



链滴

MySQL (索引)

作者: [qingshui666](#)

原文链接: <https://ld246.com/article/1594692447491>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

索引是什么

索引是一个特殊的文件，他是实际存在在文件系统上的，记录着数据表里所有数据的引用指针

索引是一个数据结构，是数据库操作系统的一种排序数据结构，能帮助我们快速查询，更新我们据表的数据

-

- 优点

-

-

- 创建索引的原因是为了帮助用户 `快速地检索数据`

-

-

- 缺点

-

-

- 创建索引能加快检索速度，但也意味着数据库增删改时需要对索引 `进行维护`，`会增加增删改的性能消耗，降低执行效率`。

- 索引是实际存在系统中的，会占用系统的存储空间。

-

索引使用场景

-

-

where

因为主键索引中存储或者包含了行数据的引用地址，一般情况下，主键索引是最快的。`如果`个where 语句中包含多个索引，MySQL会选择最优的命中。

-

-

orderBy

在我们对某个字段进行 `orderBy` 时，如果这个字段没有建立索引，MySQL 会使用 `外部排序`，即是将查询到的 `结果集分批从硬盘当中读取内存中进行排序`，这个操作不仅要进行 IO 操作还要占用内存进行排序所以它是非常影响性能的。

如果存在索引的情况下，MySQL 会直接根据索引的排序和映射逐条取出数据。如果是分页的话接取索引某个范围进行读取。不再需要读入内存中排序后再进行截取某一部分数据。

-

-

join

在我们设计表结构的时候，我们要 join 的字段应该是一个外键并且应该加上索引，这样能提高 join 的查询效率，如果外键不存在索引的情况下，join 的表可能会出现全表扫描。严重损耗检索效率

-

-

索引覆盖

如果我们一个 select 语句中，需要查询的字段都建立过索引，那么 MySQL 会直接从索引页中获数据，而不再去查询原始数据，这个就是索引覆盖。索引我们在写查询语句的时候尽量 select 需要的段，提高索引覆盖的几率。

索引的几种类型

-

-

主键索引：数据表中的唯一标识，不允许为 null

-

-

唯一索引：数据表的的列不允许重复，多个列可以聚合，允许为 null

-

-

普通索引：基础的索引，多个列可以聚合，允许为 null

<p>全文索引：一种全文搜索索引</p>

<h3 id="索引的两种算法">索引的两种算法</h3>

<p>b+tree
BTree 是最常用的 mysql 数据库索引算法，也是 mysql 默认的算法。因为它不仅可以被用在 =, >, <, =, <, < 和 between 这些比较操作符上，而且还可以用于 like 操作符，只要它的查询条件是个不以通配符开头的常量，例如：</p>

<p>hash 算法
Hash 索引只能用于对等比较，例如 =, < 和 > (相当于 =) 操作符。由于是一次定位数据，不像 BTree 索引需要从根节点到枝节点，最后才能访问到页节点这样多次 IO 访问，所以检索效率远高于 BTree 索引。</p>

<h3 id="索引的设计原则">索引的设计原则</h3>

<p>适合索引的字段应该是出现在 where 语句中，或者 join 连接的列中。</p>

<p>数据过少的表不适合创建索引</p>

<p>尽量是用短索引，有时需要索引很长的字符列，它会使索引变大并且变慢。索引字符串的前半部能有效地节约索引空间。</p>

<p>不要过度索引，索引会占用磁盘空间，并且会降低写性能。索引的创建只要保证查询性能即可。</p>

<h3 id="索引的创建原则">索引的创建原则</h3>

<p>最左前匹配原则，是聚合索引中非常重要的原则，MySQL 会一直向右匹配直到遇到范围查询 (>, <, between, like) 就停止匹配。例如组合索引 abc, 查询语句为 a=1, b>2, c=3。这样 c 是使不了索引的。</p>

<p>字段较为频繁查询的应该使用索引。</p>

<p>频繁更新的字段不适合创建索引。</p>

<p>不能有效区分的列不适合创建索引。(如性别，男女未知，最多也就三种，区分度实在太低)</p>

-

<p>尽量扩展索引，而不是去新建索引。如系统上有 a 索引，要增加一个 ab 索引，应该直接拓展索引，将 a 索引修改为 ab 索引。</p>

-

<p>有外键的列一定要建立索引。</p>

-

<p>对 text,image,bit 或者数据过长的字段不要建立索引</p>

-

- <p>不要设置可空字段，因为可空字段很难被查询优化，同事会使索引排序运算更加复杂，可以使用个特殊的值或者 0 或者空字符串代替。</p>

-

<p>取离散值最大的字段（数据表值唯一值越多的离散值越大）</p>

-

<p>索引字段越小越好，字段过长影响索引效率，占用更多内存空间。</p>

-

- <p>顾名思义，就是最左优先，在创建多列索引时，要根据业务需求，where 子句中使用最频繁的一放在最左边。</p> - <p>最左前缀匹配原则，非常重要的原则，mysql 会一直向右匹配直到遇到范围查询(>、<、between、like)就停止匹配，比如 a = 1 and b = 2 and c > 3 and d = 4 如果建立(a,b,c,d)顺序的索引，d 是用不到索引的，如果建立(a,b,d,c)的索引则都可以用到，a,b,d 的顺序可以任意调整。</p> - <p>=和 in 可以乱序，比如 a = 1 and b = 2 and c = 3 建立(a,b,c)索引可以任意顺序，mysql 的查询优化器会帮你优化成索引可以识别的形式</p> - - <p>聚簇索引会将索引和数据放到一块，找到了索引，就找到了数据。</p> - <p>非聚簇索引记录着数据的引用地址，除非是在索引覆盖的情况下否则都会，会触发一次回表查询</p> -