HBase中如果希望按指定的字段和条件进行查询，要想办法把字段放入的Row Key中。从这也能看出，HBase比较适合存储一些较为简单的数据，例如项目中的Friendship和NewsFeed的属性都较少，否则查询时就比较麻烦。

HBase擅长于存储结构简单的海量数据但索引能力有限，而Oracle等传统关系型数据库(RDBMS)能够提供丰富的查询能力，但却疲于应对TB级别的海量数据存储，HBase对传统的RDBMS并不是取代关系，而是一种补充。

HBase使用Row Key对表进行横向切割，将不同的Row Key存储在不同的HRegion上，对HBase进行读写时，都使用Row Key去定位HRegion，进而定位到HRegionServer。在HRegion中有start-key和end-key，定位时就是判断要找的Row Key是否在start-key和end-key之间，由于Row Key是按照字典序排列，如果不对Row Key进行处理，就有可能会带来热点问题。例如：

java3y111

java3y222

java3y333

java3y444

java3y555

aaa

bbb

java3y777

java3y666

以java3yxxxx开头的Row Key很对，其它的Row Key很少，就会导致存储java3yxxx的HRegion上数据量很大，而分配到其它HRegion上的就会很少。

避免热点的方法：

1. 反转Row Key的部分字段，例如项目中对user\_id进行反转并补0
2. 加盐：在Row Key前添加随机的bytes，保证Region之间负载均衡。缺点是查询时，要枚举所有可能的盐作为前缀进行查询，增加了查询时间。

