

Syntax: EXPLAIN <SELECT STATEMENT>;

RESULTS	
id	Identifier
select_type	See SELECT_TYPE
table	Table name/alias
type	JOIN type (see TYPE)
possible_keys	Possible indexes MySQL could use
key	Index actually used
key_len	Key length, in bytes
ref	Which index fields are used
rows	Approximation of # of rows returned
Extra	See EXTRA

TYPE	
system	Table has 1 value (a system table)
const	Evaluates to 1 row using PRIMARY or UNIQUE KEY
eq_ref	1 row from this tables for each set of rows from previous tables. KEY columns compared with =
ref	JOIN uses key prefix; JOIN uses KEY that isn't PRIMARY or UNIQUE; KEY compared with = or <=>
fulltext	FULLTEXT index used
ref_or_null	as for ref, with an extra pass for NULL values
index_merge	Index Merge optimization used; key and key_len columns are lists
unique_subquery	Subquery on a PRIMARY or UNIQUE KEY of 1 table
index_subquery	Subquery on a non-unique KEY of one table
range	KEY compared with =, <, <=, >, >=, IS NULL, <=>, BETWEEN, or IN([list])
index	Full scan of the INDEX
all	Full scan of the table

SELECT_TYPE	
SIMPLE	One table or JOINS
PRIMARY	First SELECT in a UNION / outer query in a subquery
UNION	Second and later SELECTs in a UNION
DEPENDENT UNION	Second and later SELECTs in a UNION, dependent on outer query
UNION RESULT	Result of a UNION
SUBQUERY	First SELECT in a subquery
DEPENDENT SUBQUERY	First SELECT in a subquery, dependent on outer query; re-evaluated for each different variable set from outer context
DERIVED	Derived table — SELECT subquery in FROM clause
UNCACHEABLE SUBQUERY	Result of subquery cannot be cached, must be re-evaluated for each outer query row

EXTRA		
Distinct	Stops after first row match	
Full scan on NULL key	No index lookup on subquery	
Impossible WHERE noticed after reading const tables	WHERE clause always false	
No tables	No FROM or FROM DUAL	
Not exists	Stops after first row match for each row set from previous tables	
Range checked for each record (index map: N)	No good index; there might be one after values from previous tables are known	
Select tables optimized away	Aggregate functions resolved by index or metadata	
Using (...)	filesort	Extra pass through data for sort
	index	Uses index only, no table read
	index for group-by	GROUP BY or DISTINCT resolved by index or metadata
	intersection	uses index_merge join type
	sort_union	uses index_merge join type
	temporary	Temporary table used
	union	uses index_merge join type
	where	WHERE clause exists
	where with pushed condition	NDB Cluster only; pushes condition to data nodes

