# mysql索引

## 索引是什么

官方文档写了索引的作用和没有索引会带来全表扫描，非常费时间。Indexes are used to find rows with specific column values quickly. Without an index, MySQL must begin with the first row and then read through the entire table to find the relevant rows. 简单的说索引是提高查询速度。这个很好理解，就像是以前的英文词典，找单词如果没有前面目录的话，效率很低，得全文找一遍。

## 为什么需要索引

* 索引是数据表中一个或者多个列进行排序的数据结构
* 索引能够大幅提升检索速度
* 创建、更新索引本身也会耗费空间和时间

## 创建索引的方式

直接创建索引，例如使用CREATE INDEX 语句或者使用创建索引向导。 间接创建索引，例如在表中定义主键约束或者唯一性键约束时，同时也创建了索引。

## 索引的优缺点

### 优点

* 协助快速查询、更新数据库表中数据。
* 通过创建唯一性索引，可以保证数据库表中每一行数据的唯一性。
* 可以大大加快数据的检索速度
* 可以加速表和表之间的连接，特别是在实现数据的参考完整性方面特别有意义
* 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，可以在查询的过程中，使用优化隐藏器，提高系统的性能

### 缺点

* 增加了数据库的存储空间
* 在插入和修改数据时要花费较多的时间（因为索引也要随之变动）
* 创建索引和维护索引要耗费时间，这种时间随着数据量的增加而增加。
* 索引需要占物理空间，除了数据表占数据空间之外，每一个索引还要占一定的物理空间，如果要建立聚簇索引，那么需要的空间就会更大。
* 当对表中的数据进行增加、删除和修改的时候，索引也要动态的维护，这样就降低了数据的维护速度。

## 创建索引的原则

* 最左前缀匹配原则（一直向右匹配直到遇到范围查询就停止匹配）；
* =和in 可以乱序（建立索引是可以任意顺序的，mysql 的查询优化器会帮你优 化成索引可以识别的形式）；
* 尽量选择区分度高的列作为索引（区分度的公式是count(distinctcol)/count(\*)，表示字段不重复的比例，比例越大我们扫描的记录数越少，唯一 键的区分度是1，）；
* 索引列不能参与计算（保持列“干净”，）；
* 尽量的扩展索引，不要新建索引（如a->（a，b）只需要修改原来的索引）；
* 选择唯一性索引（唯一性索引的值是唯一的，可以更快速的通过该索引来确定某 条记录。）；
* 为经常需要排序、分组和联合操作的字段建立索引；
* 为常作为查询条件的字段建立索引；
* 限制索引的数目；
* 尽量使用数据量少的索引；
* 尽量使用前缀来索引；
* 删除不再使用或者很少使用的索引。

## 索引的实现原理

要搞清楚索引的实现原理，先看看索引的底层实现，MySQL索引大部分采用B-Tree实现，B-Tree又有B-树和B+树。还有一些使用Hash索引。

### 二叉搜索树

理解二叉搜索树，对于后面理解B-和B+树很有帮助，因为这2种有些特性跟二叉搜索树很像。二叉搜索树的特点是左孩子的值小于父亲节点的值，父亲节点的值小于右孩子的值，即按二叉树的中序遍历，刚好是一个按小到大排序的。二叉搜索树的查找就可以使用二分查找，如果要查找10，因为10比27小，所以往左孩子找，10<14，还在左孩子找。最坏的情况下，查找的次数等于树的高度。

### B-树，B+树

## 连接

* 内连接（inner join）：两个表都存在匹配时，才会返回匹配行
* 外连接（left join right join）：返回一个表的行，即使另一个没有匹配
* 全连接（full join）: 只要某一个表存在匹配就返回

**内连接**

select \* from A inner join B on a.id = b.id

**外连接**

select \* from A left join B on a.id = b.id

## 

## 参考资料

https://juejin.im/post/5c754718e51d4525f05461b8 https://juejin.im/post/5c8bd3ebf265da2db2797d92 https://juejin.im/post/5c7a8f2e6fb9a049cd54e8ff