DELETE 语句删除一个表中的行数据。

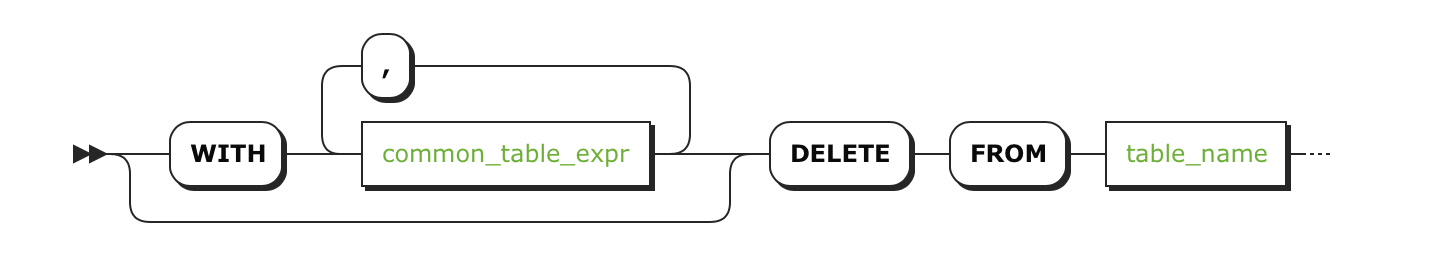
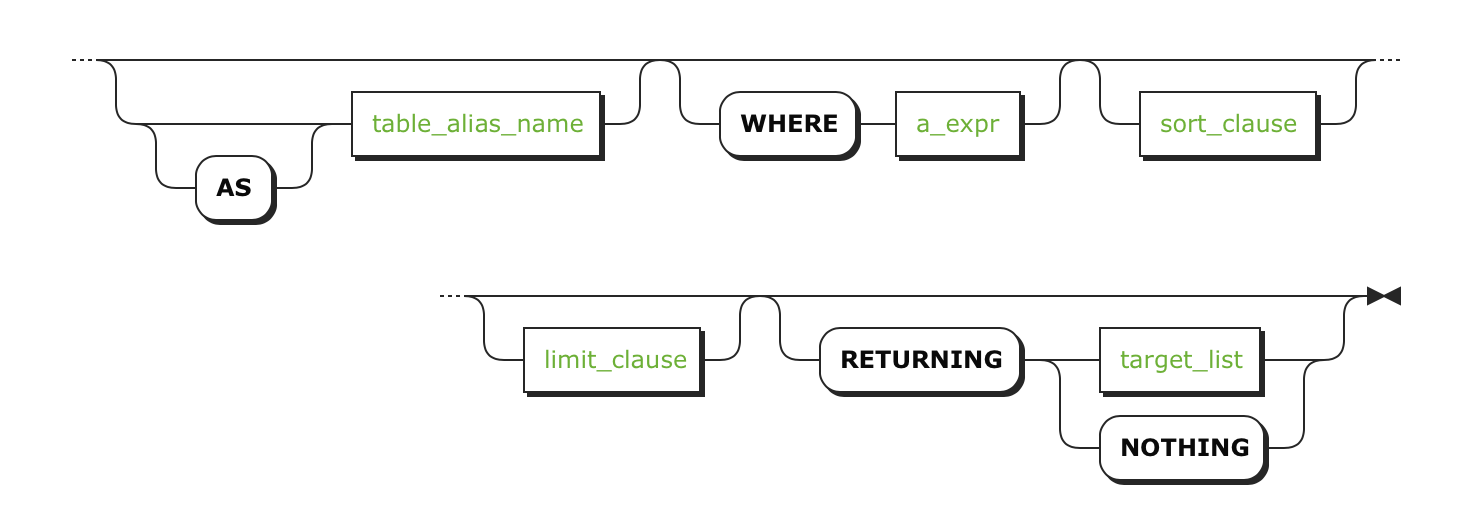
Warning： 如果你要删除由[foreign key constraint](http://doc.cockroachchina.baidu.com/#develop/sql-statements/DELETE/foreign-key.html) 引用的行， 而且有 [ON DELETE action](http://doc.cockroachchina.baidu.com/#develop/sql-statements/DELETE/foreign-key.html#foreign-key-actions-new-in-v2-0), 所有依赖这行数据的行将被删除或更新。

Note: 如果要删除列数据, 请看[DROP COLUMN](http://doc.cockroachchina.baidu.com/#develop/sql-statements/DELETE/drop-column.html)

所需权限

要执行DELETE操作，用户必须拥有DELETE与SELECT权限。

概要

参数

| **Parameter** | **Description** |
| --- | --- |
| common\_table\_expr | 请看 [Common Table Expressions](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/common-table-expressions/). |
| table\_name | 要删除的列所在的表名 |
| AS table\_alias\_name | 实际表名的一个别名. |
| WHERE a\_expr | a\_expr 必须是一个使用columns的表达式，这个表达式返回bool值(e.g. <column> = <value>)，删除使表达式返回 TRUE的行。  **如果DELETE语句中没有 WHERE条件 , DELETE 将会删除表中的所有数据.** |
| sort\_clause | 一个 ORDER BY 条件. 更多相关信息，请看 [Ordering Query Results](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/ordering-query-results/). |
| limit\_clause | 一个 LIMIT 条件. 更多相关信息，请看 [Limiting Query Results](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/limit-query-results/). |
| RETURNING target\_list | 返回删除的行数据的值, 可以是特定列的值，\* 返回所有列的值, 或者使用 [scalar expressions](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/scalar-expressions/)计算出新的值.  如果不想返回任何内容, 使用 RETURNING NOTHING. |

返回值

DELETE 语句成功执行后， 会返回下面中的一个:

| **Response** | **Description** |
| --- | --- |
| DELETE *int* | 删除的行数.  返回DELETE 0，表示DELETE 语句没有删除任何数据。 在使用 RETURNING NOTHING 之后, 返回值不会出现这个信息。 |
| Retrieved table | 检索删除的行数据，返回RETURNING条件中定义的列字段数据.  [See an example.](http://doc.cockroachchina.baidu.com/#develop/sql-statements/DELETE/#return-deleted-rows) |

Disk Space Usage After Deletes

删除行数据后，并不会立刻释放其所占的物理空间，因为CockroachDB拥有[the ability to query tables historically](https://www.cockroachlabs.com/blog/time-travel-queries-select-witty_subtitle-the_future/).

如果对于业务来说，释放磁盘空间很重要，有两个解决办法。第一，减少zone的[the time-to-live](http://doc.cockroachchina.baidu.com/#deploy/configure-replication-zones/)(TTL)，不过这会造成garbage collection 频率较高 。第二，因为[truncate](http://doc.cockroachchina.baidu.com/#develop/sql-statements/TRUNCATE/)语句能删除整张表，如果可以使用TRUNCATE ，就使用TRUNCATE

Select Performance on Deleted Rows

对于那些已经删除，但是还没有被回收机制回收的行数据，一个查询扫描一张表时，还是会扫描这些数据。所以，对于存在大量删除数据，而且时常要扫描表的使用场景时，我们需要调小[time-to-live](http://doc.cockroachchina.baidu.com/#deploy/configure-replication-zones/)参数的值。

举例

删除所有列的数据

当我们在DELETE语句中不使用WHERE条件时， 我们可以删除所有列的数据。

> **DELETE** **FROM** account\_details;

**DELETE** 7

这和 [TRUNCATE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/TRUNCATE/) 语句的功能很相似。

> TRUNCATE account\_details;

TRUNCATE

从上面可以看出, 不带WHERE条件的DELETE语句与TRUNCATE 语句不同之处为TRUNCATE 语句不返回删除的行数

TRUNCATE 语句会通过 drop table操作删除一张表的所有数据，然后会创建一张与删除表结构相同的新表。TRUNCATE 操作的性能比一行行的删除操作的性能好。

删除指定行数据

当我们要删除表中特定行的数据时，通过在WHERE中的条件来控制我们要删除那些行的数据。在WHERE条件的使用中，我们要注意使用不同列字段的作用，因为有些列有[Primary Key](http://doc.cockroachchina.baidu.com/#develop/constraints/primary-key/)/[Unique](http://doc.cockroachchina.baidu.com/#develop/constraints/unique/) 约束（保证列值的唯一性），而有些列的值不是唯一的。

使用主键/唯一键删除数据

使用有[Primary Key](http://doc.cockroachchina.baidu.com/#develop/constraints/primary-key/) or [Unique](http://doc.cockroachchina.baidu.com/#develop/constraints/unique/) 约束的列作为条件删除数据可以确保语句是确定的—某个列值不会出现在两行数据中，因此不会无意中删除其他行的数据。

在这个例子中，account\_id是表的主键，我们要删除account\_id等于1的数据。由于只会有一行数据的account\_id等于1，所以我们不会删除其他行的数据

> **DELETE** **FROM** account\_details **WHERE** account\_id = 1 RETURNING \*;

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

| account\_id | balance | account\_type |

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

| 1 | 32000 | Savings |

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

使用非唯一的列删除数据

使用非唯一的列来删除数据时，如果表中该列字段值使WHERE条件后a\_expr.表达式的值为TRUE，就删除该行数据。使用非唯一的列字段删除数据，很容易导致删除计划之外的数据。

> **DELETE** **FROM** account\_details **WHERE** balance = 30000 RETURNING \*;

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

| account\_id | balance | account\_type |

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

| 2 | 30000 | Checking |

| 3 | 30000 | Savings |

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

示例语句删除了两行，这可能是意外的。

返回删除数据

为了知道删除了哪些数据，可以使用RETURNING字段取回已删除的数据。

返回所有列数据

在 RETURNING后使用\*，返回删除数据的所有列数据

> **DELETE** **FROM** account\_details **WHERE** balance < 23000 RETURNING \*;

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

| account\_id | balance | account\_type |

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

| 4 | 22000 | Savings |

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

返回特定列的数据

在 RETURNING后加上特定列的名字，取回删除的具体列的数据

> **DELETE** **FROM** account\_details **WHERE** account\_id = 5 RETURNING account\_id, account\_type;

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

| account\_id | account\_type |

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

| 5 | Checking |

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

改变返回的列字段名字

在使用RETURNING在增加要取回列的名字时，可以使用AS更改返回数据中列的名字

> **DELETE** **FROM** account\_details **WHERE** balance < 22500 RETURNING account\_id, balance **AS** final\_balance;

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

| account\_id | final\_balance |

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

| 6 | 23500 |

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

See Also

* [INSERT](http://doc.cockroachchina.baidu.com/#develop/sql-statements/INSERT/)
* [UPDATE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/UPDATE/)
* [UPSERT](http://doc.cockroachchina.baidu.com/#develop/sql-statements/UPSERT/)
* [TRUNCATE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/TRUNCATE/)
* [ALTER TABLE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/ALTER-TABLE/)
* [DROP TABLE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/DROP-TABLE/)
* [DROP DATABASE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/DROP-DATABASE/)
* [Other SQL Statements](http://doc.cockroachchina.baidu.com/#develop/sql-statements/overview/)
* [Limiting Query Results](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/limit-query-results/)