

SQLServer索引结构及其使用之五 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/142/2021_2022_SQLServer_E7_c102_142998.htm 一、深入浅出理解索引结构二、改善SQL语句实现小数据量和海量数据的通用分页显示存储过程 建立一个 Web 应用，分页浏览功能必不可少。这个问题是数据库处理中十分常见的问题。经典的数据分页方法是:ADO 纪录集分页法，也就是利用ADO自带的分页功能（利用游标）来实现分页。但这种分页方法仅适用于较小数据量的情形，因为游标本身有缺点：游标是存放在内存中，很费内存。游标一建立，就将相关的记录锁住，直到取消游标。游标提供了对特定集合中逐行扫描的手段，一般使用游标来逐行遍历数据，根据取出数据条件的不同进行不同的操作。而对于多表和大表中定义的游标（大的数据集合）循环很容易使程序进入一个漫长的等待甚至死机。更重要的是，对于非常大的数据模型而言，分页检索时，如果按照传统的每次都加载整个数据源的方法是非常浪费资源的。现在流行的分页方法一般是检索页面大小的块区的数据，而非检索所有的数据，然后单步执行当前行。最早较好地实现这种根据页面大小和页码来提取数据的方法大概就是“俄罗斯存储过程”。这个存储过程用了游标，由于游标的局限性，所以这个方法并没有得到大家的普遍认可。后来，网上有人改造了此存储过程，下面的存储过程就是结合我们的办公自动化实例写的分页存储过程：
CREATE procedure pagination1(@pagesize int, --页面大小，如每页存储20条记录
@pageindex int --当前页码)
as
set nocount on
begin
declare @indextable table(id int identity(1,1),nid int) --定义

表变量 declare @PageLowerBound int --定义此页的底码
declare @PageUpperBound int --定义此页的顶码
set @PageLowerBound=(@pageindex-1)*@pagesizeset
@PageUpperBound=@PageLowerBound @pagesizeset rowcount
@PageUpperBoundinsert into @indextable(nid) 0select gid from
TGongwen where fariqi >dateadd(day,-365,getdate()) order by
fariqi desc0select O.gid,O.mid,O.title,O.fadanwei,O.fariqi from
TGongwen O,@indextable t where O.gid=t.nid and
t.id>@PageLowerBound and t.idendset nocount off 以上存储过程
运用了SQL SERVER的最新技术——表变量。应该说这个存储
过程也是一个非常优秀的分页存储过程。当然，在这个过程中，
您也可以把其中的表变量写成临时表：CREATE TABLE #Temp。
但很明显，在SQL SERVER中，用临时表是没有用表变量快的。
所以笔者刚开始使用这个存储过程时，感觉非常的不错，速度也
比原来的ADO的好。但后来，我又发现了比此方法更好的方法。
笔者曾在网上看到了一篇小短文《从数据表中取出第n条到第m
条的记录的方法》，全文如下：
从publish表中取出第n条到第m条的记录：SELECT TOP
m-n 1 * FROM publish WHERE (id NOT IN (SELECT TOP n-1 id
FROM publish)) id为publish表的关键字 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