

Oracle数据库执行计划的一些基本概念（2）PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/143/2021\\_2022\\_Oracle\\_E6\\_95\\_B0\\_E6\\_c102\\_143561.htm](https://www.100test.com/kao_ti2020/143/2021_2022_Oracle_E6_95_B0_E6_c102_143561.htm)

三.表之间的连接 Join是一种试图将两个表结合在一起的谓词，一次只能连接2个表，表连接也可以被称为表关联。在后面的叙述中，我们将会使用”row source”来代替”表”，因为使用row source更严谨一些，并且将参与连接的2个row source分别称为row source1和row source 2。Join过程的各个步骤经常是串行操作，即使相关的row source可以被并行访问，即可以并行的读取做join连接的两个row source的数据，但是在将表中符合限制条件的数据读入到内存形成row source后，join的其它步骤一般是串行的。有多种方法可以将2个表连接起来，当然每种方法都有自己的优缺点，每种连接类型只有在特定的条件下才会发挥出其最大优势。 row source(表)之间的连接顺序对于查询的效率有非常大的影响。通过首先存取特定的表，即将该表作为驱动表，这样可以先应用某些限制条件，从而得到一个较小的row source，使连接的效率较高，这也就是我们常说的要先执行限制条件的原因。一般是在将表读入内存时，应用where子句中对该表的限制条件。根据2个row source的连接条件的中操作符的不同，可以将连接分为等值连接(如WHERE A.COL3 = B.COL4)、非等值连接(WHERE A.COL3 > B.COL4)、外连接(WHERE A.COL3 = B.COL4( ))。上面的各个连接的连接原理都基本一样，所以为了简单期间，下面以等值连接为例进行介绍。在后面的介绍中，都已：SELECT A.COL1, B.COL2 FROM A, B WHERE A.COL3 = B.COL4.为例进行说明，假设A

表为Row Source1，则其对应的连接操作关联列为COL 3；B表为Row Source2，则其对应的连接操作关联列为COL 4；连接类型：目前为止，无论连接操作符如何，典型的连接类型共有3种：排序 - - 合并连接(Sort Merge Join (SMJ)) 嵌套循环(Nested Loops (NL)) 哈希连接(Hash Join) 排序 - - 合并连接(Sort Merge Join, SMJ) 内部连接过程：1) 首先生成row source1需要的数据，然后对这些数据按照连接操作关联列(如A.col3)进行排序。2) 随后生成row source2需要的数据，然后对这些数据按照与sort source1对应的连接操作关联列(如B.col4)进行排序。3) 最后两边已排序的行被放在一起执行合并操作，即将2个row source按照连接条件连接起来 下面是连接步骤的图形表示：MERGE\SORTSORT||Row Source 1Row Source 2如果row source已经在连接关联列上被排序，则该连接操作就不需要再进行sort操作，这样可以大大提高这种连接操作的连接速度，因为排序是个极其费资源的操作，特别是对于较大的表。预先排序的row source包括已经被索引的列(如a.col3或b.col4上有索引)或row source已经在前面的步骤中被排序了。尽管合并两个row source的过程是串行的，但是可以并行访问这两个row source(如并行读入数据，并行排序)。SMJ连接的例子：SQL> explain plan for0select /\* ordered \*/ e.deptno, d.deptno from emp e, dept d where e.deptno = d.deptno order by e.deptno, d.deptno. Query Plan-----SELECT STATEMENT [CHOOSE] Cost=17MERGE JOIN SORT JOIN TABLE ACCESS FULL EMP [ANALYZED] SORT JOIN TABLE ACCESS FULL DEPT [ANALYZED]排序是一个费时、费资源的操作，特别对

于大表。基于这个原因，SMJ经常不是一个特别有效的连接方法，但是如果2个row source都已经预先排序，则这种连接方法的效率也是蛮高的。嵌套循环(Nested Loops, NL) 这个连接方法有驱动表(外部表)的概念。其实，该连接过程就是一个2层嵌套循环，所以外层循环的次数越少越好，这也就是我们为什么将小表或返回较小row source的表作为驱动表(用于外层循环)的理论依据。但是这个理论只是一般指导原则，因为遵循这个理论并不能总保证使语句产生的I/O次数最少。有时不遵守这个理论依据，反而会获得更好的效率。如果使用这种方法，决定使用哪个表作为驱动表很重要。有时如果驱动表选择不正确，将会导致语句的性能很差、很差。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)