**mysql聚集索引和非聚集索引**

一级索引：聚集索引。叶子结点就存储了数据。innodb存储引擎。

二级索引：非聚集索引。叶节点仍然是[索引节点](http://baike.baidu.com/view/3043965.htm)，并保留一个链接指向对应[数据块](http://baike.baidu.com/view/702806.htm)。MyIsam存储引擎。

**聚集索引的特点：**

聚簇索引主键的插入速度要比非聚簇索引主键的插入速度慢很多。聚簇索引适合排序，聚簇索引本身已经是按照物理顺序放置的，排序很快。当你需要取出一定范围内的数据时，用聚簇索引也比用非聚簇索引好。

**非聚簇索引特点：**

非聚簇索引（也叫二级索引）不适合用在排序的场合。非聚簇索引则没有按序存放，需要额外消耗资源来排序。二级索引需要两次索引查找，而不是一次才能取到数据，因为存储引擎第一次需要通过二级索引找到索引的叶子节点，从而找到数据的存储地址，然后在聚簇索引中用主键再次查找索引，再找到数据。

**聚集索引的优点与缺点：**

**优点：**  
聚簇索引的优点,就是**提高数据访问性能**。聚簇索引把索引和数据都保存到同一棵B+树数据结构中，并且同时将索引列与相关数据行保存在一起。这意味着，当你访问同一数据页不同行记录时，已经把页加载到了Buffer中，再次访问的时候，会在内存中完成访问，不必访问磁盘。不同于MyISAM引擎，它将索引和数据没有放在一块，放在不同的物理文件中，索引文件是缓存在key\_buffer中，索引对应的是磁盘位置，不得不通过磁盘位置访问磁盘数据。  
**缺点：**  
1) **维护索引很昂贵**，特别是插入新行或者主键被更新导至要分页(page split)的时候。建议在大量插入新行后，选在负载较低的时间段，通过OPTIMIZE TABLE优化表，因为必须被移动的行数据可能造成碎片。使用独享表空间可以弱化碎片  
2) 表因为使用UUId作为主键，使数据存储稀疏，这就会出现聚簇索引有可能有比全表扫面更慢，所以建议使用int的auto\_increment作为主键

3) 如果**主键比较大的话，那辅助索引将会变的更**大，因为辅助索引的叶子存储的是主键值；过长的主键值，会导致非叶子节点占用占用更多的物理空间