

解读命令输出

目录

1

explain

2

show engine innodb status

构造数据

```
create table t1(id int primary key auto_increment, a int , b int, c int ,  
index (a), index bc(b,c))engine=innodb;  
insert into t1(a,b,c) values(1,1,1),(2,2,2),(3,3,3),(4,4,4);  
insert into t1(a,b,c) select a,b,c from t1;//重复插入到16384行  
insert into t1(a,b,c) values(5,5,5);  
insert into t1(a,b,c) select a,b,c from t1;  
insert into t1(a,b,c) select a,b,c from t1; //总共65540行
```

```
create table t2 like t1;  
insert into t2 select * from t1;
```

explain

```
explain select * from t1 where id =12000;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	t1	NULL	const	PRIMARY	PRIMARY	4	const	1	100.00	NULL

1 row in set, 1 warning (0.00 sec)

explain

explain select * from t1 where a =12000;

```
mysql> explain select * from t1 where a= 12000;
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | t1 | NULL | ref | a | a | 5 | const | 1 | 100.00 | NULL |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set, 1 warning (0.01 sec)
```

explain analyze select * from t1 where a =12000\G

```
mysql> explain analyze select * from t1 where a= 12000\G
***** 1. row *****
EXPLAIN: -> Index lookup on t1 using a (a=12000) (cost=0.35 rows=1) (actual time=0.297..0.297 rows=0 loops=1)

1 row in set (0.00 sec)
```


explain

explain select * from t1 where c =12000;

```
mysql> explain select * from t1 where c =12000;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	t1	NULL	ALL	NULL	NULL	NULL	NULL	65758	10.00	Using where

```
1 row in set, 1 warning (0.00 sec)
```

explain analyze select * from t1 where c =12000\G

```
mysql> explain analyze select * from t1 where c =12000\G
```

```
***** 1. row *****
```

```
EXPLAIN: -> Filter: (t1.c = 12000) (cost=6616 rows=6576) (actual time=378..378 rows=0 loops=1)
```

```
-> Table scan on t1 (cost=6616 rows=65758) (actual time=56.5..358 rows=65540 loops=1)
```

```
1 row in set (0.38 sec)
```

explain

explain select id from t1 where c =12000;

```
mysql> explain select id from t1 where c =12000;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | t1 | NULL | index | b | b | 10 | NULL | 65758 | 10.00 | Using where; Using index |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set, 1 warning (0.01 sec)
```

explain analyze select id from t1 where c =12000\G

```
mysql> explain analyze select id from t1 where c =12000\G
***** 1. row *****
EXPLAIN: -> Filter: (t1.c = 12000) (cost=6616 rows=6576) (actual time=5765..5765 rows=0 loops=1)
        -> Covering index scan on t1 using b (cost=6616 rows=65758) (actual time=588..5746 rows=65540 loops=1)

1 row in set (5.77 sec)
```


explain

explain select * from t1 a, t2 b where a.id=b.id

```
mysql> explain select * from t1 a, t2 b where a.id=b.id and a.a=12000;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	a	NULL	ref	PRIMARY,a	a	5	const	1	100.00	NULL
1	SIMPLE	b	NULL	eq_ref	PRIMARY	PRIMARY	4	test.a.id	1	100.00	NULL

2 rows in set, 1 warning (0.00 sec)

explain analyze select * from t1 a, t2 b where a.id=b.id and a.a=12000\G

```
mysql> explain analyze select * from t1 a, t2 b where a.id=b.id and a.a=12000\G
```

```
***** 1. row *****
```

```
EXPLAIN: -> Nested loop inner join (cost=0.7 rows=1) (actual time=0.26..0.26 rows=0 loops=1)
  -> Index lookup on a using a (a=12000) (cost=0.35 rows=1) (actual time=0.257..0.257 rows=0 loops=1)
  -> Single-row index lookup on b using PRIMARY (id=a.id) (cost=0.35 rows=1) (never executed)
```

```
1 row in set (0.01 sec)
```

explain

explain select c from t1 where a=2 and c=2 order by

```
mysql> explain select c from t1 where a=2 and c=2 order by b;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	t1	NULL	ref	a	a	5	const	29006	10.00	Using where; Using filesort

1 row in set, 1 warning (0.00 sec)

explain analyze select c from t1 where a=2 and c=2

order by b\G

```
mysql> explain analyze select c from t1 where a=2 and c=2 order by b\G
```

```
***** 1. row *****
```

```
EXPLAIN: -> Sort: t1.b (cost=411 rows=29006) (actual time=1492..1496 rows=16384 loops=1)
```

```
-> Filter: (t1.c = 2) (cost=411 rows=29006) (actual time=271..1478 rows=16384 loops=1)
```

```
-> Index lookup on t1 using a (a=2) (cost=411 rows=29006) (actual time=271..1473 rows=16384 loops=1)
```

```
1 row in set (1.51 sec)
```

explain

explain select count(*), c from t1 where a>2 group by c;

```
mysql> explain select count(*), c from t1 where a>2 group by c;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	t1	NULL	ALL	a,b	NULL	NULL	NULL	65758	50.00	Using where; Using temporary

1 row in set, 1 warning (0.01 sec)

explain analyze select count(*), c from t1 where a>2 group by c;

```
| -> Table scan on <temporary> (actual time=405..405 rows=3 loops=1)
    -> Aggregate using temporary table (actual time=405..405 rows=3 loops=1)
        -> Filter: (t1.a > 2) (cost=6616 rows=32879) (actual time=51..339 rows=32772 loops=1)
            -> Table scan on t1 (cost=6616 rows=65758) (actual time=51..319 rows=65540 loops=1)
|
```


explain

explain select distinct c from t1 where a>1;

```
mysql> explain select distinct c from t1 where a>1;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	t1	NULL	ALL	a,b	NULL	NULL	NULL	65758	50.00	Using where; Using temporary

1 row in set, 1 warning (0.01 sec)

explain analyze select distinct c from t1 where

```
| -> Table scan on <temporary> (cost=9904..10317 rows=32879) (actual time=441..441 rows=4 loops=1)
    -> Temporary table with deduplication (cost=9904..9904 rows=32879) (actual time=441..441 rows=4 loops=1)
        -> Filter: (t1.a > 1) (cost=6616 rows=32879) (actual time=51.1..340 rows=49156 loops=1)
            -> Table scan on t1 (cost=6616 rows=65758) (actual time=51.1..321 rows=65540 loops=1)
```

explain

explain select distinct id from t1 where a>1;

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	t1	NULL	range	PRIMARY,a,b	a	5	NULL	32879	100.00	Using where; Using index

1 row in set, 1 warning (0.00 sec)

explain analyze select distinct id from t1

where a>1;

```

-> Filter: (t1.a > 1) (cost=6585 rows=32879) (actual time=20.1..242 rows=49156 loops=1)
  -> Covering index range scan on t1 using a over (1 < a) (cost=6585 rows=32879) (actual time=20.1..229 rows=49156 loops=1)

```


explain

explain select distinct id+0 from t1 where a>1;

```
mysql> explain select distinct id+0 from t1 where a>1;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	t1	NULL	range	PRIMARY,a,b	a	5	NULL	32879	100.00	Using where; Using index; Using temporary

1 row in set, 1 warning (0.00 sec)

explain analyze select distinct id+0 from t1

```

-> Table scan on <temporary> (cost=9873..10286 rows=32879) (actual time=315..339 rows=49156 loops=1)
  -> Temporary table with deduplication (cost=9873..9873 rows=32879) (actual time=315..315 rows=49156 loops=1)
    -> Filter: (t1.a > 1) (cost=6585 rows=32879) (actual time=20.1..229 rows=49156 loops=1)
      -> Covering index range scan on t1 using a over (1 < a) (cost=6585 rows=32879) (actual time=20.1..214 rows=49156 loops=1)

```

explain

```
set optimizer_trace =1;
```

```
explain analyze select distinct id+0 from t1 where a>1;
```

```
select * from information_schema.optimizer_trace\G
```

```
{
  "join_execution": {
    "select#": 1,
    "steps": [
      {
        "creating_tmp_table": {
          "tmp_table_info": {
            "table": "<temporary>",
            "in_plan_at_position": 1,
            "columns": 1,
            "row_length": 9,
            "key_length": 8,
            "unique_constraint": false,
            "makes_grouped_rows": true,
            "cannot_insert_duplicates": false,
            "location": "TempTable"
          }
        }
      }
    ]
  },
  {
    "materialize": {
      "select#": 1,
      "steps": [
        ]
      }
    }
  ]
}
```

explain

```
set optimizer_trace =1;
```

```
set tmp_table_size=1024;explain analyze select distinct id+0 from t1  
where a>1;
```

```
... optimizer_trace\G
```

```
{  
  "converting_tmp_table_to_ondisk": {  
    "cause": "memory_table_size_exceeded",  
    "tmp_table_info": {  
      "table": "<subquery2>",  
      "in_plan_at_position": 0,  
      "columns": 1,  
      "row_length": 9,  
      "key_length": 9,  
      "unique_constraint": false,  
      "makes_grouped_rows": false,  
      "cannot_insert_duplicates": true,  
      "location": "disk (InnoDB)",  
      "record_format": "fixed"  
    }  
  }  
}
```

换个脑子休息一下

```
set transaction_isolation='repeatable-read';  
start transaction with consistent snapshot;
```

```
set transaction_isolation='read-committed';  
start transaction with consistent snapshot;
```


explain

```
explain select c from t1 where a=2 and b>1
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	t1	NULL	ref	a,b	a	5	const	29006	50.00	Using where; Using filesort

1 row in set, 1 warning (0.00 sec)

```
explain analyze select c from t1 where a=2 and b>1 order by
```

```
EXPLAIN: -> Limit/Offset: 10/9000 row(s) (cost=1571 rows=10) (actual time=1497..1497 rows=10 loops=1)
-> Sort: t1.c, limit input to 9010 row(s) per chunk (cost=1571 rows=29006) (actual time=1493..1496 rows=9010 loops=1)
-> Filter: (t1.b > 1) (cost=1571 rows=29006) (actual time=270..1477 rows=16384 loops=1)
-> Index lookup on t1 using a (a=2) (cost=1571 rows=29006) (actual time=270..1472 rows=16384 loops=1)
```


explain

explain select c from t1 where a=3 and b>1
order by b limit 10;

```
mysql> explain select c from t1 where a=3 and b>1 order by b limit 10;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	t1	NULL	range	a,b	b	5	NULL	32879	45.94	Using index condition; Using where

1 row in set, 1 warning (0.00 sec)

explain analyze select c from t1 where a=3 and b>1
order by b limit 10;

```
mysql> explain analyze select c from t1 where a=3 and b>1 order by b limit 10\G
***** 1. row *****
EXPLAIN: -> Limit: 10 row(s) (cost=1631 rows=10) (actual time=1521..1521 rows=10 loops=1)
        -> Filter: (t1.a = 3) (cost=1631 rows=15103) (actual time=1521..1521 rows=10 loops=1)
            -> Index range scan on t1 using b over (1 < b), with index condition: (t1.b > 1) (cost=1631 rows=32879) (actual time=324..1516 rows=16394 loops=1)
```

分explain

explain select c from t1 where a=3 and b>1 order by b limit 300,10;

```
mysql> explain select c from t1 where a=3 and b>1 order by b limit 300,10;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	t1	NULL	range	a,b	b	5	NULL	32879	45.94	Using index condition; Using where

1 row in set, 1 warning (0.01 sec)

explain analyze select c from t1 where a=3 and b>1 order by b limit 300,10;

```
***** 1. row *****
EXPLAIN: -> Limit/Offset: 10/300 row(s) (cost=1993 rows=10) (actual time=1808..1808 rows=10 loops=1)
-> Filter: (t1.a = 3) (cost=1993 rows=15103) (actual time=1517..1808 rows=310 loops=1)
-> Index range scan on t1 using b over (1 < b), with index condition: (t1.b > 1) (cost=1993 rows=32879) (actual time=328..1803 rows=16694 loops=1)

1 row in set (1.82 sec)
```

explain

explain select * from t1,t2 where t1.b=t2.b and t1.id<t2.id

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	t1	NULL	range	PRIMARY,a,b	a	5	NULL	32879	100.00	Using index condition; Using where
1	SIMPLE	t2	NULL	ref	PRIMARY,b	b	5	test.t1.b	16264	33.33	Using index condition

2 rows in set, 1 warning (0.00 sec)

explain analyze select * from t1,t2 where t1.b=t2.b and t1.id<t2.id and t1.a>2 limit 20\c

```
EXPLAIN: -> Limit: 20 row(s) (cost=176e+6 rows=20) (actual time=455..455 rows=20 loops=1)
  -> Nested loop inner join (cost=176e+6 rows=535e+6) (actual time=455..455 rows=20 loops=1)
    -> Filter: (t1.b is not null) (cost=14796 rows=32879) (actual time=302..302 rows=1 loops=1)
      -> Index range scan on t1 using a over (2 < a), with index condition: (t1.a > 2) (cost=14796 rows=32879) (actual time=302..302 rows=1 loops=1)
        -> Index lookup on t2 using b (b=t1.b), with index condition: (t1.id < t2.id) (cost=483 rows=16265) (actual time=153..153 rows=20 loops=1)
```


show engine innodb status\G

```
-----
TRANSACTIONS
-----
Trx id counter 105833
Purge done for trx's n:o < 105832 undo n:o < 0 state: running but idle
History list length 1
Total number of lock structs in row lock hash table 2
LIST OF TRANSACTIONS FOR EACH SESSION:
---TRANSACTION 421889187314632, not started
0 lock struct(s), heap size 1192, 0 row lock(s)
---TRANSACTION 421889187313624, not started
0 lock struct(s), heap size 1192, 0 row lock(s)
---TRANSACTION 421889187312616, not started
0 lock struct(s), heap size 1192, 0 row lock(s)
---TRANSACTION 421889187310600, not started
0 lock struct(s), heap size 1192, 0 row lock(s)
---TRANSACTION 421889187309592, not started
0 lock struct(s), heap size 1192, 0 row lock(s)
---TRANSACTION 105829, ACTIVE 204 sec
4 lock struct(s), heap size 1192, 2 row lock(s), undo log entries 1
MySQL thread id 15, OS thread handle 140412481377984, query id 156 127.0.0.1 root
```

show engine innodb status\G

```
begin;  
select * from t1 where id=12000  
for share;  
select * from t1 where id=12000  
for update;  
update t1 set c=c+1 where  
id=12000;
```

```
update t1 set c=c+1 where  
id=12000;
```


Q&A

THANKS