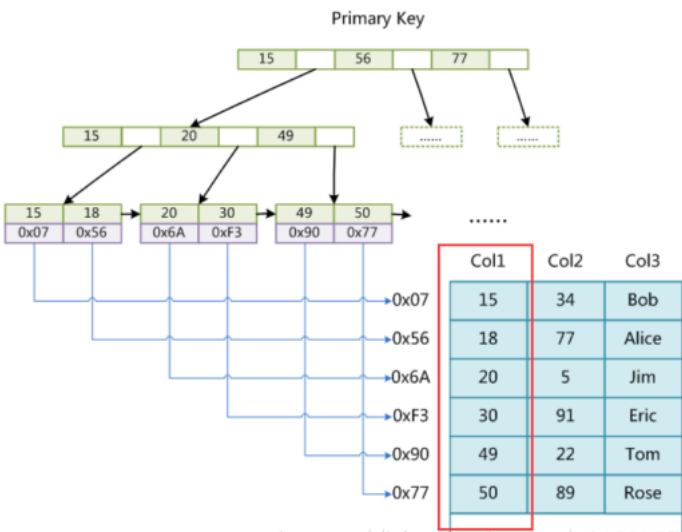


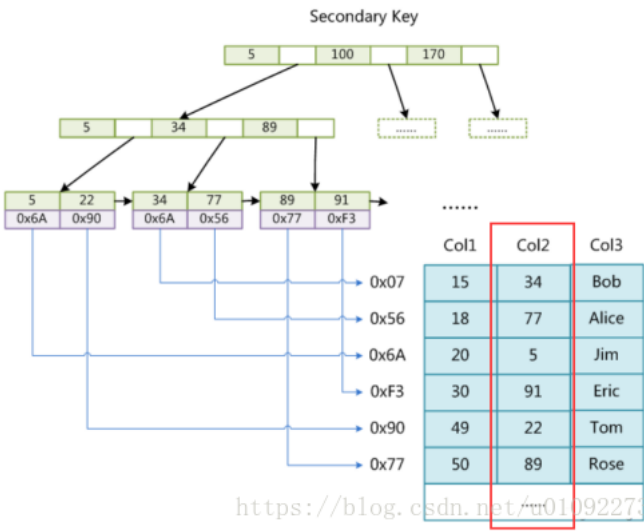
MyISAM

- 主键索引（引擎的索引文件和数据文件是分离的）



(图片来自: <https://blog.csdn.net/u010922732/article/details/82994253>)

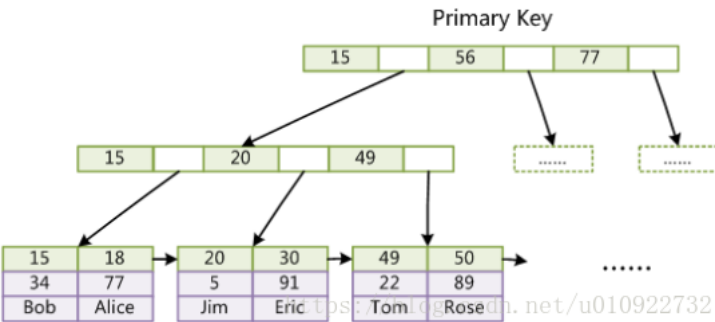
- 非主键索引（索引结构跟上面的主键索引的结构是相同的。）



(图片来自: <https://blog.csdn.net/u010922732/article/details/82994253>)

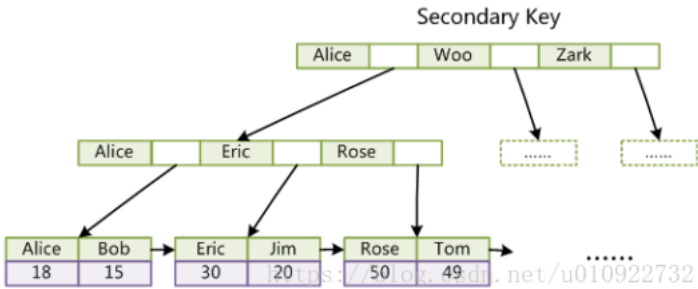
InnoDB:

- 主键索引（索引结构是在同一个树节点中同时存放索引和数据）



(图片来自: <https://blog.csdn.net/u010922732/article/details/82994253>)

- 非主键索引（在最底层的叶子结点有两行数据，第一行的字符串是辅助索引，按照ASCII码进行排序，第二行的整数是主键的值）



- InnoDB索引实现（聚集）
 - 表数据文件本身就是按B+Tree组织的一个索引结构文件
 - 聚集索引 - 叶子节点包含了完整的数据记录
 - 为什么InnoDB表必须有主键，并且推荐使用整型的自增主键？
 - 为什么非主键索引结果叶子节点存储的是主键值？（一致性和节省存储空间）

- 总结一下：InnoDB和MyISAM的区别：

- 1》InnoDB 支持事务，MyISAM 不支持事务。这是 MySQL 将默认存储引擎从 MyISAM 变成 InnoDB 的重要原因之一；
- 2》InnoDB 支持外键，而MyISAM 不支持外键。对一个包含外键的 InnoDB 表转为 MYISAM 会失败；
- 3》InnoDB 是聚集索引，MyISAM 是非聚集索引。聚簇索引的文件存放在主键索引的叶子节点上，因此 InnoDB 必须要有主键，通过主键索引效率很高。但是辅助索引需要两次查询，先查询到主键查询到数据。因此，主键不应该过大，因为主键太大，其他索引也都会很大。而 MyISAM 是非聚集索引，数据文件是分离的，索引保存的是数据文件的指针。主键索引和辅助索引是独立的。
- 4》InnoDB 不保存表的具体行数，执行 select count(*) from table 时需要全表扫描。而MyISAM 用一个变量保存了整个表的行数，执行上述语句时只需要读出该变量即可，速度很快；
- 5》InnoDB 最小的锁粒度是行锁，MyISAM 最小的锁粒度是表锁。一个更新语句会锁住整张表，导致其他查询和更新都会被阻塞，因此并发访问受限。这也是 MySQL 将默认存储引擎从 MyISAM ；要原因之一；