

利用Oracle执行计划机制提高查询性能 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/207/2021_2022__E5_88_A9_E7_94_A8Orac_c102_207506.htm 消耗在准备利用Oracle执行计划机制提高查询性能新的SQL语句的时间是Oracle SQL语句执行时间的重要组成部分。但是通过理解Oracle内部产生执行计划的机制，你能够控制Oracle花费在评估连接顺序的时间数量，并且能在大体上提高查询性能。准备执行SQL语句当SQL语句进入Oracle的库缓存后，在该语句准备执行之前，将执行下列步骤：1) 语法检查：检查SQL语句拼写是否正确和词序。2) 语义分析：核实所有的与数据字典不一致的表和列的名字。3) 轮廓存储检查：检查数据字典，以确定该SQL语句的轮廓是否已经存在。4) 生成执行计划：使用基于成本的优化规则和数据字典中的统计表来决定最佳执行计划。5) 建立二进制代码：基于执行计划，Oracle生成二进制执行代码。一旦为执行准备好了SQL语句，以后的执行将很快发生，因为Oracle认可同一个SQL语句，并且重用那些语句的执行。然而，对于生成特殊的SQL语句，或嵌入了文字变量的SQL语句的系统，SQL执行计划的生成时间就很重要了，并且前一个执行计划通常不能够被重用。对那些连接了很多表的查询，Oracle需要花费大量的时间来检测连接这些表的适当顺序。评估表的连接顺序在SQL语句的准备过程中，花费最多的步骤是生成执行计划，特别是处理有多个表连接的查询。当Oracle评估表的连接顺序时，它必须考虑到表之间所有可能的连接。例如：六个表的之间连接有720（6的阶乘，或 $6 * 5 * 4 * 3 * 2 * 1 = 720$ ）种可能的连接线路。当一个查询中含有超

过10个表的连接时，排列的问题将变得更为显著。对于15个表之间的连接，需要评估的可能查询排列将超过1万亿（准确的数字是1,307,674,368,000）种。使用optimizer_search_limit参数来设定限制 通过使用optimizer_search_limit参数，你能够指定被优化器用来评估的最大的连接组合数量。使用这个参数，我们将能够防止优化器消耗不定数量的时间来评估所有可能的连接组合。如果在查询中表的数目小于optimizer_search_limit的值，优化器将检查所有可能的连接组合。例如：有五个表连接的查询将有120（ $5! = 5 * 4 * 3 * 2 * 1 = 120$ ）种可能的连接组合，因此如果optimizer_search_limit等于5（默认值），则优化器将评估所有的120种可能。optimizer_search_limit参数也控制着调用带星号的连接提示的阈值。当查询中的表的数目比optimizer_search_limit小时，带星号的提示将被优先考虑。另一个工具：参数optimizer_max_permutations 初始化参数optimizer_max_permutations定义了优化器所考虑组合数目的上限，且依赖于初始参数optimizer_search_limit。optimizer_max_permutations的默认值是80,000。参数optimizer_search_limit和optimizer_max_permutations一起来确定优化器所考虑的组合数目的上限：除非（表或组合数目）超过参数optimizer_search_limit 或者optimizer_max_permutations设定的值，否则优化器将生成所有可能的连接组合。一旦优化器停止评估表的连接组合，它将选择成本最低的组合。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