

Oracle中表的连接及其调整 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_E4_B8_AD_E8_c102_143089.htm 在日常基于数据库应用的开发过程中，我们经常需要对多个表或者数据源进行关联查询而得出我们需要的结果集。那么Oracle到底存在着哪几种连接方式？优化器内部又是怎样处理这些连接的？哪种连接方式又是适合哪种查询需求的？只有对这些问题有了清晰的理解后，我们才能针对特定的查询需求选择合适的连接方式，开发出健壮的数据库应用程序。选择合适的表连接方法对SQL语句运行的性能有着至关重要的影响。下面我们就Oracle常用的一些连接方法及适用情景做一个简单的介绍。

一、嵌套循环连接（Nested Loop）

嵌套循环连接的工作方式是这样的：

- 1、Oracle首先选择一张表作为连接的驱动表，这张表也称为外部表（Outer Table）。由驱动表进行驱动连接的表或数据源称为内部表（Inner Table）。
- 2、提取驱动表中符合条件的记录，与被驱动表的连接列进行关联查询符合条件的记录。在这个过程中，Oracle首先提取驱动表中符合条件的第一条记录，再与内部表的连接列进行关联查询相应的记录行。在关联查询的过程中，Oracle会持续提取驱动表中其他符合条件的记录与内部表关联查询。这两个过程是并行进行的，因此嵌套循环连接返回前几条记录的速度是非常快的。在这里需要说明的是，由于Oracle最小的IO单位为单个数据块，因此在这个过程中Oracle会首先提取驱动表中符合条件的单个数据块中的所有行，再与内部表进行关联连接查询的，然后提取下一个数据块中的记录持续地循环连接下去。当然，如果单行记录

跨越多个数据块的话，就是一次单条记录进行关联查询的。

3、 嵌套循环连接的过程如下所示： Code highlighting

produced by Actipro CodeHighlighter

(freeware)[http://www.CodeHighlighter.com/NESTED LOOP](http://www.CodeHighlighter.com/NESTED_LOOP)

Outer Loop>Inner Loop> 我们可以看出这里面存在着两个循环，一个是外部循环，提取驱动表中符合条件的每条记录。另外一个为内部循环，根据外循环中提取的每条记录对内部表进行连接查询相应的记录。由于这两个循环是嵌套进行的，故此

种连接方法称为嵌套循环连接。嵌套循环连接适用于查询的选择性强、约束性高并且仅返回小部分记录的结果集。通常要求驱动表的记录（符合条件的记录，通常通过高效的索引访问）较少，且被驱动表连接列有唯一索引或者选择性强的非唯一索引时，嵌套循环连接的效率是比较高的。比如下面

这个查询是选用嵌套循环连接的典型例子： Code highlighting

produced by Actipro CodeHighlighter

(freeware)<http://www.CodeHighlighter.com/SQL>> 0select

e.empno,e.ename,e.job,d.dname2 from emp e,dept d3 where

e.deptno=d.deptno4 and e.empno=7900.EMPNO ENAME JOB

DNAME----- 7900 JAMES

CLERK SALESExecution

Plan----- 0

SELECT STATEMENT Optimizer=CHOOSE1 0 NESTED

LOOPS2 1 TABLE ACCESS (BY INDEX ROWID) OF EMP3 2

INDEX (UNIQUE SCAN) OF PK_EMP (UNIQUE)4 1 TABLE

ACCESS (BY INDEX ROWID) OF DEPT5 4 INDEX (UNIQUE

SCAN) OF PK_DEPT (UNIQUE) 在这个查询中，优化器选

择emp作为驱动表，根据唯一性索引PK_EMP快速返回符合条件empno为7900的记录，然后再与被驱动表dept的deptno关联查询相应的dname并最终返回结果集。由于dept表上面的deptno有唯一索引PK_DEPT，故查询能够快速定位deptno对应dname为SALES的记录并返回。嵌套循环连接驱动表的选择也是连接中需要着重注意的一点，有一个常见的误区是驱动表要选择小表，其实这是不对的。假如有两张表A、B关联查询，A表有1000000条记录，B表有10000条记录，但是A表过滤出来的记录只有10条，这时候显然用A表当做驱动表是比较合适的。因此驱动表是由过滤条件限制返回记录最少的那张表，而不是根据表的大小来选择的。在外连接查询中，如果走嵌套循环连接的话，那么驱动表必然是没有符合条件关联的那张表，也就是后面不加()的那张表。这是由于外连接需要提取可能另一张表没符合条件的记录，因此驱动表需要是那张我们要返回所有符合条件记录的表。比如下面这个查询，就是选择了emp表做为驱动表进行连接：嵌套循环连接返回前几行的记录是非常快的，这是因为使用了嵌套循环后，不需要等到全部循环结束再返回结果集，而是不断地将查询出来的结果集返回。在这种情况下，终端用户将会快速地得到返回的首批记录，且同时等待Oracle内部处理其他记录并返回。如果查询的驱动表的记录数非常多，或者被驱动表的连接列上无索引或索引不是高度可选的情况，嵌套循环连接的效率是非常低的。

二、排序合并连接(Sort Merge)

排序合并连接的方法非常简单。在排序合并连接中是没有驱动表的概念的，两个互相连接的表按连接列的值先排序，排序完后形成的结果集再互相进行合并连接提取符合条

件的记录。相比嵌套循环连接，排序合并连接比较适用于返回大数据量的结果。以下为排序合并连接的例子：Code highlighting produced by Actipro CodeHighlighter

```
(freeware)http://www.CodeHighlighter.com/Roby@XUE> 0select emp.ename,dept.dname2 from emp,dept3 where emp.deptno=dept.deptno4 /ENAME DNAME-----
```

```
----- CLARK ACCOUNTINGKING
```

```
ACCOUNTINGMILLER ACCOUNTINGJONES
```

```
RESEARCHSCOTT RESEARCHFORD RESEARCHADAMS
```

```
RESEARCHTURNER SALESJAMES SALESWARD
```

```
SALESMARTIN SALESBLAKE SALES12 rows 0selected.Execution
```

```
Plan-----
```

```
----- | Id | Operation | Name | Rows | Bytes | Cost (%CPU)|
```

```
Time
```

```
|-----
```

```
----- | 0 | SELECT STATEMENT | | 12 | 264 | 8 (25)| 00:00:01 || 1 |
```

```
MERGE JOIN | | 12 | 264 | 8 (25)| 00:00:01 || 2 | SORT JOIN | | 4 |
```

```
52 | 4 (25)| 00:00:01 || 3 | TABLE ACCESS FULL| DEPT | 4 | 52 | 3
```

```
(0)| 00:00:01 ||* 4 | SORT JOIN | | 12 | 108 | 4 (25)| 00:00:01 ||* 5 |
```

```
TABLE ACCESS FULL| EMP | 12 | 108 | 3 (0)| 00:00:01 |
```

可以看得出来上述查询首先按dept、emp两张表的deptno先排序，然后

排序好的结果集再进行合并连接返回最终的记录。排序合并

连接在数据表预先排序好的情况下效率是非常高的，也比较

适用于非等值连接的情况，比如>、>=、