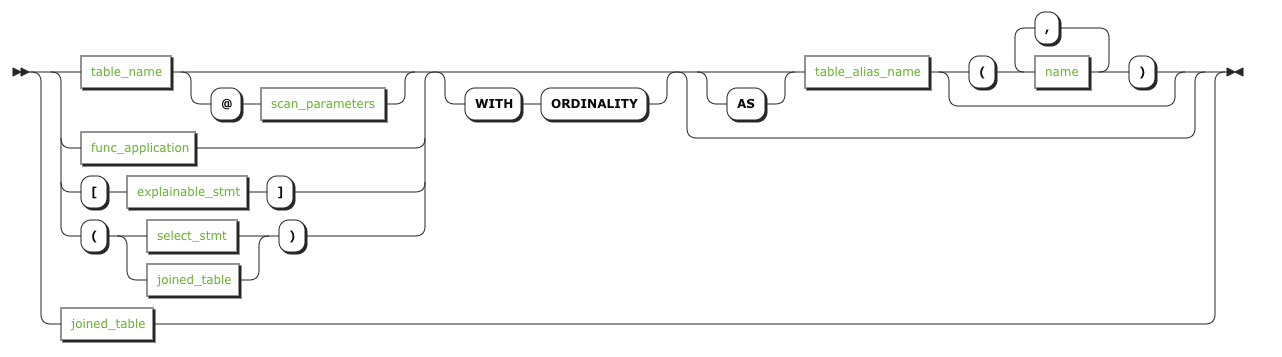
table表达式在简单SELECT子句的FROM子句中定义数据源，或者作为TABLE的参数定义数据源。

[SQL Joins](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/join-expressions/)是一种特殊的table表达式。

概要



参数

| **Parameter** | **Description** |
| --- | --- |
| table\_name | 表或视图名 |
| table\_alias\_name | 要在别名table表达式中使用的名称。 |
| name | 列名的一个或多个别名，用于别名table表达式。 |
| scan\_parameters | 强制索引选择的可选语法。 |
| func\_application | 一个函数的结果。 |
| explainable\_stmt | 使用可解释语句的结果行。 |
| select\_stmt | 用作子查询的选择查询。 |
| joined\_table | 一个join表达式 |

Table表达式语言

上面的概要确实定义了一种迷你语言，用于从更简单的部分构造复杂的table表达式。

| **Construct** | **Description** | **Examples** |
| --- | --- | --- |
| table\_name [@ scan\_parameters] | 访问一个表或视图 | accounts, accounts@name\_idx |
| function\_name ( exprs ... ) | 使用scalar函数或表生成器函数生成表数据。 | sin(1.2), generate\_series(1,10) |
| <table expr> [AS] name [( name [, ...] )] | 重命名表和可选的列。 | accounts a, accounts AS a, accounts AS a(id, b) |
| <table expr> WITH ORDINALITY | 枚举结果行。 | accounts WITH ORDINALITY |
| <table expr> JOIN <table expr> ON ... | Join表达式 | orders o JOIN customers c ON o.customer\_id = c.id |
| (... subquery ...) | 用作子查询的选择查询。 | (SELECT \* FROM customers c) |
| [... statement ...] | 使用可解释语句的结果行。  这是一个CockroachDB扩展。 | [SHOW COLUMNS FROM accounts] |

以下部分提供了有关每个选项的详细信息。

生成数据的table表达式

以下部分描述了生成数据的主table表达式。

访问表或视图

表或视图名

语法:

identifier

identifier.identifier

identifier.identifier.identifier

table表达式上下文中的单个SQL标识符指定当前数据库中具有该名称的表，视图或序列的内容，由SET DATABASE配置

Changed in v2.0:如果名称由两个或多个标识符组成，则应用[名称解析](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/name-resolution/)规则。

举个例子：

> **SELECT** \* **FROM** users; *-- uses table `users` in the current database*

> **SELECT** \* **FROM** mydb.users; *-- uses table `users` in database `mydb`*

强制索引选择

通过使用显式索引注释，您可以覆盖CockroachDB的索引选择，并在从命名表中读取时使用特定索引。

索引选择可能会影响性能，但不会更改查询的结果。

> **SHOW** INDEXES **FROM** accounts;

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

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

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

| accounts | primary | true | 1 | id | ASC | false | false |

| accounts | accounts\_name\_idx | false | 1 | name | ASC | false | false |

| accounts | accounts\_name\_idx | false | 2 | id | ASC | false | true |

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

(3 rows)

> **SELECT** name, balance

**FROM** accounts@accounts\_name\_idx

**WHERE** name = 'Edna Barath';

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

| name | balance |

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

| Edna Barath | 750 |

| Edna Barath | 2200 |

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

访问通用table表达式

table表达式上下文中的单个标识符可以引用先前定义的通用table表达式。

例如：

> WITH a AS (**SELECT** \* **FROM** users)

**SELECT** \* **FROM** a; *-- "a" refers to "WITH a AS .."*

函数的结果

table表达式可以使用函数应用程序的结果作为数据源。

语法:

name ( arguments... )

函数的名称，后跟一个左括号，后跟零个或多个标量表达式，后跟一个右括号。

函数名称的解析遵循与表名解析相同的规则。 有关详细信息，请参阅[名称解析](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/name-resolution/)

标量函数作为数据源

New in v2.0:

当返回单个值的函数用作table表达式时，它将被解释为具有单个列和包含函数结果的单行的表数据。

举个例子:

> **SELECT** \* **FROM** **sin**(3.2)

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

| sin |

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

| -0.058374143427580086 |

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

CockroachDB仅支持此语法以与PostgreSQL兼容。 用于计算标量函数的规范语法是SELECT的直接目标，例如SELECT sin(3.2)。

表生成器函数

某些函数直接从单个函数应用程序生成具有多行的表数据。 这也称为“set-returning function”。

举个例子:

> **SELECT** \* **FROM** generate\_series(1, 3)

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

| generate\_series |

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

| 1 |

| 2 |

| 3 |

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

目前，CockroachDB仅支持与PostgreSQL同名生成函数兼容的一小组生成器函数。

扩展table表达式的运算符

以下部分描述了table表达式，这些table表达式可更改表数据的元数据，或添加更多数据，而无需修改基础表的数据。

别名table表达式

别名table表达式在当前查询的上下文中临时重命名表和列。

语法:

<table expr> AS <name>

<table expr> AS <name>(<colname>, <colname>, ...)

在第一种形式中，table表达式等效于其左操作数，其中包含整个表的新名称，以及列保留其原始名称的位置。

在第二种形式中，列也被重命名了。

举个例子:

> **SELECT** c.x **FROM** (**SELECT** **COUNT**(\*) **AS** x **FROM** users) **AS** c;

> **SELECT** c.x **FROM** (**SELECT** **COUNT**(\*) **FROM** users) **AS** c(x);

普通注释

语法:

<table expr> WITH ORDINALITY

指定一个等于table表达式操作数的数据源，并带有一个额外的“Ordinality”列，该列枚举数据源中的每一行。

举个例子:

> **SELECT** \* **FROM** (**VALUES**('a'),('b'),('c'));

+---------+

| column1 |

+---------+

| a |

| b |

| c |

+---------+

> **SELECT** \* **FROM** (**VALUES** ('a'), ('b'), ('c')) **WITH** ORDINALITY;

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

| column1 | ordinality |

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

| a | 1 |

| b | 2 |

| c | 3 |

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

WITH ORDINALITY必然会阻止对查询的一些优化。 如果考虑性能，请谨慎使用它，并在出现疑惑时始终检查EXPLAIN的输出。

Join 表达式

join表达式根据特定列的值条件组合两个或多个table表达式的结果。

查看 [Join Expressions](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/join-expressions/) 获取更多细节。

使用其他查询作为table表达式

以下部分描述如何将另一个SQL查询或语句生成的结果用作table表达式。

子查询作为table表达式

括号中包含的任何选择查询都可以用作table表达式，包括简单的SELECT子句。 这称为“子查询”。

语法:

( **...** subquery **...** )

举例:

> **SELECT** c+2 **FROM** (**SELECT** **COUNT**(\*) **AS** c **FROM** users);

> **SELECT** \* **FROM** (**VALUES**(1), (2), (3));

> **SELECT** firstname || ' ' || lastname **FROM** (**TABLE** employees);

* 有关更多详细信息和性能最佳实践，另请参阅[子查询](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/subqueries/)。
* 要使用在table表达式中生成数据的其他语句（例如SHOW），请使用[方括号表示法](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/table-expressions/#using-the-output-of-other-statements)。

使用其他语句的输出

语法:

[ <statement> ]

table表达式上下文中方括号之间的可解释语句指定执行所述语句的输出。

这是一个CockroachDB扩展。 此语法使用括号补充子查询语法，括号仅限于选择查询。 引入它是为了能够使用任何可解释的语句作为子查询，包括SHOW和其他非查询语句。

举个例子:

> **SELECT** "Field" **FROM** [**SHOW** **COLUMNS** **FROM** customer];

+---------+

| Field |

+---------+

| id |

| name |

| address |

+---------+

以下语句将Albert插入employee表中，并立即在管理表中使用自动生成的员工ID创建匹配行，而无需与SQL客户端进行交互：

> **INSERT** **INTO** management(manager, reportee)

**VALUES** ((**SELECT** id **FROM** employee **WHERE** name = 'Diana'),

(**SELECT** id **FROM** [**INSERT** **INTO** employee(name) **VALUES** ('Albert') RETURNING id]));

组合性

table表达式在选择子句的SELECT和TABLE中使用，因此可以出现在可以选择子句的任何地方。 例如：

> **SELECT** ... **FROM** <**table** expr>, <**table** expr>, ...

> **TABLE** <**table** expr>

> **INSERT** **INTO** ... **SELECT** ... **FROM** <**table** expr>, <**table** expr>, ...

> **INSERT** **INTO** ... **TABLE** <**table** expr>

> **CREATE** **TABLE** ... **AS** **SELECT** ... **FROM** <**table** expr>, <**table** expr>, ...

> UPSERT **INTO** ... **SELECT** ... **FROM** <**table** expr>, <**table** expr>, ...

有关组合查询结果的更多选项查看 [Selection Queries](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/selection-queries/).

See Also

* [Constants](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/constants/)
* [Selection Queries](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/selection-queries/)
* [Selection Clauses](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/selection-queries/)
* [Explainable Statements](http://doc.cockroachchina.baidu.com/#develop/sql-statements/EXPLAIN/#explainable-statements)
* [Scalar Expressions](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/scalar-expressions/)
* [Data Types](http://doc.cockroachchina.baidu.com/#develop/data-types/overview/)
* [Subqueries](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/subqueries/)