系统设计最核心的内容就是数据库设计。



B+ Tree的原理？

InnoDB和MyISAM的区别？

什么是事务的隔离级别？

索引：是一种特殊的数据库结构，由数据表中的1列或多列组成，可以用来快速查询数据表中有某一特定值的记录。

书籍的目录就是一种索引。

目录的特点：

1. 单独的机构，不和内容混合
2. 指向具体内容的位置
3. 加速了对书籍内容的查找
4. 有些东西没有办法放到目录里
5. 书变厚了。

MySQL中的索引：索引是帮助MySQL高效获取数据的数据结构。

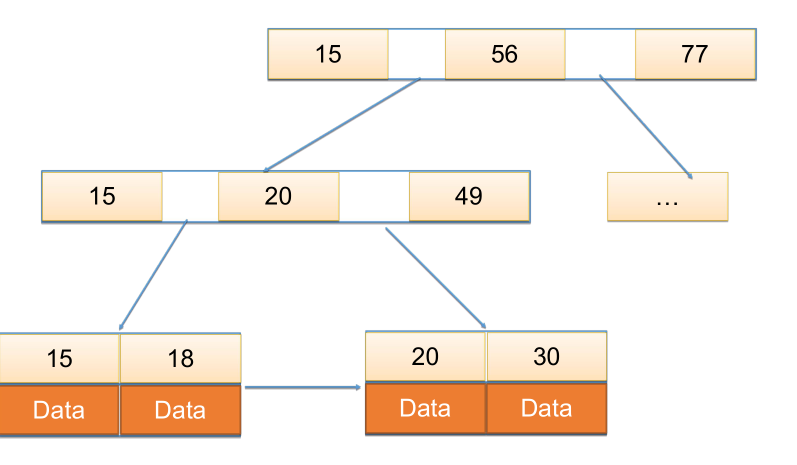
索引以索引文件的形式存储在磁盘中，索引的结构组织要尽可能的减少查找过程中磁盘I/O的存取次数。

MySQL的索引：带顺序指针的B+树（平衡树）。

一个m阶B+树的特征：

1. 根结点至少有两个子结点
2. 每个中间结点至少有ceil(m/2)(ceil是向上取整)个孩子，最多有m个孩子
3. 所有的叶子结点都位于同一层，以保证平衡性
4. 每个结点的元素从小到大排列
5. 内结点不存储数据，只存储键值和指针，叶子结点不存储指针。

注意：这里的孩子和数据结构中二叉树的孩子不是一个概念。



根结点中包含了3个孩子，15,56,77以及中间的两个白色指针。最下层的叶子结点中，存储了键值和数据部分。

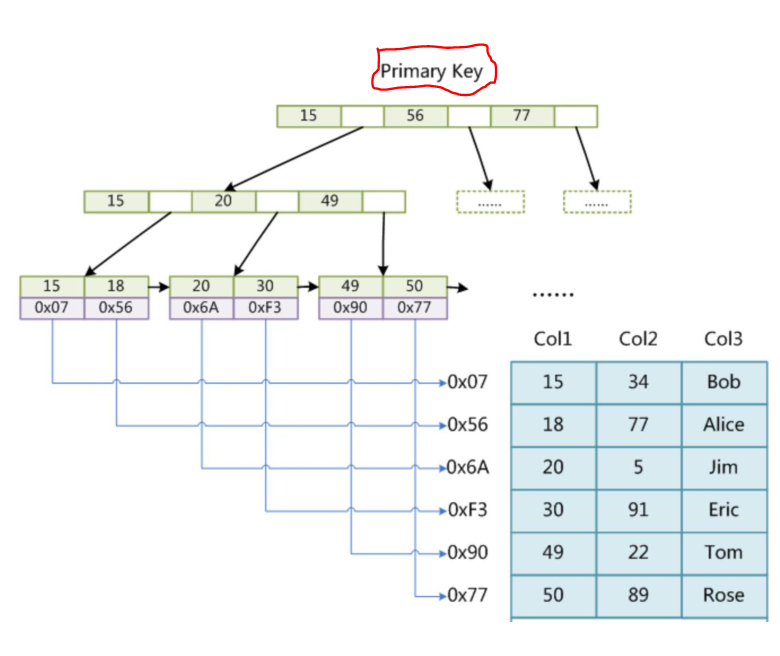
查找18-30的数据，从根结点出发，18在15-56之间，所以往根结点的子结点(15,20,49)这个结点，18在15-20之间，往(15,18)这个叶子结点，查找到18，MySQL中增加了顺序访问指针，就不用再从根结点从发取查找30了，直接从(15,18)的叶子结点往其右边的(20,30)进行查找。

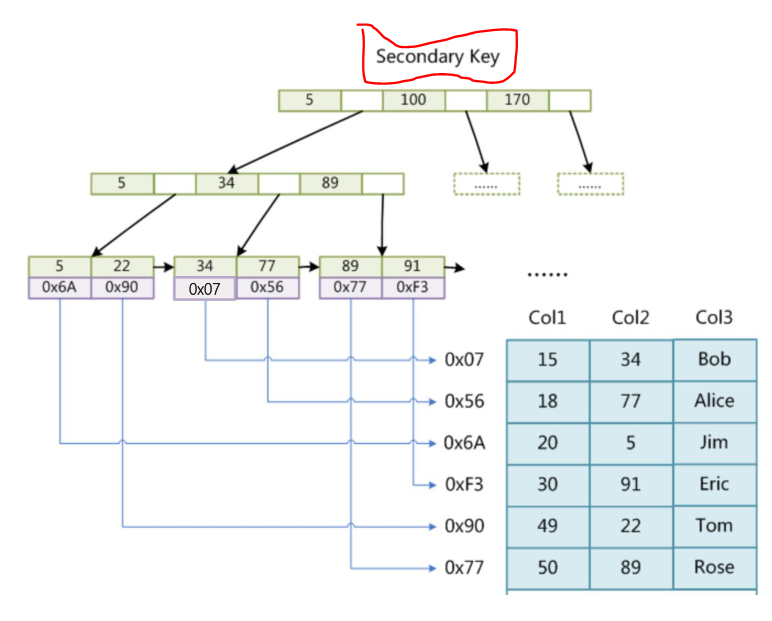
MySQL中，索引属于存储引擎级别的概念，不同存储引擎对索引的实现方式是不同的。常用的有以下两种，MyISAM和InnoDB。



MyISAM引擎使用B+树作为索引结构，叶结点的数据域存放的是实际数据记录的地址，没有把实际数据聚集在索引上，称为非聚集索引。

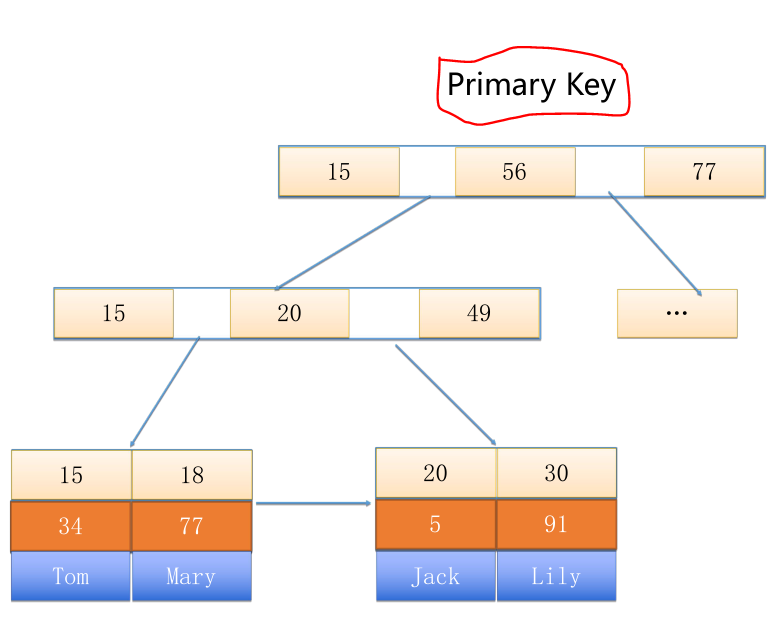
在MyISAM中，主键索引(Primary Key)和辅助索引(Secondary key)在结构上没有差别。主键索引是唯一的，辅助索引可以重复，可以没有主键索引。



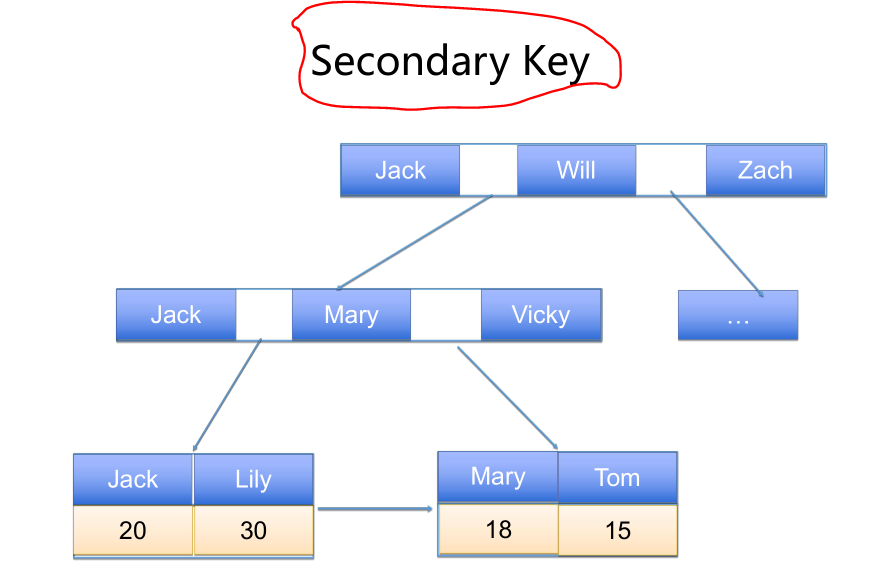


InnoDB：聚集索引，InnoDB的数据文件本身就是索引文件。树的叶结点数据域保存了完整的数据。索引的主键就是数据表的主键，所以必须有主键索引。如果没有指定主键，InnoDB会采用下面两种方式设置主键：

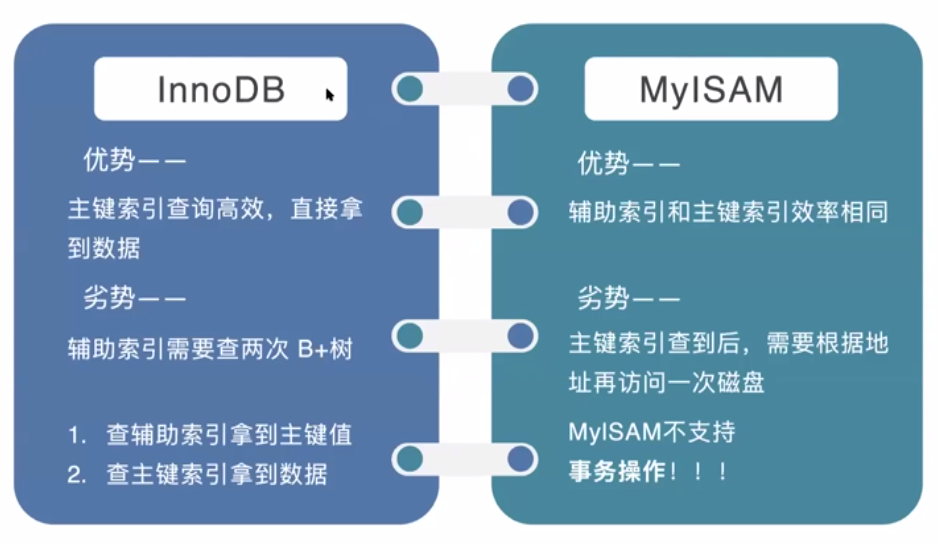
1. 自动选择表中可以唯一标识数据记录的列作为主键
2. 自动生成一个隐含字段作为主键，这个字段长度为6个字节，类型为长整型。



InnoDB中，辅助索引存储的是主键的值，而不是地址。如果通过辅助索引查询时，先查看对应的主键，然后还需要通过主键查找对应的数据，其实需要查找两次B+树。



InnoDB和MyISAM的比较：



InnoDB主键的选择：

不要使用过长的字段，如很长的字符串（UUID，描述性的字符串）

不要使用非单调的字段作为主键，如随机的字符串，随机的数字，非单调的主键使得在插入新数据时，数据库文件为了维持B+树的特性频繁的分裂调整。



索引带来的问题：

1. 索引文件本身消耗空间
2. 插入、更新、删除数据时，维护索引结构，增加负担
3. MySQL自身运行时也要消耗资源维持索引，例如预加载部分索引到内存中，如果索引过多就会大量消耗内存

不建议加索引的列：

1. 不太会查询到的列，例如全省高考的数据表，不太可能会根据地址查询，地址列就不需要建索引
2. 本身没有顺序的列，或者特别的列，学号，姓名列有顺序，但地址列没有顺序
3. 表记录比较少，就不用建索引了。例如建一个班级表，不太需要索引，表中数据条数如果没有到10000级，就不需要索引。
4. 选择性低，即重复率高，可枚举的列，不需要索引，例如性别。

