第五章 创建高性能的索引

索引：存储引擎用于快速找到记录的一种数据结构。

#### 一．索引基础

使用索引查询：MySQL先在索引上按值进行查找，然后返回所有包含该值的数据行。

1. 索引类型

B-Tree索引

B+Tree索引（B-Tree的变种）InnoDB使用该索引

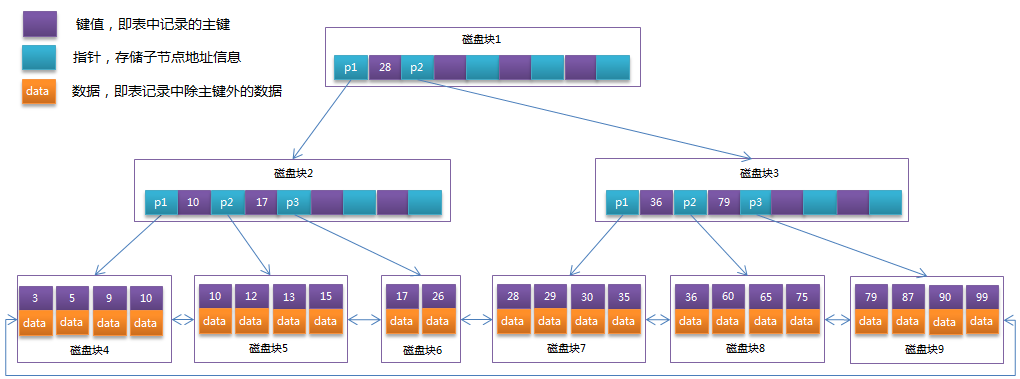
B-Tree与B+Tree的区别：

B+Tree只在叶子节点存储卫星数据（数据库中的记录），所以每个磁盘块能存储更多的节点，看起来更加的矮胖。同时叶子节点会有一个顺序链表。

B-Tree的卫星数据分布在整棵树当中，不会有相同的两个节点，所以一个磁盘块存储的数据有限，默认2~4个。

对比之下B+Tree的优点：

* 矮胖的优点是磁盘IO的次数更少。
* 所有查询都要查找到叶子节点，查询性能稳定。
* 所有叶子节点形成有序链表，便于范围查询。



组合索引：

根据索引定义的顺序，从左到右查找，所以前面的索引一定是变化小于后面的才会提高性能。而且使用组合索引查找的时候，一定要从左到右使用，否则会使得索引失效。

索引可以用于：

* 全值匹配
* 匹配最左前缀
* 匹配列前缀
* 匹配范围值
* 精确匹配某一列，并范围匹配另一列
* 只访问索引的查询

#### 二．索引优点

1. 大大减少了服务器需要扫描的数据量。

2. 索引可以帮助服务器避免排序和临时表。

3. 索引可以将随机IO变为顺序IO。

#### 三．什么情况下索引会失效

1. 索引不是独立的列

例如：

索引是表达式的一部分

索引是某个函数的参数

2. 组合索引必须顺序引用索引

3. 范围查询之后的条件不会使用索引，所以尽量把范围查询放到最后面。

4. 如果条件中有or，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)，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。

5. like查询以%开头。

6. 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。

7. 如果mysql估计使用全表扫描要比使用索引快,则不使用索引。