explain

[sql] view plain copy

1. explain select * from user

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	user	ALL	(Null)	(Null)	(Null)	(Null)	1	

[sql] view plain copy

1. explain extended select * from user

id	select_type	table	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	user	ALL	(Null)	(Null)	(Null)	(Null)	1	100	

id	SELECT识别符。这是SELECT的查询序列号
select_type	SELECT类型,可以为以下任何一种: **SIMPLE**:简单SELECT(不使用UNION或子查询) **PRIMARY**:最外面的SELECT **UNION**:UNION中的第二个或后面的SELECT语句 **DEPENDENT UNION**:UNION中的第二个或后面的SELECT语句,取决于外面的查询 **UNION RESULT**:UNION 的结果 **SUBQUERY**:子查询中的第一个SELECT **DEPENDENT SUBQUERY**:子查询中的第一个SELECT,取决于外面的查询 **DERIVED**:导出表的SELECT(FROM子句的子查询)
table	输出的行所引用的表
type	联接类型。下面给出各种联接类型,按照从最佳类型到最坏类型进行排序: **system**:表仅有一行(=系统表)。这是const联接类型的一个特例。 **const**:表最多有一个匹配行,它将在查询开始时被读取。因为仅有一行,在这行的列值可被优化器剩余部分认为是常数。const表很快,因为它们只读取一次! **eq_ref**:对于每个来自于前面的表的行组合,从该表中读取一行。这可能是最好的联接类型,除了const类型。 **ref**:对于每个来自于前面的表的行组合,所有有匹配索引值的行将从这张表中读取。 **ref_or_null**:该联接类型如同ref,但是添加了MySQL可以专门搜索包含NULL值的行。 **index_merge**:该联接类型表示使用了索引合并优化方法。 **unique_subquery**:该类型替换了下面形式的IN子查询的ref: value IN (SELECT primary_key FROM single_table WHERE some_expr) unique_subquery是一个索引查找函数,可以完全替换子查询,效率更高。 **index_subquery**:该联接类型类似于unique_subquery。可以替换IN子查询,但只适合下列形式的子查询中的非唯一索引: value IN (SELECT key_column FROM single_table WHERE some_expr) **range**:只检索给定范围的行,使用一个索引来选择行。 **index**:该联接类型与ALL相同,除了只有索引树被扫描。这通常比ALL快,因为索引文件通常比数据文件小。 **ALL**:对于每个来自于先前的表的行组合,进行完整的表扫描。
possible_keys	指出MySQL能使用哪个索引在该表中找到行
key	显示MySQL实际决定使用的键(索引)。如果没有选择索引,键是NULL。
key_len	显示MySQL决定使用的键长度。如果键是NULL,则长度为NULL。
ref	显示使用哪个列或常数与key一起从表中选择行。
rows	显示MySQL认为它执行查询时必须检查的行数。多行之间的数据相乘可以估算要处理的行数。
filtered	显示了通过条件过滤出的行数的百分比估计值。
Extra	该列包含MySQL解决查询的详细信息 **Distinct**:MySQL发现第1个匹配行后,停止为当前的行组合搜索更多的行。 **Not exists**:MySQL能够对查询进行LEFT JOIN优化,发现1个匹配LEFT JOIN标准的行后,不再为前面的的行组合在该表内检查更多的行。 **range checked for each record (index map: #)**:MySQL没有发现好的可以使用的索引,但发现如果来自前面的表的列值已知,可能部分索引可以使用。 **Using filesort**:MySQL需要额外的一次传递,以找出如何按排序顺序检索行。 **Using index**:从只使用索引树中的信息而不需要进一步搜索读取实际的行来检索表中的列信息。 **Using temporary**:为了解决查询,MySQL需要创建一个临时表来容纳结果。 **Using where**:WHERE 子句用于限制哪一个行匹配下一个表或发送到客户。 **Using sort_union(...), Using union(...), Using intersect(...)**:这些函数说明如何为index_merge联接类型合并索引扫描。 **Using index for group-by**:类似于访问表的Using index方式,Using index for group-by表示MySQL发现了一个索引,可以用来查 询GROUP BY或DISTINCT查询的所有列,而不要额外搜索硬盘访问实际的表。