

MySQL索引

- 连老师

索引面试题

- 数据库中最常见的慢查询优化方式是什么？
- 为什么加索引能优化慢查询？
- 你知道哪些数据结构可以提高查询速度？
- 那这些数据结构既然都能优化查询速度，Mysql为何选择使用B+树？

基础知识储备

▪ 局部性原理

现在，进而讨论在作业信息不全部装入主存的情况下能否保证作业的正确运行？回答是肯定的，早在 1968 年 P.Denning 就研究了程序执行时的**局部性**（**principle of locality**）原理，对程序局部性原理进行研究还有 Knuth（分析一组学生的 Fortran 程序）、Tanenbaum（分析操作系统的过程）、Huck（分析通用科学计算的程序），发现程序和数据的访问都有聚集成群的倾向，在一个时间段内，仅使用其中一小部分（称**空间局部性**），或者最近访问过的程序代码和数据，很快又被访问的可能性很大（称**时间局部性**）。这只要对程序的执行进行分析就可以发现以下一些情况：

▪ 磁盘预读(预读的长度一般为页（page）的整数倍)

- 页是存储器的逻辑块，操作系统往往将主存和磁盘存储区分割为连续的大小相等的块，每个存储块称为一页（在许多操作系统中，页大小通常为4k），主存和磁盘以页为单位交换数据。

索引是什么

- 索引是帮助 MySQL 高效获取数据的数据结构
- 索引存储在文件系统中
- 索引的文件存储形式与存储引擎有关
- 索引文件的结构
 - hash
 - 二叉树
 - B树
 - B+树

索引的分类

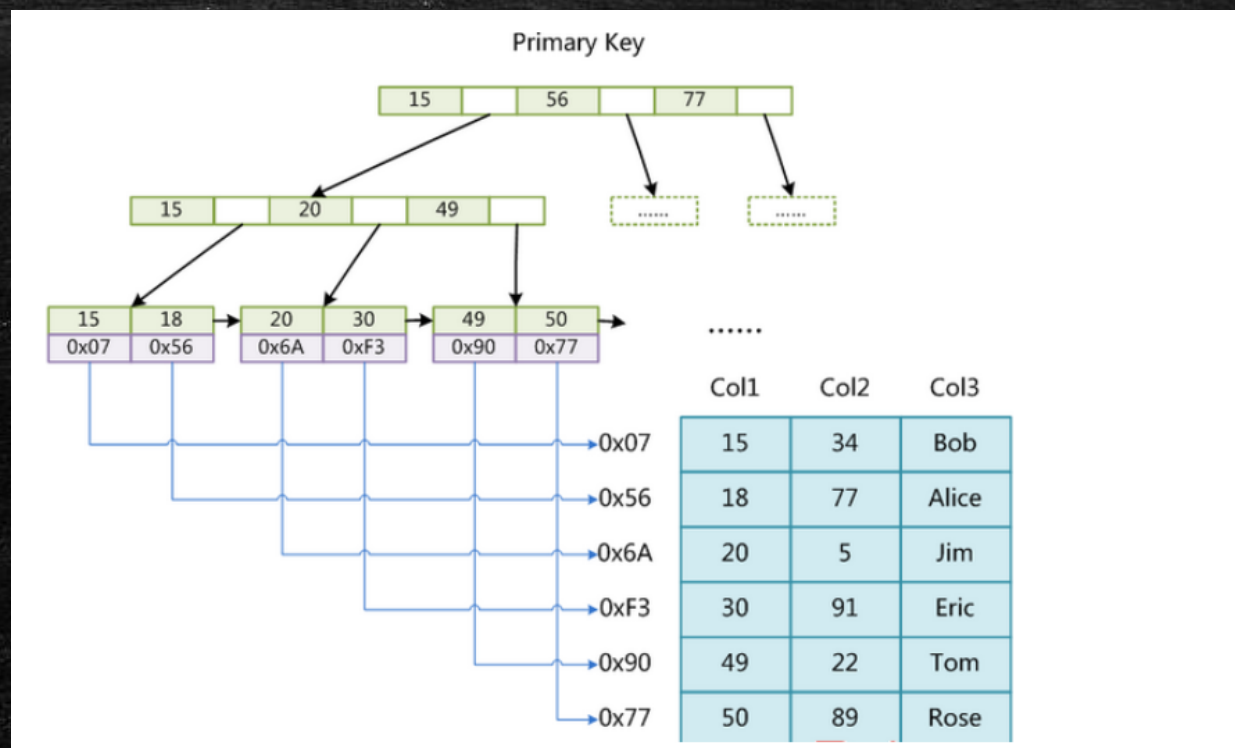
- mysql索引的五种类型：主键索引、唯一索引、普通索引和全文索引、组合索引。通过给字段添加索引可以提高数据的读取速度，提高项目的并发能力和抗压能力。
- 主键索引
 - 主键是一种唯一性索引，但它必须指定为PRIMARY KEY，每个表只能有一个主键。
- 唯一索引
 - 索引列的所有值都只能出现一次，即必须唯一，值可以为空。
- 普通索引
 - 基本的索引类型，值可以为空，没有唯一性的限制。（覆盖索引）
- 全文索引,MyISAM支持，Innodb在5.6之后支持
 - 全文索引的索引类型为FULLTEXT。全文索引可以在varchar、char、text类型的列上创建
- 组合索引
 - 多列值组成一个索引，专门用于组合搜索（最左匹配原则）

Mysql存储引擎

	MyISAM	InnoDB
索引类型	非聚簇索引	聚簇索引
支持事务	否	是
支持表锁	是	是
支持行锁	否	是
支持外键	否	是
支持全文索引	是	是（5.6后支持）
适合操作类型	大量select	大量insert、delete、update

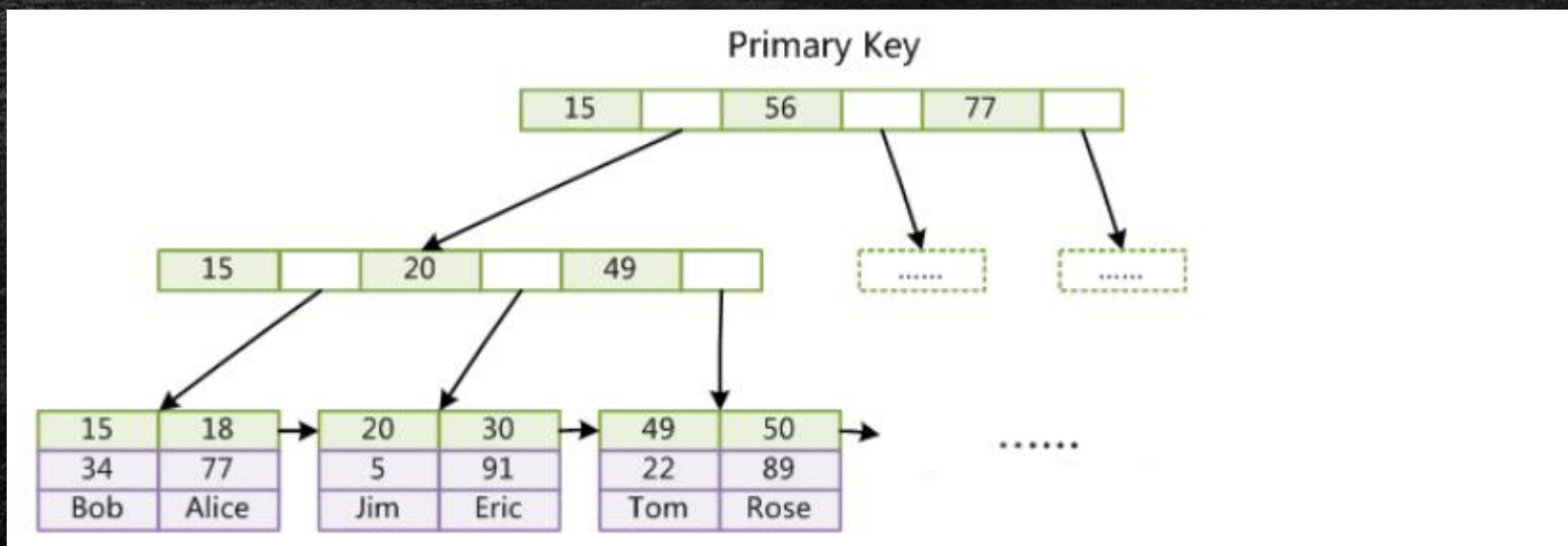
mysql索引机制

- mysql-MyISAM



Mysql索引机制

- mysql-InnoDB



索引面试难点

- 回表
- 覆盖索引
- 最左前缀
- 索引下推

索引维护

- 索引在插入新的值的时候，为了维护索引的有序性，必须要维护，在维护索引的时候需要分以下集中情况：
 - 1、如果插入一个比较大的值，直接插入即可，几乎没有成本
 - 2、如果插入的是中间的某一个值，需要逻辑上移动后续的元素，空出位置
 - 3、如果需要插入的数据页满了，就需要单独申请一个新的数据页，然后移动部分数据过去，叫做页分裂，此时性能会受影响同时空间的使用率也会降低，除了页分裂之外还包含页合并
- 尽量使用自增主键作为索引