Oracle数据库SQL语句性能调整的基本原则Oracle认证考试
 一、问题的提出 在应用系统开发初期，由于开发数据库数据比较少，对于查询SQL语句，复杂视图的的编写等体会不出SQL语句各种写法的性能优劣，但是如果将应用系统提交实际应用后，随着数据库中数据的增加，系统的响应速度就成为目前系统需要解决的最主要的问题之一。系统优化中一个很重要的方面就是SQL语句的优化。对于海量数据，劣质SQL语句和优质SQL语句之间的速度差别可以达到上百倍，可见对于一个系统不是简单地能实现其功能就可，而是要写出高质量的SQL语句，提高系统的可用性。 在多数情况下，Oracle使用索引来更快地遍历表，优化器主要根据定义的索引来提高性能。但是，如果在SQL语句的where子句中写的SQL代码不合理，就会造成优化器删去索引而使用全表扫描，一般就这种SQL语句就是所谓的劣质SQL语句。在编写SQL语句时我们应清楚优化器根据何种原则来删除索引，这有助于写出高性能的SQL语句。 二、SQL语句编写注意问题 下面就某些SQL语句的where子句编写中需要注意的问题作详细介绍。在这些where子句中，即使某些列存在索引，但是由于编写了劣质的SQL，系统在运行该SQL语句时也不能使用该索引，而同样使用全表扫描，这就造成了响应速度的极大降低。 1. IS NULL 与 IS NOT NULL 不能用null作索引，任何包含null值的列都将不会被包含在索引中。即使索引有多列这样的情况下，只要这些列中有一列含有null，该列就会从索引中排除。也就是说如果某列存在空值，即使对该

列建索引也不会提高性能。任何在where子句中使用is null或is not null的语句优化器是不允许使用索引的。 2. 联接列 对于有联接的列，即使最后的联接值为一个静态值，优化器是不会使用索引的。我们一起来看一个例子，假定有一个职工表（employee），对于一个职工的姓和名分成两列存放（FIRST_NAME和LAST_NAME），现在要查询一个叫比尔.克林顿（Bill Cliton）的职工。 下面是一个采用联接查询的SQL语句: 0select * from employss where first_name||||last_name =Beill Cliton. 上面这条语句完全可以查询出是否有Bill Cliton这个员工，但是这里需要注意，系统优化器对基于last_name创建的索引没有使用。 当采用下面这种SQL语句的编写，Oracle系统就可以采用基于last_name创建的索引。 Select * from employee where first_name =Beill and last_name =Cliton. 遇到下面这种情况又如何处理呢？如果一个变量（name）中存放着Bill Cliton这个员工的姓名，对于这种情况我们又如何避免全程遍历，使用索引呢？可以使用一个函数，将变量name中的姓和名分开就可以了，但是有一点需要注意，这个函数是不能作用在索引列上。下面是SQL查询脚本： 0select * from employee where first_name = SUBSTR(amp.name,1,INSTR(amp.name,)-1) and last_name = SUBSTR(amp.name,INSTR(amp.name ’ ,) 1) 3. 带通配符（%）的like语句 同样以上面的例子来看这种情况。目前的需求是这样的，要求在职工表中查询名字中包含cliton的人。可以采用如下的查询SQL语句： 0select * from employee where last_name like %cliton%. 这里由于通配符（%）在搜寻词首出现，所以Oracle系统不使用last_name的索引。在很多情况下可能无法

避免这种情况，但是一定要心中有底，通配符如此使用会降低查询速度。然而当通配符出现在字符串其他位置时，优化器就能利用索引。在下面的查询中索引得到了使用： 0select * from employee where last_name like c%. 4. Order by语句 ORDER BY语句决定了Oracle如何将返回的查询结果排序。Order by语句对要排序的列没有什么特别的限制，也可以将函数加入列中（象联接或者附加等）。任何在Order by语句的非索引项或者有计算表达式都将降低查询速度。 仔细检查order by语句以找出非索引项或者表达式，它们会降低性能。解决这个问题的办法就是重写order by语句以使用索引，也可以为所使用的列建立另外一个索引，同时应绝对避免在order by子句中使用表达式。 5. NOT 我们在查询时经常在where子句使用一些逻辑表达式，如大于、小于、等于以及不等于等等，也可以使用and（与）、or（或）以及not（非）。NOT可用来对任何逻辑运算符号取反。下面是一个NOT子句的例子： ... where not (status =VALID) 如果要使用NOT，则应在取反的短语前面加上括号，并在短语前面加上NOT运算符。NOT运算符包含在另外一个逻辑运算符中，这就是不等于（gt.）运算符。换句话说，即使不在查询where子句中显式地加入NOT词，NOT仍在运算符中，见下例： ... where status gt.INVALID. 再看下面这个例子： 0select * from employee where salarygt.3000. 对这个查询，可以改写为不使用NOT： 0select * from employee where salarygt.3000. 虽然这两种查询的结果一样，但是第二种查询方案会比第一种查询方案更快些。第二种查询允许Oracle对salary列使用索引，而第一种查询则不能使用索引。 100Test 下载频道开通，各类考试题目直接下载。详细请访问