顺序对列进行排序。

> **CREATE** **INDEX** **ON** products (price, stock);

要创建更好用的多列索引，我们建议你查看[best practices](http://doc.cockroachchina.baidu.com/" \l "develop/performance-optimization/indexes/#indexing-columns).

唯一索引

唯一索引不允许其列中包含重复值。

> **CREATE** **UNIQUE** **INDEX** **ON** products (name, manufacturer\_id);

这也适用于表级别的唯一约束，类似于ALTER TABLE。 上面的例子相当于：

> **ALTER** **TABLE** products **ADD** **CONSTRAINT** products\_name\_manufacturer\_id\_key **UNIQUE** (name, manufacturer\_id);

创建倒排索引（New in v2.0）

可以在JSONB列中的无模式数据上创建反向索引。

> **CREATE** INVERTED **INDEX** **ON** users (**profile**);

上面的示例等效于以下与PostgreSQL兼容的语法：

> **CREATE** **INDEX** **ON** users **USING** GIN (**profile**);

存储列

存储列可以提高检索（但不过滤）其值的查询的性能。

> **CREATE** **INDEX** **ON** products (price) STORING (name);

但是，要使用存储列，查询必须过滤同一索引中的另一列。 例如，只有当查询的WHERE子句过滤price时，SQL才能从上面的索引中检索name值。

更改列排序顺序

要按降序对列进行排序，必须在创建索引时显式设置该选项。 （默认值是升序）

> **CREATE** **INDEX** **ON** products (price **DESC**, stock);

列的排序方式会影响使用索引查询返回的行的顺序，这尤其会影响使用LIMIT的查询。

查询特定索引

通常，CockroachDB选择它计算将扫描最少的行的索引。 但是，你可以覆盖该选择并指定要使用的索引的名称。 要查看索引名称，请使用SHOW INDEX

> **SHOW** **INDEX** **FROM** products;

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

| Table | Name | Unique | Seq | Column | Direction | Storing | Implicit |

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

| products | primary | true | 1 | id | ASC | false | false |

| products | products\_price\_idx | false | 1 | price | ASC | false | false |

| products | products\_price\_idx | false | 2 | id | ASC | false | true |

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

(3 rows)

> **SELECT** name **FROM** products@products\_price\_idx **WHERE** price > 10;

See Also

* [Indexes](http://doc.cockroachchina.baidu.com/#develop/performance-optimization/indexes/)
* [SHOW INDEX](http://doc.cockroachchina.baidu.com/#develop/sql-statements/SHOW-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/)
* [Other SQL Statements](http://doc.cockroachchina.baidu.com/#develop/sql-statements/overview/)