

数据库索引

索引是一个单独存储在磁盘上的数据库结构，它们包含着对数据表里所有记录的引用指针，使用索引可以提高数据库特定数据的查询速度。索引是在存储引擎中实现的，因此每种存储引擎的索引不一定完全相同，并且每种存储引擎也不一定支持所有索引类型。

+

索引的存储类型有两种：BTREE和HASH，具体和表的存储引擎有关。MyISAM和InnoDB存储引擎只支持BTREE；MEMORY/HEAD存储索引可以支持HASH和BTREE索引。

1.索引类型：

联合索引：

- 联合索引又叫复合索引。对于复合索引:MySQL从左到右的使用索引中的字段，一个查询可以只使用索引中的一部份，但只能是最左侧部分。例如索引是key index (a,b,c). 可以支持a | a,b| a,b,c 3种组合进行查找，但不支持 b,c进行查找。当最左侧字段是常量引用时，索引就十分有效。
- 当我们需要查询多个组合列时，不要试图分别基于单个列建立单个单列索引（因为虽然有多个单列索引，但MySQL数据库每次只用一个它认为最有效率的索引）
- 联合索引的好处是已经对第二个键值进行了排序（例如取出某个用户按时间排序的记录）

InnoDB聚集索引：

- 按照每张表的主键构造一个B+树，同时叶子结点存放整张表的行记录数据
- 聚集索引的存储并不是物理上连续的而是逻辑上连续的
- 聚集索引对于主键的排序查找和范围查找速度很快

非聚集索引：

叶子结点不包含行记录全部数据，只包含主键值，每张表上可以有多个非聚集索引

2.索引的优点:

- 通过创建唯一索引，可以保证数据库表中每行数据的唯一性.
- 可以加快数据的查询速度.
- 在实现数据的参考完整性方面，可以加速表和表之间的连接.
- 在使用分组和排序子句进行数据检索时，同样可以显著减少查询中分组和排序的时间
- 通过使用索引，可以在查询中使用优化隐藏器，提高系统的性能。

3.索引的缺点:

- 创建索引和维护索引要耗费时间，并且随着数据量的增加耗费时间也增加.
- 索引需要占空间内存.
- 在对表中数据进行增加,删除和修改的时候，索引也需要动态维护，这样降低了数据维护速度.

4.应该对如下的列建立索引:

- 在作为主键的列上，强制该列的唯一性和组织表中数据的排列结构。
- 在经常用在连接的列上，这些列主要是一些外键，可以加快连接的速度。
- 在经常需要根据范围进行搜索的列上创建索引，因为索引已经排序，其指定的范围是连续的。
- 在经常需要排序的列上创建索引，因为索引已经排序，这样查询可以利用索引的排序，加快排序查询时间。
- 在经常使用在where子句中的列上面创建索引，加快条件的判断速度。

5.有些列不应该创建索引:

- 在查询中很少使用或者作为参考的列不应该创建索引。
- 对于那些只有很少数据值的列也不应该增加索引（比如性别，结果集的数据行占了表中数据行的很大比例，即需要在表中搜索的数据行的比例很大。增加索引，并不能明显加快检索速度）。
- 对于那些定义为text, image和bit数据类型的列不应该增加索引。这是因为，这些列的数据量要么相当大，要么取值很少。
- 当修改性能远远大于检索性能时，不应该创建索引，因为修改性能和检索性能是矛盾的。

6.索引失效：

- WHERE字句的查询条件里有不等于号（WHERE column!=...），MySQL将无法使用索引
- 如果WHERE字句的查询条件里使用了函数（如：WHERE DAY(column)=...），MySQL将无法使用索引
- 在JOIN操作中（需要从多个数据表提取数据时），MySQL只有在主键和外键的数据类型相同时才能使用索引，否则即使建立了索引也不会使用。
- 如果WHERE子句的查询条件里使用了比较操作符LIKE和REGEXP，MySQL只有在搜索模板的第一个字符不是通配符的情况下才能使用索引。比如说，如果查询条件是LIKE 'abc%',MySQL将使用索引；如果条件是LIKE '%abc'，MySQL将不使用索引。
- 在ORDER BY操作中，MySQL只有在排序条件不是一个查询条件表达式的情况下才使用索引。尽管如此，在涉及多个数据表的查询里，即使有索引可用，那些索引在加快ORDER BY操作方面也没什么作用。
- 如果某个数据列里包含着许多重复的值，就算为它建立了索引也不会有很好的效果。比如说，如果某个数据列里包含了净是些诸如“0/1”或“Y/N”等值，就没有必要为它创建一个索引。
- 如果条件中有or(并且其中有or的条件是不带索引的)，即使其中有条件带索引也不会使用(这也是为什么尽量少用or的原因)。注意：要想使用or，又想让索引生效，只能将or条件中的每个列都加上索引。

- 如果列类型是字符串，那一定要在条件中将数据使用引号引用起来,否则不使用索引。
- 如果mysql估计使用全表扫描要比使用索引快,则不使用索引。

聚集索引：按照每张表的主键构造一棵B+树，叶子结点存放的是整张表的行记录数据，每张表只能有一个聚集索引，聚集索引的叶子结点也称为数据页

好处：对于主键的排序查找和范围查找速度非常快

非聚集索引：叶子结点并不包含行记录数据，只包含键值与相应行数据指针，每张表中可以有多个辅助索引

联合索引：

- 联合索引又叫复合索引。对于复合索引:Mysql从左到右的使用索引中的字段，一个查询可以只使用索引中的一部份，但只能是最左侧部分。例如索引是key index (a,b,c). 可以支持a | a,b| a,b,c 3种组合进行查找，但不支持 b,c进行查找 .当最左侧字段是常量引用时，索引就十分有效。
- 当我们需要查询多个组合列时，不要试图分别基于单个列建立单个单列索引（因为虽然有多个单列索引，但mySql数据库每次只用一个它认为最有效率的索引）
- 联合索引的好处是已经对第二个键值进行了排序（例如取出某个用户按时间排序的记录）

1.聚集索引和非聚集索引的区别？

聚集索引和非聚集索引的根本区别是表中记录的排列顺序与索引的排列顺序是否一致。

a.聚集索引（CLUSTER INDEX）是指索引项的顺序与表中记录的物理顺序一致的索引组织。

优点是查询速度快，因为一旦具有第一个索引值的纪录被找到，具有连续索引值的记录也一定物理的紧跟其后。用户可以在最经常查询的列上建立聚簇索引以提高查询效率。在一个基本表上最多只能建立一个聚簇索引。

缺点是对表进行修改速度较慢，这是为了保持表中的记录的物理顺序与索引的顺序一致，而把记录插入到数据页的相应位置，必须在数据页中进行数据重排，降低了执行速度。建议使用聚集索引的场合为：

- 此列包含有限数目的不同值；
- 查询的结果返回一个区间的值；
- 查询的结果返回某值相同的大量结果集。

非聚集索引指定了表中记录的逻辑顺序，但记录的物理顺序和索引的顺序不一致，聚集索引和非聚集索引都采用了B+树的结构，但非聚集索引的叶子层并不与实际的数据页相重叠，而采用叶子层包含一个指向表中的记录在数据页中的指针的方式。非聚集索引比聚集索引层次多，添加记录不会引起数据顺序的重组。

建议使用非聚集索引的场合为：

- 此列包含了大量数目不同的值；
- 查询的结束返回的是少量的结果集；
- order by 子句中使用了该列。