- Estimated number of returned rows
- Time to return first row
- Time to return all rows (actual cost), in milliseconds

  (When there are multiple loops, this figure shows the average time per loop.)
- Number of rows returned by the iterator
- Number of loops

The query execution information is displayed using the `TREE` output format, in which nodes represent iterators. `EXPLAIN ANALYZE` always uses the `TREE` output format. In MySQL 8.0.21 and later, this can optionally be specified explicitly using `FORMAT=TREE`; formats other than `TREE` remain unsupported.

`EXPLAIN ANALYZE` can be used with `SELECT` statements, as well as with multi-table `UPDATE` and `DELETE` statements. Beginning with MySQL 8.0.19, it can also be used with `TABLE` statements.

Beginning with MySQL 8.0.20, you can terminate this statement using `KILL QUERY` or **CTRL-C**.

`EXPLAIN ANALYZE` cannot be used with `FOR CONNECTION`.

Example output:

```
mysql> EXPLAIN ANALYZE SELECT * FROM t1 JOIN t2 ON (t1.c1 = t2.c2)\G
*************************** 1. row ***************************
EXPLAIN: -> Inner hash join (t2.c2 = t1.c1)  (cost=4.70 rows=6)
(actual time=0.032..0.035 rows=6 loops=1)
    -> Table scan on t2  (cost=0.06 rows=6)
(actual time=0.003..0.005 rows=6 loops=1)
    -> Hash
        -> Table scan on t1  (cost=0.85 rows=6)
(actual time=0.018..0.022 rows=6 loops=1)

mysql> EXPLAIN ANALYZE SELECT * FROM t3 WHERE i > 8\G
*************************** 1. row ***************************
EXPLAIN: -> Filter: (t3.i > 8)  (cost=1.75 rows=5)
(actual time=0.019..0.021 rows=6 loops=1)
    -> Table scan on t3  (cost=1.75 rows=15)
(actual time=0.017..0.019 rows=15 loops=1)

mysql> EXPLAIN ANALYZE SELECT * FROM t3 WHERE pk > 17\G
*************************** 1. row ***************************
EXPLAIN: -> Filter: (t3.pk > 17)  (cost=1.26 rows=5)
(actual time=0.013..0.016 rows=5 loops=1)
    -> Index range scan on t3 using PRIMARY  (cost=1.26 rows=5)
(actual time=0.012..0.014 rows=5 loops=1)
```

The tables used in the example output were created by the statements shown here:

```
CREATE TABLE t1 (
    c1 INTEGER DEFAULT NULL,
    c2 INTEGER DEFAULT NULL
);

CREATE TABLE t2 (
    c1 INTEGER DEFAULT NULL,
    c2 INTEGER DEFAULT NULL
);

CREATE TABLE t3 (
    pk INTEGER NOT NULL PRIMARY KEY,
```