通俗理解：

利用索引中的附加列，您可以缩小搜索的范围，但使用一个具有两列的索引 不同于使用两个单独的索引。复合索引的结构与电话簿类似，人名由姓和名构成，电话簿首先按姓氏对进行排序，然后按名字对有相同姓氏的人进行排序。如果您知道姓，电话簿将非常有用；如果您知道姓和名，电话簿则更为有用，但如果您只知道名不姓，电话簿将没有用处。

所以说创建复合索引时，应该仔细考虑列的顺序。对索引中的所有列执行搜索或仅对前几列执行搜索时，复合索引非常有用；仅对后面的任意列执行搜索时，复合索引则没有用处。

重点：

多个单列索引在多条件查询时优化器会选择最优索引策略，可能只用一个索引，也可能将多个索引全用上！ 但多个单列索引底层会建立多个B+索引树，比较占用空间，也会浪费一定搜索效率，故如果只有多条件联合查询时最好建联合索引！

最左前缀原则：

顾名思义是最左优先，以最左边的为起点任何连续的索引都能匹配上，

注：如果第一个字段是范围查询需要单独建一个索引

注：在创建联合索引时，要根据业务需求，where子句中使用最频繁的一列放在最左边。这样的话扩展性较好，比如 userid 经常需要作为查询条件，而 mobile 不常常用，则需要把 userid 放在联合索引的第一位置，即最左边

同时存在联合索引和单列索引（字段有重复的），这个时候查询mysql会怎么用索引呢？

这个涉及到mysql本身的查询优化器策略了，当一个表有多条索引可走时, Mysql 根据查询语句的成本来选择走哪条索引；

有人说where查询是按照从左到右的顺序，所以筛选力度大的条件尽量放前面。网上百度过，很多都是这种说法，但是据我研究，mysql执行优化器会对其进行优化，当不考虑索引时，where条件顺序对效率没有影响，真正有影响的是是否用到了索引！

联合索引本质：

当创建\*\*(a,b,c)联合索引时，相当于创建了(a)单列索引\*\*，(a,b)联合索引以及\*\*(a,b,c)联合索引\*\*

想要索引生效的话,只能使用 a和a,b和a,b,c三种组合；当然，我们上面测试过，a,c组合也可以，但实际上只用到了a的索引，c并没有用到！

注：这个可以结合上边的 通俗理解 来思考！

其他知识点：

1、需要加索引的字段，要在where条件中

2、数据量少的字段不需要加索引；因为建索引有一定开销，如果数据量小则没必要建索引（速度反而慢）

3、避免在where子句中使用or来连接条件,因为如果俩个字段中有一个没有索引的话,引擎会放弃索引而产生全表扫描

4、联合索引比对每个列分别建索引更有优势，因为索引建立得越多就越占磁盘空间，在更新数据的时候速度会更慢。另外建立多列索引时，顺序也是需要注意的，应该将严格的索引放在前面，这样筛选的力度会更大，效率更高。