

SQLSever索引简介 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/140/2021_2022_SQLSever_E2_80_c100_140920.htm

表的索引与字典中的索引非常相似。它可以极大地提高查询的速度。对一个较大的表来说，通过加索引，一个通常要花费几个小时来完成的查询只要几分钟就可以完成。（对于包含索引的数据库，SQL Sever需要一个可观的额外空间。例如，要建立一个聚簇索引，需要大约 1.2 倍于数据大小的空间。速度是需要付出代价的。）聚簇索引和非聚簇索引假设你已经通过字典的索引找到了一个字所在的页码。一旦已经知道了页码后，你很可能随机的翻寻字典，直至找到正确的页码。这里还有一种找到页码的更有效的方法。首先，把字典翻到大概一半的地方，如果要找的页码比半本字典处的页码小，就翻到四分之一处，否则，就把书翻到四分之三的地方。通过这种方法，你可以继续把字典分成更小的部分，直至找到正确的页码附近。这是找到书页的非常有效的一种方法。（呵呵，到处都是这个例子，跟Hello world有一拼）SQL Sever的表索引以类似的方式工作。一个表索引由一组页组成，这些页构成了一个树形结构。根页通过指向另外两个页，把一个表的记录从逻辑上分成两个部分。而根页所指向的两个页又分别把记录分割成更小的部分。每个页都把记录分成更小的分割，直至到达叶级页。索引有两种类型：聚簇索引和非聚簇索引。在聚簇索引中，索引树的叶级页包含实际的数据：记录的索引顺序与物理顺序相同。在非聚簇索引中，叶级页指向表中的记录：记录的物理顺序与逻辑顺序没有必然的联系。聚簇索引非常象目录表，目

录表的顺序与实际的页码顺序是一致的。非聚簇索引则更象书的标准索引表，索引表中的顺序通常与实际的页码顺序是不一致的。一本书也许有多个索引。例如，它也许同时有主题索引和作者索引。同样，一个表可以有多个非聚簇索引。通常情况下，你使用的是聚簇索引，但是你应该对两种类型索引的优缺点都有所理解。每个表只能有一个聚簇索引，因为一个表中的记录只能以一种物理顺序存放。通常你要对一个表按照标识字段建立聚簇索引。但是，你也可以对其它类型的字段建立聚簇索引，如字符型，数值型和日期时间型字段。从建立了聚簇索引的表中取出数据要比建立了非聚簇索引的表快。当你需要取出一定范围内的数据时，用聚簇索引也比用非聚簇索引好。例如，假设你用一个表来记录访问者在你网点上的活动。如果你想取出在一定时间段内的登录信息，你应该对这个表的DATETIME型字段建立聚簇索引。对聚簇索引的主要限制是每个表只能建立一个聚簇索引。但是，一个表可以有不止一个非聚簇索引。实际上，对每个表你最多可以建立249个非聚簇索引。你也可以对一个表同时建立聚簇索引和非聚簇索引。假如你不仅想根据日期，而且想根据用户名从你的网点活动日志中取数据。在这种情况下，同时建立一个聚簇索引和非聚簇索引是有效的。你可以对日期时间字段建立聚簇索引，对用户名字段建立非聚簇索引。如果你发现你需要更多的索引方式，你可以增加更多的非聚簇索引。非聚簇索引需要大量的硬盘空间和内存。另外，虽然非聚簇索引可以提高从表中取数据的速度，它也会降低向表中插入和更新数据的速度。每当你改变了一个建立了非聚簇索引的表中的数据时，必须同时更新索引。因此你对一个表

建立非聚簇索引时要慎重考虑。如果你预计一个表需要频繁地更新数据，那么不要对它建立太多非聚簇索引。另外，如果硬盘和内存空间有限，也应该限制使用非聚簇索引的数量。索引属性这两种类型的索引都有两个重要属性：你可以用两者中任一种类型同时对多个字段建立索引（复合索引）；两种类型的索引都可以指定为唯一索引。你可以对多个字段建立一个复合索引，甚至是复合的聚簇索引。假如有一个表记录了你的网点访问者的姓和名字。如果你希望根据完整姓名从表中取数据，你需要建立一个同时对姓字段和名字字段进行的索引。这和分别对两个字段建立单独的索引是不同的。当你希望同时对不止一个字段进行查询时，你应该建立一个对多个字段的索引。如果你希望对各个字段进行分别查询，你应该对各字段建立独立的索引。两种类型的索引都可以被指定为唯一索引。如果对一个字段建立了唯一索引，你将不能向这个字段输入重复的值。一个标识字段会自动成为唯一值字段，但你也可以对其它类型的字段建立唯一索引。假设你用一个表来保存你的网点的用户密码，你当然不希望两个用户有相同的密码。通过强制一个字段成为唯一值字段，你可以防止这种情况的发生。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