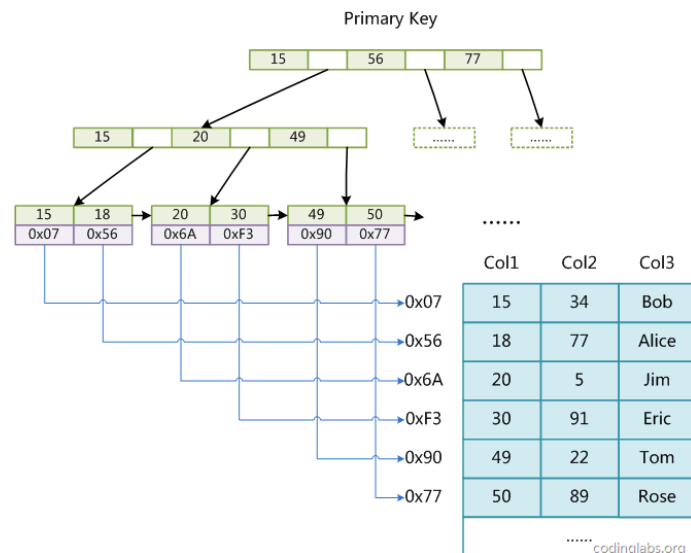


MyISAM 和 InnoDB 索引引擎的 B+树索引实现

MyISAM 索引实现

1) 主键索引:

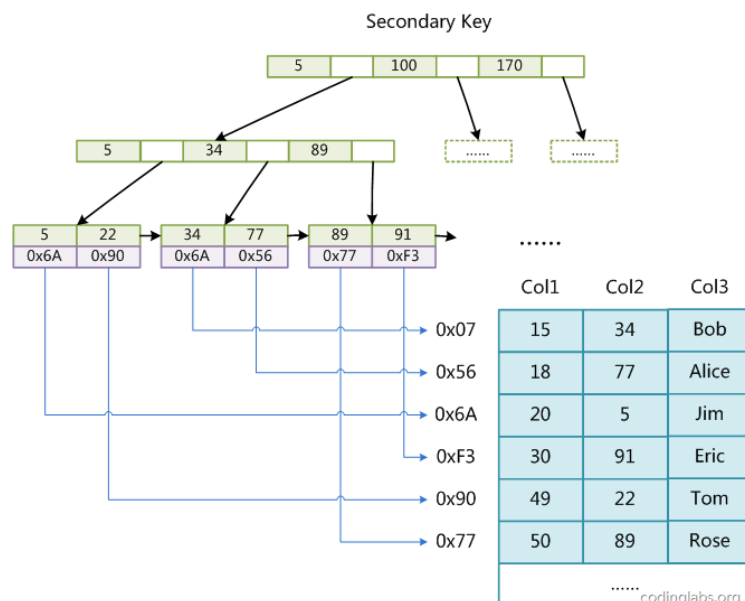
MyISAM 引擎使用 B+Tree 作为索引结构，叶节点的 data 域存放的是数据记录的地址。
下图是 MyISAM 主键索引的原理图：



这里设表一共有三列，假设我们以 Col1 为主键，图 myisam1 是一个 MyISAM 表的主索引 (Primary key) 示意。可以看出 MyISAM 的索引文件仅仅保存数据记录的地址。

2) 辅助索引 (Secondary key)

在 MyISAM 中，主索引和辅助索引 (Secondary key) 在结构上没有任何区别，只是主索引要求 key 是唯一的，而辅助索引的 key 可以重复。如果我们在 Col2 上建立一个辅助索引，则此索引的结构如下图所示：



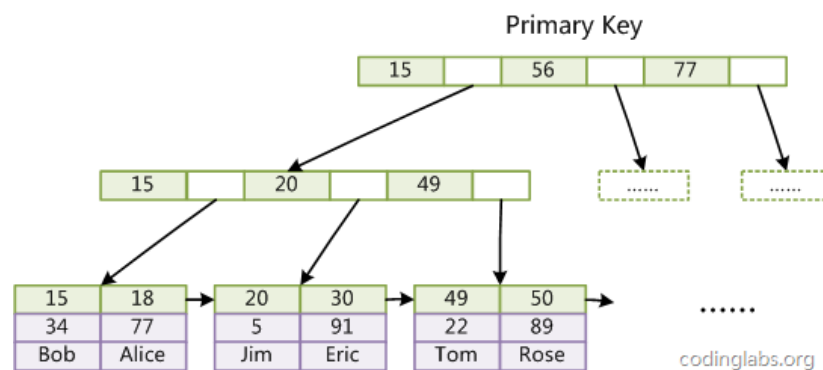
同样也是一颗 B+Tree，data 域保存数据记录的地址。因此，MyISAM 中索引检索的算法为首先按照 B+Tree 搜索算法搜索索引，如果指定的 Key 存在，则取出其 data 域的值，然后以 data 域的值作为地址，读取相应数据记录。

MyISAM 的索引方式也叫做“非聚集”的，之所以这么称呼是为了与 InnoDB 的聚集索引区分。

InnoDB 索引实现

1) 主键索引：

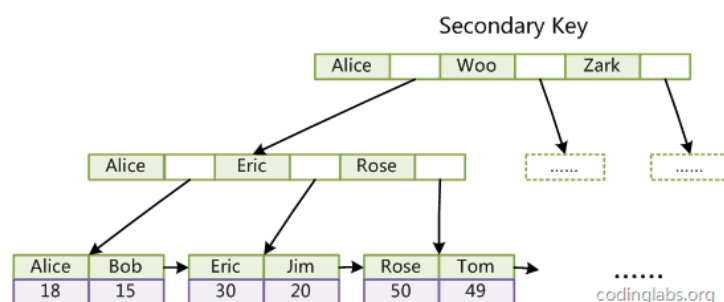
MyISAM 索引文件和数据文件是分离的，索引文件仅保存数据记录的地址。而在 InnoDB 中，表数据文件本身就是按 B+Tree 组织的一个索引结构，这棵树的叶节点 data 域保存了完整的数据记录。这个索引的 key 是数据表的主键，因此 InnoDB 表数据文件本身就是主索引。



(图 innodb 主键索引) 是 InnoDB 主索引 (同时也是数据文件) 的示意图，可以看到叶节点包含了完整的数据记录。这种索引叫做聚集索引。因为 InnoDB 的数据文件本身要按主键聚集，所以 InnoDB 要求表必须有主键 (MyISAM 可以没有)，如果没有显式指定，则 MySQL 系统会自动选择一个可以唯一标识数据记录的列作为主键，如果不存在这种列，则 MySQL 自动为 InnoDB 表生成一个隐含字段作为主键，这个字段长度为 6 个字节，类型为长整形。

2) . InnoDB 的辅助索引

InnoDB 的所有辅助索引都引用主键作为 data 域。例如，下图为定义在 Col3 上的一个辅助索引：



InnoDB 表是基于聚集索引建立的。因此 InnoDB 的索引能提供一种非常快速的主键查找性能。不过，它的辅助索引 (Secondary Index，也就是非主键索引) 也会包含主键列，所以，如果主键定义得比较大，其他索引也将很大。如果想在表上定义、很多索引，则争取尽量把主键定义得小一些。InnoDB 不会压缩索引。

文字的 ASCII 码作为比较准则。**聚集索引**这种实现方式使得按主键的搜索十分高效，但是**辅助索引搜索需要检索两遍索引：首先检索辅助索引获得主键，然后用主键到主索引中检索获得记录。**

不同存储引擎的索引实现方式对于正确使用和优化索引都非常有帮助，例如知道了 InnoDB 的索引实现后，就很容易明白为什么不建议使用过长的字段作为主键，因为所有辅助索引都引用主索引，过长的主索引会令辅助索引变得过大。再例如，用非单调的字段作为主键在 InnoDB 中不是个好主意，因为 InnoDB 数据文件本身是一颗 B+Tree，非单调的主键会造成在插入新记录时数据文件为了维持 B+Tree 的特性而频繁的分裂调整，十分低效，而使用自增字段作为主键则是一个很好的选择。

InnoDB 索引和 MyISAM 索引的区别：

一是主索引的区别，InnoDB 的数据文件本身就是索引文件。而 MyISAM 的索引和数据是分开的。

二是辅助索引的区别：InnoDB 的辅助索引 data 域存储相应记录主键的值而不是地址。而 MyISAM 的辅助索引和主索引没有多大