1. static讲一下，static是线程安全的吗？

答：四种使用场景，1.修饰成员变量和成员方法，属于类，不属于单个类的某个对象，被类中所有对象共享，可以通过类名调用。2.静态代码块，静态代码块定义在类中方法外，静态代码块在非静态代码块之前执行（静态代码块、菲静态代码块、构造方法），不管创建多少对象，静态代码快只执行一次。3.静态内部类，非静态内部类在编译完成之后会隐含地保存着一个引用，改引用是指向创建他的外围类，但是静态内部类就没有，意味着他的创建不需要依赖外围类的创建，他不能使用任何外围类的非静态成员变量和方法。4.静态导包，格式import static，并且不需要使用类名调用类中静态成员，可以直接使用类中静态成员变量和成员方法。

1. HashMap为什么不是线程安全的？
2. AtomicInteger了解吗

并发包java.util.concurrent的原子类都存放在java.util.concurrent.atomic下，原子类具有原子/原子操作特征的类。JUC包中的原子类有哪四个，有基本类型（使用原子方式更新基本类型），数组类型，引用类型，对象的属性修改类型。AtomicInteger整形原子类，常用的方法有get（），getAndSet（），getAndIncrement（）…都被final修饰，使用AtomicInteger之后，不用对increment（）方法加锁也可以保证线程安全。AtomicInteger线程安全原理分析：AtomicInteger类主要利用CAS+volatile和native方法来保证原子操作，从而避免Synchronized的高开销，执行效率大为提升。CAS的原理是拿期望值和原本的值作比较，如果相同更新成新的值。然后本地方式是用Unsafe类的方法来拿到“原来的值”的内存地址，value是一个volatile变量，在内存中可见，因此JVM能保证任何时刻任何线程总能拿到该变量的最新值。

1. 两种存储引擎的主键索引和非主键索引？主键索引叶子节点存储的数据非常大非常多，要爆炸怎么办
2. JVM参数
3. 设计模式了解啥
4. HashMap底层是什么？扩容机制？为什么是2的N次幂
5. 乐观锁和悲观锁？CAS
6. 持久化机制有几种？有什么区别
7. Synchronized主要的三种使用方式：修饰实例方法，修饰静态方法，修饰代码块。

修饰实例方法，作用于当前对象实例加锁，进入同步代码前要获得当前对象实例的锁；修饰静态方法，给当前类加锁，会作用于类的所有对象实例，因为静态成员不属于任何一个实例对象，是类成员；所以如果一个线程A调用一个实例对象的非静态synchronized方法，而线程B需要调用这个实例所属类的静态synchronized方法，是允许的，因为访问静态方法占用的锁是当前类的锁，而访问非静态synchronized方法占用的锁是当前实例对象锁；修饰代码块，指定加锁对象，对给定对象加锁。总结：synchronized关键字加到static静态方法和代码块都是给class类上锁，synchronized关键字加到实例方法上是给对象实例上锁。Synchronized同步语句块的实现使用的是monitorenter和monitorexit指令，线程尝试获取锁就是获取monitor的持有权，当计数器为0则可以成功获取，获取后将锁计数器加1，monitorexi执行后，锁计数器为0.。synchronized修饰方法是使用ACC\_SYNCHRONIZED标识。表明是同步方法。

1. volatile可见性？java内存模型下，线程可以把变量保存在本地内存中，而不是直接从主存中进行读写，所以可能会导致一个线程在主存中修改了一个变量的值，而另外一个线程继续使用寄存器中的变量值的拷贝，造成数据不一致，用volatile，可以保证每次都到主存中进行读取。
2. redis为什么快
3. AQS

AQS的全称叫AbstractQueuedSynchronizer,这个类在java.util.concurrent.locks包下面;用来构建锁和同步器的框架，使用AQS能简单且高效地构造出应用广泛的大量的同步器，比如Reentrantlocl,Semaphore,FutureTask等都是基于AQS的。AQS的核心思想是，如果被请求的共享资源空闲，则将当前请求资源的线程设置为有效的工作线程，并且将共享资源设置为锁定状态。如果被请求的共享资源被占用，那么就需要一套线程阻塞等待以及被唤醒时所分配的机制，这个机制AQS是用CLH队列锁实现的，即将暂时后去不到锁的线程加入到队列中。CLH是一个虚拟的双向队列，即不存在实例，仅存在节点之间的关联关系。AQS是将每条请求共享资源的线程封装成一个CLH队列的一个节点来实现锁的分配。AQS使用一个int state成员变量表示同步状态，且被volatile修饰保证线程可见性，通过CLH内置的队列来完成获取资源线程的排队工作，AQS使用CAS对该同步状态进行原子操作实现对其值的修改，原子的CAS操作将同步状态值设置为给定值update如果当前同步状态的值等于expect(期望值)。

AQS对资源的共享方式：

Exclusive（独占）：只有一个线程能执行，如Reentrantlock

公平锁：按照线程在队列中的排队顺序，先到者先拿到锁

非公平锁：当线程要获取锁时，无视队列顺序直接去抢锁，谁抢到就是谁的。

Share（共享）：多个线程可同时执行，如Semaphore/CountDownLatch,CyclicBarrier, ReadWriteLock..。ReentrantReadWriteLock可以看成是组合式，因为ReentrantReadWriteLock也就是读写锁允许多个线程同时对某一资源进行读。自定义同步器在实现时只需要实现共享资源state的获取与释放方式即可。

AQS底层使用了模板方法模式，如果需要自定义同步器一般需要：1.使用者继承AbstractQueuedSynchronizer并重写指定的方法，即对于共享资源state的获取和释放。2.将AQS组合在自定义同步组件的实现中，并调用其模板方法，而这些模板方法会调用使用者重写的方法。

1. explain查看是否使用索引？

使用简单，把explain写在select前面就可以知道有没有用到索引了，主要看type类型就可以，如果type是all，那说明没有用到索引，而是遍历了所有行，用到索引的话，有system,const,eq\_ref,ref,fulltext,ref\_or\_null,unique\_subquery,index\_subquery,range,index\_merge,index.。除了index\_merge之外其他的type类型只可以使用一个索引。

还有key字段是真正用到的索引。