

聚簇索引：表数据按照索引的顺序来存储的，也就是说索引顺序结构与数据存储物理结构一致。对于聚簇索引，叶子节点即存储了真实的数据行，不再有另外单独的数据页。在一张表上最多只能创建一个聚簇索引，因为真实数据的物理顺序只能有一种。

非聚簇索引：索引顺序结构与数据存储物理结构无关，对于聚簇索引，叶节点包含索引字段值及指向数据页数据行的逻辑指针，其行数量与数据表行数据量一致。

如果一张表中没有聚簇索引，那么它被称为“堆集”。这样的表中的数据行没有特定的顺序，所有的新行将被添加到表的末尾位置。而建立了聚簇索引的数据表则不同：最简单的情况下，插入操作根据索引找到对应的数据页，然后通过挪动已有的记录为新数据腾出空间，最后插入数据。如果数据页已满，则需要拆分数据页，调整指针。

聚簇索引：

1. 一个索引项直接对应实际数据记录的存储页，可谓“直达”
2. 主键缺省使用它
3. 索引项的排序和物理存储项一致，利用这一点，想要修改数据的存储顺序，可以通过改变主键的方法。
4. 一个表只能有一个聚簇索引。

非聚簇索引：

1. 不能“直达”，可链式的访问多级页表后，才能定位到数据页
2. 一个表可以有多个非聚簇索引。

建立索引是为了加快对表中记录的查找或排序（聚簇索引适合排序，非聚簇索引不适合用在排序的场合。因为聚簇索引本身已经是按照物理顺序放置的，排序很快），但是添加索引后，查询速度不一定更快，因为：

1. 增加了数据库的存储空间。
2. 在插入和修改数据时要花费较多的时间。