

告诉你影响SQLServer数据库性能设计的关键计算机等级考试
PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/644/2021_2022__E5_91_8A_

E8_AF_89_E4_BD_A0_E5_c98_644542.htm 1 逻辑数据库和表的设计

数据库的逻辑设计、包括表与表之间的关系是优化关系型数据库性能的核心。一个好的逻辑数据库设计可以为优化数据库和应用程序打下良好的基础。标准化的数据库逻辑设计包括用多的、有相互关系的窄表来代替很多列的长数据表。下面是一些使用标准化表的一些好处。A:由于表窄，因此可以使排序和建立索引更为迅速；B:由于多表，所以多值的索引成为可能；C:更窄更紧凑的索引；D:每个表中可以有少一些的索引，因此可以提高insert 0update 0delete等的速度，因为这些操作在索引多的情况下会对系统性能产生很大的影响；E:更少的空值和更少的多余值，增加了数据库的紧凑性由于标准化，所以会增加了在获取数据时引用表的数目和其间的连接关系的复杂性。太多的表和复杂的连接关系会降低服务器的性能，因此在这两者之间需要综合考虑。定义具有相关关系的主键和外来键时应该注意的事项主要是:用于连接多表的主键和参考的键要有相同的数据类型。2 索引的设计 A: 尽量避免表扫描 检查你的查询语句的where子句，因为这是优化器重要关注的地方。包含在where里面的每一列(column)都是可能的侯选索引，为能达到最优的性能，考虑在下面给出的例子:对于在where子句中给出了column1这个列。下面的两个条件可以提高索引的优化查询性能! 第一:在表中的column1列上有一个单索引。 第二:在表中有多索引，但是column1是第一个索引的列。 避免定义多索引而column1是第二个或后

面的索引，这样的索引不能优化服务器性能。例如:下面的例子用了pubs数据库。SELECT au_id, au_lname, au_fname FROM authors WHERE au_lname = ' White ' 按下面几个列上建立的索引将会是对优化器有用的索引 au_lname au_lname, au_fname 而在下面几个列上建立的索引将不会对优化器起到好的作用 au_address au_fname, au_lname 考虑使用窄的索引在一个或两个列上，窄索引比多索引和复合索引更能有效。用窄的索引，在每一页上将会有更多的行和更少的索引级别(相对与多索引和复合索引而言)，这将推进系统性能。对于多列索引，SQL Server维持一个在所有列的索引上的密度统计(用于联合)和在第一个索引上的histogram(柱状图)统计。根据统计结果，如果在复合索引上的第一个索引很少被选择使用，那么优化器对很多查询请求将不会使用索引。有用的索引会提高select语句的性能，包括 insert,update,delete。但是，由于改变一个表的内容，将会影响索引。每一个insert,update,delete语句将会使性能下降一些。实验表明，不要在一个单表上用大量的索引，不要在共享的列上(指在多表中用了参考约束)使用重叠的索引。在某一系列上检查唯一的数据的个数，比较它与表中数据的行数做一个比较。这就是数据的选择性，这比较结果将会帮助你决定是否将某一系列作为候选的索引列，如果需要，建哪一种索引。你可以用下面的查询语句返回某一系列的不同值的数目。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