

如何让你的SQL运行得更快 PDF转换可能丢失图片或格式，  
建议阅读原文

[https://www.100test.com/kao\\_ti2020/470/2021\\_2022\\_\\_E5\\_A6\\_82\\_E4\\_BD\\_95\\_E8\\_AE\\_A9\\_E4\\_c67\\_470704.htm](https://www.100test.com/kao_ti2020/470/2021_2022__E5_A6_82_E4_BD_95_E8_AE_A9_E4_c67_470704.htm) 人们在使用SQL时往往会陷入一个误区，即太关注于所得的结果是否正确，而忽略了不同的实现方法之间可能存在的性能差异，这种性能差异在大型的或是复杂的数据库环境中（如联机事务处理OLTP或决策支持系统DSS）中表现得尤为明显。笔者在工作实践中发现，不良的SQL往往来自于不恰当的索引设计、不充份的连接条件和不可优化的where子句。在对它们进行适当的优化后，其运行速度有了明显地提高！下面我将从这三个方面分别进行总结：为了更直观地说明问题，所有实例中的SQL运行时间均经过测试，不超过1秒的均表示为（测试环境--主机：HPLH II 主频：330MHZ 内存：128兆 操作系统

：Operserver5.0.4 数据库：Sybase11.0.3 一、不合理的索引设计例：表record有620000行，试看在不同的索引下，下面几个SQL的运行情况：1.在date上建有一非个群集索引 0select count(\*) from record where date > 19991201 and date < 2000 (25秒) 0select date,sum(amount) from record group by date (55秒) 0select count(\*) from record where date > 19990901 and place in ( BJ , SH ) (27秒) 分析：date上有大量的重复值，在非群集索引下，数据在物理上随机存放在数据页上，在范围查找时，必须执行一次表扫描才能找到这一范围内的全部行。2.在date上的一个群集索引 0select count(\*) from record where date > 19991201 and date < 2000 ( 14秒 ) 0select date,sum(amount) from record group by date ( 28秒 ) 0select

count(\*) from record where date > 19990901 and place in ( BJ , SH ) ( 14秒 ) 分析：在群集索引下，数据在物理上按顺序在数据页上，重复值也排列在一起，因而在范围查找时，可以先找到这个范围的起末点，且只在这个范围内扫描数据页，避免了大范围扫描，提高了查询速度。

3.在place , date , amount上的组合索引 0select count(\*) from record where date > 19991201 and date 2000 ( 26秒 ) 0select date,sum(amount) from record group by date ( 27秒 ) 0select count(\*) from record where date > 19990901 and place in ( BJ, SH ) ( 分析：这是一个不很合理的组合索引，因为它的前导列是place，第一和第二条SQL没有引用place，因此也没有利用上索引；第三个SQL使用了place，且引用的所有列都包含在组合索引中，形成了索引覆盖，所以它的速度是非常快的。

4.在date , place , amount上的组合索引 0select count(\*) from record where date > 19991201 and date 2000( 19990901 and place in ( BJ , SH ) ( 分析：这是一个合理的组合索引。它将date作为前导列，使每个SQL都可以利用索引，并且在第一和第三个SQL中形成了索引覆盖，因而性能达到了最优。

5.总结：缺省情况下建立的索引是非群集索引，但有时它并不是最佳的；合理的索引设计要建立在对各种查询的分析和预测上。一般来说：  
·有大量重复值、且经常有范围查询 ( between, > , = , < )  
·经常同时存取多列，且每列都含有重复值可考虑建立组合索引；  
·组合索引要尽量使关键查询形成索引覆盖，其前导列一定是使用最频繁的列。

二、不充份的连接条件：例：表card有7896行，在card\_no上有一个非聚集索引，表account有191122行，在 account\_no上

有一个非聚集索引，试看在不同的表连接条件下，两个SQL的执行情况：  
0select sum(a.amount) from account a, card b where a.card\_no = b.card\_no ( 20秒 )  
将SQL改为：0select sum(a.amount) from account a, card b where a.card\_no = b.card\_no and a.account\_no=b.account\_no ( 分析：在第一个连接条件下，最佳查询方案是将account作外层表，card作内层表，利用card上的索引，其I/O次数可由以下公式估算为：外层表account上的22541页 ( 外层表account的191122行\*内层表card上对应外层表第一行所要查找的3页 ) =595907次I/O 在第二个连接条件下，最佳查询方案是将card作外层表，account作内层表，利用account上的索引，其I/O次数可由以下公式估算为：外层表card上的1944页 ( 外层表card的7896行\*内层表account上对应外层表每一行所要查找的4页 ) = 33528次I/O 可见，只有充份的连接条件，真正的最佳方案才会被执行。 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)