



MySQL知识点

锁

查询

日志

事务

7

索引

索引原理

- 把无序的数据变成有序的查询
- 把创建了索引的列的内容进行排序
- 对排序结果生成倒排表
- 在倒排表内容上拼上数据地址链
- 查询的时候, 先拿到倒排表内容, 再取出数据地址链, 从而拿到具体数据

索引分类

按数据结构分类

- B+Tree索引
 - MySQL默认索引, 可用于查找、分组、排序
- Hash索引
 - 把数据进行哈希算法获得哈希值, 把这些哈希值放到一张哈希表中, 哈希表放到内存
 - 基于哈希表实现, 适合新增和等值查询, 不是有序的, 范围查询很慢, 必须全表扫描;
- Full-Text索引
 - 大量文件检索时, 需要用全文索引, 因为它的速度是 like的 N倍(MySQL 5.6版本InnoDB支持)
- R-Tree索引
 - 空间数据索引会从所有维度来索引数据, 主要用于地理数据存储

按物理存储分类

- InnoDB
 - 聚簇索引
 - 每个叶子节点存储了一行完整的表数据
 - 默认在主键字段上建立聚簇索引
 - 没有主键的话, 第一个非空的唯一索引将被建立为聚簇索引
 - 在前两者都没有的情况下, 将自动生成一个隐式的自增id列, 并在此列上建立聚簇索引
 - 二级索引
 - 聚簇索引以外的其他索引
 - 需要回表查询
 - 需要注意的是, 通过二级索引查询时, 回表不是必须的过程, 当SELECT的所有字段在单个二级索引中都能够找到时, 就不需要回表, MySQL称此时的二级索引为覆盖索引或触发了索引覆盖。
- MyISAM
 - 无聚簇索引

按字段特性分类

- 主键索引
- 普通索引
- 前缀索引
- 单列索引

按字段数量分类

- 联合索引
 - 最左前缀原则
 - 最左匹配原则是指, 在执行查询过程中, 不需要索引的全部定义, 只需要满足最左前缀, 就可以利用索引来加速检测。
 - 最左前缀可以是联合索引的最左N个字段, 也可以是字符串索引的最左M个字符

索引下推是 MySQL5.6 引入的特性, 指在索引遍历的过程中, 在符合最左匹配原则情况下, 对索引中包含的字段先进行判断, 直接过滤掉不满足条件的记录, 减少回表次数。

例子: /* 联合索引 (name, age) */
SELECT * FROM T WHERE name like 'A%' AND age = 10

- 在无索引下推的情况下, 找到符合 like A% 的行, 直接回表查主键的行记录, 然后在判断 age=10
- 在有索引下推的情况下, 先判断 age = 10, 再回表