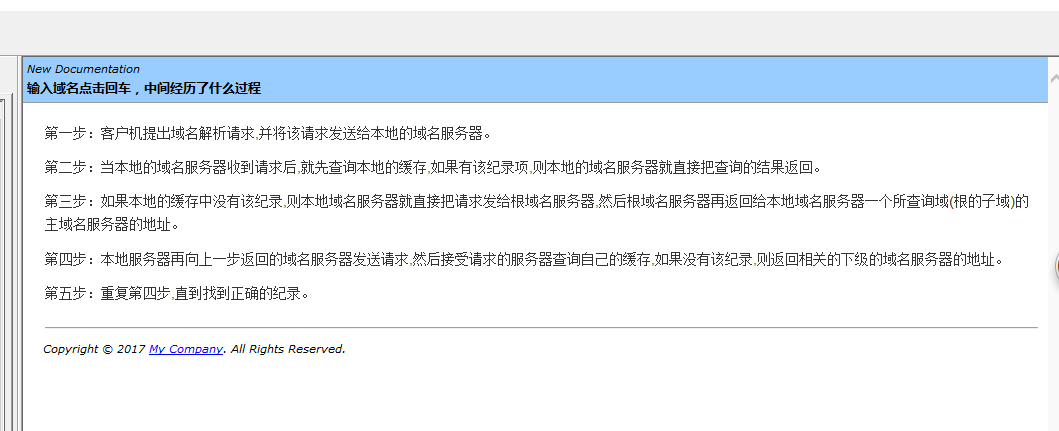
**互怼提问 – 架构优化**



1. 主从复制的原理是什么？主从复制中断了怎么办？主从复制出现延迟怎么处理？主库崩溃了怎么办？Mha

原理：

原理：主从复制一共有三个进程，从库生成两个线程，一个I/O线程，一个SQL线程；

i/o线程去请求主库 的binlog，并将得到的binlog日志写到relay log（中继日志） 文件中；

主库会生成一个 log dump 线程，用来给从库 i/o线程传binlog；

SQL 线程，会读取relay log文件中的日志，并解析成具体操作，来实现主从的操作一致，而最终数据一致;

主从复制出现错误？

错误一：

错误原因：从库用来连接主库的用户权限或者密码不对

解决方法：首先在主库上检查用来主从复制的用户权限，如果没有问题在检查从库使用的密码是否正确。

错误二：

错误原因：这个可能是从库的master.info文件有损坏。

解决方法：reset slave

错误三：

错误原因：可能是从库的约束比主库更多写造成的。

解决方法：

Mysql > stop slave;

Mysql > set global sql\_slave\_skip\_counter =1 ;

Mysql > start slave;

错误四：

错误原因：slave上缺少错误中的表。

解决方法：在slave上添加上对应的表，然后start slave。

错误五：

错误原因：从库上对应的表上缺少字段。

解决方法：根据主库上表结构，在从库对应表上添加缺少的字段，然后start slave。

错误六：

错误原因：主库删除的表在从库中不存在，导致从库在遇到删除不存在表的错误时无法继续同步。

解决方法：利用slave-skip-errors参数，跳过对于的1146错误（这个参数是一个只读的，需要在配置文件中修改，并重启从库）

1、在my.cnf的[mysqld]下面添加slave\_skip\_errors=1146

2、重启从库 service mysq jhl restart

3、在从库上启动同步

4、去掉my.cnf中的slave\_skip\_errors=1146

5、重启从库

6、启动从库复制`1

**中断：**

Show slave status\G；查看错误原因

1.show slave status\G,提示中继日志损坏,按以往的做法，根据提示重新指定合适的日志文件以及pos点。

2.  从MySQL5.5.X版本开始，增加了relay\_log\_recovery参数，这个参数的作用是：当slave从库宕机后，假如relay-log损坏了，导致一部分中继日志没有处理，则自动放弃所有未执行的relay-log，并且

重新从master上获取日志，这个参数是默认关闭的。做主从的时候没有开启这项参数。修改my.cnf，添加这两项。（skip-slave-start ，mysql服务启动跳过自动启动主从复制，以免产生新的问题）,relay\_log\_recovery不支持动态修改。所以修改配置文件，重启MySQL服务,启动主从复制线程

**a、主从同步延迟与系统时间的关系，查看主从两台机器间系统 时间差**

**c、主从同步延迟与lock锁的关系（myisam表读时会堵塞写），尽量避免使用myisam表。一个实例里面尽量减少数据库的数量。**

**d、主从复制发生异常而中断，过很久之后才发现复制异常。可通过查看master与slave的status估算相差的日志。如果相差太大，则可以考虑重做从库。**

产生的原因：当主库的并发量比较大的时候，产生的DDL数量，超过线程所承受的范围，延时就产生了，还有可能是与slave的大型query语句产生了锁等待;

解决办法:在架构上做优化，尽量让主库的DDL快速执行，因为从库只是读取数据，不需要那么高的安全性，所以可以将sync\_binlog设置为0，或者关闭binlog日志，或者是使用比主库更好的硬件设备作为slave

**可以暂时存缓存里**

**从库不能执行大量写入（5.6是支持库的并发，5.7支持了一个库多个表的并发执行）**

https://www.cnblogs.com/gomysql/p/3675429.html

MHA :原理

（1）从宕机崩溃的master保存二进制日志事件（binlog events）;

（2）识别含有最新更新的slave；

（3）应用差异的中继日志（relay log）到其他的slave；

（4）应用从master保存的二进制日志事件（binlog events）；

（5）提升一个slave为新的master；

（6）使其他的slave连接新的master进行复制；

1. Redis有哪些数据类型？

**String ，hash，set，zset, list**

1. Redis和memcache的区别是什么？

**从存储大小 Memcached单个key-value大小有限，一个value最大只支持1MB，key 最大250个字符，而Redis最大支持512MB**

**从可靠性 Memcached只是个内存缓存，对可靠性无要求；而Redis更倾向于内存数据库，因此对对可靠性方面要求比较高**

区别：

1：memcache和redis都是将数据存储到内存中，都是内存数据库，不过memcache还可以缓存其他东西，图片视频等。

2：存储类型不同，memcache只支持k-v结构，redis支持五种数据类似，字符串，链表，哈希，集合，有序集合。

3：虚拟内存，当redis物理内存使用完后，可以将一些很久没有使用过的value交换到磁盘、

4：分布式设定memcache集群，利用magenta做一主多从，redis可以做一主多从，也可以一主一从。

5：存储数据redis持久化，memcache关机后数据消失

6：数据恢复，memcache挂掉后，数据不可恢复，redis数据丢失后可以通过AOF日志恢复。

7：应用场景不同，redis除啦作为数据库使用之外，还能做消息队列，数据堆栈和数据缓存等，memcache适用于缓存sql语句，数据集，用户临时性数据，延迟查询数据 session等

1. Redis在项目中怎么用的？

**查询数据多的页面做缓存比如页面缓存首页**

1. Mysql数据库优化做过哪些工作

1：建表时选择适合字段的数据类型，合适的长度

2：查询多的字段加索引 sql语句优化

3：分页优化

4：分区分表

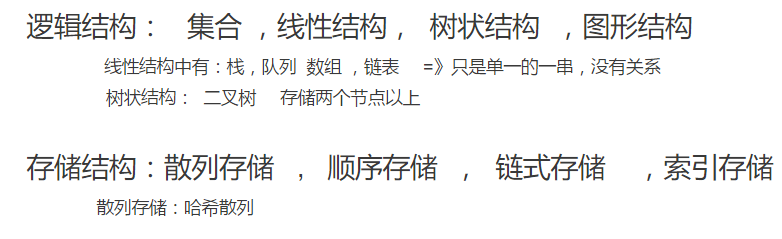
5：主从复制，读写分离

1. Myisam和innodb的区别是什么？

innodb写操作快、支持事务、数据行锁、 支持外键， InnoDB的主键范围更大，最大是MyISAM的2倍。

MyISAM不支持事务和外键，读操作快、 支持数据锁表、 GIS数据 、全文索引（因为它不支持中文分词，所以 MyISAM的全文索引其实没啥用）。

1. 你了解哪些数据结构？



1. 你了解哪些算法？

**冒泡，快速，二分，归并**

1. 负载均衡怎么搭建的？nginx属于第几层负载均衡？nginx对于请求的分配有哪些策略模式？

**反向代理 第7层**

**轮询，加权轮询，IP负载均衡，cdn负载均衡**

1. 你们的项目用了几台服务器？怎么分配的？

6

1. 处理过高并发吗？请说明业务场景和解决办法

秒杀 ，文件锁

1. 请说一下你所处理过的最复杂的业务逻辑
2. 在之前的开发过程当中你碰到过的最困难的问题是什么？怎么解决的？
3. 网络七层协议有哪些？

**物理层，数据链路层，网络层，传输层，会话层，表示层，应用层**

**互怼提问 – PHP**

1. 请说出10个字符串函数和10个数组函数

**Substr（截取）,strlen（长度）,str\_replace替换，strpos（查找并返回首个匹配项的位置），substr\_count(出现次数)，explode(字符串分割数组)，**

**Rtrim(去除右面空格)，ltrim(去除左边空格)，trim(去除两边)，str\_split （函数把字符串分割到数组中。）**

**In\_array(), is\_array(), implode(), array\_merge(), array\_keys(), array\_values(), sort(), array\_pop(删除数组最后一个元素)， array\_shift(删除数组第一个元素，并返回值) array\_push(插入数组末尾)**

你了解哪些设计模式？

**单例模式，工厂模式，简单工厂模式，策略模式，观察者模式**

1. 请说一下get和post的区别

get和post区别，唯一区别一个是用过url实现，另外一个是通过数据包；本质是http的原因，

关于GET/POST大小说法，这个是关于浏览器（发送http请求）和服务器（接收http请求）的关系，

（大多数浏览器url限制2K个字节，大多数服务器只处理64k之内的url）

**Get不是特别安全，post安全性更高。**

1. 请描述一下session和cookie的区别,session信息的存储方式，如何进行遍历？

1：session存储在服务器端，cookie存储在客户端

2:session默认存储在服务器的一个文件里（不是内存）

3：session的运行依赖session id ，而session id是存在cookie中的，也就是说如果浏览器禁用了cookie，同时session也会失效，（但是可以通过其他方式实现session ，比如在url中传递session id

）

4：session 可以放在文件，数据库，或者是内存中，默认是存储在文件中，都可以

5：用户验证一般都会用session

1. 请说一下你对mvc的理解

**M model是应用程序中用于处理应用程序数据逻辑的部分。**

**模型层主要于数据库进行交互，**

**V view是应用程序中处理数据显示的部分。**

**视图层展示给用户看的页面**

**C是应用程序中处理用户交互的部分。**

**通常控制器负责从视图读取数据，控制用户输入，并向模型发送数据。**

1. 请描述一下php-fpm的工作原理

**FastCGI进程管理器**

请求到nginx通过fastcgi协议，fastcgi会找php-fpm解析php文件，php-fpm有两个进程，master进程和worker进程，worker进程会解析php代码最后将结果返回给nginx

**互怼提问 – Mysql**

1. varchar和char的区别

**char 长度固定，varchar长度是变化的。**

**比如： char (10) 表示存储的字节就是占10字节**

**但是varchar(10) ，10是最大值，字节按照存储字节算**

1. 什么是数据库索引？mysql有哪些索引？主键索引和唯一索引有什么区别？

索引是对数据库表中一列或多列的值进行排序的一种结构，使用索引可快速访问数据库表中的特定信息数据库[索引](http://baike.baidu.com/view/262241.htm" \t "http://blog.csdn.net/sap_bw/article/details/_blank)好比是一本书前面的目录，能加快数据库的查询速度。

**普通索引，**

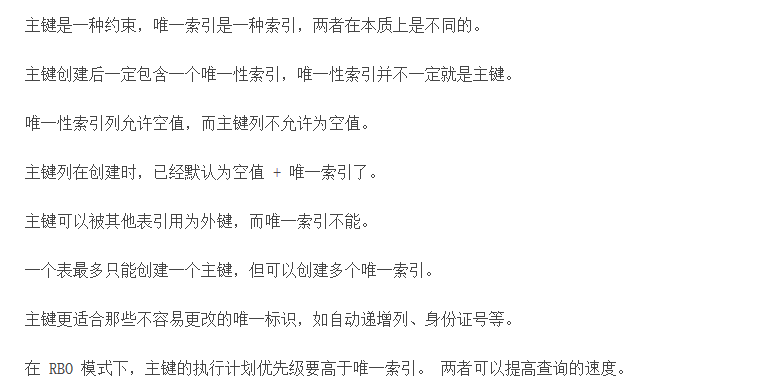
**主键索引，（要求主键中的每个值都唯一。）**

**唯一索引，（唯一索引是不允许其中任何两行具有相同索引值的索引。）**

**复合索引，**

**全文索引。**

区别：http://blog.csdn.net/tanga842428/article/details/52790651



1. Myisam和innodb的区别是什么？为什么innodb比myisam写入快？

innodb写操作快、支持事务、数据行锁、 支持外键 InnoDB的主键范围更大，最大是MyISAM的2倍。

MyISAM不支持事务和外键，读操作快、 支持数据锁表、 GIS数据 、

在高并发的情况下，因为innodb支持行锁可以多行操作，而myisam只支持表锁只能一个一个操作所以慢

1. 请列举出你所了解的会引起索引失效的情况

1.如果条件中有or，即使其中有条件带索引也不会使用(这也是为什么尽量少用or的原因)

注意：要想使用or，又想让索引生效，只能将or条件中的每个列都加上索引

2.对于多列索引，不是使用的第一部分，则不会使用索引（最左原则）

3.like查询是以%开头

4.如果列类型是字符串，那一定要在条件中将数据使用引号引用起来,否则不使用索引

5.如果mysql估计使用全表扫描要比使用索引快,则不使用索引

6对索引列运算，运算包括（+、-、\*、/、！、<>、%、like'%\_'（%放在前面）、in、exist等），导致索引失效。

7.在JOIN、UNION操作中，主键和外键的数据类型不相同，MYSQL将无法使用索引

8.is null 索引失效，is not null Betree索引生效。导致的原因：mysql没有在null写进索引

9.对索引应用内部函数，这种情况下应该建立基于函数的索引

explain select \* from demo\_range t where round(t.title)=1

1. 在mysql中值为空和值为null有什么区别？

**空值('')是不占用空间的 NULL其实是占用空间的。**

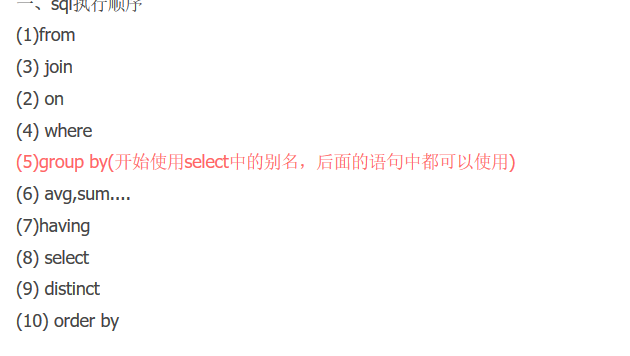
**在进行count()统计某列的记录数的时候，如果采用的NULL值，会别系统自动忽略掉，但是空值是会进行统计到其中的。**

1. left join、right join、inner join的区别是什么？

**left join(左联接) 返回包括左表中的所有记录和右表中联结字段相等的记录   
right join(右联接) 返回包括右表中的所有记录和左表中联结字段相等的记录  
inner join(等值连接) 只返回两个表中联结字段相等的行**

1. 请描述一下sql语句的执行顺序

https://www.cnblogs.com/spring-hailong/p/6182227.html



（11）limit

1. 请描述一下你对mysql连接池的理解

**数据库连接池负责分配、管理和释放数据库连接，它允许应用程序重复使用一个现有的数据库连接，而不是再重新建立一个**

**1. 最小连接数**

**是连接池一直保持的数据库连接，所以如果应用程序对数据库连接的使用量不大，将会有大量的数据库连接资源被浪费。**

**2. 最大连接数**

**是连接池能申请的最大连接数，如果数据库连接请求超过此数，后面的数据库连接请求将被加入到等待队列中，这会影响之后的数据库操作。**

1. 事物的四大特性和事物的隔离级别

## *原子性（Atomicity）*

*原子性是指事务包含的所有操作要么全部成功，要么全部失败回滚，*

## *一致性（Consistency）*

*一致性是指事务必须使数据库从一个一致性状态变换到另一个一致性状态，也就是说一个事务执行之前和执行之后都必须处于一致性状态。*

## *隔离性（Isolation）*

*隔离性是当多个用户并发访问数据库时，比如操作同一张表时，数据库为每一个用户开启的事务，不能被其他事务的操作所干扰，多个并发事务之间要相互隔离。*

## *持久性（Durability）*

*持久性是指一个事务一旦被提交了，那么对数据库中的数据的改变就是永久性的，即便是在数据库系统遇到故障的情况下也不会丢失提交事务的操作。*

*不考虑事务的隔离性，会发生的几种问题：*

## *脏读*

*脏读是指在一个事务处理过程里读取了另一个未提交的事务中的数据。*

## *不可重复读*

*不可重复读是指在对于数据库中的某个数据，一个事务范围内多次查询却返回了不同的数据值，这是由于在查询间隔，被另一个事务修改并提交了*

## *虚读(幻读)*

*幻读是事务非独立执行时发生的一种现象。例如事务T1对一个表中所有的行的某个数据项做了从“1”修改为“2”的操作，这时事务T2又对这个表中插入了一行数据项，而这个数据项的数值还是为“1”并且提交给数据库。而操作事务T1的用户如果再查看刚刚修改的数据，会发现还有一行没有修改，其实这行是从事务T2中添加的，就好像产生幻觉一样，这就是发生了幻读。*

*现在来看看MySQL数据库为我们提供的四种隔离级别：*

*① Serializable (串行化)：可避免脏读、不可重复读、幻读的发生。*

*② Repeatable read (可重复读)：可避免脏读、不可重复读的发生。*

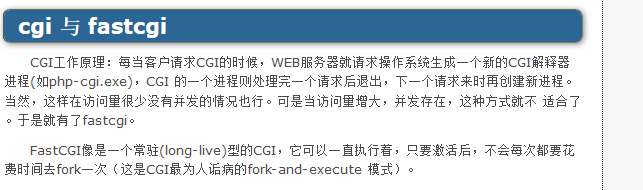
*③ Read committed (读已提交)：可避免脏读的发生。*

*④ Read uncommitted (读未提交)：最低级别，任何情况都无法保证。*

**互怼提问 – Linux**

1. 如何使nginx和php关联起来？如何使Apache和php关联起来？

Fastcgi



PHP的所有应用程序都是通过WEB服务器(如IIS或Apache)和PHP引擎程序解释执行完成的，工作过程：  
  
(1)当用户在浏览器地址中输入要访问的PHP页面文件名，然后回车就会触发这个PHP请求，并将请求传送化支持PHP的WEB服务器。  
  
(2)WEB服务器接受这个请求，并根据其后缀进行判断如果是一个PHP请求，WEB服务器从硬盘或内存中取出用户要访问的PHP应用程序，并将其发送给PHP引擎程序。  
  
(3)PHP引擎程序将会对WEB服务器传送过来的文件从头到尾进行扫描并根据命令从后台读取，处理数据，并动态地生成相应的HTML页面。  
  
(4)PHP引擎将生成HTML页面返回给WEB服务器。WEB服务器再将HTML页面返回给客户端浏览器。

简言之：apache是WEB服务器，它可以解析php语言。

1. 请描述PHP-FPM的工作原理

PHP-FPM是一个PHP FastCGI进程管理器，是只用于PHP的

PHP-FPM提供了更好的PHP进程管理方式，可以有效控制内存和进程、可以平滑重载PHP配置

请求到nginx通过fastcgi协议，fastcgi会找php-fpm解析php文件，php-fpm有两个进程，master进程和worker进程，worker进程会解析php代码最后将结果返回给nginx

1. 如何安装php扩展？

## 1：使用apt或者yum命令安装（如果PHP是通过 apt 或者 yum 命令安装的，则扩展也可以通过 apt 或者 yum 安装）

## 2：使用pecl安装

## 3：源码编译安装（一般是安装PHP自带的扩展，以安装pcntl扩展为例）

## 4：phpize安装（要安装的扩展在php源码ext目录中没有，那么这个扩展需要到[http://pecl.php.net](http://pecl.php.net/" \t "http://blog.csdn.net/abel004/article/details/_blank) 搜索下载）

1. nginx和Apache的区别是什么？

1：nginx是轻量级，比apache占用更少的内存以及资源

2：apache比nginx的rewrite强大，模块多，bug少，稳定性强，

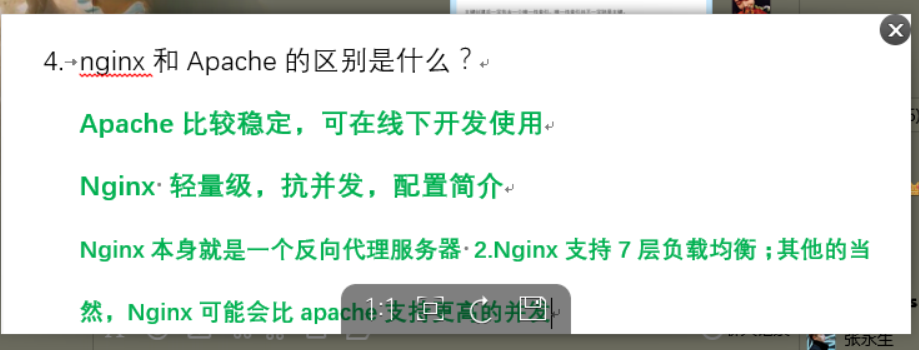
3：apache对php支持比较简单，nginx需要配合其他后端用，apache组件比nginx多，现在nginx是web服务器的首选

4：最核心的区别是apache是同步多进程模型，一个连接对应一个进程，多个连接（千万级别）可以对应一个进程

5：nginx：配置文件写的很简洁，正则配置让很多事情变得简单运行效率高，占用资源少，代理功能强大，很适合做前端响应服务器

6：Apache在处理动态有优势，Nginx并发性比较好，CPU内存占用低，如果rewrite频繁，那还是Apache吧

7：动态请求要apache去做，nginx只适合静态和反向



1. 请说出常用的10个Linux操作命令（ll ls cd除外）

top 查看内存

df-lh 查看磁盘 ps -a 查看所有进程

修改权限 chomd

修改所有者 chown 压缩 tar -zcvf

解压 tar -zxv 查看日志 tail -f

查找 find locate 复制 dd  删除 rm

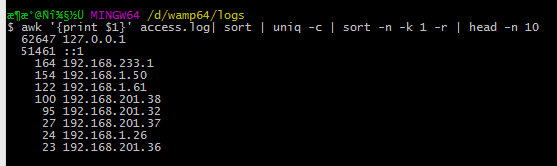
杀死进程 kill 创建文件夹 mkdir

移动：mv **报告现行正在工作的或当前目录 pwd**

**Tail 末端打开文件**

1. 1G的访问日志文件中，查出访问日志中访问次数前10的IP

Awk（是文本分析工具）



1. 如何查看一个进程是否在运行

**Ps aux |grep 查看的那个 | ps -ef | grep xxx**

1. 如何查看服务器当前CPU使用情况

**CPU占用率 top**

1. 什么是fast\_cgi,什么是cgi.什么是php-fpmcgi, php cgi,他们之间的区别

Cgi 看做是一个面向过程 php\_cgi（php里的面向过程）

fast\_cgi 就是一个面向对象(mvc) php\_fpm (各种框架)

1. session 于cookie的 原理及token 的作用？

Cookie概念

**在浏览某些 网站 时,这些网站会把 一些数据存在 客户端 , 用于使用网站 等跟踪用户,实现用户自定义 功能.**

Session的概念

Session 是存放在服务器端的类似于HashTable结构来存放用户数据;

作用：实现网页之间数据传递，是一个存储在服务器端的对象集合。  
  
 原理：当用户请求一个Asp.net页面时，系统将自动创建一个Session;退出应用程序或关闭服务器时，该Session撤销。系统在创建Session时将为其分配一个长长的字符串标识，以实现对Session进行管理与跟踪。

**Token**

**1:客户端使用用户名跟密码请求登录**

**2:服务端收到请求，去验证用户名与密码**

**3:验证成功后，服务端会签发一个 Token，再把这个 Token 发送给客户端**

**4:客户端收到 Token 以后可以把它存储起来，比如放在 Cookie 里或者 Local Storage 里**

**5:客户端每次向服务端请求资源的时候需要带着服务端签发的 Token**

**6:服务端收到请求，然后去验证客户端请求里面带着的 Token，如果验证成功，就向客户端返回请求的数据**

1. 数据库优化做了哪些 项目优化做了哪些？

mysql优化，负载均衡，缓存，静态页

1. 数组实现队列用哪些函数？

**队列先进先出**

**array\_push()尾部添加 和array\_shift() 删除数组第一个元素，并返回值**

1. redis 实现对列 关键点是什么 用了什么函数？

用了list

**Rpush(队列左边添加) lpush（队尾取）**

1. Mysql日志有哪些 ，这个 日志 默认文件格式 是什么 和文件格式 有哪些？

**错误日志：     -log-err    
查询日志：     -log    
慢查询日志:   -log-slow-queries    
更新日志:     -log-update**

**二进制日志： -log-bin**

1. 客户端禁掉 cookie session还能用吗？

**一般Session是用Session ID来确定当前对话所对应的服务器Session，而Session ID是通过Cookie来传递的，禁用Cookie相当于失去了Session ID，也就得不到Session了。**

**但是可以通过其他方式来存储sessionID,eg:redis或文件里，就可以用了**

1. 知道的PHP加密有哪些？

**Md5() 不可解的**

**Sha1() 哈希加密**

**Base64()**

**URL加密()**

1. 用过的webserver 有哪些？

Web Server中文名称叫网页服务器或[web服务器](https://baike.baidu.com/item/web%E6%9C%8D%E5%8A%A1%E5%99%A8" \t "https://baike.baidu.com/item/web%20server/_blank)。[WEB服务器](https://baike.baidu.com/item/WEB%E6%9C%8D%E5%8A%A1%E5%99%A8" \t "https://baike.baidu.com/item/web%20server/_blank)也称为WWW(WORLD WIDE WEB)服务器，主要功能是提供网上信息浏览服务

Nginx apache

1. 主主复制和主从复制的区别？

https://www.cnblogs.com/Richardzhu/archive/2013/11/14/3422840.html

主主数据能互相同步，主主复制一用作个般高容灾方

1. redis 持久化了解多少？

Aof(日志)，rdb(快照)

1. 用户表 做分表 hash分表之后 产生重复的数据怎么办？

主表判断

1. 了解的设计模式 单例模式怎么实现的？

三私一公

1. 抽象类和接口类的区别？

抽象类要被子类继承，接口要被类实现。

抽象类里可以没有抽象方法

如果一个类里有抽象方法，那么这个类只能是抽象类

抽象方法要被实现，所以不能是静态的，也不能是私有的。

接口可继承接口，并可多继承接口，但类只能单根继承、

1. 怎么防止form 表单多次提交？

disabled

1. 用过哪些版本控制软件？介绍一下？

Git和svn

git和其他[版本控制系统](https://zh.wikipedia.org/wiki/%E7%89%88%E6%9C%AC%E6%8E%A7%E5%88%B6%E7%B3%BB%E7%BB%9F" \t "http://blog.csdn.net/wenniuwuren/article/details/_blank" \o "版本控制系统)（如 [CVS](https://zh.wikipedia.org/wiki/%E5%8D%94%E4%BD%9C%E7%89%88%E6%9C%AC%E7%B3%BB%E7%B5%B1" \t "http://blog.csdn.net/wenniuwuren/article/details/_blank" \o "协作版本系统)）有不少的差别，git本身关心文件的整体性是否有改变，但多数的 CV S或 [Subversion](https://zh.wikipedia.org/wiki/Subversion" \t "http://blog.csdn.net/wenniuwuren/article/details/_blank" \o "Subversion) 系统则在乎文件内容的差异。因此git更像一个文件系统，直接在本机上获取数据，不必连接到主机端获取数据。

  SVN 是集中式或者有中心式版本控制系统，版本库是集中放在中央服务器的，而干活的时候，用的都是自己的电脑，所以首先要从中央服务器哪里得到最新的版本，然后干活，干完后，需要把自己做完的活推送到中央服务器。集中式版本控制系统是必须联网才能工作，如果在局域网还可以，带宽够大，速度够快，如果在互联网下，如果网速慢的话，就纳闷了。

      Git 是分布式版本控制系统，那么它就没有中央服务器的，每个人的电脑就是一个完整的版本库，这样，工作的时候就不需要联网了，因为版本都是在自己的电脑上。既然每个人的电脑都有一个完整的版本库，那多个人如何协作呢？比如说自己在电脑上改了文件A，其他人也在电脑上改了文件A，这时，你们两之间只需把各自的修改推送给对方，就可以互相看到对方的修改了。

1. 了解的魔术方法有哪些？

\_\_construct()　　 实例化对象时被调用，当\_\_construct和以类名为函数名的函数同时存在时，\_\_construct将被调用，另一个不被调用。

　　\_\_destruct()　　  当删除一个对象或对象操作终止时被调用。

　　\_\_call()　　　　　 对象调用某个方法，若方法存在，则直接调用；若不存在，则会去调用\_\_call函数。

　　\_\_get()　　　　   读取一个对象的属性时，若属性存在，则直接返回属性值；若不存在，则会调用\_\_get函数。

　　\_\_set()　　　　    设置一个对象的属性时，若属性存在，则直接赋值；若不存在，则会调用\_\_set函数。

　　\_\_toString()　　  打印一个对象的时被调用。如echo $obj;或print $obj;

　　\_\_clone()　　　    克隆对象时被调用。如：$t=new Test();$t1=clone $t;

　　\_\_sleep()　　　    serialize之前被调用。若对象比较大，想删减一点东东再序列化，可考虑一下此函数。

　　\_\_wakeup()　　   unserialize时被调用，做些对象的初始化工作。

　　\_\_isset()　　　    检测一个对象的属性是否存在时被调用。如：isset($c->name)。

　　\_\_unset()　　　　unset一个对象的属性时被调用。如：unset($c->name)。

　　\_\_set\_state()　　调用var\_export时，被调用。用\_\_set\_state的返回值做为var\_export的返回值。

　　\_\_autoload()　　  实例化一个对象时，如果对应的类不存在，则该方法被调用。

1. 怎么保证接口的安全，做了哪些操作？

1、当用户登录APP时，使用https协议调用后台相关接口，服务器端根据用户名和密码时生成一个access\_key，并将access\_key保存在session中，将生成的access\_key和session\_id返回给APP端。

2、APP端将接收到的access\_key和session\_id保存起来

3、当APP端调用接口传输数据时，将所传数据和access\_key使用加密算法生成签名signature，并将signature和session\_id一起发送给服务器端。

4、服务器端接收到数据时，使用session\_id从session中获取对应的access\_key，将access\_key和接收到的数据使用同一加密算法生成对应signature，如果生成的签名和接收到的signature相同时，则表明数据合法

1. sql注入和xss攻击了解多少，怎么实现的攻击？

Xss攻击是跨站脚本工具 Csrf攻击是跨站请求伪造 sql注入 DDOS流量攻击？

防止xss脚本攻击：

XSS攻击即跨站脚本攻击，通过篡改网页，注入恶意的HTML脚本，控制 用户浏览器进行恶意操作的一种攻击。

防止: script注入，转义过滤script标签。Htmlentities(把字符串转为html实体)

防止csrf跨站请求伪装：

CSRF的防御可以从服务端和客户端两方面着手，防御效果是从服务端着手效果比较好，现在一般的CSRF防御也都在服务端进行。

1：.Cookie Hashing(所有表单都包含同一个伪随机值)：

2：验证码

3：One-Time Tokens(不同的表单包含一个不同的伪随机值)

防止sql注入：

1执行sql语句是使用addslashes进行sql语句转换

2 sql语句书写尽量不要省略双引号和单引号

3 过滤掉sql语句里的关键词 ：update insert delete select \*

4 PHP配置文件中设置register\_global为off，关闭全局变量注册

5 控制错误信息 不要把错误信息输出到浏览器上 应保存在错误日

防止DDOS流量攻击：

1 使用工具:DDoS deflate . 自动查封IP.

2 解析域名到127.0.0.1 让攻击方自己攻击自己

3 把网站做成静态页面

4 限制S YN/ICMP流量

1. 面向过程，面向对象的区别？

把面向过程封装起来就是面向对象

1. php 怎么屏蔽错误?

方法一：@  
 在任何错误语句之前加上@符号，即可屏蔽！  
  
方法二：error\_reporting  
  
 在PHP文件第一行加上：error\_reporting(0); 即可屏蔽！  
  
方法三：display\_errors  
 打开[php.ini](https://www.baidu.com/s?wd=php.ini&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1d9myF-PWI9rADknWmvmW0k0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnWfsrjfvrjcYn1mYnjR3rj6Y" \t "https://zhidao.baidu.com/question/_blank)文件，搜索display\_errors = on,默认的应该是on，改为off 即可屏蔽！

1. memcache 的原理?

在Web服务开发中，服务端缓存是服务实现中所常常采用的一种提高服务性能的方法。其通过记录某部分计算结果来尝试避免再次执行得到该结果所需要的复杂计算，从而提高了服务的运行效率。

1. mysql连接池什么时候初始化？

第一次连接数据库时

1. curl 和 file\_get\_contronts()的区别？

**file\_get\_contronts() 只能使用get方式获取数据】**

**curl 模拟多种请求**

1. php超全局变量有哪些？

　$GLOBALS　　储存全局作用域中的变量

　　$\_SERVER　　获取服务器相关信息

　　$\_REQUEST　　获取POST和GET请求的参数

　　$\_POST　　获取表单的POST请求参数

　　$\_GET　　获取表单的GET请求参数

　　$\_FILES　　获取上传文件的的变量

　　$\_ENV　　获取服务器端环境变量的数组

　　$\_COOKIE　　  浏览器cookie的操作

　　　　　　　　　　设置cookie:setcookie(name, value, expire, path, domain);

　　　　　　　　　　获取cookie：$\_COOKIE["user"];

　　　　　　　　　　删除cookie：setcookie("user", "", time()-3600);//设置过期时间

　　$\_SESSION　　服务端session的操作

　　　　　　　　　　使用session前一定要session\_start()启动session

　　　　　　　　　　储存session：$\_SESSION["name"]="King";//数组操作

　　　　　　　　　　销毁session：unset($\_SESSION["name"]);//销毁一个

　　　　　　　　　　session\_destroy()和unset($\_SESSION);//销毁所有的session

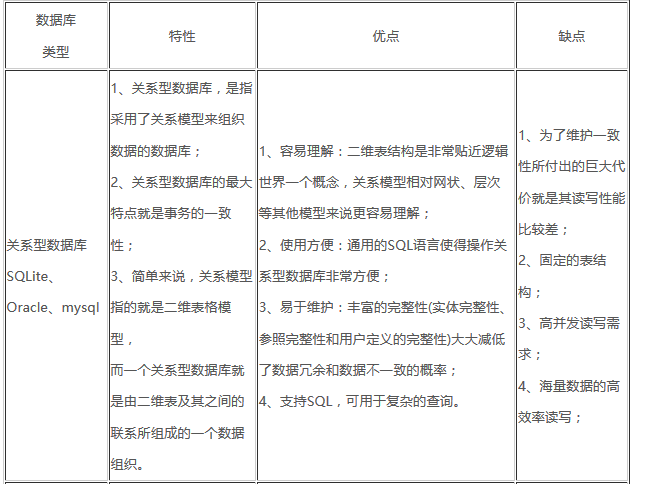
1. tcp和udp的区别？

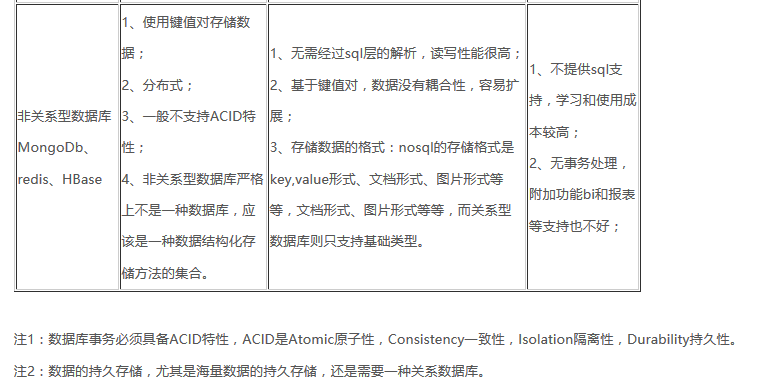
*小结TCP与UDP的区别：  
  1.基于连接与无连接；  
  2.对系统资源的要求（TCP较多，UDP少）；  
  3.UDP程序结构较简单；  
  4.流模式与数据报模式 ；  
  5.TCP保证数据正确性，UDP可能丢包，TCP保证数据顺序，UDP不保证。*

1. myisam 和 innodb 谁支持事物 谁支持外键？

Innodb 支持事务，innodb 支持外键

1. linux awk命令？
2. 关系型数据库非关系数据库的区别?

**

**

1. 说下常见的 http 返回状态码?

所有 HTTP 状态代码及其定义。  
　代码  指示    
2xx  成功    
200  正常；请求已完成。    
201  正常；紧接 POST 命令。    
202  正常；已接受用于处理，但处理尚未完成。    
203  正常；部分信息 — 返回的信息只是一部分。    
204  正常；无响应 — 已接收请求，但不存在要回送的信息。

3xx  重定向    
301  已移动 — 请求的数据具有新的位置且更改是永久的。    
302  已找到 — 请求的数据临时具有不同 URI。    
303  请参阅其它 — 可在另一 URI 下找到对请求的响应，且应使用 GET 方法检索此响应。    
304  未修改 — 未按预期修改文档。    
305  使用代理 — 必须通过位置字段中提供的代理来访问请求的资源。    
306  未使用 — 不再使用；保留此代码以便将来使用。

4xx  客户机中出现的错误

400  错误请求 — 请求中有语法问题，或不能满足请求。    
401  未授权 — 未授权客户机访问数据。    
402  需要付款 — 表示计费系统已有效。    
403  禁止 — 即使有授权也不需要访问。    
404  找不到 — 服务器找不到给定的资源；文档不存在。    
407  代理认证请求 — 客户机首先必须使用代理认证自身。    
415  介质类型不受支持 — 服务器拒绝服务请求，因为不支持请求实体的格式。

5xx  服务器中出现的错误

500  内部错误 — 因为意外情况，服务器不能完成请求。    
501  未执行 — 服务器不支持请求的工具。    
502  错误网关 — 服务器接收到来自上游服务器的无效响应。    
503  无法获得服务 — 由于临时过载或维护，服务器无法处理请求。

-------------------------------------------------------------------------------

HTTP 400 - 请求无效

HTTP 401.1 - 未授权：登录失败

HTTP 401.2 - 未授权：服务器配置问题导致登录失败   
HTTP 401.3 - ACL 禁止访问资源   
HTTP 401.4 - 未授权：授权被筛选器拒绝   
HTTP 401.5 - 未授权：ISAPI 或 CGI 授权失败

HTTP 403 - 禁止访问

HTTP 403 - 对 Internet 服务管理器 (HTML) 的访问仅限于 Localhost   
HTTP 403.1 禁止访问：禁止可执行访问   
HTTP 403.2 - 禁止访问：禁止读访问   
HTTP 403.3 - 禁止访问：禁止写访问   
HTTP 403.4 - 禁止访问：要求 SSL   
HTTP 403.5 - 禁止访问：要求 SSL 128   
HTTP 403.6 - 禁止访问：IP 地址被拒绝   
HTTP 403.7 - 禁止访问：要求客户证书   
HTTP 403.8 - 禁止访问：禁止站点访问   
HTTP 403.9 - 禁止访问：连接的用户过多   
HTTP 403.10 - 禁止访问：配置无效   
HTTP 403.11 - 禁止访问：密码更改   
HTTP 403.12 - 禁止访问：映射器拒绝访问   
HTTP 403.13 - 禁止访问：客户证书已被吊销   
HTTP 403.15 - 禁止访问：客户访问许可过多   
HTTP 403.16 - 禁止访问：客户证书不可信或者无效   
HTTP 403.17 - 禁止访问：客户证书已经到期或者尚未生效

HTTP 404.1 - 无法找到 Web 站点   
HTTP 404 - 无法找到文件   
HTTP 405 - 资源被禁止   
HTTP 406 - 无法接受   
HTTP 407 - 要求代理身份验证   
HTTP 410 - 永远不可用   
HTTP 412 - 先决条件失败   
HTTP 414 - 请求 - URI 太长

HTTP 500 - 内部服务器错误

HTTP 500.100 - 内部服务器错误 - ASP 错误   
HTTP 500-11 服务器关闭   
HTTP 500-12 应用程序重新启动   
HTTP 500-13 - 服务器太忙   
HTTP 500-14 - 应用程序无效   
HTTP 500-15 - 不允许请求 global.asa   
Error 501 - 未实现   
HTTP 502 - 网关错误

1. php 语言的优势?

开源免费跨平台

1. hash 索引主要是为了解决什么问题？

hash索引仅满足“=”、“IN”和“<=>”查询，不能使用范围查询

hash索引无法被用来进行数据的排序操作

对于组合索引，Hash 索引在计算 Hash 值的时候是组合索引键合并后再一起计算 Hash 值，而不是单独计算 Hash 值，所以通过组合索引的前面一个或几个索引键进行查询的时候，Hash 索引也无法被利用

Hash 索引遇到大量Hash值相等的情况后性能并不一定就会比B-Tree索引高

1. rbac概念？如何优化？

权限管理控制，简化

1. composer 是什么？

Composer 是 PHP5.3以上 的一个依赖管理工具。它允许你声明项目所依赖的代码库，它会在你的项目中为你安装他们。Composer 不是一个包管理器。是的，它涉及 "packages" 和 "libraries"，但它在每个项目的基础上进行管理，在你项目的某个目录中（例如 vendor）进行安装。默认情况下它不会在全局安装任何东西。因此，这仅仅是一个依赖管理。

1. 负载均衡 有哪几种方式？

任何的负载均衡技术都要想办法建立某种一对多的映射机制: 一个请求的入口映射到多个处理请求的节点，从而实现分而治之

* DNS轮询
* CDN
* IP负载均衡

1. 实现session 存入 redis 需要设置什么参数？

http://blog.csdn.net/xkweiguang/article/details/51854343

1. sphinx 的实现原理？

Sphinx的整个工作流程就是Indexer程序到数据库里面提取数据，对数据进行分词，然后根据生成的分词生成单个或多个索引，并将它们传递给searchd程序。然后客户端可以通过API调用进行搜索。

1. xss全名？

[跨站脚本攻击](https://baike.baidu.com/item/%E8%B7%A8%E7%AB%99%E8%84%9A%E6%9C%AC%E6%94%BB%E5%87%BB" \t "https://baike.baidu.com/item/xss/_blank)(Cross Site Scripting)，为了不和[层叠样式表](https://baike.baidu.com/item/%E5%B1%82%E5%8F%A0%E6%A0%B7%E5%BC%8F%E8%A1%A8" \t "https://baike.baidu.com/item/xss/_blank)(Cascading Style Sheets, [CSS](https://baike.baidu.com/item/CSS/5457" \t "https://baike.baidu.com/item/xss/_blank))的缩写混淆，故将跨站脚本攻击缩写为XSS。恶意攻击者往Web页面里插入恶意Script代码，当用户浏览该页之时，嵌入其中Web里面的Script代码会被执行，从而达到恶意攻击用户的目的。

1. pdo 和 mysqli 的区别？

PDO和MySQLi都是通过面向对象的形式提供API，但是同时MySQLi也提供了面向过程的API，

PDO比MySQLi最大的优点就是PDO支持很多种数据库，而MySQLi只支持MySQLi。

PDO能够支持其它非MySQL的数据库，而MySQLi专门针对MySQL设计的，所以MySQLi相对于PDO性能稍微好一些。但是PDO和MySQLi都还是没有PHP原生的MySQL扩展快。

1. tcp 与udp 的 应用场景？

*Tcp：  
  http协议是基于tcp协议  
  tcp协议是传输控制协议*

*udp  
  udp是用户数据报协议  
  UDP信息包的标题很短，只有8个字节，相对于TCP的20个字节信息包的额  外开销很小。  
  UDP是面向报文的*

*小结TCP与UDP的区别：  
  1.基于连接与无连接；  
  2.对系统资源的要求（TCP较多，UDP少）；  
  3.UDP程序结构较简单；  
  4.流模式与数据报模式 ；  
  5.TCP保证数据正确性，UDP可能丢包，TCP保证数据顺序，UDP不保证。*

1. 应用层协议了解么？

应用层协议(application layer protocol)定义了运行在不同端系统上的应用程序进程如何相互传递报文。

https://zhidao.baidu.com/question/578099657.html

1. mysql 内置函数 及 想获取当前时间 只要时分秒？

mysql获取当前时间可以使用CURRENT\_TIMESTAMP，用法：SELECT CURRENT\_TIMESTAMP( ) 或 SELECT CURRENT\_TIMESTAMP，

年月日 时分秒：select DATE\_FORMAT(now(),'%Y-%m-%d %T') from dual;

时分秒：select DATE\_FORMAT(now(),'%T') from dual;

或者

年月日 时分秒 select now()

年月日 select curdate

时分秒 select curtime()

1. memcache 内存满了怎么办？

要么就增加内存空间，要么将一些临时用的数据操作完之后，立即销毁，避免占内存

1. linux 上 实时监控一个文件 命令 tail –f？

tail命令

1. fopen 所有参数 含义？

**

1. 什么是管道符？

管道是把一个命令的输出作为下一个命令的输入，类似linux中管道服的

ps -aux | grep

1. 魔术常量？

　 \_\_LINE\_\_　　文件中的当前行号。

　　\_\_FILE\_\_　　文件的完整路径和文件名。如果用在被包含文件中，则返回被包含的文件名。

　　\_\_DIR\_\_　　文件所在的目录。如果用在被包括文件中，则返回被包括的文件所在的目录。

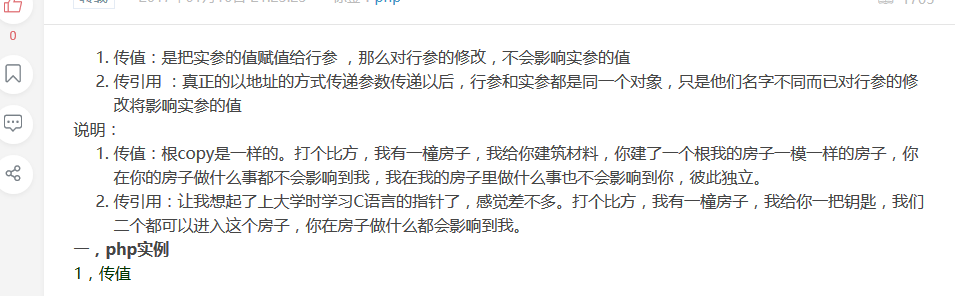
　　\_\_FUNCTION\_\_　　常量返回该函数被定义时的名字 \_\_function\_\_

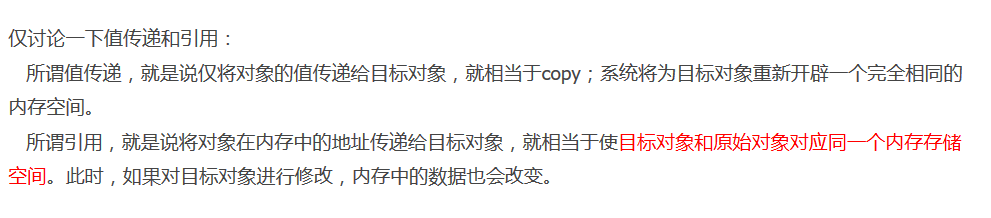
　　\_\_CLASS\_\_　　常量返回该类被定义时的名字（区分大小写）。

　　\_\_METHOD\_\_　　类的方法名（PHP 5.0.0 新加）。返回该方法被定义时的名字（区分大小写）。

　　\_\_NAMESPACE\_\_　　当前命名空间的名称（区分大小写）。此常量是在编译时定义的（PHP 5.3.0 新增）。

1. php的 按值传递 和 引用值传递？





1. 说出两个聚合函数？

聚合函数是对一组值执行计算并返回单一的值的函数，它经常与[SELECT语句](https://www.baidu.com/s?wd=SELECT%E8%AF%AD%E5%8F%A5&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1dBPH9BPvc1nW-9PWT3uHTz0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3En16LPjnYnW0d" \t "https://zhidao.baidu.com/question/_blank)的GROUP BY子句一同使用，SQL SERVER 中具体有哪些聚合函数呢？我们来一一看一下：  
1. AVG 返回指定组中的平均值，空值被忽略。  
 例：select prd\_no,avg(qty) from sales group by prd\_no  
   
2. COUNT 返回指定组中项目的数量。  
 例：select count(prd\_no) from sales   
   
3. MAX 返回指定数据的最大值。  
 例：select prd\_no,max(qty) from sales group by prd\_no   
   
4. MIN 返回指定数据的最小值。  
 例：select prd\_no,min(qty) from sales group by prd\_no  
   
5. SUM 返回指定数据的和，只能用于数字列，空值被忽略。  
 例：select prd\_no,sum(qty) from sales group by prd\_no  
   
6. COUNT\_BIG 返回指定组中的项目数量，与[COUNT函数](https://www.baidu.com/s?wd=COUNT%E5%87%BD%E6%95%B0&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1dBPH9BPvc1nW-9PWT3uHTz0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3En16LPjnYnW0d" \t "https://zhidao.baidu.com/question/_blank)不同的是COUNT\_BIG返回bigint值，而COUNT返回的是int值。  
 例：select count\_big(prd\_no) from sales  
   
7. GROUPING 产生一个附加的列，当用CUBE或ROLLUP运算符添加行时，输出值为1.当所添加的行不是由CUBE或ROLLUP产生时，输出值为0.  
 例：select prd\_no,sum(qty),grouping(prd\_no) from sales group by prd\_no with rollup  
   
8. BINARY\_CHECKSUM 返回对表中的行或表达式列表计算的二进制校验值，用于检测表[中行](https://www.baidu.com/s?wd=%E4%B8%AD%E8%A1%8C&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1dBPH9BPvc1nW-9PWT3uHTz0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3En16LPjnYnW0d" \t "https://zhidao.baidu.com/question/_blank)的更改。  
 例：select prd\_no,binary\_checksum(qty) from sales group by prd\_no  
   
9. CHECKSUM\_AGG 返回指定数据的校验值，空值被忽略。  
 例：select prd\_no,checksum\_agg(binary\_checksum(\*)) from sales group by prd\_no  
   
10. CHECKSUM 返回在表的行上或在表达式列表上计算的校验值，用于生成哈希索引。  
   
11. STDEV 返回给定表达式中所有值的统计标准偏差。  
 例：select stdev(prd\_no) from sales  
   
12. STDEVP 返回给定表达式中的所有值的填充统计标准偏差。  
 例：select stdevp(prd\_no) from sales  
   
13. VAR 返回给定表达式中所有值的统计方差。  
 例：select var(prd\_no) from sales  
   
14. VARP 返回给定表达式中所有值的填充的统计方差。  
 例：select varp(prd\_no) from sales

1. 通过 session 入库 实现 session共享 后 在第一次设置免登陆 的情况 下怎么确保下次登录？

在cookie设置时间或查数据库

1. linux 查看文件内容命令？

Cat

1. 说下你理解的线程与进程？

1。单进程单线程：一个人在一个桌子上吃菜。  
2。单进程多线程：多个人在同一个桌子上一起吃菜。  
3。多进程单线程：多个人每个人在自己的桌子上吃菜。

**进程和线程都是一个时间段的描述，是CPU工作时间段的描述，不过是颗粒大小不同**

 定义方面：进程是程序在某个数据集合上的一次运行活动；线程是进程中的一个执行路径。

 角色方面：在支持线程机制的系统中，进程是系统资源分配的单位，线程是系统调度的单位。

 资源共享方面：进程之间不能共享资源，而线程共享所在进程的地址空间和其它资源。同时线程还有自己的栈和栈指针，程序计数器等寄存器。

 独立性方面：进程有自己独立的地址空间，而线程没有，线程必须依赖于进程而存在。

1. session默认的生存时间？

php中session过期时间设置

1. 修改php配置文件中的session.gc\_maxlifetime
2. ＜?php  
   session\_start();   
   // 保存一天   
   $lifeTime = 24 \* 3600;   
   setcookie(session\_name(), session\_id(), time() + $lifeTime, "/");   
   ?＞
3. 其实 Session 还提供了一个函数 session\_set\_cookie\_params(); 来设置 Session 的生存期的，该函数必须在 session\_start() 函数调用之前调用：  
   ＜?php   
   // 保存一天   
   $lifeTime = 24 \* 3600;   
   session\_set\_cookie\_params($lifeTime);   
   session\_start();  
   $\_SESSION["admin"] = true;   
   ?＞
4. 单点登录的实现原理？

*单点登录的技术实现机制：当用户第一次访问应用系统1的时候，因为还没有登录，会被引导到认证系统中进行登录;根据用户提供的登录信息，认证系统进行身份效验，如果通过效验，应该返回给用户一个认证的凭据--ticket;用户再访问别的应用的时候，就会将这个ticket带上，作为自己认证的凭据，应用系统接受到请求之后会把ticket送到认证系统进行效验，检查ticket的合法性。如果通过效验，用户就可以在不用再次登录的情况下访问应用系统2和应用系统3了。*

1. yii自带的操作方法？
2. 说下你对于命名空间的理解？

目的：解决重名问题，PHP中不允许两个函数或者类出现相同的名字，否则会产生一个致命的错误。

命名空间将代码划分出不同的空间（区域），每个空间的常量、函数、类的名字互不影响

1. 查看主从库的状态 命令？

Show master status\G

1. 用过事务吗？及事务的应用场景？

添加订单、银行转账

**BEGIN** 开始一个事务

**ROLLBACK** 事务回滚

**COMMIT** 事务确认

1. 输入俩人的名字 , 求出 俩人的缘分值？

MD5

1. mysql 临时表 派生表？

派生表是从[SELECT语句](http://www.yiibai.com/mysql/select-statement-query-data.html" \t "http://www.yiibai.com/mysql/_blank" \o "SELECT语句)返回的虚拟表。派生表类似于临时表，但是在SELECT语句中使用派生表比[临时表](http://www.yiibai.com/mysql/temporary-table.html" \t "http://www.yiibai.com/mysql/_blank" \o "临时表)简单得多，因为它不需要创建临时表的步骤。

**术语:\***派生表\*和[子查询](http://www.yiibai.com/mysql/subquery.html" \t "http://www.yiibai.com/mysql/_blank" \o "子查询)通常可互换使用。当SELECT语句的FROM子句中使用独立子查询时，我们将其称为派生表。

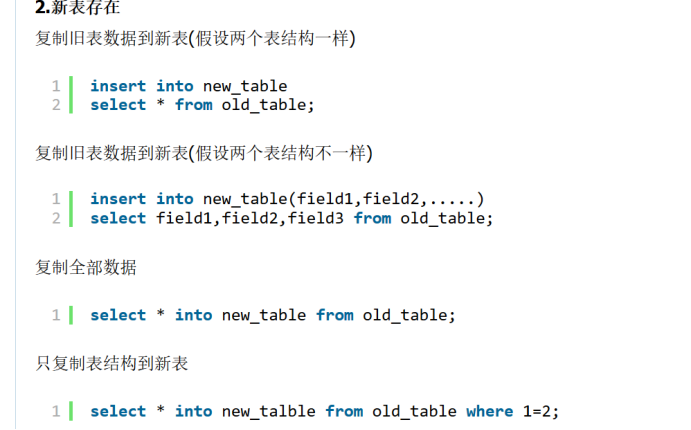
1. linux 查看内存空间命令？

top

1. 往一张数据量比较大的表中添加字段，怎么添加？

先新建一张表，然后把旧表中的数据copy到新表中，





1. memcache 和 memcached 的区别

PHP有两个memcache客户端：php memcache和php memcached。 memcache完全在PHP框架内开发的，提供了memcached的接口，memecached扩展是使用了libmemcached库提供的api与memcached服务端进行交互。 memcache提供了面向过程及面向对象的接口，memached只支持面向对象的接口。 memcached 实现了更多的 memcached 协议。 memcached 支持 Binary Protocol，而 memcache 不支持，意味着 memcached 会有更高的性能。

73、防止session\_id泄露的方法

1. 每次请求都会判断客户端是否存在session\_id吗

En

1. 虚拟内存是计算机系统内存管理的一种技术。它使得应用程序认为它拥有连续的可用的内存（一个连续完整的地址空间），而实际上，它通常是被分隔成多个物理内存碎片，还有部分暂时存储在外部磁盘存储器上，在需要时进行数据交换。目前，大多数操作系统都使用了虚拟内存，如Windows家族的“虚拟内存”；Linux的“交换空间”等。
2. 简述浏览器的工作原理：

HTTP通信的流程，大体分为三个阶段：

1、 连接 服务器通过一个ServerSocket类对象对8000端口进行监听，监听到之后建立连接，打开一个socket虚拟文件。

2、 请求 创建与建立socket连接相关的流对象后，浏览器获取请求，为GET请求，则从请求信息中获取所访问的HTML文件名，向服务器发送请求。

3、 应答 服务收到请求后，搜索相关目录文件，若不存在，返回错误信息。若存在，则想html文件，进行加HTTP头等处理后响应给浏览器，浏览器解析html文件，若其中还包含图片，视频等请求，则浏览器再次访问web服务器，异常获取图片视频等，并对其进行组装显示出来。

1. ajax技术利用了什么协议？简述ajax的工作机制

ajax利用地http协议

（1）对象初始化

（2）发送请求

（3）服务器接收请求并进行处理

（4）服务器返回响应数据

（5）客户端接收

（6）依据响应数据修改客户端页面内容

79. http 和 https 的 区别

1.https协议需要到ca申请证书，一般免费证书较少，因而需要一定费用。

　　2、http是超文本传输协议，信息是明文传输，https则是具有安全性的ssl加密传输协议。

　　3、http和https使用的是完全不同的连接方式，用的端口也不一样，前者是80，后者是443。

　　4、http的连接很简单，是无状态的；HTTPS协议是由SSL+HTTP协议构建的可进行加密传输、身份认证的网络协议，比http协议安全。

80 linux 关机命令

1、halt 立刻关机   
2、poweroff 立刻关机   
3、shutdown -h now 立刻关机(root用户使用)   
4、shutdown -h 10 10分钟后[自动关机](https://www.baidu.com/s?wd=%E8%87%AA%E5%8A%A8%E5%85%B3%E6%9C%BA&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1dBuAcknWcduADzuhc3PjcL0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHn1n1TLP1nz" \t "https://zhidao.baidu.com/question/_blank)   
如果是通过shutdown命令设置关机的话，可以用shutdown -c命令取消重启  
推荐使用shutdown命令

1. php 的工作原理?

说到php的运行机制还要先给大家介绍php的模块，PHP总共有三个模块：内核、Zend引擎、以及扩展层；PHP内核用来处理请求、文件流、错误处理等相关操作；Zend引擎（ZE）用以将源文件转换成机器语言，然后在虚拟机上运行它；扩展层是一组函数、类库和流，PHP使用它们来执行一些特定的操作。比如，我们需要mysql扩展来连接MySQL数据库；当ZE执行程序时可能会需要连接若干扩展，这时ZE将控制权交给扩展，等处理完特定任务后再返还；

最后，ZE将程序运行结果返回给PHP内核，它再将结果传送给SAPI层，最终输出到浏览器上。

1. redis集群怎么做的,官方的方案 还是 其他的 。

redis前端分片

85 接口怎么加密？

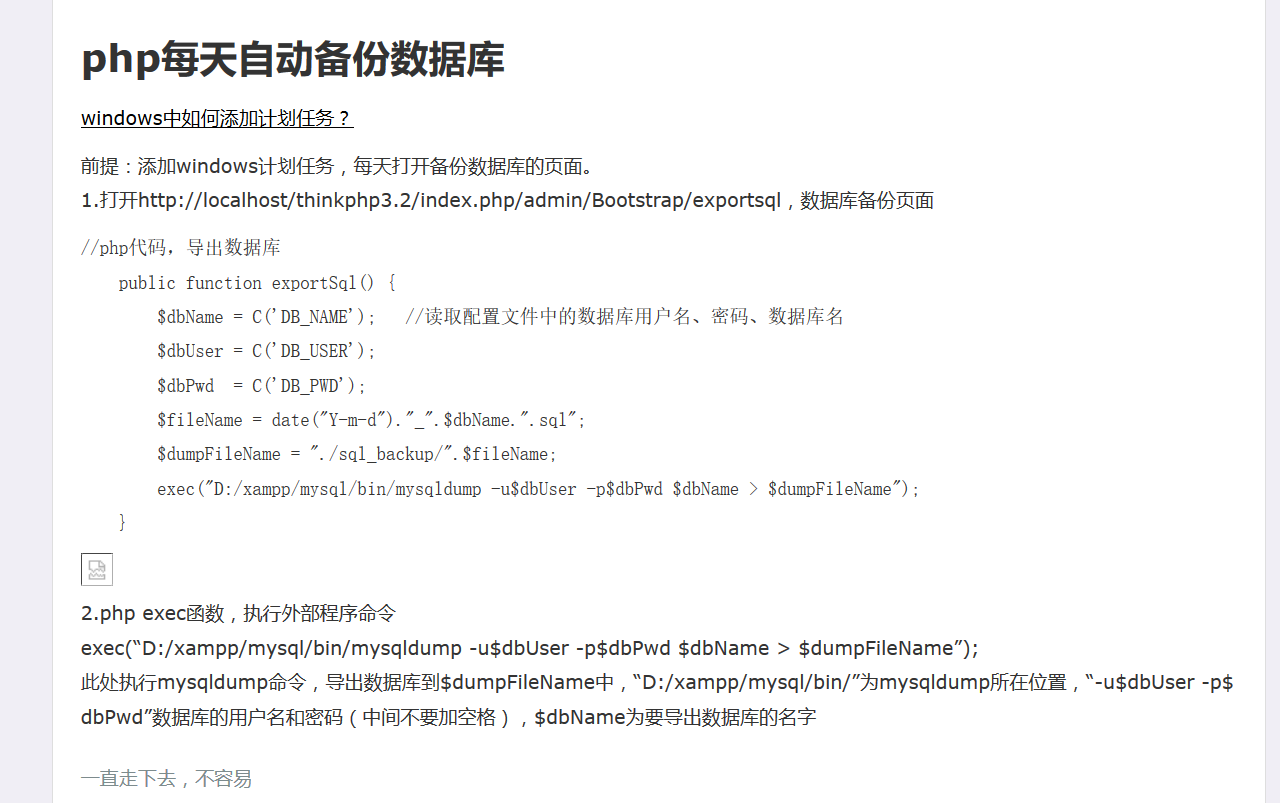
哈希加盐碰撞

1. aof持久化 方案 及 具体两种方案的同步方式

按时间 (默认是1秒) 按操作次数 操作系统

87 支付宝 接口同步显示成功异步失败相信那个？

88. 数据库怎么备份？



1. empty 的返回值？

Empty(0/false/’’/null) 返回的是true

100. include 和 require 的区别？

Include 出错，不会停止。 遇到这个函数才会加载

Require 出错，就会立即停止。 只读取一次 。

102 数组去重了解哪些方式？

Array\_filp(键值互换) array\_unique(直接去重)

1. count函数中能写啥？

Count(\*) count(1) count(clomun)

1. tp5和tp3 的路由？

107 出现死锁了咋办？

死锁的四个条件   
1.**互斥**：其他进程不能访问已分配给其他进程的资源   
2.**占有并等待**：当一个进程等待其他进程时，继续占有已分配资源   
3.**不可抢占**：不能强行占用进程已占有的资源   
4.**循环等待**：存在一个循环的进程链，使得每个进程至少占有此链中下一个进程所需的一个资源

* 解决  
  以上所有方法并没有直接提出解决死锁的方法，而是在抱有侥幸的心理，在系统里飙车。遇到了死锁怎么办？（以下充分展示了操作系统这位辣妈的实力）   
  1.取消所有的死锁进程   
  2.把每个死锁基础回滚到前面定义的某些检查点，并且重新启动所有进程。   
  3.连续取消死锁进程直到不再存在死锁。   
  4.连续抢占资源直到不存在死锁。   
  对于1.而言，这是最简(bao)单(li)的方式，对于2.而言，这需要操作系统的一些机制，不幸的是，这种恢复方式可能再次面临死锁。对于3.和4.都需要一个标准（每次取消代价最小的进程），并且重新调用检测算法，测试是否依旧有死锁存在。对于4.还要回滚到获得资源之前的某个状态。
* 就像上面对每一个方法的描述的那样，各有优缺点；同样，对于每一个进程在不同的情况下，也有各自的特点。既然一刀切，合理的解决每个进程的问题。如果将进程分类，并为每一类进程，配备一个适合的死锁处理算法，这样既能保证进程正常运行，又能提升系统效率。

108 索引覆盖？

索引覆盖是指：如果查询的列恰好是索引的一部分，那么查询只需要在索引文件上进行，不需要回行到磁盘再找数据，这种查询速度非常快，称为“索引覆盖”

1. mysql5.7版本和以前版本有什么不同？？

极大的提高了 innodb 的读取性能，减少了 MDL 锁的开销；改善了 innodb 的读写负载性能， InnoDB 的索引锁替换成了“块锁”

5.7 的 mysql 增加了 JSON 这个数据类型，并内置了 JSON 函数

开启全局事务无需重启服务器。因此，读写的操作在此过程

中是可行的，并且服务器不需要同步或者重启。

mysql5.7 以后默认密码为空将不复存在， mysql 在安装过程中将随机创建一个 root 帐号的密码，并将该密码标记为过期状态。

1. session\_id 和 服务器的session文件的关系？

Session\_id 和session同时生成,并且session文件是以session\_id所命名

1. 二进制文件有哪些格式https://www.2cto.com/database/201703/607569.html
2. 基于段的日志格式 - statement
3. 基于行的日志格式 - row
4. 混合日志格式 - mixed

113.mysql count(1)和count(\*)的区别

ount(\*)对行的数目进行计算，包含NULL  
count(column)对特定的列的值具有的行数进行计算，不包含NULL值。  
count()还有一种使用方式，count(1)这个用法和count(\*)的结果是一样的。  
  
性能问题  
  
1.任何情况下SELECT COUNT(\*) FROM tablename是最优选择；  
2.尽量减少SELECT COUNT(\*) FROM tablename WHERE COL = ‘value’ 这种查询；  
3.杜绝SELECT COUNT(COL) FROM tablename WHERE COL2 = ‘value’ 的出现。  
  
如果表没有主键，那么count（1）比count（\*）快。  
如果有主键，那么count（主键，联合主键）比count（\*）快。  
如果表只有一个字段，count（\*）最快。

count(1)跟count(主键)一样，只扫描主键。count(\*)跟count(非主键)一样，扫描整个表。明显前者更快一些

1. 如何检测php脚本执行的效率

# WebGrind 中间件 或

$start=microtime(true);在运行开始加一个时间  
结束再加一个 $end=microtime(true); $end-$start 得到运行时间秒

115.如何实时查看文件的内容

cat /var/log/\*.log

如果日志在更新，如何实时查看 tail -f /var/log/messages

还可以使用 watch -d -n 1 cat /var/log/messages

-d表示高亮不同的地方，-n表示多少秒刷新一次。

该指令，不会直接返回命令行，而是实时打印日志文件中新增加的内容，这一特性，对于查看日志是非常有效的。如果想终止输出，按 Ctrl+C 即可。

1. 如何把文件的前五行内容写入到另一个文件

>是覆盖 >>是追加

执行head -5 1.tt >2.tt ，就可以将1.tt中的前5行内容保存到2.tt中

117.如何防止接口重复被调用；

设置一个过期时间

1. 服务器优化的好处？

使用缓存改善网站性能

使用应用服务器集群，改善我网站的并发处理能力，使用反向代理和cdn加速网站响应

119.zend引擎在版本几中出现的，解决了什么问题？

第一版的 Zend 引擎在 1999年 伴随著 PHP 第四版问世。它是高度最佳化的后台模组。效能、可靠与延展性是它让PHP更强更大众化的主要原因。

zend engine引擎的出现使得php有了质的提高：

1、把边解释边运行的方式变为先进行预编译（compile），再执行（execute）的方式极大提高了php的运行效率。

2、使得执行效率大幅提高

3、由于实行功能分离，降低了模块间的耦合度，扩展性大大加强

Zend Engine 最主要的特性就是把 PHP 的边解释边执行的运行方式改为先进行预编译(Compile)，然后再执行(Execute)。这两者的分开给 PHP 带来了革命性的变化：执行效率大幅提高；由于实行了功能分离，降低了模块间耦合度，可扩展性也大大增强。此时 PHP 已经能很方便的应付各种各样的 BT 需求了，而伴随 PHP 4.4.x ―可能是 PHP4 系列的最后一个分支―的发布，PHP 的大部分开发人员已经将注意力放在了 PHP5 或者 PHP6 上面，以后发布的基本上就是一些 Bug Fix Release。可以说第一代的 Zend Engine 是已经在站最后一班岗了。

120.了解php 钩子函数吗？

钩子就像一个”陷阱”、”监听器”,当A发送一个消息到B时，当消息还未到达目的地B时，被钩子拦截调出一部分代码做处理，这部分代码也叫钩子函数或者回调函数

作用：

钩子函数可以截获并处理其他应用程序的消息。每当特定的消息发出，在没有到达目的窗口前，钩子程序就先捕获该消息，亦即钩子函数先得到控制权。这时钩子函数即可以加工处理（改变）该消息，也可以不作处理而继续传递该消息，还可以强制结束消息的传递。

121.php内核的 垃圾回收机制 和 内存管理机制？

php的内存管理机制是：预先给出一块空间，用来存储变量，当空间不够时，再申请一块新的空间

1.存储变量名，存在符号表。

2.变量值存储在内存空间。

3.在删除变量的时候，会将变量值存储的空间释放，而变量名所在的符号表不会减小。

php垃圾回收机制

PHP变量存储是存储在一个zval容器里面的

1.类型 2.值 3.is\_ref 代表是否有地址引用 4.refcount 指向该值的变量数量

1.变量赋值的时候：is\_ref为false  refcount为1

.将变量a的值赋给变量b,变量b不会立刻去在内存中存储值，而是先指向变量a的值，一直到变量a有任何操作的时候

.因为程序又操作了变量a，所以变量b会自己申请一块内存将值放进去。所以变量a的zavl容器中refcount会减1变为1,变量c指向a，所以refcount会加1变为2

垃圾回收：

1.在5.2版本或之前版本，PHP会根据refcount值来判断是不是垃圾

如果refcount值为0，PHP会当做垃圾释放掉

这种回收机制有缺陷，对于环状引用的变量无法回收

2.在5.3之后版本改进了垃圾回收机制

如果发现一个zval容器中的refcount在增加，说明不是垃圾

如果发现一个zval容器中的refcount在减少，如果减到了0，直接当做垃圾回收

如果发现一个zval容器中的refcount在减少，并没有减到0，PHP会把该值放到缓冲区，当做有可能是垃圾的怀疑对象。

当缓冲区达到了临界值，PHP会自动调用一个方法去遍历每一个值，如果发现是垃圾就清理

122.集群和分布式部署有什么区别？

分布式：一个业务分拆多个子业务，部署在不同的服务器上

集群：同一个业务，部署在多个服务器上

1. 怎么开启事务？

1、用 BEGIN, ROLLBACK, COMMIT来实现

**BEGIN** 开始一个事务

**ROLLBACK** 事务回滚

**COMMIT** 事务确认

2、直接用 SET 来改变 MySQL 的自动提交模式:

**SET AUTOCOMMIT=0** 禁止自动提交

**SET AUTOCOMMIT=1** 开启自动提交

1. 六种网站优化的方案？
2. **HTML静态化**
3. **图片服务器分离**
4. **数据库集群,库表散列**
5. **缓存**
6. **镜像**
7. **负载均衡**

126.如何在linux 下配置环境变量？

什么是环境变量?环境变量一般是指在操作系统中用来指定操作系统运行环境的一些参数，比如[临时文件夹](https://www.baidu.com/s?wd=%E4%B8%B4%E6%97%B6%E6%96%87%E4%BB%B6%E5%A4%B9&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1dBm1w-mHDLuADsrj79mhR10ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHR1PWcdPWfd" \t "https://zhidao.baidu.com/question/_blank)位置和系统文件夹位置等。这点有点类似于DOS时期的默认路径，当你运行某些程序时除了在当前文件夹中寻找外，还会到设置的默认路径中去查找。简单地说这里的“Path”就是一个变量，里面存储了一些常用命令所存放的目录路径。

***环境变量都有哪些类型****环境变量有很多种类一般有2种划分形式按变量的周期划分*

***永久变量****：*

*通过修改配置文件，配置之后变量永久生效*

***临时性变量****：*

*使用命令如export等命令设置，设置之后马上生效。当关闭shell的时候失效（这种主要用于测试比较多）*

127.怎么分区？

**list :条件值为一个数据列表。**

**Range（范围）**

这种模式允许将数据划分不同范围。例如可以将一个表通过年份划分成若干个分区

**Hash**(**按照某个表达式的值进行取余)**

128.PHP的运行原理？

*代码  
        请求------>apache(服务器)------>php--->zend api(扩展)  
                                                      |   
                                                      |  
               优化：真静态 opcode缓存              zend(内核)  
                                                      |  
                                                     |  
                                       结果 <-------opcode(php的机器语言，编译成php能解析的)*

129.tcp 和 upd 的应用场景？

从特点上我们已经知道，TCP 是可靠的但传输速度慢 ，UDP 是不可靠的但传输速度快。因此在选用具体协议通信时，应该根据通信数据的要求而决定。   
　　若通信数据完整性需让位与通信实时性，则应该选用 TCP 协议（如文件传输、重要状态的更新等）；反之，则使用 UDP 协议（如视频传输、实时通信等）。

130.mysql 的运行原理？

*sql语句 ---->分析器----->优化器  
                 |                  |  
         （先判断有没有缓存）                          |   
             mysql缓存             可执行代码    
                  |                 |  
                                   |                                   
                 结果  <------存储引擎（myisam(查) /innodb(增删改 事物)）*

首先程序的请求会通过mysql的connectors与其进行交互，请求到处后，会暂时存放在连接池（connection pool)中并由处理器（Management Serveices & Utilities）管理。当该请求从等待队列进入到处理队列，管理器会将该请求丢给SQL接口（SQL Interface）。SQL接口接收到请求后，它会将请求进行hash处理并与缓存中的结果进行对比，如果完全匹配则通过缓存直接返回处理结果；否则，需要完整的走一趟流程：

   (1)由SQL接口丢给后面的解释器（Parser），上面已经说到，解释器会判断SQL语句正确与否，若正确则将其转化为数据结构。

   (2)解释器处理完，便来到后面的优化器（Optimizer），它会产生多种执行计划,最终数据库会选择最优化的方案去执行,尽快返会结果。

   (3)确定最优执行计划后，SQL语句此时便可以交由存储引擎（Engine）处理，存储引擎将会到后端的存储设备中取得相应的数据，并原路返回给程序。

1. 文件锁 排他锁？

可以使用 LOCK\_SH 或 LOCK\_EX 常量，分别对应共享锁和排他锁

当使用 LOCK\_SH 共享锁时，多个进程可以都使用共享锁锁定同一个文件，从而实现多个进程对文件的并行读取。由此可见，LOCK\_SH 共享锁类似于多线程读写锁 pthread\_rwlock\_t 类型中的读锁。当使用LOCK\_EX 排他锁时，同一时刻只能有一个进程锁定成功，其余进行只能阻塞（阻塞与非阻塞在下文介绍），这种行为与多线程读写锁中的写锁类似。

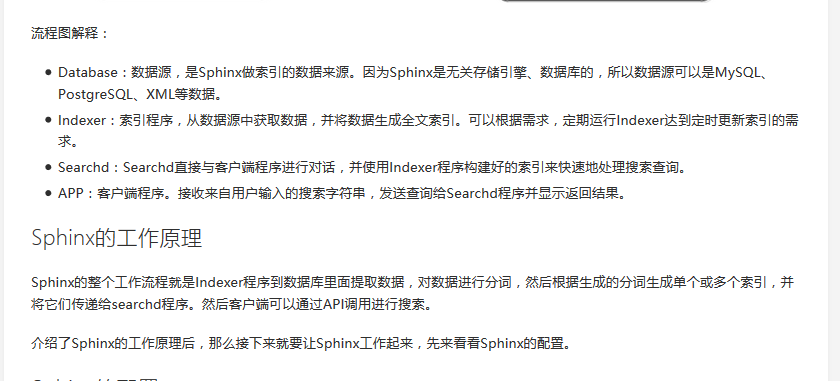
132.什么是hash？https://baike.baidu.com/item/Hash/390310?fr=aladdin

Hash，一般翻译做“散列”，也有直接音译为“哈希”的，就是把任意长度的[输入](https://baike.baidu.com/item/%E8%BE%93%E5%85%A5/5481954" \t "https://baike.baidu.com/item/Hash/_blank)（又叫做预映射， pre-image），通过散列算法，变换成固定长度的[输出](https://baike.baidu.com/item/%E8%BE%93%E5%87%BA" \t "https://baike.baidu.com/item/Hash/_blank)，该输出就是散列值。这种转换是一种压缩映射，也就是，散列值的空间通常远小于输入的空间，不同的输入可能会散列成相同的输出，所以不可能从散列值来唯一的确定输入值。简单的说就是一种将任意长度的消息压缩到某一固定长度的[消息摘要](https://baike.baidu.com/item/%E6%B6%88%E6%81%AF%E6%91%98%E8%A6%81" \t "https://baike.baidu.com/item/Hash/_blank)的函数。

133.一个文件的文件权限是 rwx rwx rwx每个位置所对应的是谁的权限？

用户 本组用户 其他用户

135.怎么使用的sphinx sphinx 的原理？

**

136.事务的隔离级别？

*现在来看看MySQL数据库为我们提供的四种隔离级别：*

*① Serializable (串行化)：可避免脏读、不可重复读、幻读的发生。*

*② Repeatable read (可重复读)：可避免脏读、不可重复读的发生。*

*③ Read committed (读已提交)：可避免脏读的发生。*

*④ Read uncommitted (读未提交)：最低级别，任何情况都无法保证。*

137.你了解哪些中间件?

# Mha（主从复制） keepalived（双机热备）  atlas（读写分离） WebGrind(php执行效率性能测试)

138.读写操作怎么分离 就是问的怎么配置的能让他该访问主库的访主库，该访问从库的访从库?

atlas

139.项目在进行读写分离的时候,程序如何区分是读还是写??

判断 或用插件 atlas

140.主主复制两条数据同时往一张表插入数据，怎么防止id 冲突?

判断奇偶

141.现在要做一个红包功能，怎么保证是数字随机的且加起来的数字符合设定的数值?

1. 一级域名如何和子域名共享session?

同服务器，session保存位置相同，只要sessionid一样就行了，

1. 把sessionid保存在cookie里
2. 用session\_set\_cookie\_params();

143.Echo print printf print\_r的区别？

Echo和print都是属于语言结构，

Echo打印字符串，没有返回值，

Print 打印字符串，返回int类型

Print\_r打印字符串,返回值是布尔类型

Printf是出格式化的字符串

1. 支付流程

Sdk 主要是签名

1. 第三方登陆流程

比如qq : 申请第三方接入，会给你一个id 和key ,填写一个回掉地址，通过aouth协议，第一次用户未授权请求返回个code 第二次用户授权请求 把id key code 传过去，如果验证通过返回个access tokey

并登陆成功跳到回掉地址

1. 注册到登陆流程
2. 购物车流程
3. redis事物

Multi exec watch

.148. 管理员20分钟没有操作需要重新登录功能

 采用session“垃圾回收机制”可以设置时间限制 在这路设置二十分钟后 当未进行页面操作时触发这个回收机制 连续二十分钟未操作后 对于登陆记录进行销毁 销毁后进行重新登陆 如果操作 就不会触发

 在php判断页面鼠标键盘无操作，就可以了

1. 同一账户不同用户登录互踢功能

Sessionid 入库 或者token

150. 管理员三次密码输入失败隔天登录功能

用户登录时将用户登录错误次数记录到表中或者缓存中一天的次数大于三次限制登录第二天清空缓存或表

1. 记录访问次数及独立ip（pv，uv）

找到ip访问日志：写脚本

查ip 访问次数

 [root@qunar logs]# cat a.sh   
#!/bin/bash

#将28/Jan/2015全天的访问日志放到a.txt文本  
cat access.log |sed -rn '/28\/Jan\/2015/p' > a.txt

#统计a.txt里面有多少个ip访问  
cat a.txt |awk '{print $1}'|sort |uniq > ipnum.txt

#通过shell统计每个ip访问次数  
for i in `cat ipnum.txt`  
do   
iptj=`cat  access.log |grep $i | grep -v 400 |wc -l`  
echo "ip地址"$i"在2015-01-28日全天(24小时)累计成功请求"$iptj"次，平均每分钟请求次数为："$(($iptj/1440)) >> result.txt  
done

**执行脚本：**

 [root@qunar logs]#sh a.sh

1. 不允许超级管理员同时登录,如果管理员登录时已经有其他管理员登陆了,则提示"您已通过其他浏览器登录,请退出后再登录或者强制下线",如果点击"强制下线"则进入后台管理中心而以前登录的管理员下线,简述实现思路
2. 创建一个数据表 里面存有用户名 密码 session\_id
3. 检查该账号是否登录成功然后判断此次登录的session\_id与库中的是否一致如果一致是一个浏览器登录如果不是证明是其他浏览器
4. 如果是登陆中提示,则提示"您已通过其他浏览器登录,请退出后再登录或者强制下线",如果点击"强制下线"删除该用户在表中的session\_id
5. 邮箱激活功能,前台用户注册后需要登录邮箱激活账号才能够登录的,请简述激活过程,重点是安全性。
6. 在提交了注册信息后，同时在数据库中写入激活码，同时程序把激活码（链接）发送到注册者的信箱里，注册者点击激活链接后，程序会和数据库里的数据进行对比，对比通过即完成激活。
7. 每次有新用户注册，记录并验证 IP、MAIL 地址注册次数。 2、发激活 MAIL 后记录此地址发信时间。
8. 目的只有一个，就是防止有人用激活邮件做为攻击别人邮箱的工具。
9. 在激活页面提供登陆渠道: 在邮箱中点链接之后进入激活页面，忽略cookie之类的信息，让用户登陆一次，根据用户名、密码和激活码三个来激活账号
10. 给这类注册链接设置一个过期时间，一般1到3天较为妥当，这样可以尽量避免误触发的情况
11. 绑定手机号功能,注册后在个人中心有一个绑定手机号的菜单,输入手机号码,手机接收短信验证码(有效期5分钟),,简述实现思路。
12. 页面一个验证码文本输入框，后面跟一个“获取验证码”按钮，同时限制按钮的点击时间，一般是限制1分钟才能点击一次，点击按钮，将用户ID，生成的验证码，存入数据库，接下来调用短信发送接口将信息发送出去，表单提交时，判断该验证码是否正确（是否在表中存在同时验证码是否失效，一般是五分钟内失效）
13. 总体思路
14. 时效限制: [5-10min]，使用次数限制: 1次，IP次数限制: 防止恶意刷手机验证码短信，手机号限制: 防止短信轰炸，跨域请求限制: 进一步限制恶意刷短信，<span style="font-family: Arial, Helvetica, sans-serif;">验证码验证: 进一步限制恶意刷短信</span>
15. 前端优化

图片：在页面上图片实际大小与展示大小保持一致

图片延迟加载

Js/css:压缩 js尽量不要往头部加载，分布规律（js代码回阻塞加载） 合并加载（用一个http请求把js css 全部加载进来）

Cdn部署

1. 订单过期处理

定时任务

1. 依赖注入，控制反转

控制反转：

1. Php缓存有几个

全页面静态化缓存 页面部分缓存 数据缓存 查询缓存 按内容变更进行缓存 内存式缓存 apache缓存模块 apache缓存模块 php APC缓存扩展 Opcode缓存

1. 数据库触发器？

触发器是一种用来保障参照完整性的特殊的存储过程，它维护不同表中数据间关系的有关规则。当对指定的表进行某种特定操作（如：Insert,Delete或Update）时，触发器产生作用。触发器可以调用存储过程。

触发器：监视某种情况，并触发某正操作

四要素：监视地点，监视事件，出发时间，触发事件



1. 数据库存储过程？

存储过程是一组为了完成特定功能的SQL语句集，是利用SQL Server所提供的Transact-SQL语言所编写的程序。经编译后存储在数据库中。存储过程是数据库中一个重要的对象

存储过程是为了完成某个数据库中的特定功能而编写的语句集，该语句集包括SQL语句（对数据的增删改查）、条件语句和循环语句等。

160.数据存放文件中 PHP引入

File\_put\_contents

161.项目写完后你怎么弄到线上？

GIT 自动化部署

162.线上代码运行正常，线下出现空白怎么办

环境 服务器

163.面向对象

1. redis是如何切库的

165.用户请求都存储在队列时怎么触发队列

异步请求

166.redis缓存数据的时候怎么缓存到指定的文件夹

167.设置一个签到领金币

数据库设计（签到次数 签到时间），每次判断上次签到时间是不是在昨天的范围之内

168.http 的数据传输方式

1）GET：获取资源

GET方法用来请求访问已被URL识别的资源

2）POST：传输实体主体

POST方法用来请求服务器传输信息实体的主体

169.单利模式工厂模式应用场景

数据库

170.敏感词过滤

171.laravel的第三方类库放在哪

vendor

172.yii的钩子

Yii 框架里面的ar模型支持before、after这样的钩子功能，

例如：beforeFind()就是在使用find这一系列方法的前面执行该函数。

173.单元测试

**phpunit**

174.json和xml区别

Xml 可读性强;json 生成数据方便 传输速度快

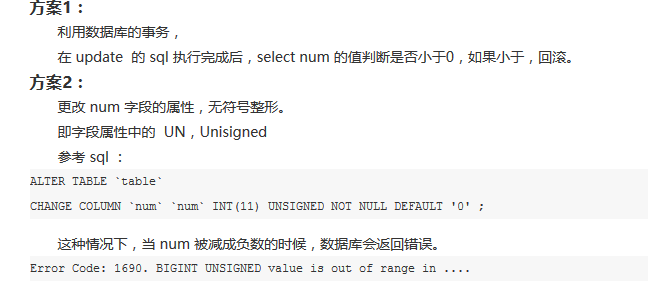
175.防盗链

判断请求的url后缀是不是图片，如果是就阻止访问

1. cookie存购物车最大存多少

4kb

1. 怎么设置数据字段不能是负数



1. tp3.2支持几种路由





179.autoload什么情况下面用，

当调用未实例化类时自动调用

180.explain里面的参数

http://blog.csdn.net/zhuxineli/article/details/14455029

181.ACID的应用场景

Mysql

182.innodb为什么支持事物

innodb事务可以分为以下几种：

1.扁平事务    2.带有保存点的扁平事务    3.链事务    4.嵌套事务   5.分布式事务

1. 怎么改session存储位置、

Php.ini

1. msvc
2. 代码发布工具

1. 支付宝异步失败怎么办
2. Rest ful协议

一种软件架构风格，设计风格而**不是**标准，只是提供了一组设计原则和约束条件。它主要用于客户端和服务器交互类的软件。

REST 指的是一组架构[约束条件](https://baike.baidu.com/item/%E7%BA%A6%E6%9D%9F%E6%9D%A1%E4%BB%B6" \t "https://baike.baidu.com/item/RESTful/_blank)和原则。满足这些约束条件和原则的应用程序或设计就是 RESTful。

Web 应用程序最重要的REST 原则是，客户端和服务器之间的交互在请求之间是无状态的。从客户端到服务器的每个请求都必须包含理解请求所必需的信息

1. App和pc端的区别，怎么实现后台一致？

189.乐观锁和悲观锁的区别

悲观锁(Pessimistic Lock), 顾名思义，就是很悲观，每次去拿数据的时候都认为别人会修改，所以每次在拿数据的时候都会上锁，这样别人想拿这个数据就会block直到它拿到锁。传统的关系型数据库里边就用到了很多这种锁机制，比如行锁，表锁等，读锁，写锁等，都是在做操作之前先上锁。它指的是对数据被外界（包括本系统当前的其他事务，以及来自外部系统的事务处理）修改持保守态度，因此，在整个数据处理过程中，将数据处于锁定状态。悲观锁的实现，往往依靠数据库提供的锁机制（也只有数据库层提供的锁机制才能真正保证数据访问的排他性，否则，即使在本系统中实现了加锁机制，也无法保证外部系统不会修改数据）。  
  
        乐观锁(Optimistic Lock), 顾名思义，就是很乐观，每次去拿数据的时候都认为别人不会修改，所以不会上锁，但是在更新的时候会判断一下在此期间别人有没有去更新这个数据，可以使用版本号等机制。乐观锁适用于多读的应用类型，这样可以提高吞吐量，像数据库如果提供类似于write\_condition机制的其实都是提供的乐观锁。

1. php为什么是弱类型语言？和其他语言的区别?

弱类型定义变量是没有那么严格

动态弱类型的好处是比较灵活.

无类型： 汇编  
弱类型、静态类型 ： C/C++  
弱类型、动态类型检查： Perl/PHP  
强类型、静态类型检查 ：Java/C#  
强类型、动态类型检查 ：Python, Scheme  
静态**显式**类型 ：Java/C  
静态隐式类型 ：Ocaml, Haskell

1. 运算符优先级

算术运算符(+ - \* /)》判断运算符(< > <> !=)》逻辑运算符(|| &&)》赋值运算符(=)

1. Curl(模拟get和post请求)代码实现

1 Get方式实现

复制代码 代码如下:

　　//初始化

　　$ch = curl\_init();

　　//设置选项，包括URL

　　curl\_setopt($ch, CURLOPT\_URL, "http://www.jb51.net");

　　curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, 1);

crul\_setopt($ch,CURLOPT\_SSL\_VERIFYPEER,FALSE);//绕过https请求的验证

crul\_setopt($ch,CURLOPT\_SSL\_VERIFYHOST,FALSE);

　　curl\_setopt($ch, CURLOPT\_HEADER, 0);

　　//执行并获取HTML文档内容

　　$output = curl\_exec($ch);

　　//释放curl句柄

　　curl\_close($ch);

　　//打印获得的数据

　　print\_r($output);

3.2 Post方式实现

复制代码 代码如下:

　　 $url = "http://localhost/web\_services.php";

　　$post\_data = array ("username" => "bob","key" => "12345");

　　$ch = curl\_init();

　　curl\_setopt($ch, CURLOPT\_URL, $url);

　　curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, 1);

　　// post数据

　　curl\_setopt($ch, CURLOPT\_POST, 1);

　　// post的变量

　　curl\_setopt($ch, CURLOPT\_POSTFIELDS, $post\_data);

　　$output = curl\_exec($ch);

　　curl\_close($ch);

　　//打印获得的数据

　　print\_r($output);

1. $a=5;Function A(&$b){$b=6;Return $b;}A($a); echo A($a);Echo $a; 66
2. JS闭包

闭包就是一个函数引用另外一个函数的变量，因为变量被引用着所以不会被回收，因此可以用来封装一个私有变量。这是优点也是缺点，**不必要的闭包只会徒增内存消耗！**另外使用闭包也要注意变量的值是否符合你的要求，因为他就像一个静态私有变量一样。

1. Php5种运行模式

1）CGI（通用网关接口 / Common Gateway Interface）  
  
2）FastCGI（常驻型CGI / Long-Live CGI）  
  
3）CLI（命令行运行 / Command Line Interface）  
  
4）Web模块模式（Apache等Web服务器运行的模式）  
  
5）ISAPI（Internet Server Application Program Interface）

1. pdo防止sql注入原理

在php5.3.6之后，pdo不会在本地对sql进行拼接然后将拼接后的sql传递给mysql server处理（也就是不会在本地做转义处理）。pdo的处理方法是在prepare函数调用时，将预处理好的sql模板（包含占位符）通过mysql协议传递给mysql server，告诉mysql server模板的结构以及语义。当调用execute时，将两个参数传递给mysql server。由mysql server完成变量的转移处理。将sql模板和变量分两次传递，即解决了sql注入问题。

198.通过php函数的方式对目录进行遍历，写出程序？

循环

*echo "<pre>";  
print\_r(read\_dir\_queue('../dayi/'));  
function read\_dir\_queue($dir){  
  $files=array();  
  $queue=array($dir);  
  while($data=each($queue)){  
   $path=$data['value'];  
   if(is\_dir($path) && $handle=opendir($path)){  
    while($file=readdir($handle)){  
     if($file=='.' || $file=='..')continue;  
     $files[]=$real\_path=$path.'/'.$file;  
     if(is\_dir($real\_path)) $queue[]=$real\_path;  
    }  
   }  
   closedir($handle);  
  }  
  return $files;  
}*

199.Linux的目录进行遍历，编写shell脚本？

200.数据库合计时，常遇到的性能瓶颈有哪些，常有的解决方案？

瓶颈主要有：  
  
1、磁盘搜索 优化方法是：将数据分布在多个磁盘上  
  
2、磁盘读/写 优化方法是：从多个磁盘并行读写。  
  
3、CPU周期 优化方法：扩充内存  
  
4、内存带宽

1. 通过js创建一个当前窗口的子窗口，通过程序实现子窗口对父窗口的操作实例？
2. 设定网站的用户数量在千万级，但是活跃用户的数量只有1%，如何通过优化数据库提高活跃用户的访问速度？

我们可以根据用户的活跃程度，把活跃的用户提取出来放到另外一张表里面，每次活跃的用户登陆的时候就直接到活跃用户表中进行查询，这样就提高了数据库的查询速度

1. 服务器在运行的过程中，随着用户访问数量的增长，如何通过优化，保证性能？如果数据库已经达到最大化，请设计出继续升级的解决方案？

负载均衡/网站优化 数据库主从复制

1. 实现无限极分类的数据库设计？
2. &&和& ，|和||的区别？

&&和||是短路运算符，&和|是非短路运算符

&&与&

区别：两者都表示“与”运算，但是&&运算符第一个表达式不成立的话，后面的表达式不运算，直接返回。而&对所有表达式都得判断。

|| 与|

区别：两者都表示“或”运算，但是||运算符第一个表达式成立的话，后面的表达式不运算，直接返回。而|对所有表达式都得判断。

1. redis数据类型及应用场景
2. Redis持久化机制对比

https://www.cnblogs.com/Fairy-02-11/p/6182478.html

1. Aof 3种模式对比

在Redis的配置文件中存在三种同步方式，它们分别是：

appendfsync always     #每次有数据修改发生时都会写入AOF文件。

appendfsync everysec  #每秒钟同步一次，该策略为AOF的缺省策略。

appendfsync no       #从不同步。高效但是数据不会被持久化。

1. RDB 和 Aof 复制过程
2. Memcache 的存储原理和LRU机制

预分配

LRU原理：当某个单元被请求的时候，维护一个计数器，通过计数器来判断最近谁最少使用，那就把谁踢出去。

APP

1. 登录：token (cookie session 原理)
2. 接口安全验证（签名验证）
3. 数据加密（对称，非对称，https）
4. 脚本无限访问怎么办

Ip限制

1. 抓包工具抓取一次正确请求后无限访问请求怎么办
2. 你在项目中主要负责什么？

第三方接入/功能实现  
购物车怎么做的？

订单有哪些状态？

未支付 已支付 未发货 已发货 取消订单  
订单拆分怎么做的？

1. laravel 门面 脚手架

脚手架：按照一定得规则生成相应得程序

门面：

简单说为了简化书写。  
比如缓存类，没有门面类的话。这么写  
$c = new Cache(xxxx); $c->get();  
或者服务容器取出来app('cache')->get();

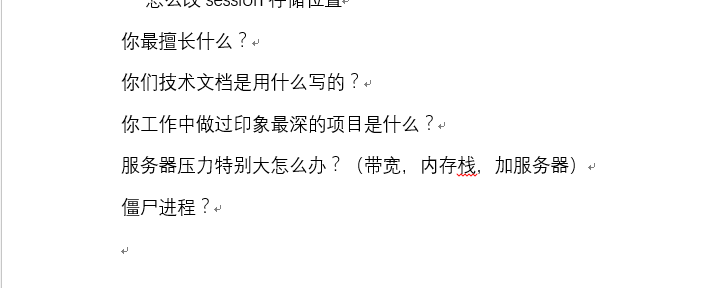
使用门面的话  
Cache::get();

218.僵尸进程

1 什么是僵尸进程：   
当子进程比父进程先结束，而父进程又没有回收子进程，释放子进程占用的资源，此时子进程将成为一个僵尸进程。如果父进程先退出   
，子进程被init接管，子进程退出后init会回收其占用的相关资源。

219.innodb事务可以分为以下几种：

1.扁平事务    2.带有保存点的扁平事务    3.链事务    4.嵌套事务   5.分布式事务



1. 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

优化程序，优化数据库，如果程序和数据库已经最优化，使用以下解决方法：  
  
1、确定当前服务器设备是否满足流量需求。  
  
2、使用Memcache缓存技术，把动态内容缓存到文件中，动态网页直接调用这些文件，而不必再访问数据库。  
  
3、禁止外部盗链，图片和文件外部盗链会给服务器带来大量的负载压力，可以通过refer来禁止外部盗链，或者使用apache来配置禁止盗链。  
  
4、控制大文件的下载，大文件的下载对于非SCSI硬盘来说会占用大量的资源，导致服务器的响应能力下降。  
  
5、使用不同的主机分流主要流量，使服务器均衡负载。  
  
6、使用流量统计软件统计分析网站流量，可以知道哪些地方耗费了大量的流量，哪些页面需要再进行优化。

1. 简述如何得到当前执行脚本路径，包括所得到参数

用$\_SERVER['DOCUMENT\_ROOT'].$\_SERVER['REQUEST\_URI'];取得当前页面的完整路径和参数。  
  
取得参数：$\_SERVER['QUERY\_STRING'];

222.客户端打开页面是空白页

第一是检查客户端是不是自己网速的问题，换个网站试试

第二是在linux上 用top 检查cpu 看看是哪里出了问题了

第三是如果是服务器就重启 如果是数据库 通过慢查询查看哪个sql语句时间长优化 如果不是隔断检查

1. 打开页面慢怎么办？

先打开filebug 查看哪个请求时间慢

然后在Linux系统上用top 查看 是服务器还是数据库

解决：负载均衡------》缓存-----》主从复制

224.**cookie和sessiond的主要区别**

**(1)、保存位置稍有区别**

cookie数据存放在客户的浏览器上，服务器端不用保存。session数据放在服务器上，本地内存也有一份。

**(2)、安全性不同**

cookie安全性不如session。因为普通cookie保存在本地硬盘上，黑客可以伪造url等方式发起xss攻击，获取本地硬盘保存状态的cookie，进而窃取用户的敏感信息。

session则不同，只有在用户登录此网站时发起xss攻击才能获取session信息，关闭浏览器之后，session即被销毁，安全性较cookie要好

**(3).跨域支持上的不同**

Cookie支持跨域名访问，例如将domain属性设置为“.biaodianfu.com”，则以“.biaodianfu.com”为后缀的一切域名均能够访问该Cookie。跨域名Cookie如今被普遍用在网络中，例如Google、Baidu、Sina等。而Session则不会支持跨域名访问。Session仅在他所在的域名内有效。

(4).**服务器压力的不同**

Session是保管在服务器端的，每个用户都会产生一个Session。假如并发访问的用户十分多，会产生十分多的Session，耗费大量的内存。因而像Google、Baidu、Sina这样并发访问量极高的网站，是不太可能运用Session来追踪客户会话的。考虑到减轻服务器性能方面，应当使用COOKIE。

(5). **存取方式的不同**

Cookie中只能保管ASCII字符串，假如需求存取Unicode字符或者二进制数据，需求先进行编码。Cookie中也不能直接存取Java对象。若要存储略微复杂的信息，运用Cookie是比拟艰难的。

而Session中能够存取任何类型的数据，包括而不限于String、Integer、List、Map等。Session中也能够直接保管Java Bean乃至任何Java类，对象等，运用起来十分便当。能够把Session看做是一个Java容器类。

**(6).cookie的保存内容大小有限制**

单个cookie保存的数据不能超过4K，很多浏览器都限制一个站点最多保存20个cookie。

225.什么是高内聚低耦合？

高内聚：在一个模块内，让每个元素之间都尽可能的紧密相连。如果某个元素与该模块的关系比较疏松的话，可能该模块的结构还不够完善，或者是该元素是多余的。

低耦合：一个完整的系统，模块与模块之间，尽可能的使其独立存在。也就是说，让每个模块，尽可能的独立完成某个特定的子功能。如果某两个模块间的关系比较复杂的话，最好首先考虑进一步的模块划分。

1. php的语言结构和函数有什么区别？

答：语言结构：就是PHP语言的关键词，语言语法的一部分；它不可以被用户定义或者添加到语言扩展或者库中；它可以有也可以没有变量和返回值。

函数：由代码块组成的，可以复用。从源码的角度来说，也就是基于Zend引擎的基础来实现的，ext拓展库中的函数都是这样实现的。

区别：语言结构比php的函数快，原因是函数要先被php解析器分解成语言结构，由此可见，函数比语言结构多了一层解析器解析

语言结构列表：echo() print() die() isset() unset() array() list() empty()

include()，注意，include\_once()是函数

require()，注意，require\_once()是函数

227.关系型数据库和非关系型数据库的区别？

关系型**[数据库](http://lib.csdn.net/base/mysql" \t "http://blog.csdn.net/longxingzhiwen/article/details/_blank" \o "MySQL知识库)**有**[Oracle](http://lib.csdn.net/base/oracle" \t "http://blog.csdn.net/longxingzhiwen/article/details/_blank" \o "Oracle知识库)**、DB2、Microsoft SQL Server、Microsoft Access、**[MySQL](http://lib.csdn.net/base/mysql" \t "http://blog.csdn.net/longxingzhiwen/article/details/_blank" \o "MySQL知识库)**等

非关系型数据库有 NoSql、Cloudant、redis 、memcache。

非关系型数据库的优势：1. 性能NOSQL是基于键值对的，可以想象成表中的主键和值的对应关系，而且不需要经过SQL层的解析，所以性能非常高。2. 可扩展性同样也是因为基于键值对，数据之间没有耦合性，所以非常容易水平扩展。

关系型数据库的优势：1. 复杂查询可以用SQL语句方便的在一个表以及多个表之间做非常复杂的数据查询。2. 事务支持使得对于安全性能很高的数据访问要求得以实现。对于这两类

关系型数据库的优势：

1. 保持数据的一致性（事务处理）

2.由于以标准化为前提，数据更新的开销很小（相同的字段基本上都只有一处）

3. 可以进行Join等复杂查询

其中能够保持数据的一致性是关系型数据库的最大优势。

数据库，对方的优势就是自己的弱势，反之亦然。

关系型数据库的不足：

不擅长的处理

1. 大量数据的写入处理

2. 为有数据更新的表做索引或表结构（schema）变更

3. 字段不固定时应用

4. 对简单查询需要快速返回结果的处理

228.网站的攻击有哪些？你怎么防御？

答：Xss攻击是跨站脚本工具 Csrf攻击是跨站请求伪造 sql注入 DDOS流量攻击？

防止xss脚本攻击：

XSS攻击即跨站脚本攻击，通过篡改网页，注入恶意的HTML脚本，控制 用户浏览器进行恶意操作的一种攻击。

防止: script注入，转义过滤script标签。

防止csrf跨站请求伪装：

CSRF的防御可以从服务端和客户端两方面着手，防御效果是从服务端着手效果比较好，现在一般的CSRF防御也都在服务端进行。

1：.Cookie Hashing(所有表单都包含同一个伪随机值)：

2：验证码

3：One-Time Tokens(不同的表单包含一个不同的伪随机值)

防止sql注入：

1执行sql语句是使用addslashes进行sql语句转换

2 sql语句书写尽量不要省略双引号和单引号

3 过滤掉sql语句里的关键词 ：update insert delete select \*

4 PHP配置文件中设置register\_global为off，关闭全局变量注册

5 控制错误信息 不要把错误信息输出到浏览器上 应保存在错误日

防止DDOS流量攻击：

1 使用工具:DDoS deflate . 自动查封IP.

2 解析域名到127.0.0.1 让攻击方自己攻击自己

3 把网站做成静态页面

4 限制S YN/ICMP流量

229.memcache和redis的区别？

答：

1：memcache和redis都是将数据存储到内从中，都是内存数据库，不过memcache还可以缓存其他东西，图片视频等。

2：存储类型不同，memcache只支持k-v结构，redis支持五种数据类似，字符串，链表，哈希，集合，有序集合。

3：虚拟内存，当redis物理内存使用完后，可以将一些很久没有使用过的value交换到磁盘

4：分布式设定memcache集群，利用magenta做一主多从，redis可以做一主多从，也可以一主一从。

5：存储数据redis持久化，memcache关机后数据消失

6：数据恢复，memcache挂掉后，数据不可恢复，redis数据丢失后可以通过AOF日志恢复

7：应用场景不同，redis除啦作为数据库使用之外，还能做消息队列，数据堆栈和数据缓存等，memcache适用于缓存sql语句，数据集，用户临时性数据，延迟查询数据 session等

230.原子操作是什么？

假定有两个操作A和B，如果从执行A的线程来看，当另一个线程执行B时，

要么将B全部执行完，要么完全不执行B，那么A和B对彼此来说是原子的。

原子操作是指对于方位同一个状态的所有操作（包括该操作本身）来说，

这个操作是一个以原子方式执行的操作。

总结：原子操作是不可分割的，在执行完毕之前不会被任何其它任务或事件中断

231.：PHP支持单继承吗？

答：php支持单继承，没有支持多继承的属性

232.：interface接口支持单继承还是多继承？

答：在php中接口可以继承接口，接口和类不同支持多继承，继承用extends关键字，多继承时只需要用逗号把继承的接口隔开就可以了。

Ps：需要注意的是当你接口继承其它接口时候，直接继承父接口的静态常量属性和抽象方法，所以类实现接口时必须实现所有相关的抽象方法。

233.：抽象类abstract什么时候用？

答：首先抽象类不能被实例化，同理抽象类不能产生实例对象，通常是将抽象方法做为子类方法重载的模板使用的，且要把继承的抽象类里的方法都实现。实际上抽象类是方便继承而引入的。

234：什么是webserver？

答：网页服务器或者是web服务器，主要是提供网上信息浏览业务（可以解析http协议）

235：nginx和apache的区别？

答：

1：nginx是轻量级，比apache占用更少的内存以及资源，nginx处理请求时异步非阻塞类型的，而apache是阻塞型的，在高并发下nginx是低资源低消耗高性能的设计相对简单

2：apache比nginx的rewrite强大，模块多，bug少，稳定性强，

3：apache对php支持比较简单，nginx需要配合其他后端用，apache组件比nginx多，现在nginx是web服务器的首选

4：最核心的区别是apache是同步多进程模型，一个连接对应一个进程，多个连接（千万级别）可以对应一个进程

5：nginx：配置文件写的很简洁，正则配置让很多事情变得简单运行效率高，占用资源少，代理功能强大，很适合做前端响应服务器

6：Apache在处理动态有优势，Nginx并发性比较好，CPU内存占用低，如果rewrite频繁，那还是Apache吧

7：动态请求要apache去做，nginx只适合静态和反向

236：数组的底层结构是什么？

答：php的数组本质上是一个哈希表

237：php-fpm和fastCGI的区别？

答：fastCGI是CGI的升级版，是WEB服务器与处理程序之间通信的一种协议，php-fpm是针对fastCGI的一个进程管理器，实现了一个支持fastCGI协议的server程序

238.mysql优化？

### 一.sql语句优化

### ☆选择合适的数据类型

1. 使用可存下数据的最小的数据类型，整型 < date,time < char,varchar < blob  
   （2）使用简单的数据类型，整型比字符处理开销更小，因为字符串的比较更复杂。如，int类型存储时间类型，bigint类型转ip函数  
   （3）使用合理的字段属性长度，固定长度的表会更快。使用enum、char而不是varchar  
   （4）尽可能使用not null定义字段  
   （5）尽量少用text，非用不可最好分表

☆ 加索引

注意

1.对于创建的多列(复合)索引，只要**查询条件使**用了**最左边的列，索引一般就会被使用**。 （**最左原则**）；

2.对于使用like的查询，查询如果是”%aaa”,不会使用到索引，‘aaa%’会使用到索引。

3.搜索中文 如：根据电影的剧情查找电影的名称，根据歌词查找歌名。like ‘%爱请%’

使用**全文索引**来解决，可以使用第三方查询工具，sphinx

## 如果条件中有or，则要求or的索引字段都必须有索引，否则不能用到索引。

1. 如果列类型是字符串，一定要在条件中将数据使用引号引用起来，否则不使用索引。
2. 优化group by语句

**默认情况下， mysql对所有的group by col1,col2进行排序。这与在查询中指定order by col1,col2类型，如果查询中包括group by 但用户想要避免排序结果的消耗，则可以使用order by null禁止排序。**

1. 当取出的数据量超过表中数据的20%，优化器就不会使用索引，而是全表扫描。

8.前缀索引

利用字段数据的前部分作为索引，称为前缀索引。减少索引长度，提高索引效率。

比如：统计密码的前7个字符，作为不相同匹配条件，几乎可以做到1:1

此时，就可以利用前7个字符做索引关键字即可（离散程度高）

语法：

**alter table 表名 add index (passwd(7)) 指定前7位作为索引关键字。**

不使用索引前缀，索引的长度。

9.使用limit对查询结果的记录进行限定  
10.避免select \*，将需要查找的字段列出来  
11.使用连接（join）来代替子查询  
12.拆分大的delete或insert语句

1. 分页优化

翻页的sql语句：

select \* from table\_name **limit offset，length**

**使用如上语句，在翻页时，翻到最后，越来越慢，**

**原因：并不是跨过offset行，取出n条，**

**是取出offset+N条数据，舍弃前面的offset行，只取出n条数据。**

**如何解决？**

**（1）**从业务上去解决：

办法：不允许翻过100页，

以百度为例，一般翻页到70页左右，谷歌40页左右

（2）不用offset，用条件查询，条件中使用id查询，使用到了索引，

select \* from user limit 10000,10;

select \* from user where id>10000 limit 10;

下一页：select \* from user where id>10000+10 limit 10

该种方式要注意：如果有数据被删除，会导致select \* from user limit 10000,10;

和select \* from user where id>10000 limit 10;语句取出的结果不一样，。

因为数据在磁盘中被删除掉了（物理删除），结构发生改变，两者的查询的结果不一致，

要想使结果一致，不能进行物理删除，给表增加一个字段来区分是否删除。

3）假如不能使用限制翻页到100页，数据有删除，还要求翻页，速度不能受影响。

思路：通过翻页，先取出id（主键），在根据id取出数据。

select name,age,email from user inner join (select id from user limit 10000,10) as tmp on tmp.id=user.id

非要物理删除，还要用offset精确查询，还不限制用户分页，怎么办

我们现在必须要查，则**只查索引，不查数据，得到id**

**再用id去查具体条目**，这种技巧就是**延迟索引**。

1. 分区.分表
2. 主从复制、读写分离

14：优化需要用到哪些中间件？

主从复制：

主库挂掉：MHA

原理：

**（1）从宕机崩溃的master保存二进制日志事件（binlog events）;**

**（2）识别含有最新更新的slave；**

**（3）应用差异的中继日志（relay log）到其他的slave；**

**（4）应用从master保存的二进制日志事件（binlog events）；**

**（5）提升一个slave为新的master；**

**（6）使其他的slave连接新的master进行复制；**

一主多从如何读数据：atlas

负载均衡一台服务器挂掉如何切换：keepalived

1. 支付原理

支付宝给的流程图还是很清晰的，其实基本流程就是

1. 用户向服务器请求一个付款
2. 服务器生成一个带签名的订单发送给客户端
3. 客户端通过这个订单向app sdk请求付款
4. sdk把用户引入支付宝付款界面进行支付
5. 支付成功后支付宝向前端返回支付成功结果，并且向服务器发送一个支付通知
6. 服务器接收通知并且验证是否是支付宝发送的成功结果

app客户端需要做的很简单：

1. 向自己的服务器请求一个订单，
2. 接收到订单后，向支付宝sdk发情一个支付请求
3. 交易结束后返回一个成功或者失败
4. xml 和html的区别

Xml比较严格，区分大小写 结束标签 html主要用于展示 xml 用于传输

241.断点续传

对于断点续传，关键点是两个：

1. 终端知道当前的文件和上一次加载的文件是不是内容发生了变化，如果有变化，需要重新从offset 0 的位置开始下载

2. 终端记录好上次成功下载到的offset，告诉server端,server端支持从特定的offset 开始吐数据

1. mysql 存储引擎（多个）

Myisam innodb

**MEMORY**

使用MySQL Memory存储引擎的出发点是速度。为得到最快的响应时间，采用的逻辑存储介质是系统内存。虽然在内存中存储表数据确实会提供很高的性能，但当mysqld守护进程崩溃时，所有的Memory数据都会丢失。获得速度的同时也带来了一些缺陷。它要求存储在Memory数据表里的数据使用的是长度不变的格式，这意味着不能使用BLOB和TEXT这样的长度可变的数据类型，VARCHAR是一种长度可变的类型，但因为它在MySQL内部当做长度固定不变的CHAR类型，所以可以使用

**MERGE**

MERGE存储引擎是一组MyISAM表的组合，这些MyISAM表结构必须完全相同，尽管其使用不如其它引擎突出，但是在某些情况下非常有用。说白了，Merge表就是几个相同MyISAM表的聚合器；Merge表中并没有数据，对Merge类型的表可以进行查询、更新、删除操作，这些操作实际上是对内部的MyISAM表进行操作。Merge存储引擎的使用场景。

**ARCHIVE**

Archive是归档的意思，在归档之后很多的高级功能就不再支持了，仅仅支持最基本的插入和查询两种功能。在MySQL 5.5版以前，Archive是不支持索引，但是在MySQL 5.5以后的版本中就开始支持索引了。Archive拥有很好的压缩机制，它使用zlib压缩库，在记录被请求时会实时压缩，所以它经常被用来当做仓库使用

1. App 更新版本怎么解决（升级/后台代码）

把旧版本拷贝一份，在这份上升级

244.composer 优缺点

245.rest ful优点 ，和其他的区别

“REST最突出的特点是用URL描述互联网上的资源。  
它主要有四个特点:  
1.资源通过URL来指定和操作。  
2.对资源的操作包括获取、创建、修改和删除，正好对应HTTP协议的GET,POST,PUT和DELETE方法。  
3.连接是无状态性的。  
4.能够利用Cache机制来提高性能。”  
上面是我当时上课记的笔记，这四个特点也是REST风格的优势。它基于HTTP协议，并不涉及新的技术，而且与SOAP,XML-RPC相比，更加简洁、高效。

综合上面的解释，我们总结一下什么是RESTful架构：

　　（1）每一个URI代表一种资源；

　　（2）客户端和服务器之间，传递这种资源的某种表现层；

　　（3）客户端通过四个HTTP动词，对服务器端资源进行操作，实现"表现层状态转化"。

1. 各种协议

TCP/IP协议族中包括上百个互为关联的协议，不同功能的协议分布在不同的协议层， 几个常用协议如下：   
  
1、Telnet（Remote Login）：提供[远程登录](https://www.baidu.com/s?wd=%E8%BF%9C%E7%A8%8B%E7%99%BB%E5%BD%95&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y4Pjf4njwWPW9hmWfzrHF-0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnW0YPWR4njDz" \t "https://zhidao.baidu.com/question/_blank)功能，一台计算机用户可以登录到远程的另一台计算机上，如同在远程主机上直接操作一样。   
2、FTP（File Transfer Protocol）：远程文件传输协议，允许用户将远程主机上的文件拷贝到自己的计算机上。   
3、SMTP（Simple Mail transfer Protocol）：简单邮政传输协议，用于传输电子邮件。   
4、NFS（Network File Server）：网络[文件服务器](https://www.baidu.com/s?wd=%E6%96%87%E4%BB%B6%E6%9C%8D%E5%8A%A1%E5%99%A8&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y4Pjf4njwWPW9hmWfzrHF-0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnW0YPWR4njDz" \t "https://zhidao.baidu.com/question/_blank)，可使多台计算机透明地访问彼此的目录。   
5、UDP（User Datagram Protocol）：用户数据包协议，它和TCP一样位于传输层，和IP协议配合使用，在传输数据时省去包头，但它不能提供数据包的重传，所以适合传输较短的文件。

247.nginx 服务器的原理

248.ajax 的原理

AJAX全称为“Asynchronous JavaScript and XML”（异步JavaScript和XML），是一种创建交互式网页应用的网页开发技术，

是几种原有技术的结合体。它由下列技术组合而成。

   1.使用CSS和XHTML来表示。

   2. 使用DOM模型来交互和动态显示。

   3.使用XMLHttpRequest来和服务器进行异步通信。

   4.使用javascript来绑定和调用。

Ajax的原理简单

Ajax的工作原理相当于在用户和服务器之间加了—个中间层(AJAX引擎),使用户操作与服务器响应异步化。并不是所有的用户请求都提交给服务器,像—些数据验证和数据处理等都交给Ajax引擎自己来做, 只有确定需要从服务器读取新数据时再由Ajax引擎代为向服务器提交请求。

Ajax的原理简单来说通过XmlHttpRequest对象来向服务器发异步请求，从服务器获得数据，然后用javascript来操作DOM而更新页面。这其中最关键的一步就是从服务器获得请求数据。要清楚这个过程和原理，我们必须对 XMLHttpRequest有所了解。

XMLHttpRequest是ajax的核心机制，它是在IE5中首先引入的，是一种支持异步请求的技术。简单的说，也就是javascript可以及时向服务器提出请求和处理响应，而不阻塞用户。达到无刷新的效果。

1. svn 原理

所有的客户端将自己的代码上传到中央服务器，中央服务器的版本数据库会保存上传的各个版本的代码，并进行统一管理。这里注意，客户端本身是不存在版本数据库的，所以，如果中央服务器离线或宕机，则集中式版本控制则无法使用。

SVN是一个增量式的版本控制，它不会讲各个版本的副本都完整的保存下来，而只会记录下版本之间的差异，然后按照顺序更新或者恢复特定版本的数据。这使得服务端的存储量会非常低。

250.退货

251.对称加密有哪些

Des 3des aes

1. mysql 索引类型

BTree索引 哈希索引，全文索引

1. 数据搜索 sphinx xunsearch

xunsearch与sphinx原理上略有不同，xunsearch当然在创建索引时，其实他是创建了一个自己的数据库， 如果你用xunsearch去查寻数据时，他不会再到你的mysql数据库中去查寻了，而是直接到它本身的数据库中去查寻，然后返回数据，而sphinx 不是这样的，sphinx创建一些索引文件（索引表），然后通过sphinx去查寻数据时，首先他到索引文件查寻数据，然后返回的是当前要查寻数据的 id（这里id是主键），然后再通过id去到mysql数据中查寻数据，然后再返回给数据。

254.mysql数据类型占多少字节：GBK 2 utf8 3 int 4 bigint 8

255.YII框架和tp框架的区别：tp初始化的时候会将内置的扩展全部加载 YII框架轻量级内部功能模块比较全

256.支付宝接口（合并订单、退款....）

支付API

API列表 类型 描述

alipay.trade.fastpay.refund.query 免费 统一收单交易退款查询

alipay.trade.order.settle 免费 统一收单交易结算接口

alipay.trade.close 免费 统一收单交易关闭接口

alipay.trade.cancel 免费 统一收单交易撤销接口

alipay.trade.refund 免费 统一收单交易退款接口

alipay.trade.precreate 免费 统一收单线下交易预创建

alipay.trade.create 免费 统一收单交易创建接口

alipay.trade.pay 免费 统一收单交易支付接口

alipay.trade.query 免费 统一收单线下交易查询

1. **php是什么语言**

**镶嵌在服务器的脚本语言**

PHP（外文名:PHP: Hypertext Preprocessor，中文名：“[超文本](https://baike.baidu.com/item/%E8%B6%85%E6%96%87%E6%9C%AC" \t "https://baike.baidu.com/item/PHP/_blank)[预处理器](https://baike.baidu.com/item/%E9%A2%84%E5%A4%84%E7%90%86%E5%99%A8" \t "https://baike.baidu.com/item/PHP/_blank)”）是一种通用[开源](https://baike.baidu.com/item/%E5%BC%80%E6%BA%90" \t "https://baike.baidu.com/item/PHP/_blank)[脚本语言](https://baike.baidu.com/item/%E8%84%9A%E6%9C%AC%E8%AF%AD%E8%A8%80" \t "https://baike.baidu.com/item/PHP/_blank)。[语法](https://baike.baidu.com/item/%E8%AF%AD%E6%B3%95" \t "https://baike.baidu.com/item/PHP/_blank)吸收了[C语言](https://baike.baidu.com/item/C%E8%AF%AD%E8%A8%80" \t "https://baike.baidu.com/item/PHP/_blank)、[Java](https://baike.baidu.com/item/Java" \t "https://baike.baidu.com/item/PHP/_blank)和[Perl](https://baike.baidu.com/item/Perl" \t "https://baike.baidu.com/item/PHP/_blank)的特点，利于学习，使用[广泛](https://baike.baidu.com/item/%E5%B9%BF%E6%B3%9B" \t "https://baike.baidu.com/item/PHP/_blank)，主要适用于[Web](https://baike.baidu.com/item/Web" \t "https://baike.baidu.com/item/PHP/_blank)开发领域。PHP 独特的[语法](https://baike.baidu.com/item/%E8%AF%AD%E6%B3%95" \t "https://baike.baidu.com/item/PHP/_blank)混合了[C](https://baike.baidu.com/item/C" \t "https://baike.baidu.com/item/PHP/_blank)、[Java](https://baike.baidu.com/item/Java" \t "https://baike.baidu.com/item/PHP/_blank)、[Perl](https://baike.baidu.com/item/Perl" \t "https://baike.baidu.com/item/PHP/_blank)以及[PHP](https://baike.baidu.com/item/PHP" \t "https://baike.baidu.com/item/PHP/_blank)自创的语法。它可以比[CGI](https://baike.baidu.com/item/CGI" \t "https://baike.baidu.com/item/PHP/_blank)或者[Perl](https://baike.baidu.com/item/Perl" \t "https://baike.baidu.com/item/PHP/_blank)更快速地执行[动态网页](https://baike.baidu.com/item/%E5%8A%A8%E6%80%81%E7%BD%91%E9%A1%B5" \t "https://baike.baidu.com/item/PHP/_blank)。用PHP做出的[动态页面](https://baike.baidu.com/item/%E5%8A%A8%E6%80%81%E9%A1%B5%E9%9D%A2" \t "https://baike.baidu.com/item/PHP/_blank)与其他的[编程语言](https://baike.baidu.com/item/%E7%BC%96%E7%A8%8B%E8%AF%AD%E8%A8%80" \t "https://baike.baidu.com/item/PHP/_blank)相比，[PHP](https://baike.baidu.com/item/PHP/9337" \t "https://baike.baidu.com/item/PHP/_blank)是将[程序](https://baike.baidu.com/item/%E7%A8%8B%E5%BA%8F" \t "https://baike.baidu.com/item/PHP/_blank)嵌入到[HTML](https://baike.baidu.com/item/HTML" \t "https://baike.baidu.com/item/PHP/_blank)（[标准通用标记语言](https://baike.baidu.com/item/%E6%A0%87%E5%87%86%E9%80%9A%E7%94%A8%E6%A0%87%E8%AE%B0%E8%AF%AD%E8%A8%80" \t "https://baike.baidu.com/item/PHP/_blank)下的一个应用）文档中去执行，执行效率比完全生成[HTML](https://baike.baidu.com/item/HTML" \t "https://baike.baidu.com/item/PHP/_blank)标记的[CGI](https://baike.baidu.com/item/CGI/607810" \t "https://baike.baidu.com/item/PHP/_blank)要高许多；PHP还可以执行[编译](https://baike.baidu.com/item/%E7%BC%96%E8%AF%91" \t "https://baike.baidu.com/item/PHP/_blank)后代码，编译可以达到[加密](https://baike.baidu.com/item/%E5%8A%A0%E5%AF%86" \t "https://baike.baidu.com/item/PHP/_blank)和[优化](https://baike.baidu.com/item/%E4%BC%98%E5%8C%96" \t "https://baike.baidu.com/item/PHP/_blank)代码运行，使代码运行更快。

**267.索引失效**

**1.如果条件中有or，即使其中有条件带索引也不会使用(这也是为什么尽量少用or的原因)**

**要想使用or，又想让索引生效，只能将or条件中的每个列都加上索引**

**2.对于多列索引，不是使用的第一部分，则不会使用索引**

**3.like查询以%开头**

**4.如果列类型是字符串，那一定要在条件中将数据使用引号引用起来,否则不使用索引**

**5.如果mysql估计使用全表扫描要比使用索引快,则不使用索引**

**268.js** 是一种解释型的脚本语言,

269.ajax的原理

Ajax的工作原理相当于在用户和服务器之间加了—个中间层(AJAX引擎),使用户操作与服务器响应异步化。并不是所有的用户请求都提交给服务器,像—些数据验证和数据处理等都交给Ajax引擎自己来做, 只有确定需要从服务器读取新数据时再由Ajax引擎代为向服务器提交请求。

Ajax的原理简单来说通过XmlHttpRequest对象来向服务器发异步请求，从服务器获得数据，然后用javascript来操作DOM而更新页面。这其中最关键的一步就是从服务器获得请求数据。要清楚这个过程和原理，我们必须对 XMLHttpRequest有所了解。

XMLHttpRequest是ajax的核心机制，它是在IE5中首先引入的，是一种支持异步请求的技术。简单的说，也就是javascript可以及时向服务器提出请求和处理响应，而不阻塞用户。达到无刷新的效果。

ajax技术利用了什么协议？简述ajax的工作机制  
ajax利用地http协议  
（1）对象初始化  
（2）发送请求  
（3）服务器接收请求并进行处理  
（4）服务器返回响应数据  
（5）客户端接收  
（6）依据响应数据修改客户端页面内容

1. SPU是标准化产品单元，区分品种；SKU是库存量单位，区分单品；商品特指与商家有关的商品，可对应多个SKU。

271. 伪事物：

在Mysql数据库里面存在了好几个存储引擎，其中有的存储引擎是支持事务管理，例如InnoDB,但是有的存储引擎是不支持事务管理，例如MyISAM存储引擎。如果用户在创建表的时候选择的存储引擎是MyISAM，用户可以选择应用表锁来代替事务，这种使用表锁来代替事务的事件被称为伪事务。表锁分为读锁（read）和写锁（write）。

1. apache运行原理

Apache是基于模块化设计的，它的核心代码并不多，大多数的功能都被分散到各个模块中，各个模块在系统启动的时候按需载入。

MPM（Multi -Processing Modules，多重处理模块）是Apache的核心组件之一，Apache通过MPM来使用操作系统的资源，对进程和线程池进行管理。Apache为了能够获得最好的运行性能，针对不同的平台 (Unix/Linux、Window)做了优化，为不同的平台提供了不同的MPM，用户可以根据实际情况进行选择，其中最常使用的MPM有 prefork和worker两种。至于您的服务器正以哪种方式运行，取决于安装Apache过程中指定的MPM编译参数,在X系统上默认的编译参数为 prefork。

1. nginx 的工作原理

Nginx由内核和模块组成，其中，内核的设计非常微小和简洁，完成的工作也非常简单，仅仅通过查找配置文件将客户端请求映射到一个location block（location是Nginx配置中的一个指令，用于URL匹配），而在这个location中所配置的每个指令将会启动不同的模块去完成相应的工作。

**274.trait**

Traits 是一种为类似 PHP 的单继承语言而准备的代码复用机制。Trait 为了减少单继承语言的限制，使开发人员能够自由地在不同层次结构内独立的类中复用方法集。Traits 和类组合的语义是定义了一种方式来减少复杂性，避免传统多继承和混入类（Mixin）相关的典型问题。

1. **orm 是什么**

  对象关系映射（Object Relational Mapping，简称ORM）模式是一种为了解决面向对象与关系数据库存在的互不匹配的现象的技术。简单的说，ORM是通过使用描述对象和数据库之间映射的元数据，将程序中的对象自动持久化到关系数据库中

1. js 框架

### Node.Js ****JQuery Mobile****