MySql索引底层知识图谱：

索引的本质：

什么叫索引：

索引就是帮助MySQL高效获取数据的一种排好序的数据结构。

查询高效、有序的数据结构

先讲个知识点，就是计算机平时是什么查询数据的：

要知道我们的数据都是存储在磁盘上的：

磁盘内部结构，以及读取原理

磁盘有几个重要组成部分：盘片（磁道）（顺时针逆时针旋转）、磁柱（磁头）（只能左右动）

查询就是通过磁柱的磁头去盘片的磁道上查询，一般来说，查询一个数据就相当于一次**I/O**操作。

由此可见正常磁盘读取数据是很慢的，中间有磁头的寻道时间，和盘片的旋转时间

假如MySQL 按照正常的数据结构去存储数据，那查找第七行数据就要进行七次**I/O**,那假如一张表数据有几十万条或者几百万条，那查询速度可想而知有多慢了。

所以MySQL 底层存储数据就用到了不一般数据结构。

说MySQL 之前补充一个知识点：、

MySQL 的表存储引擎有MyiSam（他是非聚集的，表数据文件和索引数据文件是分开的）、InnoDB（他是聚集的表数据文件和表索引文件是在一起的，下面会说到的）

当然了要知道存储引擎是针对于表的。

一般使用磁盘的**I/O**次数来评价索引结构的优劣：

讲几个数据结构：

二叉树：

结构同名字就能看出来：

从根节点按照从小到大排序

缺点：如果是数据的大小都是偏左或者都是偏右的，查询效率降很低

红黑树：

结构和二叉树差不多，在其基础上加了平衡树的作用，如当数据偏右时，三个节点又会分叉，不会像二叉树一样一直到底，起到了平衡的作用

优点：继承了二叉树的排序特点、增加了平衡树的概念

缺点：还是一样，当数据很多时树会很高，查询效率还是不高

BTree

结构在红黑树的基础上让一个节点能够存储多条索引，从而降低树的高度

优点：一个节点可以存储多条索引，降低了树的高度，也就降低了**I/O**的次数，

缺点：还不够完美，比如当要求查询索引a>20的数据，那每次查完一个还得 在从根节点往下查找，增加了**I/O**的次数。

B+Tree

结构和BTree类似，在其基础上，在每个叶子节点之间增加了指针，这样一来当查询a>20的数据时，就可以一次性的将所有的节点查询出来，从而达到了高速查询的标准，一般来说，平时我们的表数据如果加索引只会经过1~3次**I/O**。

接下来说下MySQL两个存储引擎用的数据结构：

首先说下

MyiSam他的表文件会在磁盘生成三个，分别是frm（表结构文件）

、MYD（表数据文件）、MYI（表索引文件）。

存储的数据结构为B+Tree，叶子节点和非叶子节点都会有索引标识、但是在叶子节点会存储关于索引行的数据指针，这样做的好处也是为了让非叶子节点能存储更多的索引，查询的时候先在索引表查询出该索引指向的数据地址，然后在去数据文件里查询出数据。

InnoDB：

他的表文件会在磁盘生成两个，分别是frm（表结构文件）

、IBD（表索引和数据都存储于一个文件）。

存储的数据结构也是为B+Tree，叶子节点和非叶子节点都会有索引标识、但是**不同的是在叶子节点会存储关于该索引对应的表的数据**，这样做的好处也是为了让查询操作时直接查询到该索引的数据，而不用再去数据文件中查询，效率更高。

TODO………………

一般来说cpu执行查询操作时，是先将一个节点的数据先load到cpu里，然后再进行查询，load的单位是整数的k,不会存在你要查找1.5k的数据，不管内存还是磁盘的最小存储单元为**页**，一页大小为4k，MySQL的文件页大小为16k，也就是树的节点为16k, 如果按照某种计算的话，这就已经足够千万级别的存储了，所以默认为一页16k.











