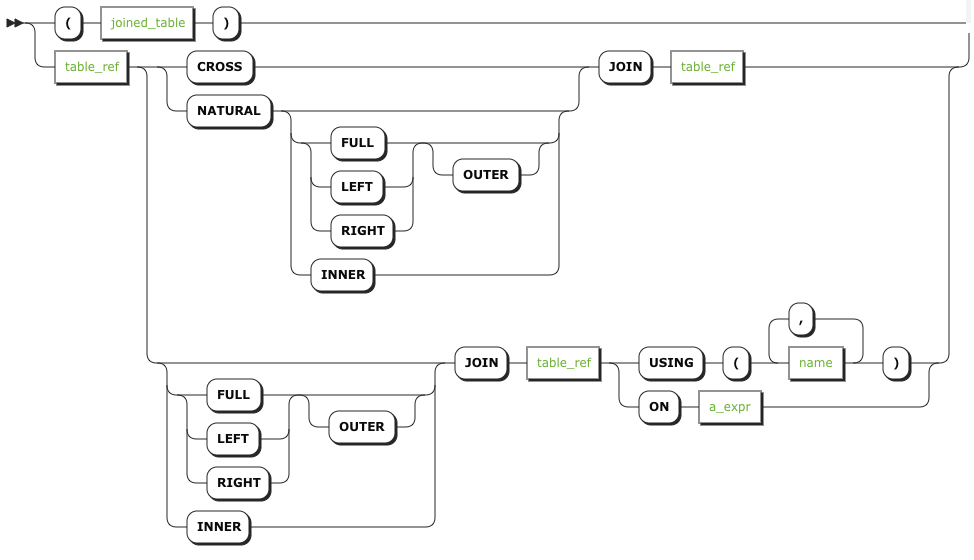
join表达式（也称为“join”）根据特定字段的条件组合两个或多个表表达式的结果。

join表达式在简单SELECT子句的FROM子句中定义数据源，或者作为TABLE的参数定义数据源。 join是一种特殊的表表达式。

概要



参数

| **Parameter** | **Description** |
| --- | --- |
| joined\_table | 另一个联接表达。 |
| table\_ref | 一个表表达式 |
| a\_expr | 用作ON连接条件的scalar表达式 |
| name | 用作USING连接条件的列名 |

支持的连接类

CockroachDB支持JOIN的以下用法。

Inner Joins

仅返回与该条件匹配的左右操作数中的行。

<table expr> [ INNER ] JOIN <table expr> ON <val expr>

<table expr> [ INNER ] JOIN <table expr> **USING**(<colname>, <colname>, ...)

<table expr> NATURAL [ INNER ] JOIN <table expr>

<table expr> CROSS JOIN <table expr>

Left Outer Joins

对于右侧没有匹配的左侧行，右侧的列返回NULL值。

<table expr> LEFT [ OUTER ] JOIN <table expr> ON <val expr>

<table expr> LEFT [ OUTER ] JOIN <table expr> **USING**(<colname>, <colname>, ...)

<table expr> NATURAL LEFT [ OUTER ] JOIN <table expr>

Right Outer Joins

对于左侧没有匹配的每一行，对于左侧的列，返回NULL值。

<table expr> RIGHT [ OUTER ] JOIN <table expr> ON <val expr>

<table expr> RIGHT [ OUTER ] JOIN <table expr> **USING**(<colname>, <colname>, ...)

<table expr> NATURAL RIGHT [ OUTER ] JOIN <table expr>

Full Outer Joins

对于在另一侧没有匹配的连接一侧的每一行，为非匹配的列返回NULL值。

<table expr> FULL [ OUTER ] JOIN <table expr> ON <val expr>

<table expr> FULL [ OUTER ] JOIN <table expr> **USING**(<colname>, <colname>, ...)

<table expr> NATURAL FULL [ OUTER ] JOIN <table expr>

Supported Join Conditions

CockroachDB支持以下条件来匹配连接中的行：

* 没有CROSS JOIN条件：左边的每一行都被认为与右边的每一行相匹配。
* ON谓词：计算布尔标量表达式以确定操作数行是否匹配。
* USING：从左右两行成对比较命名列; 如果列成对相等，则认为左右行匹配。
* NATURAL：使用左表和右表表达式中存在的所有列名生成隐式的USING条件。

注意：为了与PostgreSQL的兼容性，支持NATURAL; 不过不鼓励在新应用程序中使用它，因为当将新列添加到其中一个连接操作数时，其结果可以以不可预测的方式静默地更改。

性能最佳实践

CockroachDB目前正在进行重大更改，以便使用join提高查询性能。 随着时间的推移，本节中列出的限制和变通方法将被解决或变得不必要。

* 与非交叉表相比，交叉表上的连接通常（但不总是）处理得更有效。
* 当没有索引可用于满足连接时，CockroachDB可以在开始返回结果行之前加载内存中满足条件之一的连接操作数的所有行。 如果连接条件或其他WHERE子句选择性不足，这可能会导致连接失败。
* 外连接通常比内连接效率低。 尽可能使用内连接。 全外连接是优化最差的。
* 对包含联接的查询使用EXPLAIN来验证是否使用了索引。
* 参考[Index Best Practices](http://doc.cockroachchina.baidu.com/#develop/performance-optimization/sql-best-practices/#indexes-best-practices)

See Also

* [Scalar Expressions](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/scalar-expressions/)
* [Table Expressions](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/table-expressions/)
* [Simple SELECT Clause](http://doc.cockroachchina.baidu.com/#develop/sql-statements/SELECT/)
* [Selection Queries](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/selection-queries/)
* [EXPLAIN](http://doc.cockroachchina.baidu.com/#develop/sql-statements/EXPLAIN/)
* [Performance Best Practices - Overview](http://doc.cockroachchina.baidu.com/#develop/performance-optimization/sql-best-practices/)
* [SQL join operation (Wikipedia)](https://en.wikipedia.org/wiki/Join_(SQL))
* [CockroachDB's first implementation of SQL joins (CockroachDB Blog)](https://www.cockroachlabs.com/blog/cockroachdbs-first-join/)
* [On the Way to Better SQL Joins in CockrochDB (CockroachDB Blog)](https://www.cockroachlabs.com/blog/better-sql-joins-in-cockroachdb/)