

用智能优化限制提高Oracle数据库性能 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/142/2021_2022__E7_94_A8_E6_99_BA_E8_83_BD_E4_c102_142874.htm Oracle SQL运行时间的最主要的组成部分是花在为执行准备新的SQL语句上的时间。不过，如果了解了可执行计划产生的内在机制，你就可以控制Oracle花费在评估表的连接顺序的时间，并在总体上提高查询的性能。准备为执行提供的SQL语句在一个SQL语句进入Oracle库的cache之后、而真正被执行之前，将会依次发生如下事件：语法检查检查该SQL语句的拼写和词序是否正确。语义解析Oracle根据数据词典（data dictionary）来验证所有的表格（table）和列（column）。已保存纲要检查Oracle检查词典以确认对应该SQL语句是否已存在已保存的纲要（Stored Outline）。产生执行计划Oracle根据一种罚值（cost-based）优化算法和数据词典中的统计数据来决定如何生成最优执行计划。产生二进制代码Oracle在执行计划的基础上生成可执行的二进制代码。一旦开始准备执行SQL语句，上述的过程很快就会执行，这是因为Oracle可以识别出同样的SQL语句并对同样的SQL语句重复使用对应的可执行代码。然而，对产生ad hoc SQL的系统以及SQL中嵌入文本值（literal value）的情况，SQL执行计划的生成时间就会变得相当长，而且以前的执行计划也常常不能被再次利用。对那些牵涉到许多表格的查询，Oracle可能要花上很长的时间来决定把连接这些表格的顺序。评估连接表格的顺序生成可执行计划的时间往往是SQL的准备过程中最大的开销组成部分，尤其是在处理有多个表的连接的查询的情况下。当Oracle评估表的连接

顺序时，它必须考虑每一种可能的排序。例如，当有六个表格需要连接时，Oracle需要考虑720种（6的排列数，即 $6 \times 5 \times 4 \times 3 \times 2 \times 1 = 720$ ）可能的连接排序。当需要连接的表的数量超过10时，这个排列问题将变得非常突出：如果需要连接的表格有15个，那么需要考虑的可能的查询排列顺序超过一万亿种（精确值为1,307,674,368,000）。

在optimizer_search_limit参数中设置限制 你可以通过optimizer_search_limit参数来控制上述问题的发生，该参数用来指定优化器评估的表格连接顺序的最大数目。利用这个参数，就可以防止优化器在评估所有可能的表格连接顺序中所花费的多余时间。如果查询中的表的数量少于或者等于optimizer_search_limit，那么优化器检查所有的可能表的连接方式。例如，涉及了五个表的查询一共有120种($5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$)可能的连接顺序，所以如果参数optimizer_search_limit的值设置为5（默认值），那么优化器就会考虑所有的这120种可能的连接顺序

。optimizer_search_limit参数还用来控制启动开始连接指示（star join hint）的阈值。当查询所涉及的表格数量少于参数optimizer_search_limit的设定值，开始连接指示将被设置。

另一个工具：optimizer_max_permutations参数

optimizer_max_permutations初始参数用来设定优化器优化范围的上界（即最多考虑多少种表格连接顺序），它依赖于初始参数optimizer_search_limit。参数optimizer_max_permutations的默认值为8000。参数optimizer_search_limit

和optimizer_max_permutations一同用来设置优化器所考虑的排列数的上限。优化器不断的产生可能的表的连接的排列，直

到排列数达到参数optimizer_search_limit或者optimizer_max_permutations为止。一旦优化器停止产生新的可能连接排列，它将会从中选择出耗费最小的排列。用已排序指示来指定一种连接排序 你可以设定优化器评估的排列数的上限。但是对复杂的情况下，即使允许的排列数很大，优化器也很可能在远远没有找到一个比较合适的排列之间就已经停止优化了。你不妨回头看看我前面举的那个例子（15个需要连接的表有超过一万亿种排列）。如果设定优化器考虑80,000种排列，那么这仅仅考虑了所有可能性的0.000006%，优化器极可能没有达到最佳的排列。在Oracle SQL中解决这个问题的最好方法就是手工指定一种表格连接顺序。这里需要遵循的大原则就是表格连接顺序应该使得查询计划尽快得以建立，通常在SQL语句中使用WHERE限制子句。下面以一个对名为emp的表格的并行查询为例，例子中的代码强制查询计划执行一个嵌套循环连接（nested loop join）。注意，我使用了已排序指示来引导优化器来评估WHERE子句中给出的表格的连接顺序。

```
0select /* ordered use_nl(bonus)
parallel(e, 4) */ e.ename, hiredate, b.comm. from emp e, bonus b
where e.ename = b.ename .
```

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com