

检查Oracle数据库中不合理的sql语句 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/466/2021\\_2022\\_\\_E6\\_A3\\_80\\_E6\\_9F\\_A5Orac\\_c67\\_466309.htm](https://www.100test.com/kao_ti2020/466/2021_2022__E6_A3_80_E6_9F_A5Orac_c67_466309.htm) 代码: 0select sql\_text

,sharable\_mem from v\$sql where sharable\_mem > 100000 order by sharable\_mem . 上面的sql语句是查询shared pool中占用内存超过100K的sql语句。这个sql可以非常有效的检查出Oracle shared pool中那些严重占用内存的sql，根据我的经验，绝大多数有问题的sql语句都会在这里留下痕迹，通过在这里找出有问题的sql语句并进行修改，再反复运行这个sql脚本，直到所有有问题的sql都处理完毕，这就是对Oracle数据库在sql上面的最好的优化，可以保证不会因为程序员的sql语句问题导致Oracle数据库的性能问题。共享池主要由库缓冲区（共享SQL区和PL/SQL区）和数据字典缓冲区组成。详细的情况你随便找一本介绍Oracle内存结构的书上面都有讲到。我就不把书上的内容给你打印一遍了，自己去看。 0select \* from v\$sgastat. --显式SGA的状态信息。有的人写的SQL语句非常复杂，嵌套了好几层，SQL语句本身写的很差，就有可能占用很大的SQL区域。其实现实的很多情况是本来不需要那么复杂的sql的时候，由于程序员水平不够，写了那种很糟糕很复杂的sql，造成的数据库性能问题。另外还有一个常识问题，却很容易被忽略。比如：代码: 0select \* from table\_name where id = 1. 0select \* from table\_name where id = 2. 对于这种带参数的sql，id = ? 这个地方叫做站位符(Placeholder)。拿PHP为例，很多人喜欢这样写代码 代码: \$sql = "0select \* from table\_name where id = ". \$id=1. \$stmt = ociparse(\$conn,\$sql.\$id).

ociexecute(\$stmt). ..... \$id = 2. \$stmt = ociparse(\$conn,\$sql.\$id).  
ociexecute(\$stmt). 拿Java为例，是这样的：代码: String sql =  
"0select \* from table\_name where id = ". Statement stmt =  
conn.createStatement(). rset = stmt.executeQuery(sql "1"). ..... rset  
= stmt.executeQuery(sql "2"). 这种写法，对于Oracle数据库来说  
，完全就是两条不同的sql语句，代码: 0select \* from  
table\_name where id = 1. 0select \* from table\_name where id = 2. 每  
次查询都要进行sql语句的执行解析，并且每个sql都会分配一个  
区域来存放sql解析后的二进制可执行代码。试想，要是id  
不同的10万个sql呢？Oracle就会分配10万个sql区域来分别存  
放10万个这样的id不同的sql语句。对于一个数据库驱动的  
Web网站这样情况下，SGA开的再大，也会很快被耗  
尽share pool的，最后报一个ORA-4031错误。数据库就连接不  
上了，只好重起。正确的写法应该是：代码: \$stmt =  
ociparse(\$conn,"0select \* from table\_name where id = :id").  
ocibindbyname(\$stmt,":id",&\$id, 12). \$id =1.  
ociexecute(\$stmt). ... \$id = 2. ociexecute(\$stmt). 代码:  
PreparedStatement pstmt = conn.prepareStatement("0select \* from  
table\_name where id = ?"). pstmt.setInt(1,1). rset =  
pstmt.executeQuery(). ... pstmt.setInt(1,2). rset =  
pstmt.executeQuery(). 这样Oracle数据库就知道你实际用的  
都是同一条sql语句，会以这样的形式： 0select \* from  
table\_name where id = :1 解析执行后存放在sql区域里面，当以  
后再有一样的sql的时候，把参数替换一下，就立刻执行，不  
需要再解析sql了。既加快了sql执行速度，也不会占有过  
多SGA的share pool。可惜的是，很多程序员明知道这个问题

，却意识不到问题的严重性，因为上面那种写法，编程的时候很灵活，sql语句可以动态构造，实现起来很容易，后面那种写法，sql语句是写死的，参数不能再变了，编程经常会非常麻烦。很多数据库的性能问题都是这样造成的。有兴趣在一个生产系统中，用上面sql检查一下，看看是否选择出来的是否有很多都是一样的sql语句，只是参数不同，如果是这样的话，就说明程序员的代码写的有问题。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)