

SQLServer索引结构及其使用（二）PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/179/2021_2022_SQLServer_E7_c98_179404.htm 计算机等级考试训练软件《百宝箱》改善SQL语句 很多人不知道SQL语句在SQL SERVER中是如何执行的，他们担心自己所写的SQL语句会被SQL SERVER误解。比如：0select * from table1 where name= ' ' zhangsan ' ' and tID > 10000 和执行: 0select * from table1 where tID > 10000 and name= ' ' zhangsan ' ' 一些人不知道以上两条语句的执行效率是否一样，因为如果简单的从语句先后上看，这两个语句的确是不一样，如果tID是一个聚合索引，那么后一句仅仅从表的10000条以后的记录中查找就行了；而前一句则要先从全表中查找看有几个name= ' ' zhangsan ' ' 的，而后再根据限制条件条件tID>10000来提出查询结果。事实上，这样的担心是不必要的。SQL SERVER中有一个“查询分析优化器”，它可以计算出where子句中的搜索条件并确定哪个索引能缩小表扫描的搜索空间，也就是说，它能实现自动优化。虽然查询优化器可以根据where子句自动的进行查询优化，但大家仍然有必要了解一下“查询优化器”的工作原理，如非这样，有时查询优化器就会不按照您的本意进行快速查询。在查询分析阶段，查询优化器查看查询的每个阶段并决定限制需要扫描的数据量是否有用。如果一个阶段可以被用作一个扫描参数（SARG），那么就称之为可优化的，并且可以利用索引快速获得所需数据。SARG的定义：用于限制搜索的一个操作，因为它通常是指一个特定的匹配，一个值得范围内的匹配或者两个以上条件的AND连接。形式如下：列名 操作符 或

操作符列名 列名可以出现在操作符的一边，而常数或变量出现在操作符的另一边。如：Name= '张三' 价格>5000
5000Name= '张三' and 价格>5000 如果一个表达式不能满足SARG的形式，那它就无法限制搜索的范围了，也就是SQL SERVER必须对每一行都判断它是否满足WHERE子句中的所有条件。所以一个索引对于不满足SARG形式的表达式来说是无用的。介绍完SARG后，我们来总结一下使用SARG以及在实践中遇到的和某些资料上结论不同的经验：1、Like语句是否属于SARG取决于所使用的通配符的类型 如：name like '张%'，这就属于SARG 而：name like '%张'，就不属于SARG。原因是通配符%在字符串的开通使得索引无法使用。2、or 会引起全表扫描 Name= '张三' and 价格>5000 符号SARG，而：Name= '张三' or 价格>5000 则不符合SARG。使用or会引起全表扫描。3、非操作符、函数引起的不满足SARG形式的语句 不满足SARG形式的语句最典型的情况就是包括非操作符的语句，如：NOT、!=、<、!、NOT EXISTS、NOT IN、NOT LIKE等，另外还有函数。下面就是几个不满足SARG形式的例子：ABS(价格)Name like '%三' 有些表达式，如：WHERE 价格*2>5000 SQL SERVER也会认为是SARG，SQL SERVER会将此式转化为：WHERE 价格>2500/2 但我们不推荐这样使用，因为有时SQL SERVER不能保证这种转化与原始表达式是完全等价的。4、IN 的作用相当与OR 语句：Select * from table1 where tid in (2,3) 和 Select * from table1 where tid=2 or tid=3 是一样的，都会引起全表扫描，如果tid上有索引，其索引也会失效。5、尽量少用NOT 6、exists 和 in 的执行效率是一样的 很多资料上都显示说

，exists要比in的执行效率要高，同时应尽可能的用not exists来代替not in。但事实上，我试验了一下，发现二者无论是前面带不带not，二者之间的执行效率都是一样的。因为涉及子查询，我们试验这次用SQL SERVER自带的pubs数据库。运行前我们可以把SQL SERVER的statistics I/O状态打开：

(1) 0select title,price from titles where title_id in (0select title_id from sales where qty>30) 该句的执行结果为：表 ' ' sales ' '。扫描计数 18，逻辑读 56 次，物理读 0 次，预读 0 次。表 ' ' titles ' '。扫描计数 1，逻辑读 2 次，物理读 0 次，预读 0 次。

(2) 0select title,price from titles where exists (0select * from sales where sales.title_id=titles.title_id and qty>30) 第二句的执行结果为：表 ' ' sales ' '。扫描计数 18，逻辑读 56 次，物理读 0 次，预读 0 次。表 ' ' titles ' '。扫描计数 1，逻辑读 2 次，物理读 0 次，预读 0 次。我们从此可以看到用exists和用in的执行效率是一样的。

7、用函数charindex()和前面加通配符%的LIKE执行效率一样 前面，我们谈到，如果在LIKE前面加上通配符%，那么将会引起全表扫描，所以其执行效率是低下的。但有的资料介绍说，用函数charindex()来代替LIKE速度会有大的提升，经我试验，发现这种说明也是错误的：

0select gid,title,fariqi,reader from tgongwen where charindex(' ' 刑侦支队 ' ',reader)>0 and fariqi> ' ' 2004-5-5 ' ' 用时：7 秒，另外：扫描计数 4，逻辑读 7155 次，物理读 0 次，预读 0 次。

0select gid,title,fariqi,reader from tgongwen where reader like ' ' % ' ' ' 刑侦支队 ' ' ' % ' ' and fariqi> ' ' 2004-5-5 ' ' 用时：7秒，另外：扫描计数 4，逻辑读 7155 次，物理读 0 次，预读 0 次。

8、union并不绝对比or的执行效

率高 我们前面已经谈到了在where子句中使用or会引起全表扫描，一般的，我所见过的资料都是推荐这里用union来代替or。事实证明，这种说法对于大部分都是适用的。

```
0select gid,fariqi,neibuyonghu,reader,title from Tgongwen where fariqi= '2004-9-16' ' or gid>9990000 用时：68秒。扫描计数1，逻辑读404008次，物理读283次，预读392163次。
```

```
0select gid,fariqi,neibuyonghu,reader,title from Tgongwen where fariqi= '2004-9-16' ' union 0select gid,fariqi,neibuyonghu,reader,title from Tgongwen where gid>9990000 用时：9秒。扫描计数8，逻辑读67489次，物理读216次，预读7499次。
```

看来，用union在通常情况下比用or的效率要高的多。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