# 贝若偲java软件工程师笔试题

## 数据库

1. 查询语句不同元素（where、join、limit、group by、having等等）执行先后顺序？

.查询中用到的关键词主要包含六个，并且他们的顺序依次为 select--from--where--group by--having--order by

其中select和from是必须的，其他关键词是可选的，这六个关键词的执行顺序 与sql语句的书写顺序并不是一样的，而是按照下面的顺序来执行

from:需要从哪个数据表检索数据

where:过滤表中数据的条件

group by:如何将上面过滤出的数据分组

having:对上面已经分组的数据进行过滤的条件

select:查看结果集中的哪个列，或列的计算结果

order by :按照什么样的顺序来查看返回的数据

参考：<https://blog.csdn.net/wozhengtao/article/details/84946672>

1. 数据库的优化（从sql语句优化和索引两个部分回答）

参考： <https://blog.csdn.net/u011225629/article/details/50492403/>

1. Mybatis一对多关联查询如何实现？

参考： https://blog.csdn.net/abc5232033/article/details/79054247

1. Po,do,vo,dto理解，设计？

## Java

1. HashMap和HashTable的区别？

HashMap是非线程安全的，是因为HashMap的方法都是没有用synchronized关键字修饰的。

HashMap的key-value支持key-value，null-null，key-null，null-value四种。而Hashtable只支持key-value一种（即

key和value都不为null这种形式）。既然HashMap支持带有null的形式，那么在HashMap中不能由get()方法来判断

HashMap中是否存在某个键， 而应该用containsKey()方法来判断，因为使用get的时候，当返回null时，你无法判断到底

是不存在这个key，还是这个key就是null，还是key存在但value是null。

4、线程安全性不同，HashMap的方法都没有使用synchronized关键字修饰，都是非线程安全的，而Hashtable的方法几乎

都是被synchronized关键字修饰的。但是，当我们需要HashMap是线程安全的时，怎么办呢？我们可以通过Collections.synchronizedMap(hashMap)来进行处理，亦或者我们使用线程安全的ConcurrentHashMap。ConcurrentHashMap虽然也是线程安全的，但是它的效率比Hashtable要高好多倍。因为ConcurrentHashMap使用了分段锁，并不对整个数据进行锁定。

5、初始容量大小和每次扩充容量大小的不同

Hashtable默认的初始大小为11，之后每次扩充，容量变为原来的2n+1。HashMap默认的初始化大小为16。之后每次扩充，容量变为原来的2倍。

1. 如何保证多线程下 i++ 结果正确？

参考： <https://blog.csdn.net/hefenglian/article/details/82586243>

解决方法

1、 使用循环CAS，实现i++原子操作

Java从JDK1.5开始提供了java.util.concurrent.atomic包来提供线程安全的原子操作类。这些原子操作类都是是用CAS来实现，i++的原子性操作。以AtomicInteger为例子，讲一下 public final int getAndIncrement(){} 方法的实现。

Atomic包参考 <https://www.cnblogs.com/chenpi/p/5375805.html>

public final int getAndIncrement() {

for (;;) {

int current = get();

int next = current + 1;

if (compareAndSet(current, next))

return current;

}

}

2、使用锁机制，实现i++原子操作

// 使用Lock实现，多线程的数据同步

public static ReentrantLock lock = new ReentrantLock();

1

2

3、使用synchronized，实现i++原子操作

for (int i = 0; i < times; i++) {

// 进行自加的操作

synchronized (SynchronizedTest.class) {

count++;

}

}

1. Spring IOC和DI是什么？

依赖注入，在容器运行的时候，扫描所有的依赖关系，并为之动态的注入对应的依赖关系

Ioc容器就是为我们提供一个公共的管理平台，管理我们对象之间的依赖关系，我们需要什么就向容器要，容器之中如果要就会给我们，我们拿过来直接用，而不用去考虑对象什么时候创建，什么时候销毁，只管用就行了

1. Spring MVC 的工作原理？

### SpringMVC流程

第1步：浏览器发送指定的请求都会交给DispatcherServlet,他会委托其他模块进行真正的业务和数据处理

第2步：DispatcherServlet会查找到HandleMapping,根据浏览器的请求找到对应的Controller，并将请求交给目标Controller

第3步：目标Controller处理完业务后，返回一个ModelAndView给DispatcherServlet

第4步：DispatcherServlet通过ViewResolver视图解析器找到对应的视图对象View

第5步：视图对象View负责渲染，并返回到浏览器

1. Spring AOP实现原理?

AOP（Aspect Orient Programming），我们一般称为面向方面（切面）编程，作为面向对象的一种补充，用于处理系统中分布于各个模块的横切关注点，比如事务管理、日志、缓存等等。AOP实现的关键在于AOP框架自动创建的AOP代理，AOP代理主要分为静态代理和动态代理，静态代理的代表为AspectJ；而动态代理则以Spring AOP为代表。

1. 知道有哪些设计模式，原理是什么？