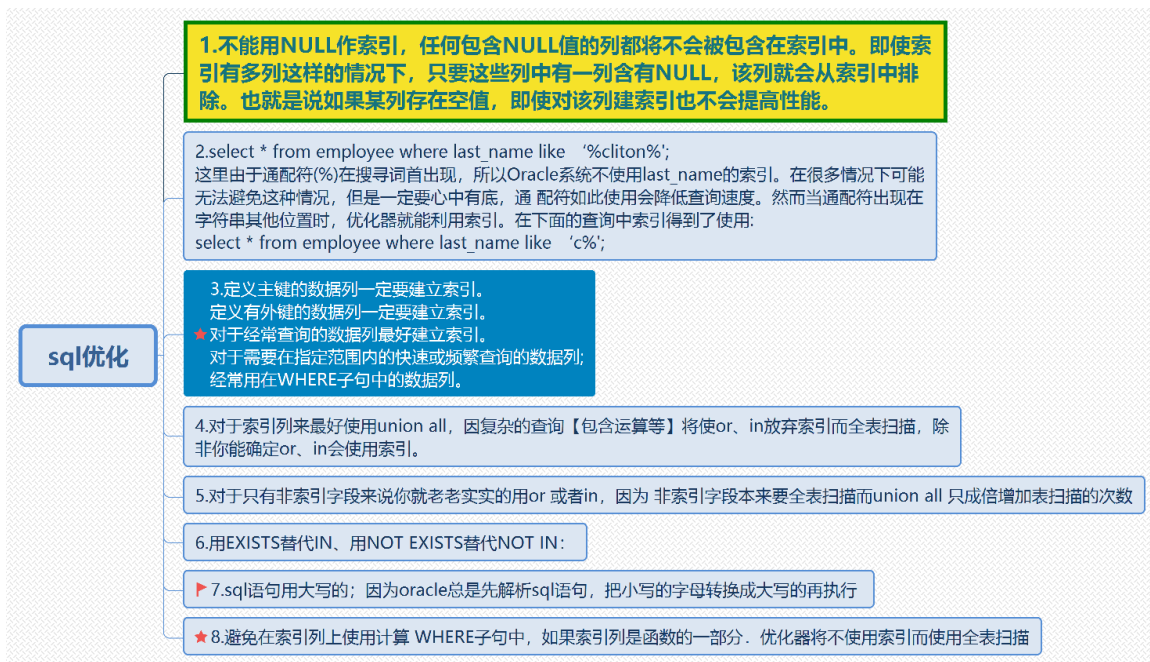


sql优化

sql优化.....	1
1.	
1.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。	2
2. 2.select * 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，但是一定要心中有数，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: select * from employee where last_name like 'c%';	2
3. 3.定义主键的数据列一定要建立索引。定义有外键的数据列一定要建立索引。对于经常查询的数据列最好建立索引。 对于需要在指定范围内的快速或频繁查询的数据列; 经常用在WHERE子句中的数据列。	2
4. 4.对于索引列来最好使用union all，因复杂的查询【包含运算等】将使or、in放弃索引而全表扫描，除非你能确定or、in会使用索引。	3
5. 5.对于只有非索引字段来说你就老老实实的用or 或者in，因为非索引字段本来要全表扫描而union all 只成倍增加表扫描的次数	3
6. 6.用EXISTS替代IN、用NOT EXISTS替代NOT IN:	3
7.	
7.sql语句用大写的；因为oracle总是先解析sql语句，把小写的字母转换成大写的再执行	3
8. 8.避免在索引列上使用计算 WHERE子句中，如果索引列是函数的一部分。优化器将不使用索引而使用全表扫描	3



1. 1.不能用null作索引，任何包含null值的列都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。

2. 2.`select * 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，但是一定要心中有数，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：

`select * from employee where last_name like 'c%';`

3. 3.定义主键的数据列一定要建立索引。
定义有外键的数据列一定要建立索引。

对于经常查询的数据列最好建立索引。

对于需要在指定范围内的快速或频繁查询的数据列；

经常用在WHERE子句中的数据列。



4. 4.对于索引列来最好使用union

all，因复杂的查询【包含运算等】将使or、in放弃索引而全表扫描，除非你能确定or、in会使用索引。

5. 5.对于只有非索引字段来说你就老老实实的用or 或者in，因为非索引字段本来要全表扫描而union all 只成倍增加表扫描的次数

6. 6.用EXISTS替代IN、用NOT EXISTS替代NOT IN：

7. 7.sql语句用大写的；因为oracle总是先解析sql语句，把小写的字母转换成大写的再执行



8. 8.避免在索引列上使用计算

WHERE子句中，如果索引列是函数的一部分。优化器将不使用索引而使用全表扫描

