今天，我们学习如何通过建立索引来对MySQL进行优化。

(Tips1：在一个数量级非常大的表中查询一条数据，第一次耗时为1.31s，第二次用相同的sql语句进行查询，发现耗时0.00s。这是因为mysql有缓存，如果发现两次sql语句一模一样的话，就会直接从缓存里拿数据)

概念2：B+树(B+的索引虽然稠密，效率高但是B+树并不是聚簇索引)

概念3：聚簇索引，非聚簇索引

比如主键索引，就是聚簇索引，它的索引指针指向的直接就是数据，在物理地址上是相连的。

非聚簇索引就是索引指针指向的数据的地址。

在一张表中，只能存在一个聚簇索引。

索引的分类：

1. 主键索引

就是最普通的索引啦…我们平常都会用到，主键自动添加索引

1. 普通索引

就是自己创建的最简单的索引

1. 全文索引
2. 组合索引

举个例子：把id+name+age+sex组合起来作为索引，就称之为组合索引。**相当于创建了id的单列索引，id+name，id+name+age，id+name+age+sex的组合索引。**

组合索引遵循**最左前缀匹配原则**，什么是**最左前缀匹配原则**？

当你查询语句以id+name+age+sex作为条件的时候，mysql会一直向右匹配直到遇到**范围查询(where in() …order by,>，<，between，like)**的时候，会停止索引的匹配。比如 select \* from tab1 where id = ? and name = ? and age > 10 and sex = 0，本来该是id+name+age+sex都作为索引的，由于范围查询符号”>”的出现，导致只有id+name+age三个作为索引了。

因此，在where子句中，索引的顺序就很重要了。

什么样的sql语句适合建立索引？

1. Where, group by, order by,on中出现的列
2. 选择离散度大(区分度大)的列，越大越适合当索引

利用count(distinct col)/count(\*)来判断离散度的大小