

找工作面试的过程中，项目将会是整个流程的核心灵魂，也是能在面试中能让面试官尽快认同你的一个强有力的依据，所以在面试中把电商项目清晰的表述出来是极为关键的。

## 1.说说你最近做的这个项目的背景，简单的介绍一下你这个项目？

我们公司之前主要以实体店为主，进行批发与零售，业务也相对比较传统。为了适应市场需求，增强公司竞争力，提升业务绩效，另一方面，也为基于互联网的商务模式创新奠定基础。所以开始 xxx 商城建设项目，其中包含商品管理、订单管理、类目管理、客户管理、合作商管理、客服管理、购物平台、内容管理等，很大程度上分担了人工的压力，对提高客户服务效率和客户满意度能够起到较好的作用。

(先大体的描述下项目，然后能够挑一两个自己最为熟悉的模块进行叙述)

一定要学会装逼，在描述的过程中引导性的说出项目中的亮点，例如分布式锁，分布式事务

## 2.本项目提供了哪些模块（功能和服务）

**商品模块：**其中包括商品管理，类型管理，属性管理，栏目管理等等 **订单模块：**其中包括下单，退单，库存，收货人信息等 **会员模块：**会员注册，会员信息管理，会员等级管理，会员权限等 **购物车模块：**购物车数据存储，增删改查购物车商品，清空购物车等 **提交订单页面：**提交用户的订单信息, 处理并发问题。 **个人中心：**包括用户的登录，个人信息的管理，收货地址的管理，用户所下的订单的管理 **支付模块：**支付方式管理（在线支付、货到付款）等

## 3.你负责的哪些模块

在项目中主要负责相关功能的开发，主要有：

- 1 1) 后台管理系统：主要实现商品管理、商品规格参数管理、订单管理、会员管理等、CMS(内容管理系统)等，并且提供了跨域支持；
- 2 2) 前台系统：主要是面向用户访问，使用 `js`、`ajax` 进行前后台数据交互（一般是用 `json` 格式数据返回）
- 3 3) 会员登录：提供和用户信息相关的接口，比如说用户注册、查询等接口（登录时需要进行多重验证，特别注意安全方面）
- 4 4) 订单功能：主要是提供和订单相关的业务接口，在订单系统了做了严格的数据校验以及高并发写的支持（这里可以说使用队列实现），并且使用了定时器实现对下单的时间控制，比如说关闭超时未付款的订单；
- 5 5) 搜索功能：主要是提供商品的搜索，可以采用 `solr`全文搜索，当然也有其他的搜索方式；
- 6 6) 会员系统：主要是维护用户的信息，已购买订单、优惠券、系统消息、修改密码、绑定手机等功能；
- 7 7) 缓存：主要是用 `Redis` 实现，并且对 `Redis` 做了集群来保证 `Redis` 服务的高可用（缓存方面除了 `redis` 外还有 `memcached`）
- 8 8) 支付系统：主要是负责订单的支付、对账等功能，主要是对接了支付宝的接口；

（根据个人的实际情况选择最为熟悉的模块，进行叙述）

what

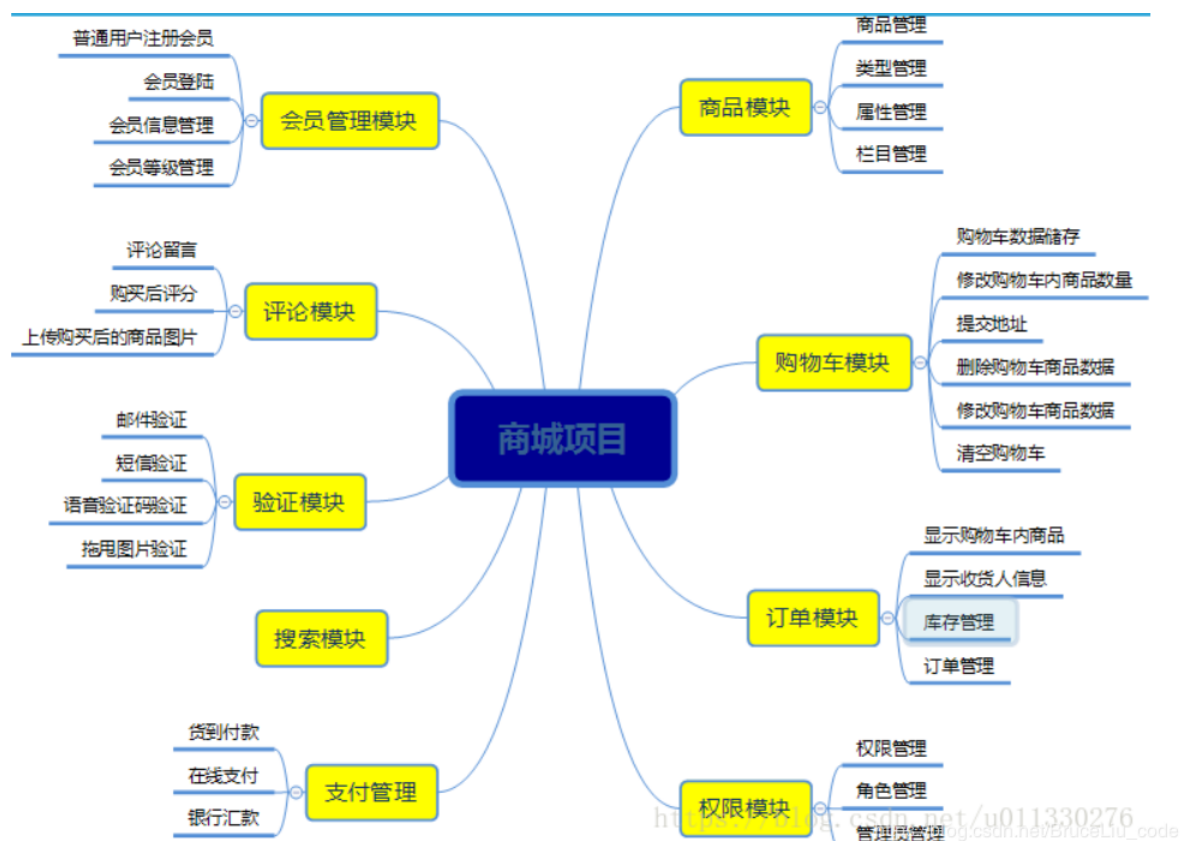
Why

Where

How

## 4.电商项目的思维图

---



以上是大部分常用模块，如有其它模块自行补充。根据上图的体系架构，一层的进行归纳记忆，从项目——模块——功能一点点的往深入熟悉记忆，并且加以理解。

## 5.项目开发流程



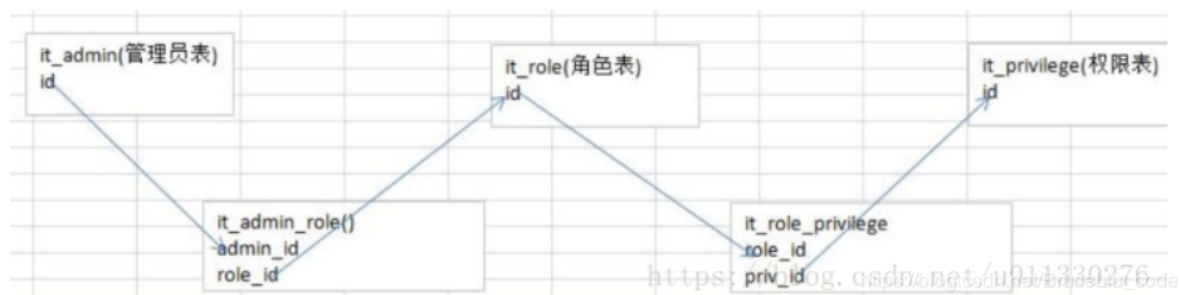
## 6.权限管理

包含功能: 管理员登录, 权限管理 (权限管理控制), 角色管理, 管理员管理

前台首页

后端管理: 权限操作! 角色, 不同的用户登录到后端管理, 那么所具备的操作功能(权限)不同

超级管理员、管理员、库存管理员、产品管理员....



## 7.你们项目里面哪些地方用到多线程？

**项目：**秒杀抢购使用了多线程: 场景，我们一共只有100个商品，在最后一刻，我们已经消耗了99个商品，仅剩最后一个。这个时候，系统发来多个并发请求，这批请求读取到的商品余量都是99个，然后都通过了这一个余量判断，最终导致超发。 **应对策略1:** 全部请求采用“先进先出”的队列方式来处理。那么新的问题来了，高并发的场景下，因为请求很多，很可能一瞬间将队列内存“撑爆”，然后系统又陷入到了异常状态。 **放弃！**

**应对策略2:** 乐观锁，采用带版本号（Version）更新。实现就是，这个数据所有请求都有资格去修改，但会获得一个该数据的版本号，只有版本号符合的才能更新成功，其他的返回抢购失败。这样的话，我们就不需要考虑队列的问题。

自定义Web服务器：平时用的tomcat，看过tomcat源码。如果说自己来实现一个web服务器：

网络编程Socket+多线程编程+文件流操作

## 8.你项目对于订单是怎么处理的，假如一个客户在下订单后没有购买怎么办，对于顾客在购买商品的时候你们怎么处理你们的库存？

我们都经常在淘宝上买东西，当我们提交订单后，如果某个时间段之内我们没有支付，淘宝肯定不会帮我们一直保留那个订单，如果超过半个小时我们未支付的话，淘宝会自动帮我们取消订单。在没有用RabbitMQ消息队列之前，我们可以通过设置一个定时任务，设定一个定时规则去轮询数据库查询超过半个小时而且未支付的订单，然后修改订单状态为已取消，这也是一个解决方案，但是需要轮询数据库，增加了对数据库的压力。

其实这种场景可以使用死信队列来做，就是用户提交订单之后，发送一条消息并且设置消息过期时间为半个小时(或其他时间)，如果超过设置的这个时间，那么消息自动变成死信，就会被转发到死信队列中，这时候我们可以监听死信队列中的消息，然后查询一下订单的状态，如果还是未支付的话，那么更新订单的状态为已取消。

- 1 1、下单后商品进入购物车，库存减，会有一个状态，，时间为半小时，超时后自动取消，库存加
- 2 2、用户付款,库存减

## 9.你平时测试的流程？

需求下来时，会先开发Dao层-service层-controller层 当持久层与业务层写完时，会使用Junit测试业务实现 然后开发页面及action，在进行本地测试。（自测通过后）上传公司内部服务器，测试人员会进行第二轮测试。（测试通过后）会走脚本，进行抗压访问测试。

测试和开发沟通的时候使用BUG管理系统工具：禅道、JIRA(付费)

持续集成CI CD

## 10.RabbitMQ在项目中如何应用的？

用户的请求数据直接写入数据库，在高并发的情况下，会对数据库造成巨大的压力，同时也使得系统响应延迟加剧。在使用ActiveMQ后，用户的请求发给队列后立即返回（当然不能直接给用户提示订单提交成功，提示：您“您提交了订单，请等待系统确认”），再由消息队列的消费者进程从消息队列中获取数据，异步写入数据库。由于消息队列的服务处理速度远快于数据库，因此用户的响应延迟可得到有效改善。

## 11.RabbitMQ如果数据提交不成功怎么办？

我们会重新发送消息，并记录日志，一般重新发送三次，如果仍然失败，我们会邮件或 短信通知相关人员进行处理。

使用本地消息表完成RabbitMQ消息一致性

## 12.忘记密码找回密码的流程？

在注册的时候会设置找回密码的问题和答案，在非登录状态忘记密码后，需要通过回答问题和答案找回密码。根据用户名找到用户设置的问题，然后回答完问题后，生成一个UUID，缓存在redis中设置有效期，并返回这个UUID。然后前端将这个UUID和新密码传入重置密码的函数，将传入的UUID和之前缓存在redis中的UUID进行比较，如果相同，则对新密码进行md5加密后更新到数据库中。

## 13.什么是sso系统(Redis实现的，非CAS)

单点登录：

单点登录是在多个应用系统中，用户只需要登录一次就可以访问所有相互信任的应用系统。登录的处理流程：

- 1 1、登录页面提交用户名密码。
- 2 2、登录成功后生成token。Token相当于原来的jsessionId，字符串，可以使用uuid。
- 3 3、把用户信息保存到redis。Key就是token，value就是TbUser对象转换成json。
- 4 4、使用String类型保存Session信息。可以使用“前缀:token”为key
- 5 5、设置key的过期时间。模拟Session的过期时间。一般半个小时。
- 6 6、把token写入cookie中。

如何判断是否登录

- 1 1.从cookie中取token
- 2 2.取不到未登录
- 3 3.取到token，到redis中查询token是否过期
- 4 4.如果过期，为登录状态
- 5 5.没有过期，登录状态

## 14.购物车逻辑

- 1、要求用户登录。
- 2、把购物车商品列表保存到数据库中。推荐使用redis。
- 3、Key：用户id，value：购车商品列表。推荐使用hash，hash的field：商品id，value：商品信息。
- 4、在用户未登录情况下写cookie。当用户登录后，访问购物车列表时，

- 1 a)把cookie中的数据同步到redis。
- 2 b)把cookie中的数据删除
- 3 c)展示购物车列表时以redis为准。
- 4 d)如果redis中有数据cookie中也有数据，需要做数据合并。相同商品数量相加，不同商品添加一个新商品。

- 5、如果用户登录状态，展示购物车列表以redis为准。如果未登录，以cookie为准。

## 15.浏览器跨域问题

跨域是指从一个域名的网页去请求另一个域名的资源。浏览器出于安全的考虑，不允许不同源的请求。JSONP解决AJAX跨域问题：JSONP是服务器与客户端跨源通信的常用方法。最大特点就是简单适用，老式浏览器全部支持，服务器改造非常小。它的基本思想是，网页通过添加一个 `<script>` 元素，向服务器请求JSON数据，这种做法不受同源政策限制；服务器收到请求后，将数据放在一个指定名字的回调函数里传回来。基于JSONP的实现原理，所以JSONP只能是“GET”请求，不能进行较为复杂的POST和其它请求，所以遇到那种情况，就得参考下面的CORS解决跨域了（所以如今它也基本被淘汰了），从Spring MVC 4.2 开始增加支持跨域访问，需要在控制器的方法上增加@CrossOrigin注解，如下：

```
1 @RequestMapping("/crossDomain2")
2 @ResponseBody
3 @CrossOrigin
4 public String crossDomain2(HttpServletRequest req,
5                             HttpServletResponse res, String name){
6     .....
7 }
```

## 16.海量数据的存储问题

如今随着互联网的发展，数据的量级也是呈指数的增长，从GB到TB到PB。对数据的各种操作也是愈加的困难，传统的关系性数据库已经无法满足快速查询与插入数据的需求。这个时候NoSQL的出现暂时解决了这一危机。它通过降低数据的安全性，减少对事务的支持，减少对复杂查询的支持，来获取性能上的提升。

但是，在有些场合NoSQL一些折衷是无法满足使用场景的，就比如有些使用场景是绝对要有事务与安全指标的。这个时候NoSQL肯定是无法满足的，所以还是需要使用关系性数据库。如果使用关系型数据库解决海量存储的问题呢？此时就需要做数据库集群，为了提高查询性能将一个数据库的数据分散到不同的数据库中存储。采用MyCat完成Mysql的集群配置！

数据库集群，横向扩展

采用分布式文件系统，HDFS，OOS

海量数据存储分布式数据库 HBASE

## 17.什么是Mycat?

简单的说，MyCAT就是：一个新颖的数据库中间件产品，支持mysql集群，提供高可用性数据分片集群。你可以像使用mysql一样使用mycat。对于开发人员来说根本感觉不到mycat的存在。 Mycat读写分离

数据库读写分离对于大型系统或者访问量很高的互联网应用来说，是必不可少的一个重要功能。对于MySQL来说，标准的读写分离是主从模式，一个写节点Master后面跟着多个读节点，读节点的数量取决于系统的压力，通常是1-3个读节点的配置。

## 18.电商活动倒计时方案(秒杀方案)：

- 1 1、确定一个基准时间。可以使用一个sql语句从数据库中取出一个当前时间。SELECT NOW();
- 2 2、活动开始的时间是固定的。
- 3 3、使用活动开始时间-基准时间可以计算出一个秒为单位的数值。
- 4 4、在redis中设置一个key（活动开始标识）。设置key的过期时间为第三步计算出来的时间。
- 5 5、展示页面的时候取出key的有效时间。Ttl命令。使用js倒计时。
- 6 6、一旦活动开始的key失效，说明活动开始。
- 7 7、需要在活动的逻辑中，先判断活动是否开始。
- 8 8、把商品的数量放到redis中。
- 9 9、秒杀时使用decr命令对商品数量减一。如果不是负数说明抢到。
- 10 10、一旦返回数值变为0说明商品已售完。

由于商城是基于SOA的架构，表现层和服务层是不同的工程。所以要实现商品列表查询需要两个系统之间进行通信。

## 19.如何实现远程通信？

- 1、Webservice：效率不高基于soap协议。项目中不推荐使用。

SOAP协议=HTTP协议+XML数据

- 2、使用restful形式的服务：http+json。很多项目中应用。如果服务太多，服务之间调用关系混乱，需要治理服务。
- 3、使用SpringCloud技术栈，实现服务之间通信
- 4、借助RabbitMQ完成模块通讯

## 20.电商项目中是如何解决高并发和高可用的？

- 1 1. 页面静态化
- 2 2. fastDFS图片服务器
- 3 3. 数据缓存服务器
- 4 4. 数据库集群、库表散列（数据库的各种优化、数据库的拆分）
- 5 5. 负载均衡
- 6 6. 消息队列流量削峰

在实际的生产环境下，一般都是集群部署

## 21.Redis集群中，某个节点宕机怎么办？你遇见过吗？你的解决思路是什么？

redis集群：一般的是至少是2台服务器，主从服务器！如果redis集群的主服务器挂了，没有关系还有备服务器。

哨兵模式和直连型模式

## 22.Redis在其中是怎么用的？起了什么作用？

redis中存储的都是key-value格式的。拿商品数据来说，key就是商品id，value是商品相关信息的json数据。

在商城系统中当并发量比较高，频繁的对数据库进行读操作的时候都需要添加缓存。例如页面中内容数据的缓存、商品数据的缓存以及用户数据的缓存等。

做商品数据的缓存时，因为商品的数据量很大，而且缓存是把数据保存到内存中，此时不可能把所有的商品数据都放到缓存中。所以需要设置商品数据缓存的有效期，当用户访问到非热点数据后，此数据放到缓存中，当缓存到期后就从缓存中删除，而且长时间不会添加到缓存。而热点数据一旦从缓存中删除会马上又添加到缓存。这样可以提高缓存的利用率，同时也减轻了数据库的压力。

各模块是怎么设计的，如商品参数，广告位等，数据库设计思路。。。

总的来说调用和设计上的问的较多，以及秒杀，比较多的是细节，比如秒杀中遇到的一些问题，怎么解决之类的（面经）

缓存预热：在项目启动的时候，事先把需要缓存的数据先放入Redis缓存中，需要监控Web项目启动的事件！

## 23.人员配置

- 1 产品经理：3人，确定需求以及给出产品原型图
- 2 项目经理：1人，项目管理
- 3 前端团队：3人，根据产品经理给出的原型制作出静态页面
- 4 后端团队：20人，实现产品功能
- 5 测试团队：4人，测试所有的功能
- 6 运维团队：3人，项目的发布以及维护

(上述是以团队为单位，有的公司技术部不止一个团队，还有运维、UI、交互设计师等等，视情况而定)

## 24.什么是FastDFS,在哪个地方使用到了fastDFS?

FastDFS是用c语言编写的一款开源的分布式文件系统。FastDFS为互联网量身定制，充分考虑了冗余备份、负载均衡、线性扩容等机制，并注重高可用、高性能等指标，使用FastDFS很容易搭建一套高性能的文件服务器集群提供文件上传、下载等服务。

使用FastDFS完成广告和商品图片的分布式存储。

## 25.对于大流量的网站,您采用什么样的方法来解决访问量问题?

- 1 ①确认服务器硬件是否足够支持当前的流量
- 2 ②优化数据库访问
- 3 ③禁止外部的盗链
- 4 ④控制大文件的下载
- 5 ⑤使用不同主机分流主要流量、集群
- 6 ⑥使用流量分析统计软件

## 26.数据库的存储引擎? 有什么区别?

MyISAM、InnoDB 构成上，MyISAM 的表在磁盘中有三个文件组成，分别是表定义文件（.frm）、数据文件（.MYD）、索引文件（.MYI），而 InnoDB 的表由表定义文件（.frm）、表空间数据和日志文件组成。安全方面，MyISAM 强调的是性能，其查询效率较高，但不支持事务和外键等安全性方面的功能，而 InnoDB 支持事务和外键等高

级功能，查询效率稍低。对锁的支持，MyISAM 支持表锁，而 InnoDB 支持行锁。

## 27.Sql 语句的优化?

- 1) 尽量选择较小的列
- 2) 将 where 中用的比较频繁的字段建立索引
- 3) select 子句中避免使用 '\*'
- 4) 避免在索引列上使用计算、not in 和 <> 等操作
- 5) 当只需要一行数据的时候使用 limit 1
- 6) 保证单表数据不超过 200W，适时分割表。针对查询较慢的语句，可以使用 explain 来分析该语句具体的执行情况。
- 7) 使用第三方监控软件监控 SQL 的执行过程，耗时时间，有针对性的进行 SQL 的优化

## 28.什么是页面静态化?

就是把一个动态的页面（操作数据库的 php 页面）变成一个静态页面，后续用户直接访问静态页面。页面静态化技术分为两种：真静态和伪静态。真静态：把一个动态的页面，实实在在的转成一个静态的页面，即 .html 文件。伪静态：所谓伪静态是从 url 地址上看是一个静态页面，但是实际上还是对应一个动态页面

以空间换时间

## 29.什么是二次开发？二次开发的弊端？

二次开发，简单的说就是在现有的软件上进行定制修改，功能的扩展，然后达到自己想要的功能，一般来说都不会改变原有系统的内核。弊端：

- 1) 插件限制太多；
- 2) 修改源文件对升级有影响。

## 30.数据库的读写分离？主从复制？

读写分离，基本的原理是让主数据库处理事务性增、改、删操作（INSERT、UPDATE、DELETE），而从数据库处理 SELECT 查询操作。数据库复制被用来把事务性操作导致的变更同步到集群中的从数据库。至少两台数据库服务器，可以分别设置主服务器和从服务器，对主服务器的任何操作都会同步到从服务器上

原理：mysql 中有一种日志，叫做 bin 日志（二进制日志），会记录下所有修改过数据库的 sql 语句。主从复制的原理实际是多台服务器都开启 bin 日志，然后主服务器会把执行过的 sql 语句记录到 bin 日志中，之后把这个 bin 日志发给从服务器，在从服务器再把 bin 日志中记录的 sql 语句同样的执行一遍。这样从服务器上的数据就和主服务器相同了。

## 31.高并发如何处理？

---

需要对服务器的架构分层，重新布局，负载均衡，集群策略。

负载均衡器（硬件和软件）**硬件**：F5-Bigip:立竿见影，价格昂贵，网游公司或大网站用的比较多**软件**：lvs(linux virtual server 虚拟服务，集成到内核中)，nginx(可以做 web 服务器，也可以做负载均衡使用)

负载均衡策略：

- 1 （1）轮询技术：把客户端的请求轮流分发给服务器。
- 2 （2）最少连接：负载均衡把请求给最空闲的服务器
- 3 （3）ip 哈希：同一地址的客户端，始终请求同一台服务器。

## 32.网站并发数：

---

经过压力测试可以支持3000左右的并发，可以满足目前的业务需求。由于我们的系统是分布式架构，支持水平扩展，如果将来并发量提高的话，可以增加服务器来提高并发量。

## 33.迭代的周期

---

采用迭代开发的方式进行，一般一次迭代的周期为二个月左右。

## 34.你说你用了redis缓存，你redis存的是什么格式的数据，是怎么存的

---

redis中存储的都是key-value格式的。拿商品数据来说，key就是商品id，value是商品相关信息的json数据。

## 35.你购物车存cookie里边 可以实现不登录就可以使用购物车 那么我现在没有登录把商品存购物车了 然后登录了 然后我换台电脑并且登录了 还能不能看见我购物车的信息？如果看不到怎么做到cookie同步，就是在另外一台电脑上可以看到购物车信息

本商城项目现阶段使用的仅仅是把购物车的商品写入cookie中，这样服务端基本上么有存储的压力。但是弊端就是用户更换电脑后购物车不能同步。打算下一步这么实现：当用户没有登录时向购物车添加商品是添加到cookie中，当用户登录后购物车的信息是存储在redis中的并且是跟用户id向关联的，此时你更换电脑后使用同一账号登录购物车的信息就会展示出来。

## 36.如果用户一直添加购物车添加商品怎么办？并且他添加一次你查询一次数据库？互联网上用户那么多，这样会对数据库造成很大压力你怎么办？

当前我们使用cookie的方式来保存购物车的数据，所以当用户往购物车中添加商品时，并不对数据库进行操作。将来把购物车商品放入redis中，redis是可以持久化的可以永久保存，此时就算是频繁的往购物车中添加数据也没什么问题。

## 37.秒杀抢购库存解决方案

- 1 把商品的数量放到redis中。
- 2 秒杀时使用decr命令对商品数量减一。如果不是负数说明抢到。
- 3 一旦返回数值变为0说明商品已售完。

## 38.支付接口是怎么做的？

面试中可以说支付这部分不是我们做的，我们项目中并没有涉及支付部分的处理。如果了解支付是如何实现可以参考之前学过的微信支付相关文档。

支付宝：

<https://doc.open.alipay.com/doc2/apiDetail.htm?spm=a219a.7629065.0.0.eeTXH8&apiId=850&docType=4#>

微信支付：

[https://mp.weixin.qq.com/cgi-bin/readtemplate?t=business/faq\\_tmpl](https://mp.weixin.qq.com/cgi-bin/readtemplate?t=business/faq_tmpl)

微信支付的流程复习一下，调用的接口有统一下单，和订单查询接口。具体传什么参数，笔记中有

## 39.业务如何说？先说业务、说表、说具体实现？

---

先说总体的业务流程，然后再说具体业务的实现方法及使用的技术。最后说你在系统中负责的内容。不需要说表结构。

## 40.单点登录系统，如果cookie禁用，你们怎么解决？

---

如果禁用cookie可以使用url中带参数，把token传递给服务端。当然此方法涉及安全性问题，其实在cookie中保存token同样存在安全性问题。推荐使用sso框架CAS实现单点登录。

## 41.你前台portal采用4台服务器集群部署，那能前台高并发访问性能提上去了，那数据库会不会造成一个瓶颈，这一块你是怎么处理的？

---

portal系统在高并发的情况下如果每次请求都请求都查询数据库确实会出现数据库的瓶颈。为了降低数据库压力，在服务层会添加一个缓存，用redis实现，这样的话请求先到缓存中查找是否有缓存的内容，如果有直接从缓存中取数据，如果没有再到数据库中查询。这样数据库的压力就不会那么大了。

## 42.安全性问题（别的网站使用爬虫技术爬你的网站怎么办？有没有安全措施）

---

单位时间内请求次数超过某个阈值就让输入验证码，可以极大降低抓取的速度，如果多次超过某个阈值可以加入黑名单。还有就是页面内容使用json返回，数据经常变一变格式，或者js动态生成页面内容。

## 43. 订单表的数据量太大,我把订单分到许多表中,那么我我想用一条sql查处所有的订单,怎么解决?

---

分库情况下：可以使用mycat数据库中间件实现多个表的统一管理。虽然物理上是把一个表中的数据保存到多个数据库中，但是逻辑上还是一个表，使用一条sql语句就可以把数据全部查询出来。

单库情况下：需要动态生成sql语句。先查询订单相关的表，然后将查询多个表的sql语句使用union连接即可。

## 44. 咱们单点登录模块中，别人伪造我们cookie中的token怎么办？

---

服务端是无法阻止伪造cookie的，如果对安全性要求高的话可以使用cas框架。

## 45. 第一个是当两个客户同时买一件商品时库存只有一个了,怎么控制？

---

可以使用mysql的行锁机制，实现乐观锁，在更新商品之前将商品锁定，其他用户无法读取，当此用户操作完毕后释放锁。当并发量高的情况下，需要使用缓存工具例如redis来管理库存。

## 46. 对数据库只是采用了读写分离,并没有完全解决数据库的压力,那么有什么办法解决？

---

如果数据库压力确实很大的情况下可以考虑数据库分片，就是将数据库中表拆分到不同的数据库中保存。可以使用mycat中间件

## 47. 同一账号以客户端登录怎么挤掉另一端。

---

用户登录后需要在Session中保存用户的id。当用户登录时，从当前所有的Session中判断是否有此用户id的存在，如果存在的话就把保存此用户id的Session销毁。

