https://blog.csdn.net/wxnjob/article/details/16855017

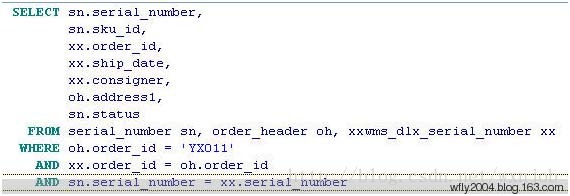
**PLSQL DEVELOPER执行计划的查看**

2013年11月21日 08:46:30 [太阳99](https://me.csdn.net/wxnjob) 阅读数：11446更多

个人分类： [Oracle](https://blog.csdn.net/wxnjob/article/category/1427896)

这里，我学到的一个很重要的东西，就是用PL/SQL DEVELOPER去看一条SELECT语句的执行计划，执行计划里面可以看到这条SELECT语句的开销、I/O操作开销等数值，可以很清晰地看到语句各个部分的执行效率。选中这条SELECT语句以后，按F5就可以。

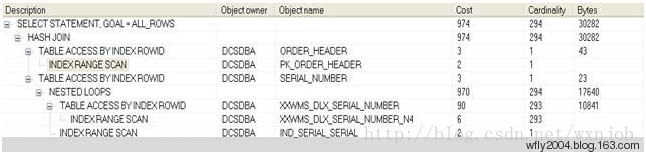
以下面的SELECT语句为例子：



从三张表中取数据，按我以前的想法，只要WHERE语句那里有能让三张表连接起来的条件就可以。所以我只使用了：

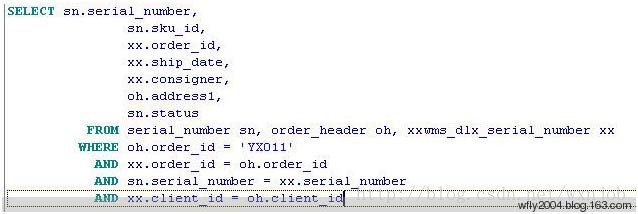
https://img-blog.csdn.net/20131121084457390?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvd3huam9i/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/SouthEast

按F5后，可以看到这样做的执行计划如下:



开销974，其实并不算高。但是加上测试环境的数据库连接速度确实很慢，导致了经常会出现数据库连接超时的提示。

修改后的SELECT语句如下：



可以看到，修改只是在WHERE里面多加了两个关联条件。此时再按F5看执行计划：



这时候COST只剩下95了。实际操作中，确实也明显感到了操作速度的提升。

执行计划的执行顺序是，从内层至外层，从上往下。比如上面修改后的实行计划中，数据库先对ORDER\_HEADER的主键做了一个索引范围检索，然后再对XXWMS\_DLX\_SERIAL\_NUMBER的非唯一性索引XXWMS\_DLX\_SERIAL\_NUMBER\_N4做索引范围检索。

在做修改之前，SERIAL\_NUMBER表和XXWMS\_DLX\_SERIAL\_NUMBER表使用了循环嵌套（NESTED LOOPS）的连接方法，得到结果后再与ORDER\_HEADER进行哈希连接。由于两个表的数据量都比较大（SERIAL\_NUMBER在测试环境中数据量达到一千四百万条左右），循环嵌套连接的步骤显然开销了很多。

可以猜想，如果循环嵌套连接中其中一个表的数据量小一点，就可以减少很多开销。于是，在连接条件中多加一条xx.client\_id = oh.client\_id。这样就使用了ORDER\_HEADER的主键去和XXWMS\_DLX\_SERIAL\_NUMBER的N4索引进行连接，得到结果以后再去和SERIAL\_NUMBER连接。结果在上图可以看到，COST只有95，大大加快了执行的效率。

这个经历让我觉得PL/SQL的优化相当神奇，效果是一样的，但只是加一个条件就可以使得执行效率有如此大的提高。就这个改变，使得弹出来的超时提醒的几率大大降低，对于用户来说这个改变是很重要的。

其它查看方法：

需要先创建plan\_table  
create table PLAN\_TABLE (  
statement\_id varchar2(30),  
timestamp date,  
remarks varchar2(80),  
operation varchar2(30),  
options varchar2(255),  
object\_node varchar2(128),  
object\_owner varchar2(30),  
object\_name varchar2(30),  
object\_instance numeric,  
object\_type varchar2(30),  
optimizer varchar2(255),  
search\_columns number,  
id numeric,  
parent\_id numeric,  
position numeric,  
cost numeric,  
cardinality numeric,  
bytes numeric,  
other\_tag varchar2(255),  
partition\_start varchar2(255),  
partition\_stop varchar2(255),  
partition\_id numeric,  
other long,  
distribution varchar2(30),  
cpu\_cost numeric,  
io\_cost numeric,  
temp\_space numeric);  
  
建表的sql在rdbms/admin下有，名字是utlxplan.sql  
我用的是9i,其他版本的名字可能不一样，  
然后执行  
explain plan for sql语句  
然后可以执行这个sql看结果  
SELECT LPAD (' ', 2 \* LEVEL) ||  
operation ||  
' ' ||  
options ||  
' ' ||  
object\_name query\_plan  
FROM plan\_table  
CONNECT BY PRIOR id = parent\_id  
START WITH id = 1  
ORDER BY id;