

1 优化器概述

优化器是Oracle内置的软件，决定了执行SQL语句的最有效的方法。

1.1 优化器操作

数据库可以以多种方式执行SQL语句，如全表扫描、索引扫描、嵌套循环、哈希连接等。在决定执行计划时，优化器会考虑对象和查询条件的很多因素，该决定是SQL处理中的一个很重要的步骤，可以极大地影响SQL的执行时间。

当用户提交一个SQL语句执行时，优化器执行下面的步骤：

- 基于可用的访问路径和提示，优化器为SQL语句生成一组潜在的执行计划；
- 根据数据字典中的统计信息，优化器评估每一个执行计划的成本。统计信息包括表、索引和分区的数据分布和存储特性；
- 优化器比较这些执行计划，并选择成本最低的执行计划；

对于Oracle数据库处理的任何SQL语句，优化器将执行如下操作：

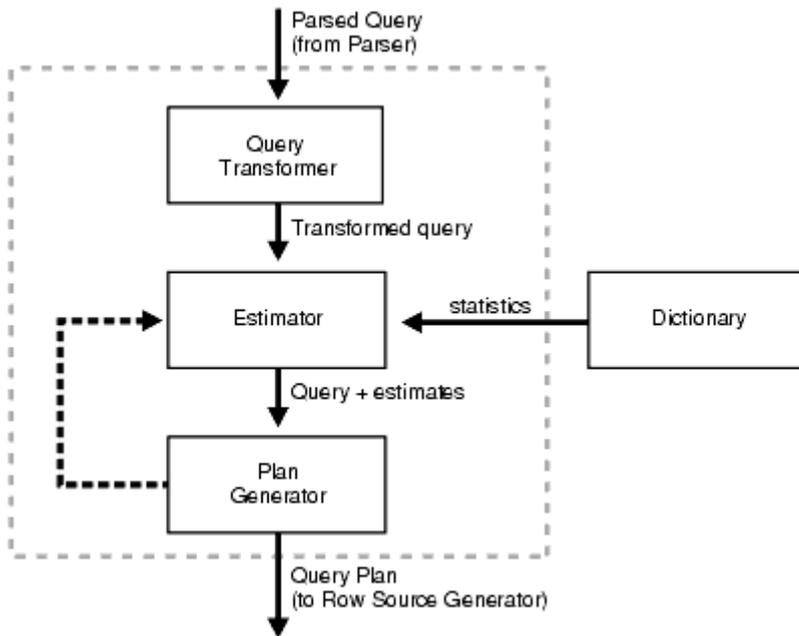
- 表达式和条件的评估；
- 语句转换；
- 优化目标的选择；
- 访问路径的选择；
- 连接顺序的选择；

1.2 查询优化器的组件

查询优化器操作包括：

- 查询转换器：视图合并、谓词推进、子查询展开、物化视图的查询重写；
- 评估器：选择性、基数、成本；
- 计划生成器；

优化器组件如图：



1.3 绑定变量演示

见实验演示

2 优化器访问路径

访问路径是从数据库中检索数据的方式。通常情况下，检索少量数据时索引访问是有效的，检索大数据量时全表扫描更有效。

访问路径有：

- 全表扫描；
- Rowid扫描；
- 索引扫描；
- 哈希访问；
- 抽样表扫描；

3 连接方式

3.1 表的连接方式

- 嵌套循环连接Nested Loop Joins

- 哈希连接Hash Joins
- 排序合并连接Sort Merge Joins
- 笛卡尔连接Cartesian Joins
- 外连接Outer Joins

3.2 连接总结

1) 嵌套循环(nest loop)

对于被连接的数据子集较小的情况，嵌套循环连接是较好的选择。在嵌套循环中，外表驱动内表，外表返回的每一行都要在内表中检索找到它匹配的行，因此整个查询返回的结果集不能太大（大于10000不合适），要把返回子集较小的表作为外表（驱动表），而且在内表的连接字段上一定要有索引。

2) 哈希连接(hash join)

哈希连接是大数据集连接时常用的方式，优化器使用两个表中较小的表，利用连接键在内存中建立散列表，然后扫描较大的表并探测散列表，找出与散列表匹配的行。

这种方式适用于较小的表完全可以放入内存的情况，这样成本就是访问两个表的成本之和。但是在表很大的情况下并不能完全放入内存，这时优化器将它分割成若干不同的分区，不能放入内存的部分就把该分区写入磁盘的临时段。

哈希连接只能应用于等值连接(如WHERE A.COL3 = B.COL4)、非等值连接(WHERE A.COL3 > B.COL4)、外连接(WHERE A.COL3 = B.COL4(+))。

3) 排序合并连接 (Sort Merge Join)

通常情况下哈希连接的效果都比排序合并连接要好。然而如果行源已经被排过序，在执行排序合并连接时不需要再排序了，这时排序归并连接的性能会优于哈希连接。

4 优化器模式

4.1 ALL_ROWS

指示优化器为SQL语句选择以最大吞吐为目标的执行计划；

4.2 FIRST_ROWS_n

指示优化器为SQL语句选择以最快反应时间为目标的执行计划，Oracle较高版本中，需为该Hint指定一个参数，first_rows (n) 表示优化器将为该Hint相关的SQL语句选择一个尽快返回前n行结果数据的执行计划；

4.3 FIRST_ROWS

同上