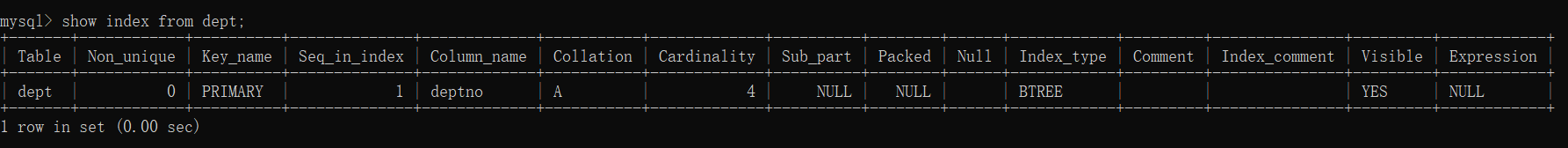
1. explain + sql语句

查看SQL语句的具体执行过程

1. type-查询数据的方式

常见的type：all、index、range、ref、eq\_ref、const

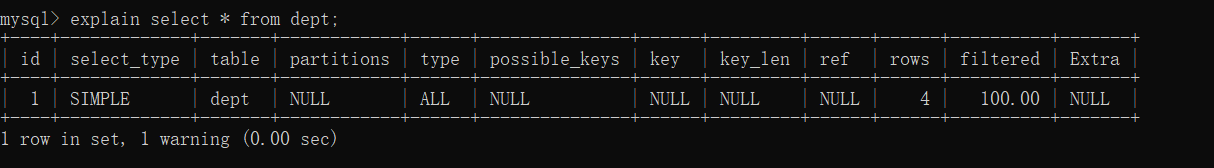


1. all

全表扫描

将遍历全表以找到匹配的行，性能最差

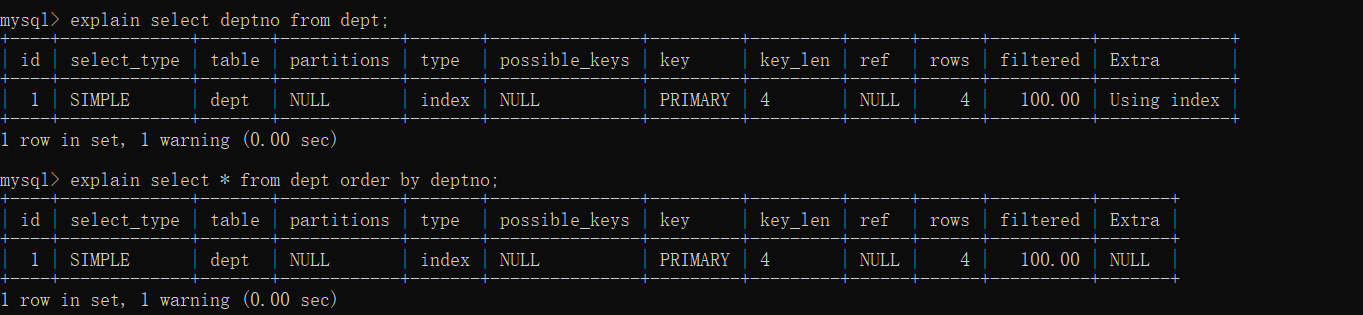
* select \* from dept;



1. index

遍历索引树读取

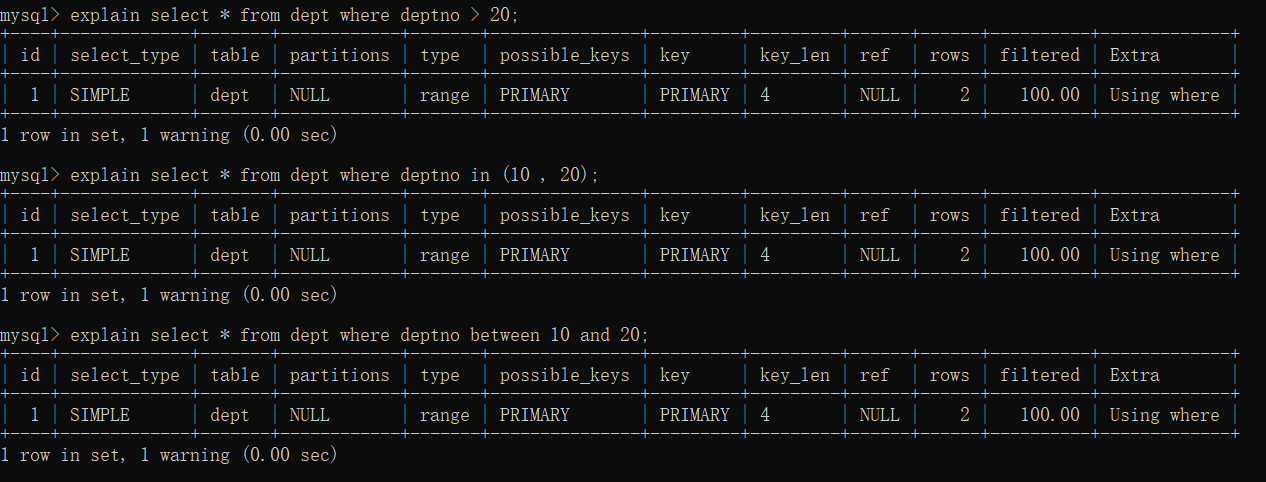
* select deptno from dept;
* select \* from dept order by depton;



1. range

对有索引的字段，按照范围检索，bettween...and、<、>、<=、in 等

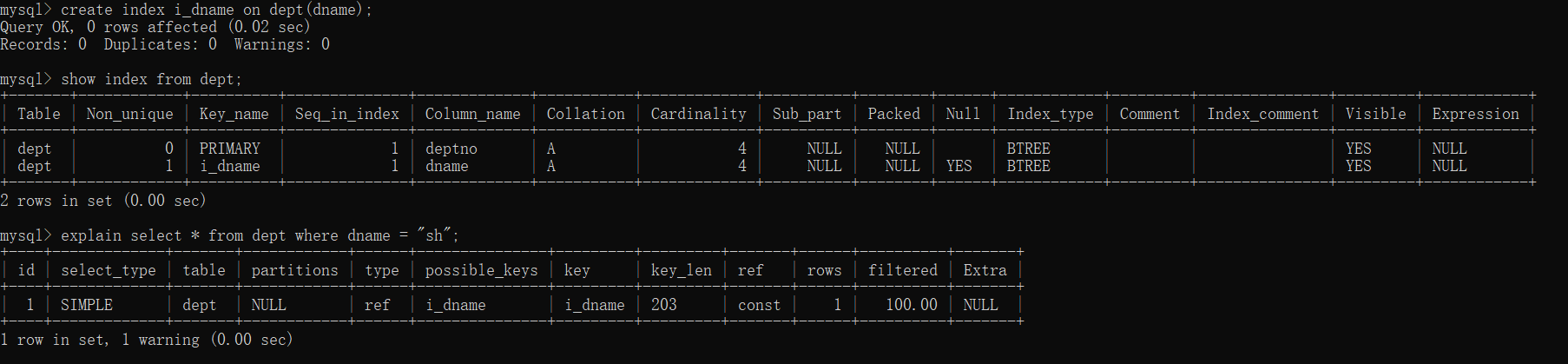
* select \* from dept where deptno > 10;



1. ref（多条）

非唯一性、非主键索引扫描，返回匹配某个单独值的所有行；

虽然使用了索引，但该索引列的值并不唯一，有重复。这样即使使用索引快速查找到了第一条数据，仍然不能停止，要进行目标值附近的小范围扫描。但它的好处是它并不需要扫全表，因为索引是有序的，即便有重复值，也是在一个非常小的范围内扫描。



1. eq\_ref（单条）

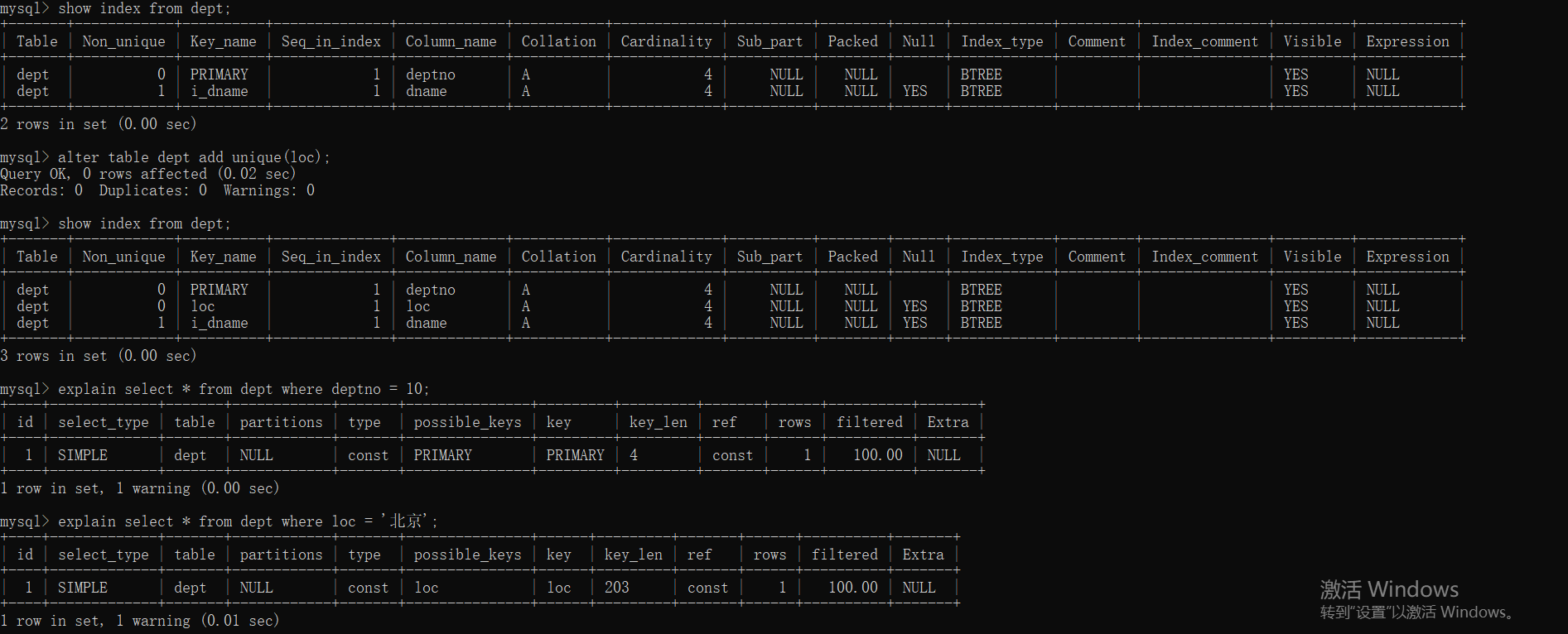
唯一性索引扫描，对于每个索引键，表中只有一条记录与之匹配，常用语主键或唯一索引扫描；

使用了主键或者唯一性索引进行查找的情况，比如根据学号查找某一学校的一名同学，在没有查找前我们就知道结果一定只有一个，所以当我们首次查找到这个学号，便立即停止了查询。

只能通过两张表，构造此条件。

1. const（单条）

当命中的是主键或者唯一索引且为常量



1. system

当表仅有一行记录时(系统表)，数据量很少，往往不需要进行磁盘IO，速度非常快

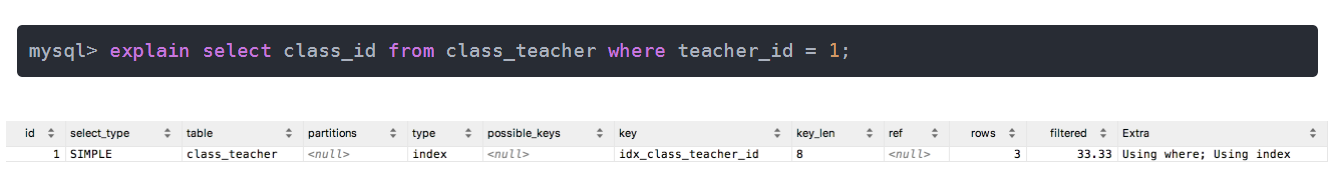
1. rows 列（重要）

MySQL 估计 sql 要查询出来的数据条数

1. Extra
2. use index：使用覆盖索引
3. use where：没有使用索引，where过滤非索引列
4. use where use index：

表示查询的列被索引列覆盖，并且 where 筛选条件是索引列

之一但不是索引的前导列，说明无法直接通过索引查找查询到符合条件的数据。



1. use temporary：使用临时表
2. use filesort：无法使用索引进行排序，只能利用外部排序
3. Using index condition：查询的列不完全被索引列覆盖

