**面试题**

1. 为什么现在函数式编程这么越来越受关注？为什么函数式编程重要？什么时候适用函数式语言？

函数式编程有什么好处，和面向对象比有什么好处，有什么坏处。

1. 内聚和耦合的区别是什么？ 内聚分为哪几类？耦合分为哪几类？
2. 重构在哪些场景下有用？为什么要进行代码重构？重构的优劣势是什么？
3. 设计(design)、架构(architecture)、功能(functionality)和美学(aesthetic)之间有什么区别？讨论一下。

5. 微软、谷歌、欧朋(opera)和火狐这类公司是如何从他们的浏览器中获利的？

通过市场占有，成为搜索引擎的导流通道；捆绑一些自家的服务，也是导流入口；

6. 封装的重要性体现在哪儿？

屏蔽掉evil的细节部分，让使用者聚焦在业务开发上。这样也就避免了一些通用功能的反复开发。

7. 实时语言(real-time language)和堆内存分配(heap memory allocation)之间的关系是什么？

堆内存使用一般比较慢，而且可能不连续，带有gc的语言其gc主要也是工作在堆区。堆内存分配可能和实时是相互矛盾的吧。

8. 不变性(Immutability)是指: (变量的)值只能在创建的时候被设置一次，之后就不能被改变。为什么不变性对写更加安全的代码有帮助？

因为不变，所以就没有了并发时的读写冲突问题。

1. 可变值(mutable values)和不可变值(immutable values)有哪些优缺点？

10. TCP和HTTP有什么区别？

tcp工作在传输层，http工作在应用层；http是建立在tcp可靠连接之上的应用层协议；

11. 在客户端渲染(client-side rendering)和服务端渲染(server-side rendering)之间，你是如何权衡的？

客户端如果性能好，可以做一些复杂的计算和渲染，服务端渲染可以提高客户端的展示速度，但要权衡资源。有一些图形计算任务都堆到服务器上的话，可能使后端资源吃紧。

12. 如何在一个不可靠的协议之上构建一个可靠的通信协议？

参考谷歌的quic协议，其实也可以自己建立ack机制，滑动窗口之类的。

13. 为什么人们会抵制变化？

变化需要学习。学习会带给很多人恐惧。

14. 作为一个软件工程师，你想要既要有创新力，又要产出具有可预测性。采用什么策略才能使这两个目标可以共存呢？

1.思考方面：多学习先人的既有做法，思考为什么要这么做。再思考这么做会不会有什么问题，如果想不明白，去社区提问。2.形成自己的代码库，缩短自己的单位“任务”的工作时间。并经常对自己完成一项特定任务的时间进行统计。这样在排期上给出的时间会较合理。

15. 什么是好的代码？

首先要易读，命名规范。遵循任意一种代码风格约束，使用静态代码检查工具帮助自己强制实现这些规范。其次函数不能太长，要能一屏幕看到底。其它还是看beautiful code。

1. SOA和MicroService之间有什么区别？
2. MicroService架构的优劣是什么？
3. Microservice与Serverless的优劣势比较。

19. Microservice与Serverless的适用场景

Microservice适合构建复杂的应用。比如：构造一个分布式数据库。

Serverless 适合构建比较简单的应用，比如上传一张图片、对一段音频/视频进行编解码、对IOT设备的请求返回一小段数据等。

20. 进程是什么？进程的概念主要有哪些？

1. 进程是一个实体，每一个进程都有它自己的地址空间，一般情况下，包括文本区域（text region）、数据区域（data region）和堆栈（stack region）。文本区域存储处理器执行的代码；数据区域存储变量和进程执行期间使用的动态分配的内存；堆栈区域存储着活动过程调用的指令和本地变量。

2. 进程是一个“执行中的程序”，程序是一个没有生命的实体，只有处理器赋予程序生命时，它才能成为一个活动的实体，我们称其为进程。

21. 进程的基本状态有哪些？

阻塞态：等待某个事件的完成；

就绪态：等待系统分配处理器以便运行；

执行态：占有处理器正在运行。

22. 垃圾回收算法有哪些？垃圾回收触发条件是什么？

复制回收算法、标记清除算法、标记整理算法、增量回收、分代回收

Young GC：当 Young Gen 中的 eden 区分配满的时候触发。把 Eden 区存活的对象将被复制到一个 Survivor 区，当这个 Survivor 区满时，此区的存活对象将被复制到另外一个 Survivor 区。

Full GC：当准备要触发一次 Young GC 时，如果发现之前 Young GC 的平均晋升大小比目前 Old Gen剩余的空间大，则不会触发 Young GC 而是转为触发 Full GC

除了 CMS 的 Concurrent Collection 之外，其它能收集 Old Gen 的GC都会同时收集整个 GC 堆，包括 Young Gen，所以不需要事先触发一次单独的Young GC如果有 Perm Gen 的话，要在 Perm Gen分配空间但已经没有足够空间时System.gc() Heap dump 并发 GC 的触发条件就不太一样。以 CMS GC 为例，它主要是定时去检查 Old Gen 的使用量，当使用量超过了触发比例就会启动一次 GC，对 Old Gen做并发收集。

23. 垃圾收集器有哪些种类？

Serial 收集器、ParNew 收集器、Parallel Scavenge 收集器、Serial Old收集器、Parallel Old 收集器、CMS 收集器、G1收集器



24. JVM 的基本架构是如何构成的？

JVM 的基本架构如上图所示，其主要包含三个大块：

类加载器：负责动态加载Java类到Java虚拟机的内存空间中。

运行时数据区：存储 JVM 运行时所有数据

执行引擎：提供 JVM 在不同平台的运行能力

25. 主要的系统线程有哪些？

Compile Threads：运行时将字节码编译为本地代码所使用的线程

GC Threads：包含所有和 GC 有关操作

Periodic Task Thread：JVM 周期性任务调度的线程，主要包含 JVM 内部的采样分析

Singal Dispatcher Thread：处理 OS 发来的信号

VM Thread：某些操作需要等待 JVM 到达 安全点（Safe Point），即堆区没有变化。比如：GC 操作、线程 Dump、线程挂起 这些操作都在 VM Thread 中进行。

26. 按照线程类型来分，在 JVM 内部有哪些种类的线程？

1. 守护线程：通常是由虚拟机自己使用，比如 GC 线程。但是，Java程序也可以把它自己创建的任何线程标记为守护线程（public final void setDaemon(boolean on)来设置，但必须在start()方法之前调用）。

2. 非守护线程：main方法执行的线程，我们通常也称为用户线程。

27. 分布式事务的实现主要有以下几种方案：

a. XA 方案

b. TCC 方案

c. 可靠消息最终一致性方案

d. 最大努力通知方案

28. 基于 ZooKeeper 的分布式锁

1. 使用 ZK 的临时节点和有序节点，每个线程获取锁就是在 ZK 创建一个临时有序的节点，比如在 /lock/ 目录下。

2. 创建节点成功后，获取 /lock 目录下的所有临时节点，再判断当前线程创建的节点是否是所有的节点的序号最小的节点。

3. 如果当前线程创建的节点是所有节点序号最小的节点，则认为获取锁成功。

4. 如果当前线程创建的节点不是所有节点序号最小的节点，则对节点序号的 前一个节点 添加一个事件监听。

缺陷

1. 羊群效应：当一个节点变化时，会触发大量的 watches 事件，导致集群响应变慢。每个节点尽量少的 watches，这里就只注册 前一个节点 的监听

2. ZK 集群的读写吞吐量不高

3. 网络抖动可能导致 Session 离线，锁被释放

29. 分库分表有哪些拆分方式？垂直拆分有哪些实现方式？水平拆分有哪些实现方式？

1. 垂直分表 也就是 大表拆小表。 2. 垂直分库 针对的是一个系统中的不同业务进行拆分。将多个业务系统的数据放在不同的数据库中。

水平拆分：1. 离散映射：如 mod 或 dayofweek ， 这种类型的映射能够很好的解决热点问题，但带来了数据迁移和历史数据问题。2. 连续映射；如按 id 或 gmt\_create\_time 的连续范围做映射。这种类型的映射可以避免数据迁移，但又带来热点问题。

1. 简单描述一下MySQL的架构
2. 第一层是 MySQL 的服务层，包含 MySQL 核心服务功能：解析、分析、优化、缓存以及内置函数，所有跨存储引擎的功能都在这一层实现：存储过程、触发器、视图等。  2.第二层是 MySQL 的 存储引擎层，MySQL 中可使用多种存储引擎：InnoDB、MyISAM、Memory。

31. MySQL Server包括哪些功能模块？

1. 连接器 连接器负责跟客户端建立连接、获取权限、维持和管理连接。

2. 查询缓存 查询缓存将查询结果按 K-V 的形式进行缓存，K 是查询的语句，V 是查询的结果。当一个表发生更新后，该表对应的所有缓存均会失效。

3. 分析器，分析器有两个功能：词法分析、语法分析。对于一个 SQL 语句，分析器首先进行词法分析，对 SQL 语句进行拆分，识别出各个字符串代表的含义。然后就是语法分析，分析器根据定义的语法规则判断 SQL 是否满足 MySQL 语法。

4. 优化器，优化器在获取到分析器的结果后，通过表结构和 SQL 语句选择执行方案，比如：多表关联时，各个表如何进行连接；当表中有索引时，应该怎样选择索引 等等。

5. 执行器 获取到执行方案后，执行器就会根据表的引擎定义，去使用这个引擎提供的接口进行查询。

32. MySQL 执行器内部执行步骤是哪些？

1. 调用引擎接口取这个表的第一行，判断该行是否满足 WHERE 子句，如果满足则将这行存在结果集中，否则跳过。

2. 调用引擎接口取下一行，重复相同的判断逻辑，直到取到这个表的最后一行。

3. 执行器将上述遍历过程中所有满足条件的行组成的记录集作为结果集返回给客户端。

4. 对于走索引的查询，执行的逻辑也差不多。第一次调用的是 取满足条件的第一行 这个接口，之后循环取 满足条件的下一行 这个接口，这些接口都是引擎中已经定义好的。

33. 请描述一下MySQL Update 的处理逻辑？

1. MySQL Server 发送更新请求到 InnoDB 引擎

2. 从 Buffer Pool 加载对应记录的 Data Page（P1）

3. 若 Buffer Pool 中没有该记录，则从磁盘加载该记录

4. 将 P1 存储到 Undo Page 中，并在 Redo Log Buffer 中记录 Undo 操作

5. 更新 P1 为 P1’ ，并将 P1’ 写入 Dirty Page ，记录变更到 Redo Log Buffer（Prepare 状态）

6. 返回 MySQL Server 执行完成

7. MySQL Server 记录 binlog

8. MySQL Server 提交 Commit

9. Redo Log Buffer 状态有 Prepare 更改为 Commit，并刷入磁盘

10. 当 Dirty Page 过多时，启动 ChcekPoint 机制，将脏页刷入磁盘

1. 什么是MySQL热点问题？如何解决MySQL热点问题？
2. 数据库的热点块，从简单了讲，就是极短的时间内对少量数据块进行了过于频繁的访问。定义看起来总是很简单的，但实际在数据库中，我们要去观察或者确定热点块的问题，却不是那么简单了。要深刻地理解数据库是怎么通过一些数据特征来表示热点块的，我们需要了解一些数据库在这方面处理机制的特性。
3. 如何解决热点问题：a). 前端做一些请求限制，验证码、不能频繁操作，反机器人验证等。b). 应用层：限流、降级、熔断、削峰、异步调用。c). 通过分库发表把热点数据分布在不同的库或表中；打限流补丁。
4. 如何做高并发下的流量控制？如何进行限流？限流算法有哪些？限流算法的原理是什么？
5. 限流、削峰、熔断等。
6. 计数器算法、漏桶算法、令牌桶算法。
7. NGINX限流如何设置？

limit\_req\_zone 用来限制单位时间内的请求数，即速率限制,采用的漏桶算法 “leaky bucket”。

limit\_req\_conn 用来限制同一时间连接数，即并发限制。

37. 事务传播特性

1. PROPAGATION\_REQUIRED: 如果存在一个事务，则支持当前事务。如果没有事务则开启

2. PROPAGATION\_SUPPORTS: 如果存在一个事务，支持当前事务。如果没有事务，则非事务的执行

3. PROPAGATION\_MANDATORY: 如果已经存在一个事务，支持当前事务。如果没有一个活动的事务，则抛出异常。

4. PROPAGATION\_REQUIRES\_NEW: 总是开启一个新的事务。如果一个事务已经存在，则将这个存在的事务挂起。

5. PROPAGATION\_NOT\_SUPPORTED: 总是非事务地执行，并挂起任何存在的事务。

6. PROPAGATION\_NEVER: 总是非事务地执行，如果存在一个活动事务，则抛出异常

7. PROPAGATION\_NESTED：如果一个活动的事务存在，则运行在一个嵌套的事务中. 如果没有活动事务,。

38. TCP协议（传输控制协议Transmission Control Protocol ）

建立链接：三次握手(1.给对方拨号 2.喂，您好，请问是哪位 3 我是小刘)

关闭链接：四次挥手保证可靠

39. WEB 前端优化？

减少HTTP请求的数量（合并css、js、图片）

利用浏览器的缓存机制

利用GZIP压缩机制：只针对文本类资源有效

把CSS文件放在HTML开头

把javascript文件放在HTML结尾

避免CSS表达式（判断浏览器）

使用javascript压缩

减少DNS查找

避免重定向

40. 简述软件开发文档?

需求分析、概要设计、详细设计、操作手册、测试计划

**Docker面试题**

镜像相关

1、如何批量清理临时镜像文件？

docker rmi $(docker images -q -f danging=true)

2、如何查看镜像支持的环境变量？

docker run IMAGE env

3、本地的镜像文件都存放在哪里？

与Docker相关的本地资源存放在/var/lib/docker/目录下，其中container目录存放容器信息，graph目录存放镜像信息，aufs目录下存放具体的镜像底层文件。

4、构建Docker镜像应该遵循哪些原则？

整体原则上，尽量保持镜像功能的明确和内容的精简，要点包括：

# 尽量选取满足需求但较小的基础系统镜像，建议选择debian:wheezy镜像，仅有86MB大小

# 清理编译生成文件、安装包的缓存等临时文件

# 安装各个软件时候要指定准确的版本号，并避免引入不需要的依赖

# 从安全的角度考虑，应用尽量使用系统的库和依赖

# 使用Dockerfile创建镜像时候要添加.dockerignore文件或使用干净的工作目录

**容器相关**

1、容器退出后，通过docker ps 命令查看不到，数据会丢失么？

容器退出后会处于终止（exited）状态，此时可以通过 docker ps -a 查看，其中数据不会丢失，还可以通过docker start 来启动，只有删除容器才会清除数据。

2、如何停止所有正在运行的容器？

docker kill $(docker ps -q)

3、如何清理批量后台停止的容器？

docker rm $（docker ps -a -q）

4、如何获取某个容器的PID信息?

docker  inspect  --format '{{ .State.Pid }}' <CONTANINERID or NAME>

5、如何获取某个容器的IP地址?

docker  inspect  --format  '{{ >NetworkSettings.IPAddress }}'  <CONTANINERID or NAME>

6、如何给容器指定一个固定IP地址，而不是每次重启容器IP地址都会变？

目前Docker并没有提供直接的对容器IP地址的管理支持，可以在网上查找容器网络配置创建点对点连接的案例，来手动配置容器的静态IP。或者在容器启动后，再手动进行修改。

7、如何临时退出一个正在交互的容器的终端，而不终止它？

按Ctrl+p，后按Ctrl+q，如果按Ctrl+c会使容器内的应用进程终止，进而会使容器终止。

8、很多应用容器都是默认后台运行的，怎么查看它们的输出和日志信息？

docker logs 容器名称或容器id

9、使用docker port 命令映射容器的端口时，系统报错Error: No public port ‘80’ published for …，是什么意思？

创建镜像时Dockerfile要指定正确的EXPOSE的端口，容器启动时指定PublishAllport=true

10、可以在一个容器中同时运行多个应用进程吗？

一般不推荐在同一个容器内运行多个应用进程，如果有类似需求，可以通过额外的进程管理机制，比如supervisord来管理所运行的进程

11、如何控制容器占用系统资源（CPU，内存）的份额？

在使用docker create命令创建容器或使用docker run 创建并运行容器的时候，可以使用-c|–cpu-shares[=0]参数来调整同期使用CPU的权重，使用-m|–memory参数来调整容器使用内存的大小。

仓库相关

1、仓库（Repository）、注册服务器（Registry）、注册索引（Index）有何关系？

首先，仓库是存放一组关联镜像的集合，比如同一个应用的不同版本的镜像，注册服务器是存放实际的镜像的地方，注册索引则负责维护用户的账号，权限，搜索，标签等管理。注册服务器利用注册索引来实现认证等管理。

2 、从非官方仓库（如：dl.dockerpool.com）下载镜像的时候，有时候会提示“Error：Invaild registry endpoint https://dl.docker.com:5000/v1/…”?

Docker 自1.3.0版本往后以来，加强了对镜像安全性的验证，需要手动添加对非官方仓库的信任。DOCKER\_OPTS=”–insecure-registry dl.dockerpool.com:5000”   重启docker服务

3.docker pull  某一个镜像很慢，比如说centos. 这种情况是因为docker官方的镜像库是国外，我们应该配置国内的镜像库或者加速器。

加速器可以用阿里云镜像加速器，进入阿里云网站在控制台可以看到，根据提示来即可。配置文件 /etc/docker/daemon.json 添加如下命令

DOCKER\_NETWORK\_OPTIONS="--registry-mirror=https://....."

地址可以为国内镜像地址，也可为公司内网镜像仓库地址。

4.docker push镜像时碰到的443问题，这是因为docker默认会走ssl加密的。

如果是公司内网的，直接在/etc/sysconfig/docker-network 里面添加如下配合

DOCKER\_NETWORK\_OPTIONS=“--insecure-registry xxx-ip”

5.碰到网络问题，无法pull镜像，命令行指定http\_proxy无效，怎么办？

在Docker配置文件中添加export http\_proxy="http://<PROXY\_HOST>:<PROXY\_PORT>"，

之后重启Docker服务即可。

**配置相关**

1、Docker的配置文件放在那里。如何修改配置？

Ubuntu系统下Docker的配置文件是/etc/default/docker，CentOS系统配置文件存放在/etc/sysconfig/docker

2、如何更改Docker的默认存储设置？

Docker的默认存放位置是/var/lib/docker,如果希望将Docker的本地文件存储到其他分区，可以使用Linux软连接的方式来做。

**Docker与虚拟化**

1、Docker与LXC（Linux Container）有何不同？

LXC利用Linux上相关技术实现容器，Docker则在如下的几个方面进行了改进：

移植性：通过抽象容器配置，容器可以实现一个平台移植到另一个平台；

镜像系统：基于AUFS的镜像系统为容器的分发带来了很多的便利，同时共同的镜像层只需要存储一份，实现高效率的存储；

版本管理：类似于GIT的版本管理理念，用户可以更方面的创建、管理镜像文件；

仓库系统：仓库系统大大降低了镜像的分发和管理的成本；

周边工具：各种现有的工具（配置管理、云平台）对Docker的支持，以及基于Docker的Pass、CI等系统，让Docker的应用更加方便和多样化。

2 、Docker与Vagrant有何不同？

两者的定位完全不同

Vagrant类似于Boot2Docker（一款运行Docker的最小内核），是一套虚拟机的管理环境，Vagrant可以在多种系统上和虚拟机软件中运行，可以在Windows。Mac等非Linux平台上为Docker支持，自身具有较好的包装性和移植性。

原生Docker自身只能运行在Linux平台上，但启动和运行的性能都比虚拟机要快，往往更适合快速开发和部署应用的场景。

3、开发环境中Docker与Vagrant该如何选择？

Docker不是虚拟机，而是进程隔离，对于资源的消耗很少，单一开发环境下Vagrant是虚拟机上的封装，虚拟机本身会消耗资源。

Other FAQ

1、Docker能在非Linux平台（Windows+MacOS）上运行吗？

可以

2 、如何将一台宿主机的docker环境迁移到另外一台宿主机？

停止Docker服务，将整个docker存储文件复制到另外一台宿主机上，然后调整另外一台宿主机的配置即可

3、Docker容器创建后，删除了/var/run/netns 目录下的网络名字空间文件，可以手动恢复它？

# 查看容器进程ID，比如1234

docker inspect --format='{{. State.pid}}' $container\_id

# 到proc目录下，把对应的网络名字空间文件链接到/var/run/netns,然后通过正常的系统命令查看操作容器的名字空间。

**MySQL面试**

1. Mysql中有哪几种锁？

1.表级锁：开销小，加锁快；不会出现死锁；锁定粒度大，发生锁冲突的概率最高，并发度最低。

2.行级锁：开销大，加锁慢；会出现死锁；锁定粒度最小，发生锁冲突的概率最低，并发度也最高。

3. 页面锁：开销和加锁时间界于表锁和行锁之间；会出现死锁；锁定粒度界于表锁和行锁之间，并发度一般。

2. Mysql中有哪些不同的表格？ 共有5种类型的表格：

MyISAM、Heap、Merge、INNODB、ISAM

3. 简述在MySQL数据库中MyISAM和InnoDB的区别

MyISAM：

1.不支持事务，但是每次查询都是原子的；

2.支持表级锁，即每次操作是对整个表加锁；

3.存储表的总行数；

4.一个MYISAM表有三个文件：索引文件、表结构文件、数据文件；

5. 采用菲聚集索引，索引文件的数据域存储指向数据文件的指针。辅索引与主索引基本一致，但是辅索引不用保证唯一性。

InnoDb：

1.支持ACID的事务，支持事务的四种隔离级别；

2.支持行级锁及外键约束：因此可以支持写并发；

3.不存储总行数；

4.一个InnoDb引擎存储在一个文件空间（共享表空间，表大小不受操作系统控制，一个表可能分布在多个文件里），也有可能为多个（设置为独立表空，表大小受操作系统文件大小限制，一般为2G），受操作系统文件大小的限制；

5.主键索引采用聚集索引（索引的数据域存储数据文件本身），辅索引的数据域存储主键的值；因此从辅索引查找数据，需要先通过辅索引找到主键值，再访问辅索引；最好使用自增主键，防止插入数据时，为维持B+树结构，文件的大调整。

4. Mysql中InnoDB支持的四种事务隔离级别名称，以及逐级之间的区别？

SQL标准定义的四个隔离级别为：

read uncommited ：读到未提交数据

read committed：脏读，不可重复读

repeatable read：可重读

serializable ：串行事物

5. CHAR和VARCHAR的区别？

1.CHAR和VARCHAR类型在存储和检索方面有所不同

2.CHAR列长度固定为创建表时声明的长度，长度值范围是1到255

当CHAR值被存储时，它们被用空格填充到特定长度，检索CHAR值时需删除尾随空格。

6. 主键和候选键有什么区别？

表格的每一行都由主键唯一标识,一个表只有一个主键。

主键也是候选键。按照惯例，候选键可以被指定为主键，并且可以用于任何外键引用。

7. myisamchk是用来做什么的？

它用来压缩MyISAM表，这减少了磁盘或内存使用。

8. MyISAM Static和MyISAM Dynamic有什么区别？

在MyISAM Static上的所有字段有固定宽度。动态MyISAM表将具有像TEXT，BLOB等字段，以适应不同长度的数据类型。

MyISAM Static在受损情况下更容易恢复。

9. 如果一个表有一列定义为TIMESTAMP，将发生什么？

每当行被更改时，时间戳字段将获取当前时间戳。

10. 列设置为AUTO INCREMENT时，如果在表中达到最大值，会发生什么情况？

它会停止递增，任何进一步的插入都将产生错误，因为密钥已被使用。

11. 怎样才能找出最后一次插入时分配了哪个自动增量？

LAST\_INSERT\_ID将返回由Auto\_increment分配的最后一个值，并且不需要指定表名称。

12. 你怎么看到为表格定义的所有索引？

索引是通过以下方式为表格定义的：

SHOW INDEX FROM <tablename>;

13. LIKE声明中的％和\_是什么意思？

％对应于0个或更多字符，\_只是LIKE语句中的一个字符。

14. 如何在Unix和Mysql时间戳之间进行转换？

UNIX\_TIMESTAMP是从Mysql时间戳转换为Unix时间戳的命令

FROM\_UNIXTIME是从Unix时间戳转换为Mysql时间戳的命令

15. 列对比运算符是什么？

在SELECT语句的列比较中使用=，<>，<=，<，> =，>，<<，>>，<=>，AND，OR或LIKE运算符。

16. BLOB和TEXT有什么区别？

BLOB是一个二进制对象，可以容纳可变数量的数据。TEXT是一个不区分大小写的BLOB。

BLOB和TEXT类型之间的唯一区别在于对BLOB值进行排序和比较时区分大小写，对TEXT值不区分大小写。

17. mysql\_fetch\_array和mysql\_fetch\_object的区别是什么？

mysql\_fetch\_array（） – 将结果行作为关联数组或来自数据库的常规数组返回。

mysql\_fetch\_object – 从数据库返回结果行作为对象。

18. MyISAM表格将在哪里存储，并且还提供其存储格式？

每个MyISAM表格以三种格式存储在磁盘上：

·“.frm”文件存储表定义

·数据文件具有“.MYD”（MYData）扩展名

19. 索引文件具有“.MYI”（MYIndex）扩展名

Mysql如何优化DISTINCT？

20. DISTINCT在所有列上转换为GROUP BY，并与ORDER BY子句结合使用。

SELECT DISTINCT t1.a FROM t1,t2 where t1.a=t2.a;

21. 如何显示前50行？

在Mysql中，使用以下代码查询显示前50行：

SELECT \* FROM LIMIT 0,50;

22. 可以使用多少列创建索引？

任何标准表最多可以创建16个索引列。

23. NOW（）和CURRENT\_DATE（）有什么区别？

NOW（）命令用于显示当前年份，月份，日期，小时，分钟和秒。

CURRENT\_DATE（）仅显示当前年份，月份和日期。

24. 什么是非标准字符串类型？

TINYTEXT

TEXT

MEDIUMTEXT

LONGTEXT

25. 什么是通用SQL函数？

CONCAT(A, B) – 连接两个字符串值以创建单个字符串输出。通常用于将两个或多个字段合并为一个字段。

FORMAT(X, D)- 格式化数字X到D有效数字。

CURRDATE(), CURRTIME()- 返回当前日期或时间。

NOW（） – 将当前日期和时间作为一个值返回。

MONTH（），DAY（），YEAR（），WEEK（），WEEKDAY（） – 从日期值中提取给定数据。

HOUR（），MINUTE（），SECOND（） – 从时间值中提取给定数据。

DATEDIFF（A，B） – 确定两个日期之间的差异，通常用于计算年龄

SUBTIMES（A，B） – 确定两次之间的差异。

FROMDAYS（INT） – 将整数天数转换为日期值。

26. MYSQL支持事务吗？

在缺省模式下，MYSQL是autocommit模式的，所有的数据库更新操作都会即时提交，所以在缺省情况下，mysql是不支持事务的。

但是如果你的MYSQL表类型是使用InnoDB Tables 或 BDB tables的话，你的MYSQL就可以使用事务处理,使用SET AUTOCOMMIT=0就可以使MYSQL允许在非autocommit模式，在非autocommit模式下，你必须使用COMMIT来提交你的更改，或者用ROLLBACK来回滚你的更改。

27. mysql里记录货币用什么字段类型好

NUMERIC和DECIMAL类型被Mysql实现为同样的类型，这在SQL92标准允许。他们被用于保存值，该值的准确精度是极其重要的值，例如与金钱有关的数据。当声明一个类是这些类型之一时，精度和规模的能被(并且通常是)指定。

例如：

salary DECIMAL(9,2) 在这个例子中，9(precision)代表将被用于存储值的总的小数位数，而2(scale)代表将被用于存储小数点后的位数。因此，在这种情况下，能被存储在salary列中的值的范围是从-9999999.99到9999999.99。

28. mysql有关权限的表都有哪几个？

MySQL服务器通过权限表来控制用户对数据库的访问，权限表存放在mysql数据库里，由mysql\_install\_db脚本初始化。这些权限表分别user，db，table\_priv，columns\_priv和host。

29. 列的字符串类型可以是什么？

字符串类型是：

SET

BLOB

ENUM

CHAR

TEXT

30. MySQL数据库作发布系统的存储，一天五万条以上的增量，预计运维三年,怎么优化？

a. 设计良好的数据库结构，允许部分数据冗余，尽量避免join查询，提高效率。

b. 选择合适的表字段数据类型和存储引擎，适当的添加索引。

c. mysql库主从读写分离。

d. 找规律分表，减少单表中的数据量提高查询速度。

e。添加缓存机制，比如memcached，apc等。

f. 不经常改动的页面，生成静态页面。

g. 书写高效率的SQL。比如 SELECT \* FROM TABEL 改为 SELECT field\_1, field\_2, field\_3 FROM TABLE.

31. 锁的优化策略

1. 读写分离

2. 分段加锁

3. 减少锁持有的时间

4. 多个线程尽量以相同的顺序去获取资源

5. 不能将锁的粒度过于细化，不然可能会出现线程的加锁和释放次数过多，反而效率不如一次加一把大锁。

32. 索引的底层实现原理和优化

B+树，经过优化的B+树

主要是在所有的叶子结点中增加了指向下一个叶子节点的指针，因此InnoDB建议为大部分表使用默认自增的主键作为主索引。

33. 什么情况下设置了索引但无法使用

1.以“%”开头的LIKE语句，模糊匹配

2. OR语句前后没有同时使用索引

3. 数据类型出现隐式转化（如varchar不加单引号的话可能会自动转换为int型）

34. 实践中如何优化MySQL,最好是按照以下顺序优化：

1.SQL语句及索引的优化

2. 数据库表结构的优化

3.系统配置的优化

4.硬件的优化

5. 详细可以查看 阿里P8架构师谈：MySQL慢查询优化、索引优化、以及表等优化总结

35. 优化数据库的方法

1. 选取最适用的字段属性，尽可能减少定义字段宽度，尽量把字段设置NOTNULL，例如’省份’、’性别’最好适用ENUM

2. 使用连接(JOIN)来代替子查询

3. 适用联合(UNION)来代替手动创建的临时表

4. 事务处理

锁定表、优化事务处理

适用外键，优化锁定表

5.建立索引，优化查询语句

简单描述mysql中，索引，主键，唯一索引，联合索引的区别，对数据库的性能有什么影响（从读写两方面）

索引是一种特殊的文件(InnoDB数据表上的索引是表空间的一个组成部分)，它们包含着对数据表里所有记录的引用指针。

普通索引(由关键字KEY或INDEX定义的索引)的唯一任务是加快对数据的访问速度。

普通索引允许被索引的数据列包含重复的值。如果能确定某个数据列将只包含彼此各不相同的值，在为这个数据列创建索引的时候就应该用关键字UNIQUE把它定义为一个唯一索引。也就是说，唯一索引可以保证数据记录的唯一性。

主键，是一种特殊的唯一索引，在一张表中只能定义一个主键索引，主键用于唯一标识一条记录，使用关键字 PRIMARY KEY 来创建。

索引可以覆盖多个数据列，如像INDEX(columnA, columnB)索引，这就是联合索引。

索引可以极大的提高数据的查询速度，但是会降低插入、删除、更新表的速度，因为在执行这些写操作时，还要操作索引文件。

36. 数据库中的事务是什么?

事务（transaction）是作为一个单元的一组有序的数据库操作。如果组中的所有操作都成功，则认为事务成功，即使只有一个操作失败，事务也不成功。如果所有操作完成，事务则提交，其修改将作用于所有其他数据库进程。如果一个操作失败，则事务将回滚，该事务所有操作的影响都将取消。

37. 事务特性：

（1）原子性：即不可分割性，事务要么全部被执行，要么就全部不被执行。

（2）一致性或可串性。事务的执行使得数据库从一种正确状态转换成另一种正确状态

（3）隔离性。在事务正确提交之前，不允许把该事务对数据的任何改变提供给任何其他事务，

（4）持久性。事务正确提交后，其结果将永久保存在数据库中，即使在事务提交后有了其他故障，事务的处理结果也会得到保存。

38. SQL注入漏洞产生的原因？如何防止？

SQL注入产生的原因：程序开发过程中不注意规范书写sql语句和对特殊字符进行过滤，导致客户端可以通过全局变量POST和GET提交一些sql语句正常执行。

39. 防止SQL注入的方式：

开启配置文件中的magic\_quotes\_gpc 和 magic\_quotes\_runtime设置

执行sql语句时使用addslashes进行sql语句转换

Sql语句书写尽量不要省略双引号和单引号。

过滤掉sql语句中的一些关键词：update、insert、delete、select、 \* 。

提高数据库表和字段的命名技巧，对一些重要的字段根据程序的特点命名，取不易被猜到的。

为表中得字段选择合适得数据类型

字段类型优先级: 整形>date,time>enum,char>varchar>blob,text

优先考虑数字类型，其次是日期或者二进制类型，最后是字符串类型，同级别得数据类型，应该优先选择占用空间小的数据类型

40. 对于关系型数据库而言，索引是相当重要的概念，请回答有关索引的几个问题：

1.索引的目的是什么？

快速访问数据表中的特定信息，提高检索速度

创建唯一性索引，保证数据库表中每一行数据的唯一性。

加速表和表之间的连接

使用分组和排序子句进行数据检索时，可以显著减少查询中分组和排序的时间

2.索引对数据库系统的负面影响是什么？

负面影响：创建索引和维护索引需要耗费时间，这个时间随着数据量的增加而增加；索引需要占用物理空间，不光是表需要占用数据空间，每个索引也需要占用物理空间；当对表进行增、删、改、的时候索引也要动态维护，这样就降低了数据的维护速度。

3.为数据表建立索引的原则有哪些？

在最频繁使用的、用以缩小查询范围的字段上建立索引。

在频繁使用的、需要排序的字段上建立索引

4.什么情况下不宜建立索引？

对于查询中很少涉及的列或者重复值比较多的列，不宜建立索引。

5. 对于一些特殊的数据类型，不宜建立索引，比如文本字段（text）等

41. 解释MySQL外连接、内连接与自连接的区别

1.先说什么是交叉连接: 交叉连接又叫笛卡尔积，它是指不使用任何条件，直接将一个表的所有记录和另一个表中的所有记录一一匹配。

2.内连接 则是只有条件的交叉连接，根据某个条件筛选出符合条件的记录，不符合条件的记录不会出现在结果集中，即内连接只连接匹配的行。

3.外连接 其结果集中不仅包含符合连接条件的行，而且还会包括左表、右表或两个表中的所有数据行，这三种情况依次称之为左外连接，右外连接，和全外连接。

4.左外连接，也称左连接，左表为主表，左表中的所有记录都会出现在结果集中，对于那些在右表中并没有匹配的记录，仍然要显示，右边对应的那些字段值以NULL来填充。右外连接，也称右连接，右表为主表，右表中的所有记录都会出现在结果集中。左连接和右连接可以互换，MySQL目前还不支持全外连接。

42.MySQL中的事务回滚机制概述

事务是用户定义的一个数据库操作序列，这些操作要么全做要么全不做，是一个不可分割的工作单位，事务回滚是指将该事务已经完成的对数据库的更新操作撤销。

要同时修改数据库中两个不同表时，如果它们不是一个事务的话，当第一个表修改完，可能第二个表修改过程中出现了异常而没能修改，此时就只有第二个表依旧是未修改之前的状态，而第一个表已经被修改完毕。而当你把它们设定为一个事务的时候，当第一个表修改完，第二表修改出现异常而没能修改，第一个表和第二个表都要回到未修改的状态，这就是所谓的事务回滚

43. SQL语言包括哪几部分？每部分都有哪些操作关键字？

SQL语言包括数据定义(DDL)、数据操纵(DML),数据控制(DCL)和数据查询（DQL）四个部分。

数据定义：Create Table,Alter Table,Drop Table, Craete/Drop Index等

数据操纵：Select ,insert,update,delete,

数据控制：grant,revoke

数据查询：select

44. 完整性约束包括哪些？

数据完整性(Data Integrity)是指数据的精确(Accuracy)和可靠性(Reliability)。

分为以下四类：

1) 实体完整性：规定表的每一行在表中是惟一的实体。

2) 域完整性：是指表中的列必须满足某种特定的数据类型约束，其中约束又包括取值范围、精度等规定。

3) 参照完整性：是指两个表的主关键字和外关键字的数据应一致，保证了表之间的数据的一致性，防止了数据丢失或无意义的数据在数据库中扩散。

4) 用户定义的完整性：不同的关系数据库系统根据其应用环境的不同，往往还需要一些特殊的约束条件。用户定义的完整性即是针对某个特定关系数据库的约束条件，它反映某一具体应用必须满足的语义要求。

与表有关的约束：包括列约束(NOT NULL（非空约束）)和表约束(PRIMARY KEY、foreign key、check、UNIQUE) 。

45. 什么是锁？

答：数据库是一个多用户使用的共享资源。当多个用户并发地存取数据时，在数据库中就会产生多个事务同时存取同一数据的情况。若对并发操作不加控制就可能会读取和存储不正确的数据，破坏数据库的一致性。

加锁是实现数据库并发控制的一个非常重要的技术。当事务在对某个数据对象进行操作前，先向系统发出请求，对其加锁。加锁后事务就对该数据对象有了一定的控制，在该事务释放锁之前，其他的事务不能对此数据对象进行更新操作。

基本锁类型：锁包括行级锁和表级锁

46. 什么叫视图？游标是什么？

答：视图是一种虚拟的表，具有和物理表相同的功能。可以对视图进行增，改，查，操作，视图通常是有一个表或者多个表的行或列的子集。对视图的修改不影响基本表。它使得我们获取数据更容易，相比多表查询。

游标：是对查询出来的结果集作为一个单元来有效的处理。游标可以定在该单元中的特定行，从结果集的当前行检索一行或多行。可以对结果集当前行做修改。一般不使用游标，但是需要逐条处理数据的时候，游标显得十分重要。

47. 什么是存储过程？用什么来调用？

答：存储过程是一个预编译的SQL语句，优点是允许模块化的设计，就是说只需创建一次，以后在该程序中就可以调用多次。如果某次操作需要执行多次SQL，使用存储过程比单纯SQL语句执行要快。可以用一个命令对象来调用存储过程。

48. 如何通俗地理解三个范式？

第一范式：1NF是对属性的原子性约束，要求属性具有原子性，不可再分解；

第二范式：2NF是对记录的惟一性约束，要求记录有惟一标识，即实体的惟一性；

第三范式：3NF是对字段冗余性的约束，即任何字段不能由其他字段派生出来，它要求字段没有冗余。。

49. 范式化设计优缺点:

优点: 可以尽量得减少数据冗余，使得更新快，体积小

缺点:对于查询需要多个表进行关联，减少写得效率增加读得效率，更难进行索引优化

50. 反范式化:

优点:可以减少表得关联，可以更好得进行索引优化

缺点:数据冗余以及数据异常，数据得修改需要更多的成本

51. 什么是基本表？什么是视图？

答：基本表是本身独立存在的表，在 SQL 中一个关系就对应一个表。 视图是从一个或几个基本表导出的表。视图本身不独立存储在数据库中，是一个虚表

52. 试述视图的优点？

答：(1) 视图能够简化用户的操作 (2) 视图使用户能以多种角度看待同一数据； (3) 视图为数据库提供了一定程度的逻辑独立性； (4) 视图能够对机密数据提供安全保护。

53. NULL是什么意思

答：NULL这个值表示UNKNOWN(未知):它不表示“”(空字符串)。对NULL这个值的任何比较都会生产一个NULL值。您不能把任何值与一个 NULL值进行比较，并在逻辑上希望获得一个答案。

使用IS NULL来进行NULL判断

54. 主键、外键和索引的区别？

主键、外键和索引的区别

定义：

1.主键–唯一标识一条记录，不能有重复的，不允许为空

2.外键–表的外键是另一表的主键, 外键可以有重复的, 可以是空值

3.索引–该字段没有重复值，但可以有一个空值

作用：

1.主键–用来保证数据完整性

2.外键–用来和其他表建立联系用的

3.索引–是提高查询排序的速度

个数：

1.主键–主键只能有一个

2.外键–一个表可以有多个外键

3.索引–一个表可以有多个唯一索引

55. 你可以用什么来确保表格里的字段只接受特定范围里的值?

答：Check限制，它在数据库表格里被定义，用来限制输入该列的值。触发器也可以被用来限制数据库表格里的字段能够接受的值，但是这种办法要求触发器在表格里被定义，这可能会在某些情况下影响到性能。

56. 说说对SQL语句优化有哪些方法？（选择几条）

（1）Where子句中：where表之间的连接必须写在其他Where条件之前，那些可以过滤掉最大数量记录的条件必须写在Where子句的末尾.HAVING最后。

（2）用EXISTS替代IN、用NOT EXISTS替代NOT IN。

（3）避免在索引列上使用计算

（4）避免在索引列上使用IS NULL和IS NOT NULL

（5）对查询进行优化，应尽量避免全表扫描，首先应考虑在 where 及 order by 涉及的列上建立索引。

（6）应尽量避免在 where 子句中对字段进行 null 值判断，否则将导致引擎放弃使用索引而进行全表扫描

（7）应尽量避免在 where 子句中对字段进行表达式操作，这将导致引擎放弃使用索引而进行全表扫描