

在Oracle中使用的hints调整机制 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/471/2021\\_2022\\_\\_E5\\_9C\\_A8](https://www.100test.com/kao_ti2020/471/2021_2022__E5_9C_A8)

Oracle\_E4\_c67\_471768.htm Oracle使用的hints调整机制一直很复杂，Oracle Technical Network对使用hints调整Oracle SQL的过程有很好的全面评述。根据对10g数据库的介绍，可使用更多新的optimizer hints来控制优化行为。现在让我们迅速了解一下这些强大的新hints: spread\_min\_analysis 使用这一hint，你可以忽略一些关于如详细的关系依赖图分析等电子表格的编译时间优化规则。其他的一些优化，如创建过滤以有选择性的定位电子表格访问结构并限制修订规则等，得到了继续使用。由于在规则数非常大的情况下，电子表格分析会很长。这一提示可以帮助我们减少由此产生的数以百小时计的编译时间。 例如: `SELECT /* SPREAD_MIN_ANALYSIS */ ...`

spread\_no\_analysis 通过这一hint，可以使无电子表格分析成为可能。同样，使用这一hint可以忽略修订规则和过滤产生。如果存在一电子表格分析，编译时间可以被减少到最低程度。 例如: `SELECT /* SPREAD_NO_ANALYSIS */ ...`

use\_nl\_with\_index 这项hint使CBO通过嵌套循环把特定的表格加入到另一原始行。只有在以下情况中，它才使用特定表格作为内部表格:如果没有指定标签，CBO必须可以使用一些标签，且这些标签至少有一个作为索引键值加入判断.反之

，CBO必须能够使用至少有一个作为索引键值加入判断的标签。 例如: `SELECT /* USE_NL_WITH_INDEX (polrecpolrind) */ ... CARDINALITY` 此hint定义了对由查询或查询部分返回的基数的评价。注意如果没有定义表格，基数是由整个查询所

返回的总行数。 例如: `SELECT /* CARDINALITY ( [tablespec] card ) */ SELECTIVITY` 此hint定义了对查询或查询部分选择性的评价。如果只定义了一个表格，选择性是在所定义表格里满足所有单一表格判断的行部分。如果定义了一系列表格，选择性是指在合并以任何顺序满足所有可用判断的全部表格后，所得结果中的行部分。 例如: `SELECT /* SELECTIVITY ( [tablespec] sel ) */` 然而，注意如果hints `CARDINALITY` 和 `SELECTIVITY`都定义在同样的一批表格，二者都会被忽略。

`no_use_nl` Hint `no_use_nl`使CBO执行循环嵌套，通过把指定表格作为内部表格，把每个指定表格连接到另一原始行。通过这一hint，只有hash join和sort-merge joins会为指定表格所考虑。 例如: `SELECT /* NO_USE_NL ( employees ) */ ...`

`no_use_merge` 此hint使CBO通过把指定表格作为内部表格的方式，拒绝sort-merge把每个指定表格加入到另一原始行。 例如: `SELECT /* NO_USE_MERGE ( employees dept ) */ ...`

`no_use_hash` 此hint使CBO通过把指定表格作为内部表格的方式，拒绝hash joins把每个指定表格加入到另一原始行。 例如: `SELECT /* NO_USE_HASH ( employees dept ) */ ...`

`no_index_ffs` 此hint使CBO拒绝对指定表格的指定标签进行fast full-index scan。 Syntax: `/* NO_INDEX_FFS ( tablespecindexspec ) */`

`no_index_ss` 此hint使CBO拒绝对指定表格的指定标签进行skip scan。 Syntax: `/* NO_INDEX_SS ( tablespecindexspec ) */`

`no_star_transformation` 此hint使CBO忽略star 询问信息。 Syntax: `/* NO_STAR_TRANSFORMATION */`

`index_ss` 此hint明确地为指定表格选择index skip scan。如果语句使用index range scan，Oracle将以对其索引值的升序排列来检查索引入口。在被分

割的索引中，其结果为对每个部分内部的升序排列。 Syntax: `/* INDEX_SS (tablespecindexspec) */ index_ss_asc` 此hint明确地为指定表格选择index skip scan。如果语句使用index range scan，Oracle将以对其索引值的升序排列来检查索引入口。在被分割的索引中，其结果为对每个部分内部的升序排列。 Syntax: `/* INDEX_SS_ASC (tablespecindexspec) */ index_ss_desc` 此hint明确为指定表格选择index skip scan。如果语句使用index range scan，Oracle将以对其索引值的降序排列来检查索引入口。在被分割的索引中，其结果为对每个部分内部的降序排列。 Syntax: `/* INDEX_SS_DESC (tablespecindexspec) */ cpu_costing` 此hint为SQL语句打开CPU costing。这是对优化器的默认评估模式。优化器评估当执行给定查询时，数据库需要运行的IO操作数、IO操作种类、以及CPU周期数。 Syntax: `/* CPU_COSTING (tablespecindexspec) */ no_cpu_costing` 此hint为SQL语句关闭CPU costing。然后CBO使用IO cost模式，此模式忽略CPU花费，仅测量在single-block reads中的所有指标。 Syntax: `/* NO_CPU_COSTING */` 随着Oracle优化器越来越成熟，Oracle专家必须不断增加自己对调整SQL语句的工具储备。当然，讨论所有复杂的Oracle10g SQL新hints远远超出了本文的范围，你可以从Mike Ault的新书Oracle Database 10g New Features中获得关于Oracle10g的更多信息 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](http://www.100test.com)