课堂主题

Mysql架构、索引介绍及原理

课堂目标

掌握Mysql的各组件及各组件的功能

理解Mysql简版执行流程和详细执行流程

掌握MyIsam和InnoDB的区别并说明使用场景

掌握Mysql日志文件及主要日志文件的作用

理解Mysql的数据文件及作用

使用命令查看mysql日志

配置my.cnf开启二进制日志、通用查询日志、慢查询日志等

掌握索引、分类、优劣势

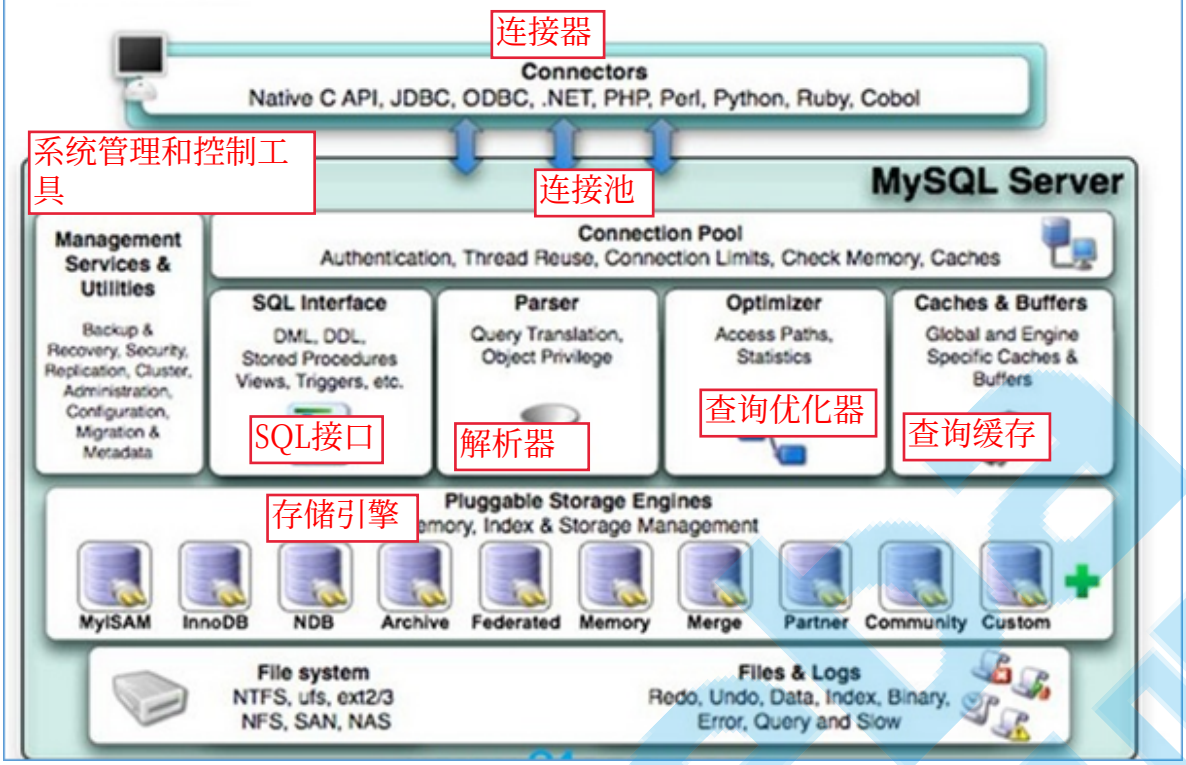
使用命令创建、查看、删除索引

理解索引的原理和存储结构

一、MySQL架构篇

逻辑架构

逻辑架构图



连接器（Connectors）

系统管理和控制工具（Management Serveices & Utilities）

连接池（Connection Pool）

SQL接口（SQL Interface）

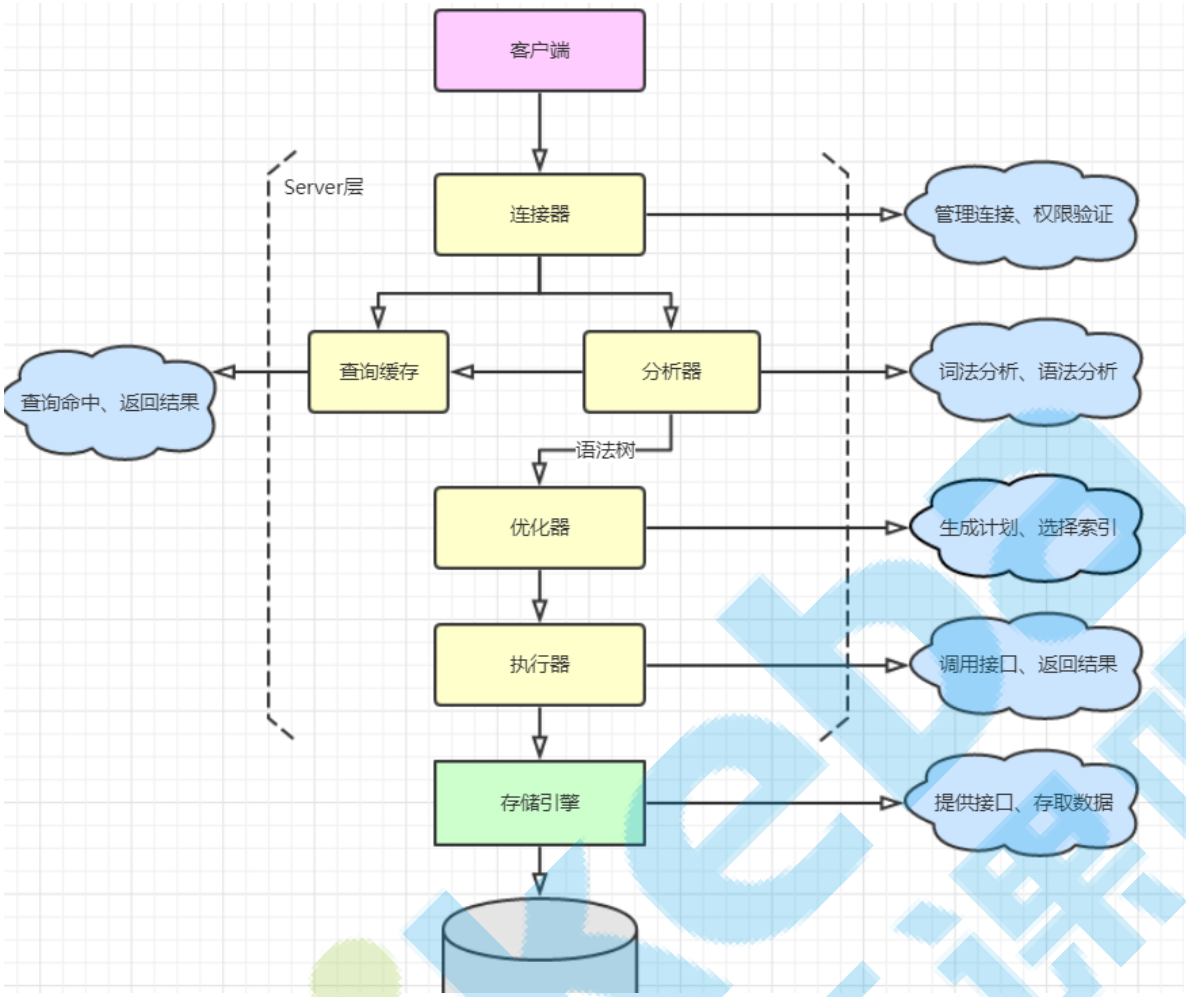
解析器（Parser）

查询优化器（Optimizer）

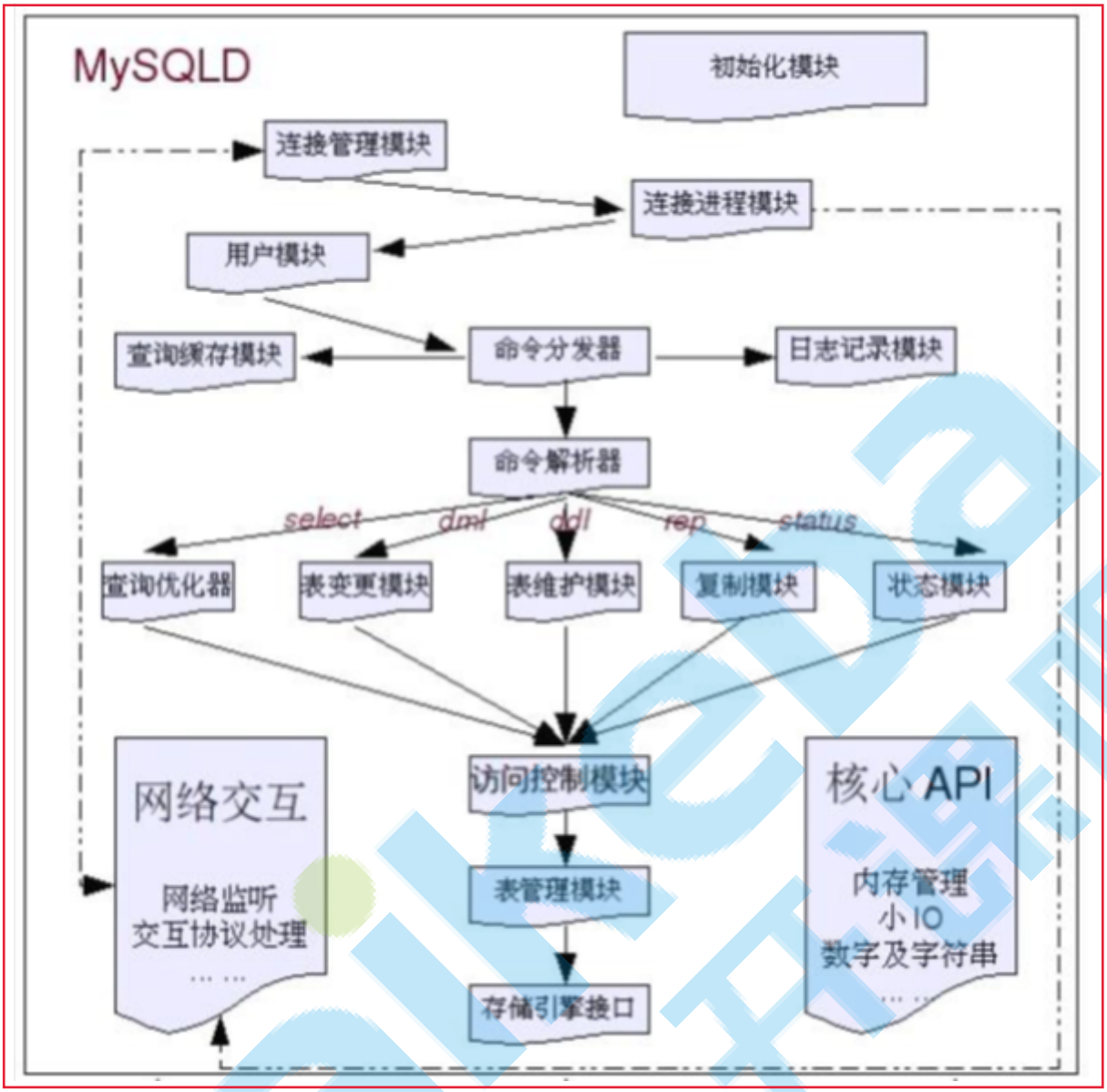
查询缓存（Cache和Buffer）

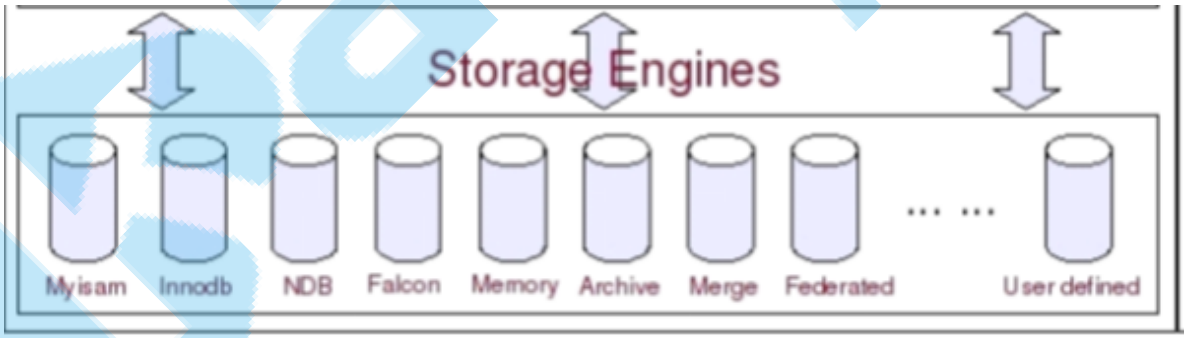
存储引擎（Pluggable Storage Engines）

简版执行流程图


客户端
Server层
连接器
管理连接、权限验证
查询缓存
分析器
查询命中、返回结果
词法分析、语法分析
语法树
优化器
生成计划、选择索引
执行器
调用接口、返回结果
存储引擎
提供接口、存取数据

详细执行流程图





物理结构

- MySQL是通过文件系统对数据和索引进行存储的。
- MySQL从物理结构上可以分为日志文件和数据索引文件。
- MySQL在Linux中的数据索引文件和日志文件都在/var/lib/mysql目录下。
- 日志文件采用顺序IO方式存储、数据文件采用随机IO方式存储。

日志文件

错误日志（errorlog）

二进制日志（bin log）

通用查询日志（general query log）

慢查询日志（slow query log）

重做日志（redo log）

回滚日志（undo log）

中继日志（relay log）

数据文件

InnoDB数据文件

- **.frm文件**：主要存放与表相关的数据信息,主要包括**表结构的定义信息**
- **.ibd**：使用**独享表空间**存储**表数据和索引**信息，一张表对应一个ibd文件。
- **ibdata文件**：使用**共享表空间**存储**表数据和索引**信息，所有表共同使用一个或者多个ibdata文件。

Mylsam数据文件

- **.frm文件**：主要存放与表相关的数据信息,主要包括**表结构的定义信息**
- **.myd文件**：主要用来存储**表数据信息。**
- **.myi文件**：主要用来存储**表数据文件中任何索引的数据树。**

二、MySQL索引篇

索引介绍

索引是什么

索引的优势和劣势

索引的分类

单列索引

组合索引

全文索引

空间索引

位图索引

索引的使用

创建索引

- 单列索引之普通索引

```
CREATE INDEX index_name ON table(column(length)) ;
ALTER TABLE table_name ADD INDEX index_name (column(length)) ;
```

- 单列索引之唯一索引

```
CREATE UNIQUE INDEX index_name ON table(column(length)) ;
alter table table_name add unique index index_name(column);
```

- 单列索引之全文索引

```
CREATE FULLTEXT INDEX index_name ON table(column(length)) ;
alter table table_name add fulltext index_name(column)
```

- 组合索引

```
ALTER TABLE article ADD INDEX index_titme_time (title(50),time(10)) ;
```

删除索引

```
DROP INDEX index_name ON table
```

查看索引

```
SHOW INDEX FROM table_name \G
```

索引原理分析

索引的存储结构

索引存储结构

- 索引是在**存储引擎中实现**的，也就是说不同的存储引擎，会使用不同的索引
- **MyISAM和InnoDB存储引擎**：只支持**B+ TREE索引**， 也就是说**默认使用BTREE，不能够更换**
- MEMORY/HEAP存储引擎：支持HASH和BTREE索引

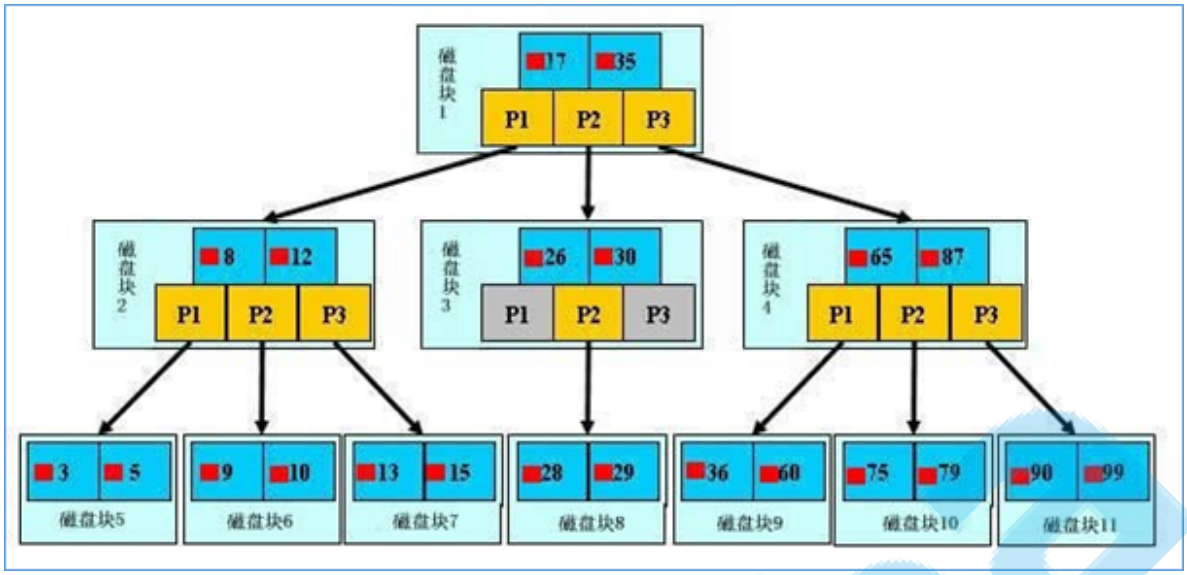
B树和B+树

数据结构示例网站：

https://www.cs.usfca.edu/~galles/visualization/Algorithms.html

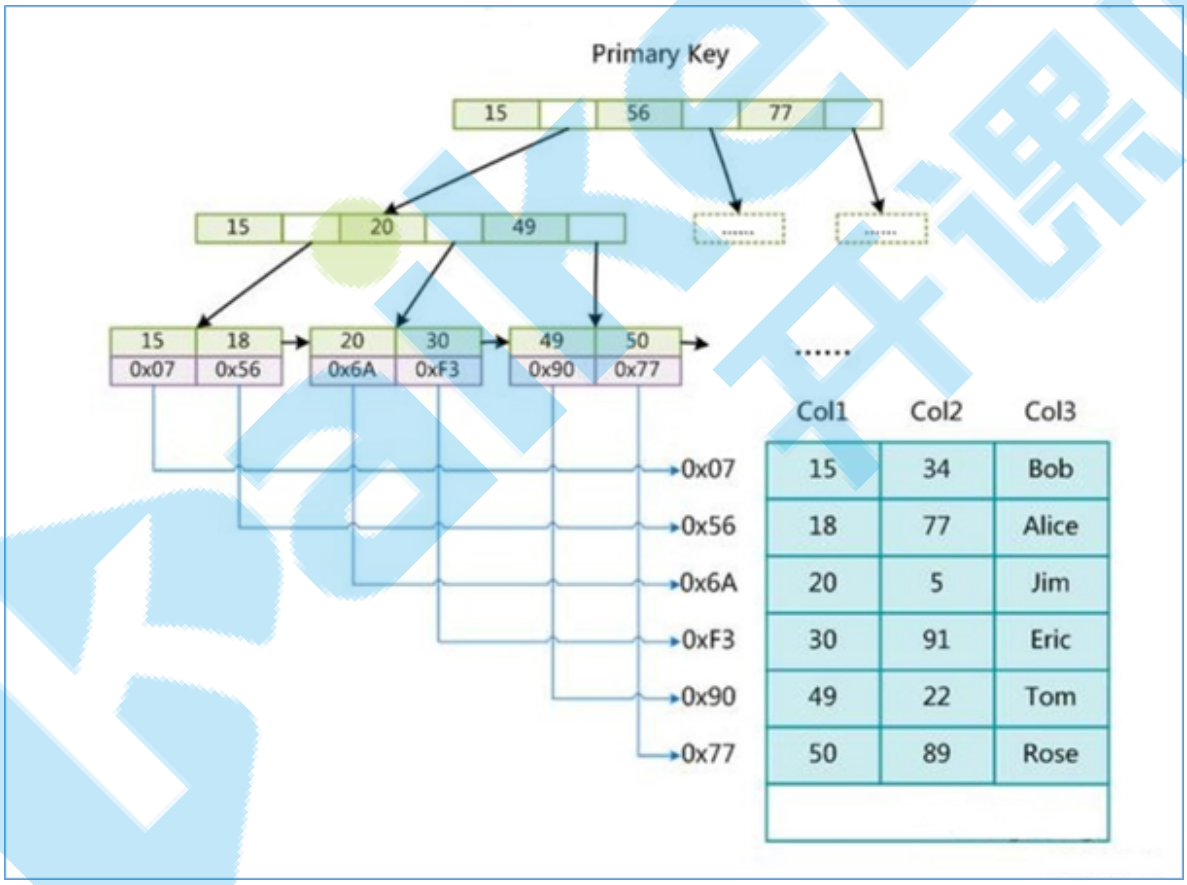
B树图示

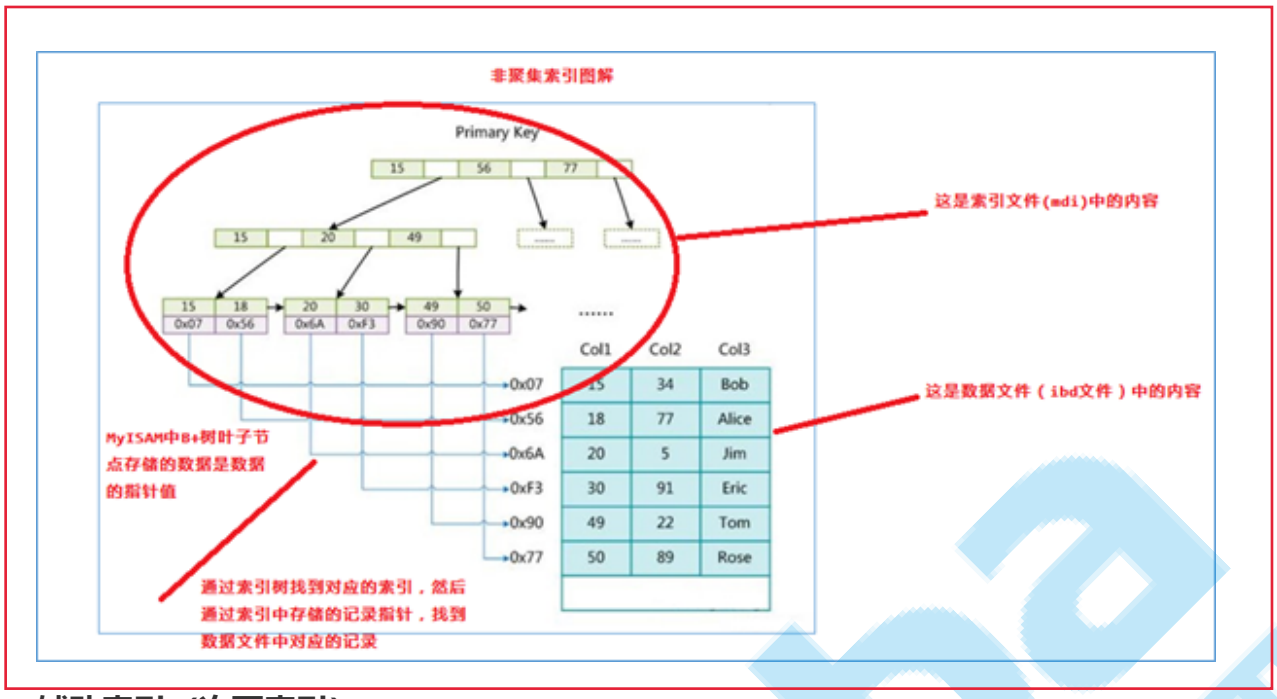
B树是为了磁盘或其它存储设备而设计的一种多叉（下面你会看到，相对于二叉，B树每个内结点有多个分支，即多叉）平衡查找树。 多叉平衡



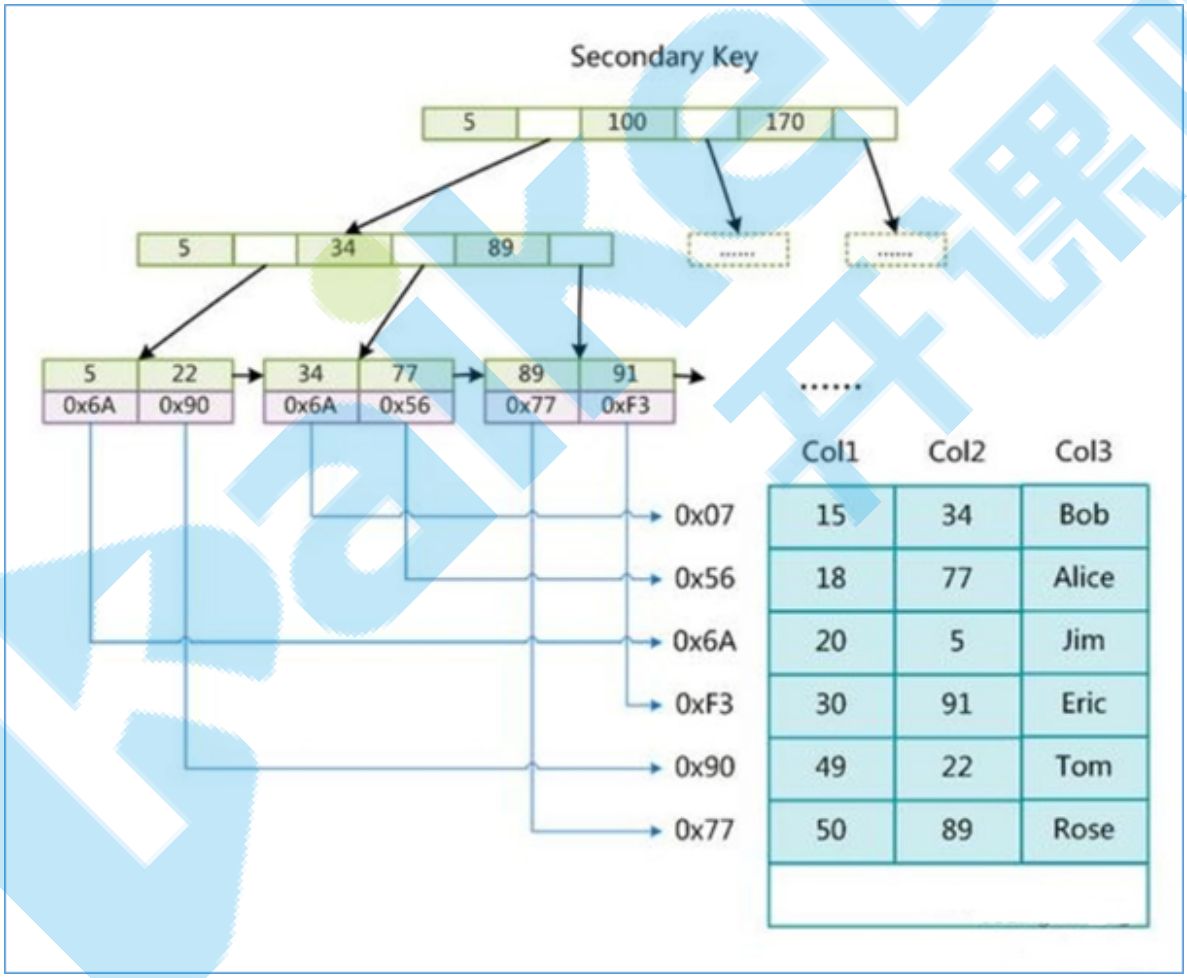
非聚集索引（MyISAM）

主键索引



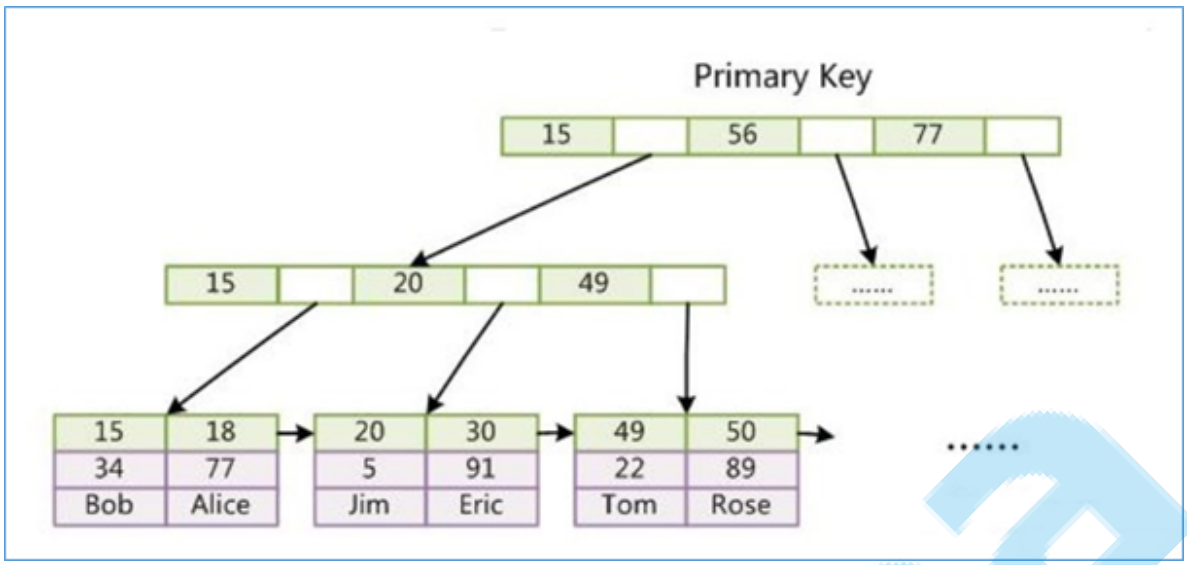


辅助索引（次要索引）

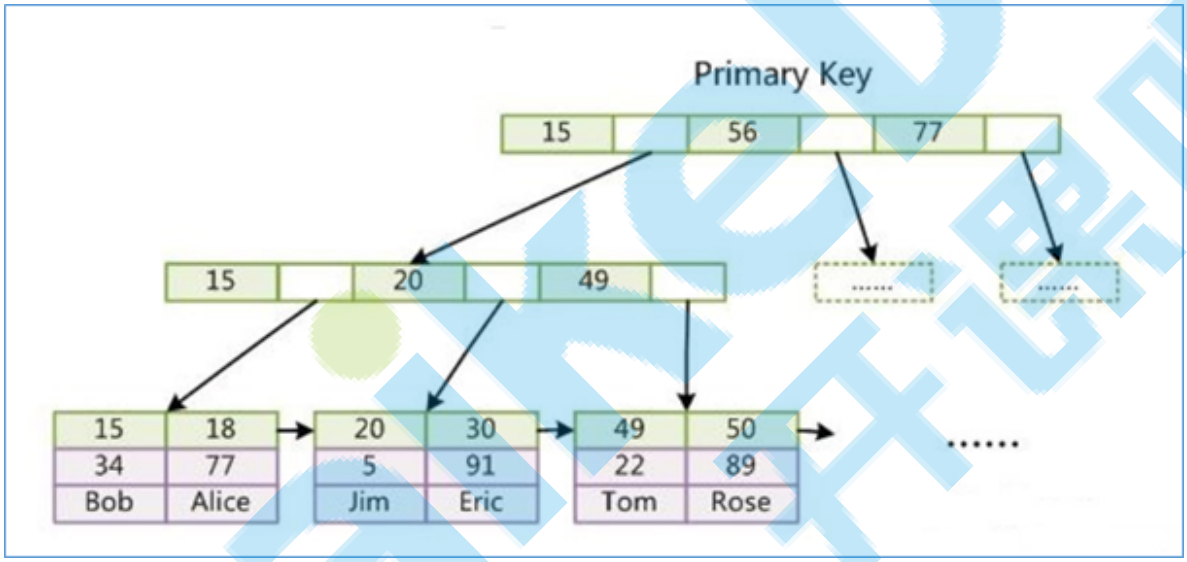


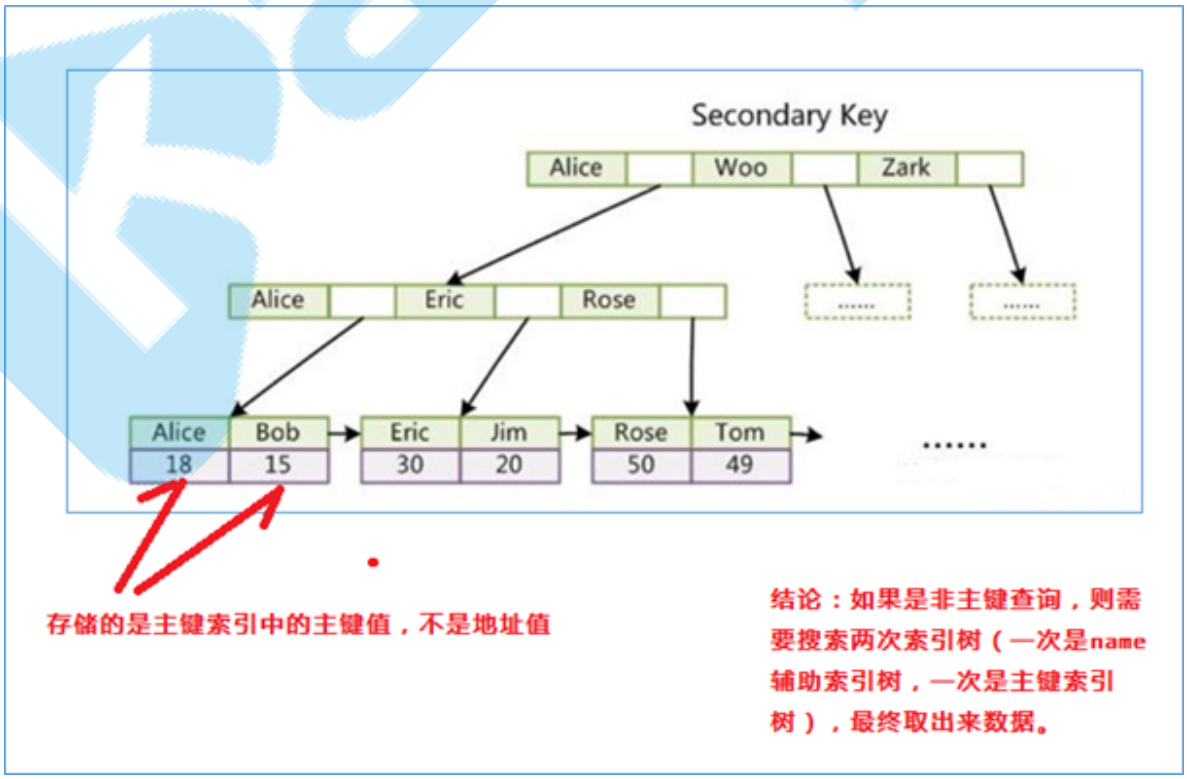
聚集索引（InnoDB）

主键索引


Primary Key
15
56
77
15
20
49
.....
.....
15
18
34
77
Bob
Alice
20
30
5
91
Jim
Eric
49
50
22
89
Tom
Rose
.....

辅助索引（次要索引）


Primary Key
15
56
77
15
20
49
.....
.....
15
18
34
77
Bob
Alice
20
30
5
91
Jim
Eric
49
50
22
89
Tom
Rose
.....


Secondary Key
Alice
Woo
Zark
Alice
Eric
Rose
.....
.....
Alice
Bob
18
15
Eric
Jim
30
20
Rose
Tom
50
49
.....
存储的是主键索引中的主键值，不是地址值
结论：如果是非主键查询，则需要搜索两次索引树（一次是name辅助索引树，一次是主键索引树），最终取出来数据。