# 认识SQL优化、基数和选择性

版本：Oracle 11G R2

环境：SQLPlus

用户：scott

**1.SQL优化，首先需要知道SQL的性能瓶颈在哪里**

我们先出个题目：

优化一个系统选择： (1)应用架构优化 (2)数据库db优化

应用架构优化运维年限太长，优化难度太大，所以我们只能从数据库db优化

**2.数据库的优化**

(1) 参数、配置、硬件优化等等占20%  
(2) SQL优化占80%

**3.做SQL优化如果知道思路，10分钟一般可以搞定，不知道思路就搞不定了。**

**4.学习SQL优化是学习思路，不是学会看执行计划。**

主要是SQL优化的核心思想，以及SQL优化的具体操作步骤和方法，重点并不是讲理论。但是这个也有一些核心的基础理论，少了这些基础理论，那后面的学习也就没意义了，因为没基础可能听不懂。前期会有一些基础理论，后期会大量讲解SQL优化的具体步骤，优化方法，以及案例分析。

**5.基数和选择性**

基数(Cardinality) 列唯一键(Distinct\_keys)的数量，比如性别，该列只有男女之分，所以这一列基数是2。主键列的基数等于行数。

选择性(Selectivity) 列唯一键(Distinct\_Keys)与行数(Num\_Rows)的比值。

举例5.1：

有一个表test有1000W行，id是主键列。

主键列的基数 ：1000W；

非主键列的基数：select count(distinct 非主键列名) from test；

举例5.2：

有一个表test有1000W行，id是主键列，gender是性别列，男女各占一半

基数很多：select \* from test where id = '10' 返回1条，是不是应该走索引？

基数很少：select \* from test where gender = 'M'返回500W条，是不是应该走全表扫描?

**6.基数的多少跟走不走索引有没有关系？**

一般情况下：

(1)基数越多，返回数据就越少

(2)基数越多，就越适合建立索引

(3)基数越少，就不适合建立索引（注：没有绝对可言）

举例6.1：

有一个表test有1000W行，gender是性别列，男人有1W，女人有999W。

select \* from test where gender = ‘M’ 走索引or全表扫描？

select \* from test where gender = ‘F’ 走全表扫描

**7.查询真实的基数和选择性**

我们在进行SQL调优的时候，可以用下面SQL去查看列真实的基数和选择性

select count(distinct column\_name),

count(\*) total\_rows,

count(distinct column\_name) / count(\*) \* 100 selectivity

from table\_name;

**8.查询基数的脚本依赖于什么？**

统计信息

# 直方图

**1、基于规则的优化方式(Rule-Based Optimization，简称为RBO)**

优化器在分析SQL语句时,所遵循的是Oracle内部预定的一些规则，对数据是不敏感的。它只借助少量的信息来决定一个sql语句的执行计划，包括：

    1) sql语句本身  
    2) sql中涉及到的table、view、index等的基本信息

    例如：我们常见的，当一个where子句中的一列有索引时去走索引。但是需要注意，走索引不一定就是优的，比如一个表只有两行数据，一次IO就可以完成全表的检索,而此时走索引时则需要两次IO,这时全表扫描(table\_access\_full)的效率更优。

 在得到目标SQL的执行计划时，RBO所用的判断原则为一组内置的规则，这些规则是硬编码在Oracle数据库的代码中，RBO根据这些规则从目标SQL可能的执行路径中选择一条作为最终的执行计划；

具体是这样的：Oracle 会在代码里事先给各种类型的执行路径定一个等级，一共有15个等级，优先级从等级1到等级15（等级1值最低，等级最高）。Oracle会认为等级值低的执行效率比等级值高的效率高，也就是说，等级越高效率越高。可是在决定目标SQL的执行计划时可能不止一条执行计划，那么RBO就会从SQL诸多可能的执行计划中选择一条等级值最低的执行路径作为最终的执行计划。

提问：RBO的等级1和等级15对应的执行路径分别是什么呢？

Oracle数据库里，对于OLTP系统的SQL来说，显然是通过rowid访问是最高效的，而通过全表扫描来访问是最低效的。

RBO Path 1: Single Row by Rowid (等级最高)

　　RBO Path 2: Single Row by Cluster Join (集群连接)

　　RBO Path 3: Single Row by Hash Cluster Key with Unique or Primary Key

　　RBO Path 4: Single Row by Unique or Primary Key

　　RBO Path 5: Clustered Join

　　RBO Path 6: Hash Cluster Key

　　RBO Path 7: Indexed Cluster Key

　　RBO Path 8: Composite Index

　　RBO Path 9: Single-Column Indexes

　　RBO Path 10: Bounded Range Search on Indexed Columns

　　RBO Path 11: Unbounded Range Search on Indexed Columns

　　RBO Path 12: Sort Merge Join

　　RBO Path 13: MAX or MIN of Indexed Column

　　RBO Path 14: ORDER BY on Indexed Column

　　RBO Path 15: Full Table Scan (等级最低)

从上图可以看出来，RBO内置的等级解释：

等级1：使用rowid定位单行。当where子句中直接嵌入rowid时，RBO走此路径。Oracle不推荐直接引用rowid，因为rowid可能由于版本的改变而变化，行迁移、行链接、EXP/IMP也会使rowid发生变化。

等级2：用Cluster Join(集群连接)定位单行，即两个表做等值连接，一方的连接字段是Cluster Key，且where中存在可以保证该语句仅返回一行记录的条件时，RBO走此路径；

等级3：用带有唯一约束或主键的Hash Cluster Key 定位单行；

等级4：用唯一约束的字段或主键的字段来定位单行

等级5：Cluster Join(集群连接)

等级6：使用Hash Cluster Key

等级7：使用索引 Cluster Key

等级8：使用复合索引

等级9：使用单字段索引

等级10：用索引进行有界限范围的查找

等级11：用索引字段进行无界限的查找

等级12：排序合并连接

等级13：对索引字段使用Max或Min函数

等级14：order by 索引字段

等级15：全表扫描

RBO在Oracle中由来已久，从Oracle 10g 开始，Oracle已经不再支持RBO，但是RBO相关规则代码并没有从Oracle数据库移除，但是我们依然可以通过修改优化器模式或者使用 Rule Hint 来继续使用RBO。

**2、基于代价的优化方式(Cost-Based Optimization，简称为CBO)**

    它是看语句的代价(Cost)，通过代价引擎来估计每个执行计划所需的代价，该代价将每个执行计划所耗费的资源进行量化，CBO根据这个代价选择出最优的执行计划。

    优化器在判断是否用这种方式时,主要参照的是表及索引的统计信息。统计信息给出表的大小、有多少行、每行的长度等信息。这些统计信息起初在库内是没有的，是做收集统计信息后才出现的，很多的时侯过期统计信息会令优化器做出一个错误的执行计划,因些应及时更新这些信息。

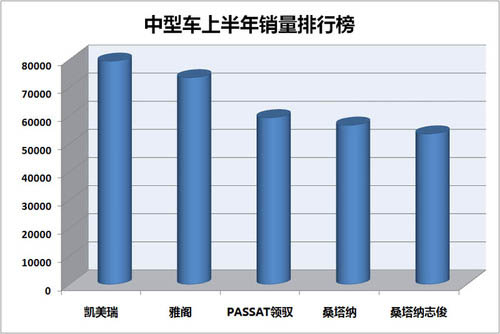
**3.什么是直方图？**

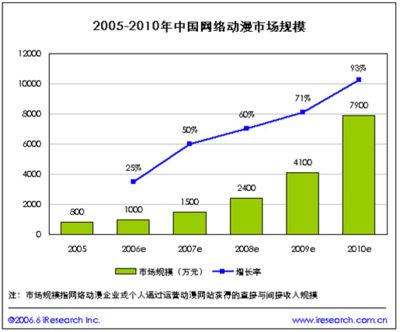
(1) 在Oracle中，CBO会默认认为目标列的数据在其最小值和最大值之间是均匀的，并且会按照这个均匀分布的原则来计算对目标列施加查询条件后的可选择率以及结果集的基数，进而据此来计算成本值并选择执行计划。

(2) 但是如果目标列的数据是均匀分布这个原则并不总是正确的。在实际系统中，我们很容易就可以看见一些目标列的数据分布是不均匀的。此时还按照均匀分布原则去选择率和基数，那么CBO选择执行计划可能是不合理的，甚至是错误的。【比如：有一张test表1000W，有gender性别列，男人1W，女人999W】



4. 其实直方图形象比如类似于柱状图，如图所示：





直方图 当某列数据分布不均衡，为了让CBO能生成最佳的执行计划，我们可能需要对表收集直方图，直方图最大的桶数(Bucket)是254。收集直方图是一个很耗时的过程，如无必要，千万别去收集直方图。

5. 直方图的类型：频率直方图 和 等高直方图

一种是频率直方图(FREQUENCY HISTOGRAM)，当列中Distinct\_keys 较少(小于254)，

如果不手工指定直方图桶数(BUCKET)，Oracle就会自动的创建频率直方图，并且桶数

(BUCKET)等于Distinct\_Keys。

另外一种是高度平衡直方图(HEIGHT BALANCED)，当列中Distinct\_keys大于254，如果

不手工指定直方图桶数(BUCKET)，Oracle就会自动的创建高度平衡直方图。

6. CBO、 统计信息 以及 直方图之间是什么关系？

CBO是依赖统计信息，而直方图就是统计信息的侦察兵。

统计信息

1. **什么是Oracle里的统计信息？**

Oracle数据库里的统计信息是这样的一组数据：它存储在数据字典里，且从多个维度描述了Oracle数据库里对象的详细信息。而CBO会利用这些统计信息来计算目标SQL各种可能的、不同的执行路径的成本，并从中选择一条成本值最小的执行路径作为目标SQL的执行计划。

1. **Oracle数据库里的统计信息类型**

* 表的统计信息
* 索引的统计信息
* 列的统计信息
* 系统统计信息
* 数据字典统计信息
* 内部对象统计信息

表的统计信息用于描述Oracle数据库里表的详细信息，它包含了一些典型的维度，如记录数、表块（表的数据块）的数量、平均行长度等。

索引的统计信息用于描述Oracle数据库里索引的详细信息，它包含了一些典型的维度，如索引的层级、叶子块的数量、聚簇因子（集群因子）等。

列的统计信息用于描述Oracle数据库里列的详细信息，它包含了一些典型的维度，如列distinct值的数量、列的null值的数量、列的最小值、列的最大值以及直方图等。

系统统计信息用于描述Oracle数据库所在数据库服务器的系统处理能力，它包含了CPU和I/O这两个维度，借助于系统统计信息，Oracle可以更清楚地知道目标数据库服务器的实际处理能力。

数据字典统计信息用于描述Oracle数据库里数据字典基表（如TAB$、IND$等）、数据字典基表上的索引，以及这些数据字典基表的列的详细信息，其实上述数据字典基表的统计信息和描述普通表、索引、列的统计信息没有本质的区别。

内部对象统计信息用于描述Oracle数据里的一些内部表（如X$系列表）的详细信息，它的维度和普通表的统计信息的维度类似，只不过其表块的数量为0，因为X$系列表实际上只是Oracle自定义的内存结构，并不占用实际的物理存储空间。

1. **Oracle收集统计信息的方法**

在Oracle数据库里，有两种方法可以用来收集统计信息：

1. 使用ANALYZE命令
2. 使用DBMS\_STATS包

其中，表、索引、列和数据字典的统计信息使用ANALYZE命令或者DBMS\_STATS包均可收集，但是系统和内部对象统计信息只能使用DBMS\_STATS包收集。

1. **DBMS\_STATS包收集**



1. **OLAP和OLTP系统，哪种系统必须收集统计信息？**

OLAP 视情况而定收集统计信息

a.使用ETL抽数据，基本上不建议搜集，因为数据量经常变，业务不复杂，会使用hint固定

b.报表使用到的表必须收集统计信息

OLTP 必须收集统计信息

a.使用where条件查询返回数据量少，必须收集统计信息

# 如何查看执行计划

1. **查看SQL执行计划的方式**
2. explain plan 命令 或者 PLSQL中的F5
3. DBMS\_XPLAN包
4. SQLPlus中的autotrace开关
5. 10046事件
6. 10053事件
7. AWR报告
8. **explain plan 命令**

(1)如何在Oracle数据库中得到一个SQL的执行计划？

直接在PL/SQL Developer中按快捷键F5就可以了。

(2)为何在PL/SQL Developer中得到SQL的执行计划如此方便容易，其原理是什么？PL/SQL Developer在背后到底做了哪些事情？

在你按下快捷键F5后，PL/SQL Developer就调用了explain plan命令，就是Oracle中得到SQL执行计划的第一种方法，PL/SQL Developer中的快捷键F5相当于封装了explain plan命令。

explain plan 命令的语法是执行如下两条命令：

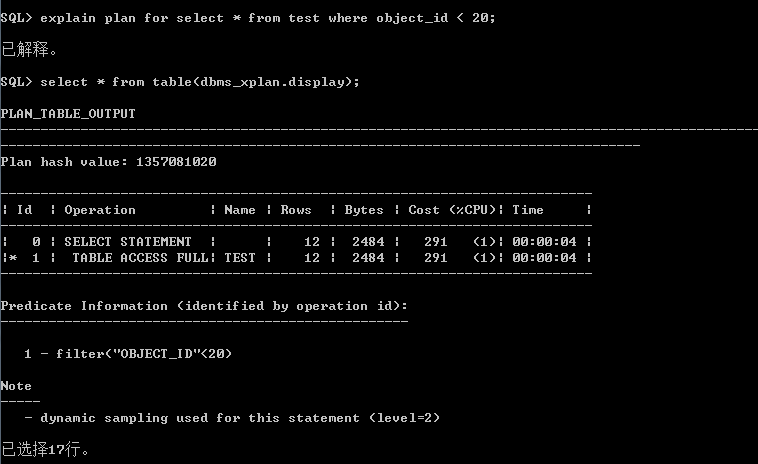
explain plan for + 目标SQL语句

select \* from table(dbms\_xplan.display)

explain plan方式：

explain plan for select \* from test where object\_id < 20;

select \* from table(dbms\_xplan.display);

****

PL/SQL Developer 的F5方式：



Explain plan命令和 快捷键F5的区别是什么？

1. Explain plan命令得到的执行计划有访问路径和谓词信息
2. 快捷键F5得到的执行计划只有访问路径
3. Explain plan命令得到的执行计划有Id列，而且有\*号

访问路径和谓词信息分别指的啥？

1. 访问路径：访问一张表的路径方式，比如：TABLE ACCESS FULL
2. 谓词信息：说直白了就是where条件列的信息

从执行计划观察一张表是否有过滤条件？

直接看执行计划Id列是否有\*号

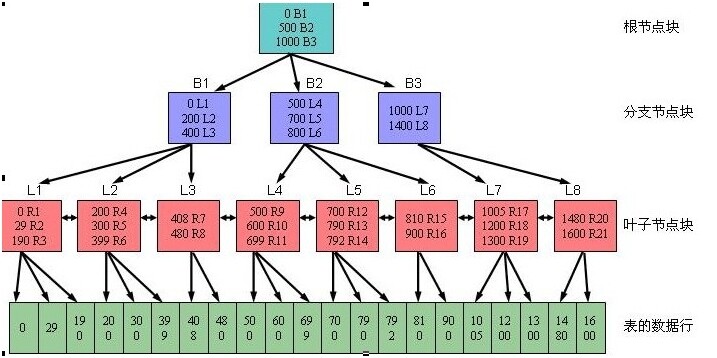
*如何从执行计划还原SQL的from部分和where部分？*

*通过执行计划和谓词信息来还原*

1. 如何看懂执行计划的顺序？



# 索引的访问方式



select \* from test where id = ‘29’;

1. 走索引，Root，Branch1，Lift1；
2. 找到了29的值，得到rowid=R2
3. 根据R2回表查询该行记录
4. 索引范围扫描



1. 索引唯一扫描



1. Rowid访问表



1. 索引全扫描



1. 索引快速全扫描



1. 索引Max/Min扫描



1. 索引跳跃扫描



1. 索引回表

