

SQLServer索引结构及其使用（四）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\\_179402.htm](https://www.100test.com/kao_ti2020/179/2021_2022_SQLServer_E7_c98_179402.htm) 计算机等级考试训练软件《百宝箱》聚集索引的重要性的和如何选择聚集索引 在上一节的标题中，笔者写的是：实现小数据量和海量数据的通用分页显示存储过程。这是因为在将本存储过程应用于“办公自动化”系统的实践中时，笔者发现这第三种存储过程在小数据量的情况下，有如下现象：1、分页速度一般维持在1秒和3秒之间。2、在查询最后一页时，速度一般为5秒至8秒，哪怕分页总数只有3页或30万页。虽然在超大容量情况下，这个分页的实现过程是很快的，但在分前几页时，这个1 - 3秒的速度比起第一种甚至没有经过优化的分页方法速度还要慢，借用户的话说就是“还没有ACCESS数据库速度快”，这个认识足以导致用户放弃使用您开发的系统。笔者就此分析了一下，原来产生这种现象的症结是如此的简单，但又如此的重要：排序的字段不是聚集索引！本篇文章的题目是：“查询优化及分页算法方案”。笔者只所以把“查询优化”和“分页算法”这两个联系不是很大的论题放在一起，就是因为二者都需要一个非常重要的东西——聚集索引。在前面的讨论中我们已经提到了，聚集索引有两个最大的优势：1、以最快的速度缩小查询范围。2、以最快的速度进行字段排序。第1条多用在查询优化时，而第2条多用在进行分页时的数据排序。而聚集索引在每个表内又只能建立一个，这使得聚集索引显得更加的重要。聚集索引的挑选可以说是实现“查询优化”和“高效分页”的最关键因素。但要既使聚集索引列既符合查询列的需要

，又符合排序列的需要，这通常是一个矛盾。笔者前面“索引”的讨论中，将fariqi，即用户发文日期作为了聚集索引的起始列，日期的精确度为“日”。这种作法的优点，前面已经提到了，在进行划时间段的快速查询中，比用ID主键列有很大的优势。但在分页时，由于这个聚集索引列存在着重复记录，所以无法使用max或min来最为分页的参照物，进而无法实现更为高效的排序。而如果将ID主键列作为聚集索引，那么聚集索引除了用以排序之外，没有任何用处，实际上是浪费了聚集索引这个宝贵的资源。为解决这个矛盾，笔者后来又添加了一个日期列，其默认值为getdate()。用户在写入记录时，这个列自动写入当时的时间，时间精确到毫秒。即使这样，为了避免可能性很小的重合，还要在此列上创建UNIQUE约束。将此日期列作为聚集索引列。有了这个时间型聚集索引列之后，用户就既可以用这个列查找用户在插入数据时的某个时间段的查询，又可以作为唯一列来实现max或min，成为分页算法的参照物。经过这样的优化，笔者发现，无论是大数据量的情况下还是小数据量的情况下，分页速度一般都是几十毫秒，甚至0毫秒。而用日期段缩小范围的查询速度比原来也没有任何迟钝。聚集索引是如此的重要和珍贵，所以笔者总结了一下，一定要将聚集索引建立在：

- 1、您最频繁使用的、用以缩小查询范围的字段上；
- 2、您最频繁使用的、需要排序的字段上。

结束语 本篇文章汇集了笔者近段在使用数据库方面的心得，是在做“办公自动化”系统时实践经验的积累。希望这篇文章不仅能够给大家的工作带来一定的帮助，也希望能让大家能够体会到分析问题方法；最重要的是，希望这篇文章能够抛砖引玉，掀起大家

的学习和讨论的兴趣，以共同促进，共同为公安科技强警事业和金盾工程做出自己最大的努力。最后需要说明的是，在试验中，我发现用户在进行大数据量查询的时候，对数据库速度影响最大的不是内存大小，而是CPU。在我的P4 2.4机器上试验的时候，查看“资源管理器”，CPU经常出现持续到100%的现象，而内存用量却并没有改变或者说没有大的改变。即使在我们的HP ML 350 G3服务器上试验时，CPU峰值也能达到90%，一般持续在70%左右。本文的试验数据都是来自我们的HP ML 350服务器。服务器配置：双Inter Xeon 超线程 CPU 2.4G，内存1G，操作系统Windows Server 2003 Enterprise Edition，数据库SQL Server 2000 SP3

从上表中，我们可以看出，三种存储过程在执行100页以下的分页命令时，都是可以信任的，速度都很好。但第一种方案在执行分页1000页以上后，速度就降了下来。第二种方案大约是在执行分页1万页以上后速度开始降了下来。而第三种方案却始终没有大的降势，后劲仍然很足。在确定了第三种分页方案后，我们可以据此写一个存储过程。大家知道SQL SERVER的存储过程是事先编译好的SQL语句，它的执行效率要比通过WEB页面传来的SQL语句的执行效率要高。下面的存储过程不仅含有分页方案，还会根据页面传来的参数来确定是否进行数据总数统计。

```
--获取指定页的数据：CREATE PROCEDURE
pagination3@tblName varchar(255), -- 表名@strGetFields
varchar(1000) = *, -- 需要返回的列 @fldName varchar(255)=, --
排序的字段名@PageSize int = 10, -- 页尺寸@PageIndex int = 1, --
页码@doCount bit = 0, -- 返回记录总数, 非 0 值则返回
@OrderType bit = 0, -- 设置排序类型, 非 0 值则降
```

```

序@strWhere varchar(1500) = -- 查询条件 (注意: 不要加
where)ASdeclare @strSQL varchar(5000) -- 主语句declare
@strTmp varchar(110) -- 临时变量declare @strOrder varchar(400)
-- 排序类型if @doCount != 0beginif @strWhere !=set @strSQL =
"0select count(*) as Total from [" @tblName "]" where "
@strWhereelseaset @strSQL = "0select count(*) as Total from ["
@tblName "]"end --以上代码的意思是如果@doCount传递过来的
的不是0，就执行总数统计。以下的所有代码都是@doCount
为0的情况：elsebeginif @OrderType != 0beginset @strTmp = "--
如果@OrderType不是0，就执行降序，这句很重要
! endelsebeginset @strTmp = ">(0select max"set @strOrder = "
order by [" @fldName "]" asc"endif @PageIndex = 1beginif
@strWhere != set @strSQL = "0select top " str(@PageSize) " "
@strGetFields " from [" @tblName "]" where " @strWhere " "
@strOrderelseaset @strSQL = "0select top " str(@PageSize) " "
@strGetFields " from [" @tblName "]" @strOrder--如果是第一页
就执行以上代码，这样会加快执行速度endelsebegin--以下代
码赋予了@strSQL以真正执行的SQL代码 set @strSQL = "0select
top " str(@PageSize) " " @strGetFields " from [" @tblName "]" where
[" @fldName "]" @strTmp "(" [" @fldName "]") from (0select top "
str((@PageIndex-1)*@PageSize) " [" @fldName "]" from ["
@tblName "]" @strOrder " as tblTmp)" @strOrderif @strWhere !=
set @strSQL = "0select top " str(@PageSize) " " @strGetFields " from
[" @tblName "]" where [" @fldName "]" @strTmp "(" [" @fldName "]")
from (0select top " str((@PageIndex-1)*@PageSize) " [" @fldName
"]" from [" @tblName "]" where " @strWhere " " @strOrder ") as

```

tblTmp) and " @strWhere " " @strOrderend end exec (@strSQL)GO

上面的这个存储过程是一个通用的存储过程，其注释已写在其中了。在大数据量的情况下，特别是在查询最后几页的时候，查询时间一般不会超过9秒；而用其他存储过程，在实践中就会导致超时，所以这个存储过程非常适用于大容量数据库的查询。笔者希望能够通过对以上存储过程的解析，能给大家带来一定的启示，并给工作带来一定的效率提升，同时希望同行提出更优秀的实时数据分页算法。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)