EXPLAIN [语句](http://doc.cockroachchina.baidu.com/#develop/sql-statements/overview/) 返回[指定语句（可解释）](http://doc.cockroachchina.baidu.com/#develop/sql-statements/EXPLAIN/#explainable-statements)的CockroachDB查询计划，用户可以使用EXPLAIN语句优化查询。

可解释语句

EXPLAIN能够解释以下语句：

* [ALTER USER](https://www.cockroachlabs.com/docs/stable/sql-grammar.html#alter_user_stmt), [ALTER TABLE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/ALTER-TABLE/), [ALTER INDEX](http://doc.cockroachchina.baidu.com/#develop/sql-statements/ALTER-INDEX/), [ALTER VIEW](http://doc.cockroachchina.baidu.com/#develop/sql-statements/alert-view/), [ALTER DATABASE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/ALTER-DATABASE/), [ALTER SEQUENCE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/ALTER-SEQUENCE/)
* [BACKUP](http://doc.cockroachchina.baidu.com/#develop/sql-statements/BACKUP/)
* [CANCEL JOB](http://doc.cockroachchina.baidu.com/#develop/sql-statements/CANCEL-JOB/), [CANCEL QUERY](http://doc.cockroachchina.baidu.com/#develop/sql-statements/CANCEL-QUERY/)
* [CREATE DATABASE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/CREATE-DATABASE/), [CREATE INDEX](http://doc.cockroachchina.baidu.com/#develop/sql-statements/CREATE-INDEX/), [CREATE TABLE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/CREATE-TABLE/), [CREATE TABLE AS](http://doc.cockroachchina.baidu.com/#develop/sql-statements/CREATE-TABLE-AS/), [CREATE USER](http://doc.cockroachchina.baidu.com/#develop/sql-statements/CREATE-USER/), [CREATE VIEW](http://doc.cockroachchina.baidu.com/#develop/sql-statements/CREATE-VIEW/), [CREATE SEQUENCE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/CREATE-SEQUENCE/)
* [DELETE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/DELETE/)
* [DROP DATABASE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/DROP-DATABASE/), [DROP INDEX](http://doc.cockroachchina.baidu.com/#develop/sql-statements/DROP-INDEX/), [DROP SEQUENCE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/DROP-SEQUENCE/), [DROP TABLE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/DROP-TABLE/), [DROP USER](http://doc.cockroachchina.baidu.com/#develop/sql-statements/DROP-USER/), [DROP VIEW](http://doc.cockroachchina.baidu.com/#develop/sql-statements/DROP-VIEW/)
* [EXECUTE](https://www.cockroachlabs.com/docs/stable/sql-grammar.html#execute_stmt)
* EXPLAIN
* [IMPORT](http://doc.cockroachchina.baidu.com/#develop/sql-statements/IMPORT/)
* [INSERT](http://doc.cockroachchina.baidu.com/#develop/sql-statements/INSERT/)
* [PAUSE JOB](http://doc.cockroachchina.baidu.com/#develop/sql-statements/PAUSE-JOB/)
* [RESET](http://doc.cockroachchina.baidu.com/#develop/sql-statements/RESET-(session-variable)/)
* [RESTORE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/RESTORE/)
* [RESUME JOB](http://doc.cockroachchina.baidu.com/#develop/sql-statements/RESUME-JOB/)
* [SELECT](http://doc.cockroachchina.baidu.com/#develop/sql-statements/SELECT/) and any [selection query](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/selection-queries/)
* [SET](http://doc.cockroachchina.baidu.com/#develop/sql-statements/SET(session-variable)/)
* [SET CLUSTER SETTING](http://doc.cockroachchina.baidu.com/#develop/sql-statements/SET-CLUSTER-SETTING/)
* [SHOW BACKUP](http://doc.cockroachchina.baidu.com/#develop/sql-statements/SHOW-BACKUP/), [SHOW COLUMNS](http://doc.cockroachchina.baidu.com/#develop/sql-statements/SHOW-COLUMNS/), [SHOW CONSTRAINTS](http://doc.cockroachchina.baidu.com/#develop/sql-statements/SHOW-CONSTRAINTS/), [SHOW CREATE TABLE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/show-create-table/), [SHOW CREATE VIEW](http://doc.cockroachchina.baidu.com/#develop/sql-statements/SHOW-CREATE-VIEW/), [SHOW CREATE SEQUENCE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/SHOW-CREATE-SEQUENCE/), [SHOW CLUSTER SETTING](http://doc.cockroachchina.baidu.com/#develop/sql-statements/SHOW-CLUSTER-SETTING/), [SHOW DATABASES](http://doc.cockroachchina.baidu.com/#develop/sql-statements/SHOW-DATABASES/), [SHOW GRANTS](http://doc.cockroachchina.baidu.com/#develop/sql-statements/SHOW-GRANTS/), [SHOW INDEX](http://doc.cockroachchina.baidu.com/#develop/sql-statements/SHOW-INDEX/), [SHOW JOBS](http://doc.cockroachchina.baidu.com/#develop/sql-statements/SHOW-JOBS/), [SHOW QUERIES](http://doc.cockroachchina.baidu.com/#develop/sql-statements/SHOW-QUERIES/), [SHOW SESSIONS](http://doc.cockroachchina.baidu.com/#develop/sql-statements/SHOW-SESSIONS/), [SHOW TABLES](http://doc.cockroachchina.baidu.com/#develop/sql-statements/SHOW-TABLES/), [SHOW TRACE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/SHOW-TRACE/), [SHOW USERS](http://doc.cockroachchina.baidu.com/#develop/sql-statements/SHOW-USERS/), [SHOW HISTOGRAM](https://www.cockroachlabs.com/docs/stable/sql-grammar.html#show_histogram_stmt)
* [UPDATE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/UPDATE/)
* [UPSERT](http://doc.cockroachchina.baidu.com/#develop/sql-statements/UPSERT/)

查询计划

使用EXPLAIN输出，用户可以从以下角度进行优化查询：

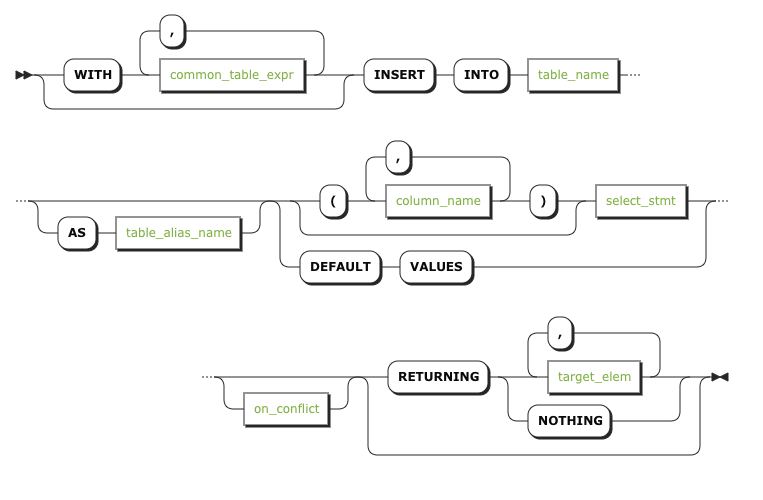
* 更少的查询嵌套层次能够使得查询执行得更快，获得更好的查询性能。
* 避免代价很高的全表扫描。用户可以[创建包含了WHERE子句涉及属性的索引](http://doc.cockroachchina.baidu.com/#develop/performance-optimization/indexes/)避免可能的全表扫描。

基于EXPLAIN语句输出，用户可以用过以下方式确定查询是否涉及全表扫描：

* 通过查看**Field**属性值为table的行的**Description**值，获取查询使用的索引。
* 通过查看**Field**属性值为spans的行的**Description**值，获得索引中需要被扫描的Key值范围。

更多细节，用户可以查看[Find the Indexes and Key Ranges a Query Uses](http://doc.cockroachchina.baidu.com/#develop/sql-statements/EXPLAIN/#find-the-indexes-and-key-ranges-a-query-uses).

概要



所需权限

用户必须拥有指定表的可解释语句的[权限](http://doc.cockroachchina.baidu.com/#deploy/access-management/privileges/) 。

参数

| **参数** | **描述** |
| --- | --- |
| EXPRS | 每个处理阶段包含的SQL语句。 |
| QUALIFY | 输出引用列所在的表名，特别在两张表基于同名属性进行JOIN行为的情况下能够避免同名歧义。  用户使用QUALIFY时需要配合EXPRS使用。 |
| METADATA | 在**Columns**中显示每层所使用的列，同时包含**Ordering**信息。 |
| VERBOSE | 同时使用EXPRS、METADATA和QUALIFY选项的缩写 |
| TYPES | 包括CockroachDB用于评估中间SQL表达式的[数据类型](http://doc.cockroachchina.baidu.com/#develop/data-types/overview/)。  TYPES选项内部实现了METADATA和EXPRS。 |
| explainable\_stmt | 指定[可解释语句](http://doc.cockroachchina.baidu.com/#develop/sql-statements/EXPLAIN/#explainable-statements)。 |

**Warning:** EXPLAIN 还包含了除查询计划以外的其他模式，这些模式仅对CockroachDB开发人员有用。

执行成功后返回的内容

执行EXPLAIN语句成功后，返回具有以下属性列的表数据。

| **列** | **简介** |
| --- | --- |
| **Tree** | 以树的形式展示查询计划的层级结构。 |
| **Field** | 查询计划中节点对应的参数名。 |
| **Description** | 对应**Field**列中显示的参数的额外信息 |
| **Columns** | 提供给往下层级查询计划使用的属性列。  需要使用METADATA选项，或是涉及METADATA的其他选项 |
| **Ordering** | 呈现给层次结构中每个级别的执行结果的顺序，同时包括该层的其他属性。  需要配合使用METADATA选项，或是其他涉及METADATA的选项 |

示例

默认查询计划

默认情况下，EXPLAIN语句将输出包含了该查询涉及到的索引和扫描的Key值范围的信息。

> **EXPLAIN** **SELECT** \* **FROM** kv **WHERE** v > 3 **ORDER** **BY** v;

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

| Tree | Field | Description |

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

| sort | | |

| │ | order | +v |

| └── scan | | |

| | table | kv@primary |

| | spans | ALL |

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

第一行将显示查询计划的树状结构，以及层次结构中涉及到节点的一系列属性。值得注意的是，用户可以看到上述示例中primary索引被扫描，且扫描范围为全表。

EXPRS选项

EXPRS选项包含了每个处理阶段涉及到的SQL表达式以及每一行对应的额外的信息。

> **EXPLAIN** (EXPRS) **SELECT** \* **FROM** kv **WHERE** v > 3 **ORDER** **BY** v;

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

| Tree | Field | Description |

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

| sort | | |

| │ | order | +v |

| └── scan | | |

| | table | kv@primary |

| | spans | ALL |

| | filter | v > 3 |

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

METADATA选项

METADATA选项包含了树状结构中每一层需要使用的属性列的详细信息。

> **EXPLAIN** (METADATA) **SELECT** \* **FROM** kv **WHERE** v > 3 **ORDER** **BY** v;

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

| Tree | Level | Type | Field | Description | Columns | Ordering |

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

| sort | 0 | sort | | | (k, v) | k!=NULL; v!=NULL; key(k); +v |

| │ | 0 | | order | +v | | |

| └── scan | 1 | scan | | | (k, v) | k!=NULL; v!=NULL; key(k) |

| | 1 | | table | kv@primary | | |

| | 1 | | spans | ALL | | |

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

**Ordering** 列包含了重要的信息，包括该级别的行的排序（上述例子为+v标记，代表升序，而降序为-标记），以及改行对应的其他信息。上述例子中CockroachDB推断k和v不能为空，且k作为键（意味着同一键只对应不超过一行的值）。

> **EXPLAIN** (METADATA) **SELECT** \* **FROM** kv **WHERE** v > 3 **ORDER** **BY** v **DESC**;

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

| Tree | Level | Type | Field | Description | Columns | Ordering |

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

| sort | 0 | sort | | | (k, v) | k!=NULL; v!=NULL; key(k); -v |

| │ | 0 | | order | -v | | |

| └── scan | 1 | scan | | | (k, v) | k!=NULL; v!=NULL; key(k) |

| | 1 | | table | kv@primary | | |

| | 1 | | spans | ALL | | |

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

**Ordering**列也包含了一部分其他信息，包含了在任何行上值相等的多个列的信息，或者在所有行上具有相同值的“常量”列的信息。 例如：

> **EXPLAIN** (METADATA) **SELECT** \* **FROM** abcd **JOIN** efg **ON** a=e **AND** c=1;

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

| Tree | Level | Type | Field | Description | Columns | Ordering |

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

| join | 0 | join | | | (a, b, c, d, e, f, g) | a=e; c=CONST; a!=NULL; key(a) |

| │ | 0 | | type | inner | | |

| │ | 0 | | equality | (a) = (e) | | |

| │ | 0 | | mergeJoinOrder | +"(a=e)" | | |

| ├── scan | 1 | scan | | | (a, b, c, d) | c=CONST; a!=NULL; key(a); +a |

| │ | 1 | | table | abcd@primary | | |

| │ | 1 | | spans | ALL | | |

| └── scan | 1 | scan | | | (e, f, g) | e!=NULL; key(e); +e |

| | 1 | | table | efg@primary | | |

| | 1 | | spans | ALL | | |

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

第二行中我们可以看到a和e列具有相同值，且c列为“常量”列。

QUALIFY选项

QUALIFY使用<table name>.<column name>格式显示查询计划中涉及的属性列，使用时需要注意配合EXPRS一起使用。

> **EXPLAIN** (EXPRS, QUALIFY) **SELECT** a.v, b.v **FROM** t.kv **AS** a, t.kv **AS** b;

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

| Tree | Field | Description |

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

| render | | |

| │ | render 0 | a.v |

| │ | render 1 | b.v |

| └── join | | |

| │ | type | cross |

| ├── scan | | |

| │ | table | kv@primary |

| │ | spans | ALL |

| └── scan | | |

| | table | kv@primary |

| | spans | ALL |

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

用户可以不指定QUALIFY选项执行EXPLAIN语句，查看输出结果之间的区别。

> **EXPLAIN** (EXPRS) **SELECT** a.v, b.v **FROM** kv **AS** a, kv **AS** b;

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

| Level | Type | Field | Description |

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

| 0 | render | | |

| 0 | | render 0 | v |

| 0 | | render 1 | v |

| 1 | join | | |

| 1 | | type | cross |

| 2 | scan | | |

| 2 | | table | kv@primary |

| 2 | scan | | |

| 2 | | table | kv@primary |

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

VERBOSE选项

VERBOSE选项是EXPRS、METADATA、QUALIFY三个选项组合的缩写。

> **EXPLAIN** (VERBOSE) **SELECT** \* **FROM** kv **AS** a **JOIN** kv **USING** (k) **WHERE** a.v > 3 **ORDER** **BY** a.v **DESC**;

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

| Tree | Level | Type | Field | Description | Columns | Ordering |

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

| sort | 0 | sort | | | (k, v, v) | k!=NULL; key(k); -v |

| │ | 0 | | order | -v | | |

| └── render | 1 | render | | | (k, v, v) | k!=NULL; key(k) |

| │ | 1 | | render 0 | a.k | | |

| │ | 1 | | render 1 | a.v | | |

| │ | 1 | | render 2 | radu.public.kv.v | | |

| └── join | 2 | join | | | (k, v, k[omitted], v) | k=k; k!=NULL; key(k) |

| │ | 2 | | type | inner | | |

| │ | 2 | | equality | (k) = (k) | | |

| │ | 2 | | mergeJoinOrder | +"(k=k)" | | |

| ├── scan | 3 | scan | | | (k, v) | k!=NULL; v!=NULL; key(k); +k |

| │ | 3 | | table | kv@primary | | |

| │ | 3 | | spans | ALL | | |

| │ | 3 | | filter | v > 3 | | |

| └── scan | 3 | scan | | | (k, v) | k!=NULL; key(k); +k |

| | 3 | | table | kv@primary | | |

| | 3 | | spans | ALL | | |

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

TYPES选项

TYPES包含了查询计划中使用的值的类型，内部涉及了METADATA和EXPRS选项：

> **EXPLAIN** (TYPES) **SELECT** \* **FROM** kv **WHERE** v > 3 **ORDER** **BY** v;

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

| Tree | Level | Type | Field | **Description** | Columns | Ordering |

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

| **sort** | 0 | **sort** | | | (k **int**, v **int**) | k!=**NULL**; v!=**NULL**; key(k); +v |

| │ | 0 | | order | +v | | |

| └── scan | 1 | scan | | | (k **int**, v **int**) | k!=**NULL**; v!=**NULL**; key(k) |

| | 1 | | table | kv@primary | | |

| | 1 | | spans | ALL | | |

| | 1 | | filter | ((v)[**int**] > (3)[**int**])[bool] | | |

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

确定查询涉及到的索引和Key值范围

用户可以执行EXPLAIN语句查看查询涉及到的索引和Key值范围，从而确定该查询是否涉及全表扫描。

> **CREATE** **TABLE** kv (k INT **PRIMARY** **KEY**, v INT);

由于v属性列未被索引，该查询将进行全表扫描来过滤出满足WHERE子句的行。

> **EXPLAIN** **SELECT** \* **FROM** kv **WHERE** v **BETWEEN** 4 **AND** 5;

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

| Level | Type | Field | Description |

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

| 0 | scan | | |

| 0 | | table | kv@primary |

| 0 | | spans | ALL |

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

如果在v属性列建立索引，则能够避免全表扫描。

> **CREATE** **INDEX** v **ON** kv (v);

> **EXPLAIN** **SELECT** \* **FROM** kv **WHERE** v **BETWEEN** 4 **AND** 5;

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

| Tree | Field | Description |

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

| scan | | |

| | table | kv@v |

| | spans | /4-/6 |

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

可以看到该查询改换全表扫描为索引v的扫描，key值范围为(4, 6]。

其他

* [ALTER TABLE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/ALTER-TABLE/)
* [ALTER SEQUENCE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/ALTER-SEQUENCE/)
* [BACKUP](http://doc.cockroachchina.baidu.com/#develop/sql-statements/BACKUP/)
* [CANCEL JOB](http://doc.cockroachchina.baidu.com/#develop/sql-statements/CANCEL-JOB/)
* [CREATE DATABASE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/CREATE-DATABASE/)
* [DROP DATABASE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/DROP-DATABASE/)
* [EXECUTE](https://www.cockroachlabs.com/docs/stable/sql-grammar.html#execute_stmt)
* [IMPORT](http://doc.cockroachchina.baidu.com/#develop/sql-statements/IMPORT/)
* [Indexes](http://doc.cockroachchina.baidu.com/#develop/performance-optimization/indexes/)
* [INSERT](http://doc.cockroachchina.baidu.com/#develop/sql-statements/INSERT/)
* [PAUSE JOB](http://doc.cockroachchina.baidu.com/#develop/sql-statements/PAUSE-JOB/)
* [RESET](http://doc.cockroachchina.baidu.com/#develop/sql-statements/RESET-(session-variable)/)
* [RESTORE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/RESTORE/)
* [RESUME JOB](http://doc.cockroachchina.baidu.com/#develop/sql-statements/RESUME-JOB/)
* [SELECT](http://doc.cockroachchina.baidu.com/#develop/sql-statements/SELECT/)
* [Selection Queries](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/selection-queries/)
* [SET](http://doc.cockroachchina.baidu.com/#develop/sql-statements/SET(session-variable)/)
* [SET CLUSTER SETTING](http://doc.cockroachchina.baidu.com/#develop/sql-statements/SET-CLUSTER-SETTING/)
* [SHOW COLUMNS](http://doc.cockroachchina.baidu.com/#develop/sql-statements/SHOW-COLUMNS/)
* [UPDATE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/UPDATE/)
* [UPSERT](http://doc.cockroachchina.baidu.com/#develop/sql-statements/UPSERT/)