SQL子查询允许重用在另一个查询的查询结果。

CockroachDB支持两种子查询：

* 关系子查询支持选择查询或table表达式两种方式。
* 标量子查询支持标量表达式。

子查询中的数据写入

当子查询包含数据修改语句（INSERT，DELETE等）时，即使范围查询仅使用结果的子集，数据修改也能执行完成。

可以使用（...）或[...]符号定义子查询，它们是使用WITH来定义。

> **SELECT** \*

**FROM** [**INSERT** **INTO** t(x) **VALUES** (1), (2), (3) RETURNING x]

**LIMIT** 1;

此查询总是在t中插入3行，即使范围查询仅使用LIMIT查看1行

相关子查询

当子查询使用在范围查询中定义的表或列名称时，它被称为“correlated”。

此时，CockroachDB仅支持非相关子查询：子查询中列出的所有表名和列名必须在子查询本身中定义。

如果你想要使用相关子查询，请考虑使用join表达式将相关子查询转换为非相关子查询:

举个例子：

# Find every customer with at least one order.

> **SELECT** c.name

**FROM** customers c

**WHERE** **EXISTS**(**SELECT** \* **FROM** orders o **WHERE** o.customer\_id = c.id);

子查询是相关的，因为它使用范围查询中定义的c。 因此，CockroachDB尚未支持它; 但是，它可以转换为等效查询：

> **SELECT** **DISTINCT** **ON**(c.id) c.name

**FROM** customers c **CROSS** **JOIN** orders o

**WHERE** c.id = o.customer\_id;

另请参阅[Stack Overflow上的这个问题：将子查询转换为连接](https://stackoverflow.com/questions/1772609/procedurally-transform-subquery-into-join)

CockroachDBs目前正在进行重大更改，以引入对相关子查询的支持。 预计此限制将在未来版本中解除。

性能最佳实践

CockroachDBs目前正在进行重大变革，以发展和改进子查询的性能。 随着时间的推移，本节中列出的限制和变更方法将被解除或变得不必要。

* 当前scalar子查询禁止执行分布式查询。 为了确保处理大量行查询的最大性能，请使客户端应用程序提前计算子查询结果，并将这些结果直接传递给范围查询。
* 当范围查询的执行开始时，scalar子查询的结果当前完全加载到内存中。 要防止因内存耗尽而导致执行错误，请确保子查询返回尽可能少的结果。

See Also

* [Selection Queries](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/selection-queries/)
* [Scalar Expressions](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/scalar-expressions/)
* [Table Expressions](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/table-expressions/)
* [Performance Best Practices - Overview](http://doc.cockroachchina.baidu.com/#develop/performance-optimization/sql-best-practices/)