**索引**

**索引的详解**

1. 索引就是一个表的目录，当我们使用索引的时候，能够大大的提升查询的效率，但是我们不能频繁的乱添加索引，因为可能会造成索引的重建、索引的失效等问题，这样就会降低了效率
2. 索引都单独的存放在一个很小的内存空间里面，是独立的，但是索引都会指向我们表中所对应的数据
3. 当我们需要对数据频繁的进行查询，或者对数据进行筛选的时候就可以考虑用索引，这样能够大大提升效率
4. 所以分为：聚簇索引、非聚簇索引
5. 聚簇索引：

-> 索引的存储顺序和表中的数据的存储数据是一样的就叫做聚簇索引

-> 一个表只能有一个聚簇索引

-> 聚簇索引创建好后会指向表中所对应的数据，他们的存储数据是一致的，并且聚簇索引会自动排好序

-> 聚簇索引最好是日期、自动增长的主键，因为这样方便排序，当然，聚簇索引也能够创建在非主键的列上

1. 非聚簇索引：

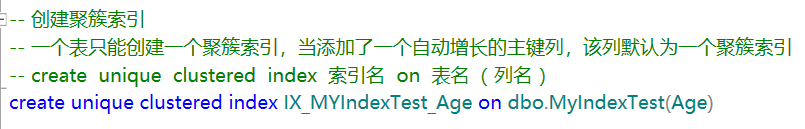
-> 非聚簇索引又分为：唯一非聚簇索引、不唯一非聚簇索引

-> 非聚簇索引的存储数据不一致，有可能是指向的聚簇索引，也有可能是指向的是表中的数据

-> 一个表中能够存在多个非聚簇索引

**索引的创建：**

1. 聚簇索引的创建



1. 唯一非聚簇索引的创建



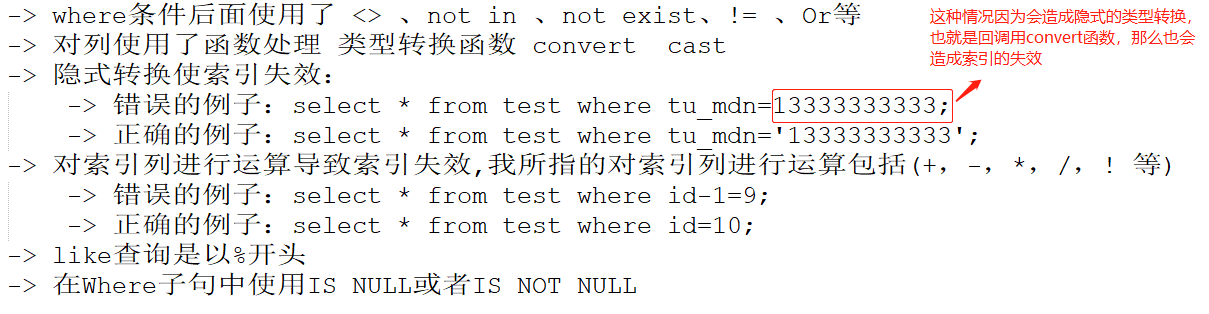
1. 不唯一非聚簇索引的创建



**索引的弊端：索引的重建、索引的失效**

1. **索引的失效**

当我们在SQL语句中满足了某些特定的条件就会造成索引的失效，如下图：



1. **索引的重建**

当我们在一个表中频繁的对数据进行修改、添加、删除等操作的时候，就有可能造成索引的重建的问题，索引当某个表存在频繁的修改、添加、删除操作的时候，我们最好不要创建索引，因为这样有可能会造成索引的重建的问题

什么是索引的重建？

比如说我们有一个表有着索引，该索引对应的ID值为1、2、3、4等，这时候我们删除了ID为2、3的数据，也就是只剩下了ID为1和4的数据了，但这时候我们又添加了一个新的数据进去，并且这个索引塞进了1和4之间，这时候就会造成索引的重建