

OracleSQL最佳实践Oracle认证考试 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/644/2021\\_2022\\_OracleSQL\\_E6\\_c102\\_644919.htm](https://www.100test.com/kao_ti2020/644/2021_2022_OracleSQL_E6_c102_644919.htm) "mkhghigh">

1.用EXISTS代替DISTINCT，消除sort operation 2.如果在GROUP BY中过滤数据，在WHERE从句中指定条件比在HAVING从句中有更好的性能，因为在GROUP之前已经过滤掉数据，因此更少的行被汇总

3.UNION会对两个SELECT语句的结果集执行一个SORT，并消除重复行，成本会昂贵，而UNION ALL则不会。因此如果应用能够处理重复，或者确信没有重复记录，那么考虑使用UNION ALL代替UNION 4.能不用UNION就不要用它 5.为了避免在一个SQL语句中混合使用LEFT JOIN和RIGHT JOIN产生混淆，应该使用一个一致的视野，例如外连接只使用FULL or LEFT OUTER JOIN，忽略掉RIGHT OUTER JOIN

例如：Sql代码 SELECT e.lname, j.function, d.name FROM job j LEFT OUTER JOIN employee e ON e.job\_id = j.job\_id RIGHT OUTER JOIN department d ON e.dept\_id = d.dept\_id. SELECT e.lname, j.function, d.name FROM job j LEFT OUTER JOIN employee e ON e.job\_id = j.job\_id RIGHT OUTER JOIN department d ON e.dept\_id = d.dept\_id. 应该转换为：

Sql代码 SELECT e.lname, j.function, d.name FROM department d LEFT OUTER JOIN (job j LEFT OUTER JOIN employee e ON e.job\_id = j.job\_id) ON e.dept\_id = d.dept\_id. SELECT e.lname, j.function, d.name FROM department d LEFT OUTER JOIN (job j LEFT OUTER JOIN employee e ON e.job\_id = j.job\_id) ON e.dept\_id = d.dept\_id.

6.SQL被Oracle执行之前需要被解析。无论一个给定

的SQL语句被执行多少次，它仅仅需要一次解析。在解析期间，下面的步骤被执行(不考虑顺序)：(1)SQL语句语法被验证(2)数据字典被搜索用以验证表和列的定义(3)数据字典被搜索用以验证在相关对象上的安全权限(4)相关对象上会获得解析锁(5)决定最佳的执行计划(6)语句被加载到SGA系统全局区中的共享池内的共享SQL区(也叫library cache库高速缓存区)。这个执行计划和解析信息被保存在这里，以防止相同的语句被再次执行。一条SQL语句被解析的条件是，仅仅如果Oracle不能在SGA的共享SQL区中找到同样的SQL语句。在解析一条SQL语句之前，Oracle在库高速缓存中搜索相同的SQL语句。如果找到了确切的匹配，则不再需要解析这条语句。然而如果相同的SQL语句没有被找到，那么Oracle会执行上面所述的步骤去解析这条语句。为了成为相同的SQL语句，必须满足下述条件：(1)有相同的大小写字符(2)有相同的空格和换行(3)使用相同的名字引用相同的对象，必须有相同的owner所有者。如果应用可能多次执行相同的或相似的SQL语句，尽一切办法尝试避免不必要的解析。这样将会提高应用的所有性能。

减少SQL解析的两种技术：(1)使用绑定变量(2)使用表别名

### 6.1 使用绑定变量

当多用户使用一个应用的时候，会经常反复的执行相同的SQL语句集合，但是会使用不同的数值。例如，一个客户代表经常执行下面语句：Sql代码  
SELECT \* FROM customer WHERE cust\_nbr = 121. SELECT \* FROM customer WHERE cust\_nbr = 121. 而另一个客户代表将会执行：Sql代码  
SELECT \* FROM customer WHERE cust\_nbr = 328. SELECT \* FROM customer WHERE cust\_nbr = 328. 这两条SQL语句相似，但是不相同，因为cust\_nbr的号不同，因

此Oracle必须解析两次。可以使用绑定变量重写应用。这样的话，有问题的SQL语句可以改写为：  
Sql代码 SELECT \* FROM customer WHERE cust\_nbr = :x. SELECT \* FROM customer WHERE cust\_nbr = :x. Oracle仅仅需要解析一次语句了。多用户并发执行的程序在同时提供不同的cust\_nbr的时候，能够共享这条SQL语句相同的拷贝，减少不必要的解析。

### 6.2使用表别名

表别名的使用能够帮助提升SQL语句的性能，提供了方便的简化符号，是查询更可读，简明，表别名的长度最大能到30个字符 容易犯的错误是在写hint的时候忘记使用表别名，这样的话这些hint将被静态忽略掉 一旦定义了别名，在查询语句的任何地方以及任何的hint都必须指定别名，而不是真实的表名。这里阐述使用表别名如何影响性能，如下查询：  
Sql代码 SELECT c.cust\_nbr, name, order\_nbr FROM customer c, cust\_order o WHERE c.cust\_nbr = o.cust\_nbr. SELECT c.cust\_nbr, name, order\_nbr FROM customer c, cust\_order o WHERE c.cust\_nbr = o.cust\_nbr. 因为name列只在customer表中有，order\_nbr列只在cust\_order表中存在，所以可以不用别名限定列，这句sql是有效的。但是如果查询没有限定列，那么当解析这个语句找到这些列属于哪个表，Oracle必须搜索customer和cust\_order这两个表。对于一个查询来说，这个搜索所需要的时间可能是可以忽略不计的，但是如果你有许多这样的查询要解析的话，那么加起来会很耗资源了。在一个查询中，使用表别名限定所有的列(甚至那些非不明确的列)是好的编程实践，这样的好处是，解析语句的时候Oracle能够避免这种额外的搜索。

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)