



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

## 第三版：Python 70 道

### threading.local 的作用

为每个线程创建一个独立的空间，使得线程对自己的空间中的数据进行操作(数据隔离)。

### 进程之间如何进行通信?

#### 1、 共享内存

通过 mmap 模块实现

#### 2、 信号

#### 3、 通过 Queue 队列

#### 4、 通过 Pipe 管道

#### 5、 通过 socket

### 什么是并发和并行

1、 并发:指应用能够交替执行不同的任务,其实并发有点类似于多线程的原理,多线程并非是同时执行多个任务,如果你开两个线程执行,就是在你几乎不可能察觉到的速度不断去切换这两个任务,已达到“同时执行效果”,其实并不是的,只是计算机的速度太快,我们无法察觉到而已。

2、 并行:指应用能够同时执行不同的任务,

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

3、 并发是多个事件在同一时间段执行，并行是多个事件在统一时间点执行。

## 解释什么是异步非阻塞

1、 同步和异步：

2、 同步：就是在发出一个功能调用时，在没有得到结果之前，该调用就不返回。

3、 异步：当一个异步过程调用发出后，调用者不会立刻得到结果。实际处理这个调用的部件是在调用发出后，通过状态、通知来通知调用者，或通过回调函数处理这个调用。

4、 阻塞和非阻塞

5、 阻塞：阻塞调用是指调用结果返回之前，当前线程会被挂起。函数只有在得到结果之后才会返回。

6、 非阻塞：指在不能立刻得到结果之前，该函数不会阻塞当前线程，而会立刻返回。

7、 阻塞，非阻塞：进程/线程要访问的数据是否就绪，进程/线程是否需要等待；

8、 同步，异步：访问数据的方式，同步需要主动读写数据，在读写数据的过程中还是会阻塞；异步只需要 I/O 操作完成的通知，并不主动读写数据，由操作系统内核完成数据的读写。

## 路由器和交换机的区别

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

交换机是一种用于电信号转发的网络设备。路由器是链接因特网中各局域网和广域网的设备。

区别

- 1、 交换机工作在第二层，数据链路层，路由器工作在第三层，网络层。
- 2、 路由器提供防火墙服务。
- 3、 传统交换机只能风格冲突域，不能分割广播域，二路由器可以分割广播域。

## 什么是域名解析

域名解析是把域名指向网站空间 IP，让人们通过注册的域名可以方便地访问到网站一种服务。IP 地址是网络上标识站点的数字地址，为方便记忆，采用域名来代替 IP 地址标识站点地址。域名解析就是域名到 IP 地址的转换过程。

## 如何修改本地 hosts 文件

进入 c:\windows\system32\drivers\etc 进行修改

## 生产者消费者模型的应用场景

说明

生产者只在仓库未滿时进行生产，仓库满时生产者进程被阻塞；消费者只在仓库非空时进行消费，仓库为空时消费者进程被阻塞；

应用场景：处理数据比较消耗时间，线程独占，生产数据不需要即时的反馈等。

比如说写入日志，将多线程产生的日志放在队列中，然后写入。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

## 什么是 cdn

cdn 全称是内容分发网络。其目的是让用户能够更快速的得到请求的数据。  
cdn 就是用来加速的，他能让用户就近访问数据，这样就更更快的获取到需要的数据。

## traceroute 使用哪种网络协议

traceroute (Windows 系统下是 tracert) 命令利用 ICMP 协议定位您的计算机和目标计算机之间的所有路由器。

## 守护线程，守护进程是什么

主进程创建守护进程

- 1、守护进程会在主进程代码运行结束的情况下，立即挂掉。
- 2、守护进程本身就是一个子进程。
- 3、主进程在其代码结束后就已经算运行完毕了（守护进程在此时就被回收），然后主进程会一直等非守护的子进程都运行完毕后回收子进程的资源(否则会产生僵尸进程)，才会结束。

守护线程

- 1、守护线程会在“该进程内所有非守护线程全部都运行完毕后,守护线程才会挂掉”。并不是主线程运行完毕后守护线程挂掉。这一点是和守护进程的区别之处！

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

2、守护线程守护的是：当前进程内所有的子线程！

3、主线程在其他非守护线程运行完毕后才算运行完毕（守护线程在此时就被回收）。因为主线程的结束意味着进程的结束，进程整体的资源都将被回收，而进程必须保证非守护线程都运行完毕后才能结束。

## 简述多进程开发中 join 和 daemon 的区别

1、join：当子线程调用 join 时，主线程会被阻塞，当子线程结束后，主线程才能继续执行。

2、daemon：当子进程被设置为守护进程时，主进程结束，不管子进程是否执行完毕，都会随着主进程的结束而结束。

## GIL 锁对 python 性能的影响

1、会降低多线程的效率。可以说 python 就是个单线程的程序。

2、如何避免：

3、用多进程代替多线程

4、使用其他解释器

## 使用 yield 实现一个协程

```
def consumer():  
    r = "
```

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

```
while True:
n = yield r
if n is None:
return
print('[CONSUMER] Consuming %s...' % n)
r = '200 OK'

def produce(c):
c.send(None)
n = 0
while n < 5:
n = n + 1
print('[PRODUCER] Producing %s...' % n)
r = c.send(n)
print('[PRODUCER] Consumer return: %s' % r)
c.close()

c = consumer()
produce(c)
```

## 使用 async 语法实现一个协程

```
import asyncio
import time

now = lambda : time.time()

async def hello():
print("hello")
await asyncio.sleep(2)
```

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

```
return "done"

start = now()
# 协程对象
h1 = hello()
h2 = hello()
h3 = hello()

# 创建一个事件 loop
loop = asyncio.get_event_loop()
# 任务 (task) 对象
tasks = [
    asyncio.ensure_future(h1),
    asyncio.ensure_future(h2),
    asyncio.ensure_future(h3),
]

# 将协程加入到事件循环 loop
loop.run_until_complete(asyncio.wait(tasks))
for task in tasks:
    print(task.result())

print(now()-start)
```

**简述线程死锁是怎么造成的。如何避免？**

- 1、死锁的产生原因？
- 2、系统资源的竞争

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

3、 进程运行推进顺序不当

4、 解决死锁

5、 加锁顺序：线程按照一定的顺序加锁

6、 加锁时限：线程尝试获取锁的时候加上一定的时限，超过时限，则放弃对该锁的请求，并释放自己占有的锁。

7、 死锁检测

## 什么是 asyncio

asyncio 是并发的一种方式，是一个协程相关的库，也叫异步 IO

## 什么是 gevent

gevent 是一个 python 网络框架，它为各种并发和网络相关的任务提供了整洁的 API

## 什么是 twisted 框架

twisted 是用 python 实现的基于事件驱动的网络引擎框架。

## 什么是 LVS

LVS 是 linux 虚拟服务器，是一个虚拟的 linux 集群系统。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题





微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

## 什么是 Nginx

nginx 是一个高性能的 HTTP 和反向代理服务，也是一个 IMAP/POP3/SMTP 服务

## 什么是 keepalived

Keepalived 是 Linux 下一个轻量级别的高可用解决方案

## 什么是 haproxy?

HAProxy 是一个使用 C 语言编写的自由及开放源代码软件，其提供高可用性、负载均衡，以及基于 TCP 和 HTTP 的应用程序代理

## 什么是负载均衡

负载均衡建立在现有网络结构之上，它提供了一种廉价有效透明的方法扩展网络设备和服务器的带宽、增加吞吐量、加强网络数据处理能力、提高网络的灵活性和可用性。

负载均衡其意思就是分摊到多个操作单元上进行执行。

## 什么是 rpc

远程过程调用 (RPC) 是一种协议，程序可使用这种协议向网络中的另一台计算机上的程序请求服务

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

1.RPC 采用客户机/服务器模式。请求程序就是一个客户机，而服务提供程序就是一个服务器。

2.首先，客户机调用进程发送一个有进程参数的调用信息到服务进程，然后等待应答信息。

2.在服务器端，进程保持睡眠状态直到调用信息到达为止。当一个调用信息到达，服务器获得进程参数，计算结果，发送答复信息，然后等待下一个调用信息，

3.最后，客户端调用进程接收答复信息，获得进程结果，然后调用执行继续进行。

## 什么是正向代理和反向代理？

### 正向代理

1、 正向代理类似一个跳板机，代理访问外部资源。

2、 正向代理 是一个位于客户端和原始服务器(origin server)之间的服务器，为了从原始服务器取得内容，客户端向代理发送一个请求并指定目标(原始服务器)，然后代理向原始服务器转交请求并将获得的内容返回给客户端。客户端必须要进行一些特别的设置才能使用正向代理。

### 正向代理作用：

1、 访问原来无法访问的资源，如 google

2、 可以做缓存，加速访问资源

3、 对客户端访问授权，上网进行认证

4、 代理可以记录用户访问记录（上网行为管理），对外隐藏用户信息

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

## 反向代理

1、反向代理 (Reverse Proxy) 实际运行方式是指以代理服务器来接受 internet 上的连接请求，然后将请求转发给内部网络上的服务器，并将从服务器上得到的结果返回给 internet 上请求连接的客户端，此时代理服务器对外就表现为一个服务器

### 反向代理的作用：

- 1、保证内网的安全，可以使用反向代理提供 WAF 功能，阻止 web 攻击
- 2、负载均衡，通过反向代理服务器来优化网站的负载

## 列举常见的关系型数据库和非关系型数据库。

- 1、关系型数据库：Oracle、DB2、Microsoft SQL Server、Microsoft Access、MySQL
- 2、非关系型数据库：mongodb, Redis, elasticsearch, NoSql

## MySQL 常见数据库引擎及区别

- 1、InnoDB：用于事务处理应用程序，具有众多特性，包括 ACID 事务支持。(提供行级锁)
- 2、MyISAM：默认的 MySQL 插件式存储引擎，它是在 Web、数据仓储和其他应用环境下最常使用的存储引擎之一。注意，通过更改 STORAGE\_ENGINE 配置变量，能够方便地更改 MySQL 服务器的默认存储引擎。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

3、 Memory：将所有数据保存再 RAM 中

## 简述事务及其特性

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

事务具有 4 个特性：原子性、一致性、隔离性和持续性。

- 1、 原子性：事务是数据库的逻辑工作单位，事务中包括的诸操作要么都做，要么都不做。
- 2、 一致性：事务执行的结果必须是使数据库从一个一致性状态变到另一个一致性状态。
- 3、 隔离性：一个事务的执行不能被其他事务干扰。即一个事务内部的操作及使用的数据对其他并发事务是隔离的，并发执行的各个事务之间不能互相干扰。
- 4、 持续性：持续性也称永久性，指一个事务一旦提交，它对数据库中数据的改变就应该是永久性的。接下来的其他操作或故障不应该对其执行结果有任何影响。

## 简述触发器、函数、视图和存储过程

- 1、 触发器：触发器是一个特殊的存储过程，它是 MySQL 在 insert、update、delete 的时候自动执行的代码块
- 2、 函数：MySQL 中提供了许多内置函数，还可以自定义函数（实现程序员需要 sql 逻辑处理）

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

3、视图：视图是由查询结果形成的一张虚拟表，是表通过某种运算得到的一个投影

4、存储过程：把一段代码封装起来，当要执行这一段代码的时候，可以通过调用该存储过程来实现（经过第一次编译后再次调用不需要再次编译，比一个个执行 sql 语句效率高）

## MySQL 索引种类

- 1、普通索引：仅加速查询
- 2、唯一索引：加速查询 + 列值唯一（可以有 null）
- 3、主键索引：加速查询 + 列值唯一（不可以有 null）+ 表中只有一个
- 4、组合索引：多列值组成一个索引，专门用于组合搜索，其效率大于索引合并
- 5、全文索引：对文本的内容进行分词，进行搜索

## 索引再什么情况下遵循最左前缀的规则？

在多字段进行索引的时候，会遵循以上原则

## MySQL 常见的函数

- 1、ABS(x)：返回 x 的绝对值
- 2、ROUND(x)：返回参数 x 的四舍五入的一个整数

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

3、 TRIM(str): 去除字符串两边的空白

4、 COUNT(): 返回值的个数

5、 AVG(): 返回平均值

6、 SUM(): 求和

### 列举创建索引但是无法命中索引的情况

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)

2、 对于多列索引, 不是使用的第一部分(第一个), 则不会使用索引

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

### 数据库的导入与导出命令

1、 导出(MySQLdump)

2、 导出数据和表结构

关注公众号: 磊哥聊编程, 回复: 面试题, 获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

3、 `MySQLdump -uroot -p dbname > dbname .sql`

4、 只导出表结构

5、 `MySQLdump -uroot -p -d dbname > dbname .sql`

6、 导入

7、 `MySQL -u 用户名 -p 密码 数据库名 < 数据库名.sql`

## 你了解哪些数据库优化方案

1、 减少数据访问-创建并正确使用索引

2、 返回更少的数据

3、 数据分页处理

4、 只返回需要的字段

5、 减少交互次数

6、 使用存储过程

7、 优化业务逻辑

8、 减少服务器的cpu 运算

9、 使用绑定变量

10、 合理使用排序

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

11、减少比较操作

12、大量复杂运算在客户端处理

13、利用更多资源

14、客户端多进程访问

15、数据库并行处理

## char 和 varchar 的区别

1、保存方式不一样

2、char 有固定的长度，而 varchar 属于可变长的字符类型

3、char 的效率比 varchar 高

## MySQL 执行计划的作用和使用方法

1、作用：用来进行查询分析，比如整个查询涉及多少表，使用哪些索引，运行时间等

2、使用方法：使用 explain 关键字，如 `explain select xxx from xxx;`

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题





微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

## 为什么数据很大的时候使用 limit offset 分页时，越往后翻速度越慢，如何优化？

使用 limit 分页时，比如 limit 10000, 20 的意思是扫描满足条件的 10020 行，扔掉前面的 10000 行，最后返回 20 行，问题就出在这，当数据量大的时候，大量时间花在了扫描上面了。

优化方法：

- 1、 当一个数据库表过于庞大，LIMIT offset, length 中的 offset 值过大，则 SQL 查询语句会非常缓慢，你需增加 order by，并且 order by 字段需要建立索引。
- 2、 如果使用子查询去优化 LIMIT 的话，则子查询必须是连续的，某种意义上讲，子查询不应该有 where 条件，where 会过滤数据，使数据失去连续性

## 什么是索引合并

- 1、 索引合并是把几个索引的范围扫描合并成一个索引。
- 2、 索引合并的时候，会对索引进行并集，交集或者先交集再并集操作，以便合并成一个索引。
- 3、 这些需要合并的索引只能是一个表的。不能对多表进行索引合并。

## 什么是覆盖索引

覆盖索引又可以称为索引覆盖。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

1、解释一：就是 select 的数据列只用从索引中就能够取得，不必从数据表中读取，换句话说查询列要被所使用的索引覆盖。

2、解释二：索引是高效找到行的一个方法，当能通过检索索引就可以读取想要的数据库，那就不需要再到数据表中读取行了。如果一个索引包含了（或覆盖了）满足查询语句中字段与条件的数据就叫做覆盖索引。

3、解释三：是非聚集组合索引的一种形式，它包括在查询里的 Select、Join 和 Where 子句用到的所有列（即建立索引的字段正好是覆盖查询语句[select 子句]与查询条件[Where 子句]中所涉及的字段，也即，索引包含了查询正在查找的所有数据）。

## 简述数据库的读写分离

读写分离就是在主服务器上修改，数据会同步到从服务器，从服务器只能提供读取数据，不能写入，实现备份的同时也实现了数据库性能的优化，以及提升了服务器安全。

## 简述数据库分库分表

垂直切分

垂直分库：

就是根据业务耦合性，将关联度低的不同表存储在不同的数据库

垂直分表：

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

基于数据库中的“列”进行，某个表字段较多，可以新建一张扩展表，将不经常用或字段长度较大的字段拆分出去到扩展表中。

### 横向切分：

水平切分分为库内分表和分库分表，是根据表内数据内在的逻辑关系，将同一个表按不同的条件分散到多个数据库或多个表中，每个表中只包含一部分数据，从而使得单个表的数据量变小，达到分布式的效果。

## 数据库锁的作用

锁分为三种：乐观锁，悲观锁和共享锁

数据库和操作系统一样，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。在实际应用中经常会遇到的与锁相关的异常情况，当两个事务需要一组有冲突的锁，而不能将事务继续下去的话，就会出现死锁，严重影响应用的正常执行。

## MySQL 的半同步复制原理

半同步复制，介于异步复制和全同步复制之间，主库在执行完客户端提交的事务后不是立刻返回给客户端，而是等待至少一个从库接收到并写到 relay log 中才返回给客户端。相对于异步复制，半同步复制牺牲了一定的性能，提高了数据的安全性。

异步复制，MySQL 默认的复制是异步的，主库在执行完客户端提交的事务后会立即将结果返回给客户端，并不关心从库是否已经接收并处理。原理最简单，性能最好，但是主从之间数据不一致的概率很大。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

全同步复制，指当主库执行完一个事务，所有的从库都执行了该事务才返回给客户端。因为需要等待所有从库执行完该事务才能返回，所以全同步复制的性能必然会收到严重的影响。

## MySQL 的增删改查

增

指定字段名

语法：INSERT INTO 表名 (字段名 1, 字段名 2, ...) VALUES (值 1, 值 2, ...);

举例：INSERT INTO student(id,name,grade) VALUES(1,'zhangshan',98);

不指定字段名

语法：INSERT INTO 表名 VALUES(值 1, 值 2, ...);

举例：INSERT INTO student VALUES (2,'lisi',62);

其他写法

语法：INSERT INTO 表名 SET 字段名 1=值 1[,字段名 2=值 2, ...]

举例：INSERT INTO student SET id=4, name='zhaoliu',grade=72;

同时添加多条数据

语法：INSERT INTO 表名[(字段名 1, 字段名 2, ...)] VALUES (值 1, 值 2, ...),  
(值 1, 值 2, ...), ... (值 1, 值 2, ...)

举例：INSERT INTO student VALUES (5, 'lilei',99), (6,'hanmeimei',87),  
(8,'poly',76);

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

删

#### 删除部分数据

语法：DELETE FROM 表名 [WHERE 条件表达式]

命令：DELETE FROM student WHERE id=7;

#### 删除全部数据

语法：DELETE FROM 表名

命令：DELETE FROM student;

#### 推荐的删除全部数据

语法：TRUNCATE [TABLE ] 表名

举例：TRUNCATE TABLE student

改

#### 更新部分数据

语法：UPDATE 表名 SET 字段名 1=值 1, [ , 字段名 2=值 2, ...] [WHERE 条件表达式 ]

命令：UPDATE student SET name= 'caocao' ,grade=50 WHERE id=1;

#### 更新全部数据

语法：UPDATE 表名 SET 字段名=值

命令：UPDATE student SET grade=80;

查

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

### 查询所有字段

语法: SELECT 字段名 1, 字段名 2, ... FROM 表名 (该语法也可以查询部分字段)

语法: SELECT FROM 表名;

### 按条件查询

语法: SELECT 字段名 1, 字段名 2, ... FROM 表名 WHERE 条件表达式

命令: SELECT id, name FROM student2 WHERE id=4;

### 带 IN 关键字的查询

1、语法: SELECT | 字段名 1, 字段名 2, ... FROM 表名 WHERE 字段名 [ NOT ] IN (元素 1, 元素 2, ...)

2、命令: SELECT FROM student2 WHERE id IN (1,2,3);

3、带 BETWEEN AND 关键字的查询

4、语法: SELECT | { 字段名 1, 字段名 2, ... } FROM 表名 WHERE 字段名 [ NOT ] BETWEEN 值 1 AND 值 2;

5、命令: SELECT id,name FROM students WHERE id BETWEEN 2 AND 5;

### 空值查询

语法: SELECT | 字段名 1, 字段名 2, ... FROM 表名 WHERE 字段名 IS [ NOT ] NULL

命令: SELECT FROM student2 WHERE gender IS NULL;

关注公众号: [磊哥聊编程](#), 回复: [面试题](#), 获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

### 带 DISTINCT 关键字的查询

语法：SELECT DISTINCT 字段名 FROM 表名；

命令：SELECT DISTINCT gender FROM student2；

### 带 LIKE 关键字的查询

语法：SELECT | 字段名 1, 字段名 2, ... FROM 表名 WHERE 字段名 [ NOT ] LIKE '匹配字符串'；

注意：%表示匹配任意长度的字符串，\_表示匹配单个字符串

3、命令：SELECT id,name FROM student2 WHERE name LIKE "S%";

4、命令：SELECT id,name FROM student2 WHERE name LIKE 'w%g';

5、命令：SELECT id,name FROM student2 WHERE name NOT LIKE '%y%';

6、命令：SELECT FROM student2 WHERE name LIKE 'wu\_ong';

### 带 AND 关键字的多条件查询

1、语法：SELECT | 字段名 1, 字段名 2, ... FROM 表名 WHERE 条件表达式 1 AND 条件表达式 2 [ ... AND 条件表达式 n ];

2、命令：SELECT id,name FROM student2 WHERE id<5 AND gender='女';

### 带 OR 关键字的多条件查询

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

1、语法：SELECT | 字段名 1, 字段名 2, ... FROM 表名 WHERE 条件表达式 1 OR 条件表达式 2 [ ... OR 条件表达式 n];

2、命令：SELECT id,name ,gender FROM student2 WHERE id<3 OR gender='女';

**AND 和 OR 一起使用时，AND 的优先级高于 OR**

聚合函数

**COUNT()函数：统计记录的条数**

语法：SELECT COUNT(\*) FROM 表名举例：

命令：SELECT COUNT(\*) FROM student2;

**SUM()函数：求出表中某个字段所有值的总和**

语法：SELECT SUM(字段名) FROM 表名；

命令：SELECT SUM(grade) FROM student2;

**AVG()函数：求出表中某个字段所有值的平均值**

语法：SELECT AVG(字段名) FROM 表名；

命令：SELECT AVG(grade) FROM student2;

**MAX()函数：求出表中某个字段所有值的最大值**

语法：SELECT MAX(字段名) FROM 表名；

命令：SELECT MAX(grade) FROM student2;

**MIN()函数：求出表中某个字段所有值的最小值**

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题





微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

语法：SELECT MIN(字段名) FROM 表名;

命令：SELECT MIN(grade) FROM student2;

### 对查询结果进行排序

语法：SELECT 字段名 1, 字段名 2, ... FROM 表名 ORDER BY 字段名 1 [ ASC | DESC ], 字段名 2 [ ASC | DESC ]

(升序)命令：SELECT FROM student2 ORDER BY grade;

(降序)命令：命令：SELECT FROM student2 ORDER BY grade DESC;

### 分组查询

语法：SELECT 字段名 1, 字段名 2, ... FROM 表名 GROUP BY 字段名 1, 字段名 2, ... [ HAVING 条件表达式 ];

### 单独使用 GROUP BY 进行分组

命令：SELECT FROM student2 GROUP BY gender;

### GROUP BY 和聚合函数一起使用

命令：SELECT COUNT(\*) ,gender FROM student2 GROUP BY gender;

### GROUP BY 和 HAVING 关键字一起使用

命令：SELECT sum(grade),gender FROM student2 GROUP BY gender HAVING SUM(grade) < 300;

### 使用 LIMIT 限制查询结果的数量

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

语法：SELECT 字段名 2, 字段名 2, ... FROM 表名 LIMIT [ OFFSET, ] 记录数

(从 0 开始的 4 条)命令：SELECT FROM student LIMIT 4;

(从第五条开始的 4 条)命令：SELECT FROM student2 ORDER BY grade DESC LIMIT 4,4;

### 为表和字段取别名

语法：SELECT FROM 表名 [ AS ] 别名;

命令：SELECT FROM student2 AS s WHERE s.gender='女';

### 为字段取别名

语法：SELECT 字段名 [ AS ] 别名 [ , 字段名 [ AS ] 别名, ... ] FROM 表名 ;

命令：SELECT name AS stu\_name, gender AS stu\_gender FROM student2;

## MySQL 的建表语句

#创建表，例子

#所谓的建表就是声明列的过程,所以要首先分析列

```
create table member(  
id int unsigned auto_increment primary key,  
username varchar(20) not null default "",  
gender char(1) not null default "",  
weight tinyint unsigned not null default 0,  
birth date not null default '0000-00-00',  
salary decimal(8,2) not null default 0.00,  
lastlogin int unsigned not null default 0  
)engine myisam charset utf8;
```

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

## MySQL 如何创建索引

1、ALTER TABLE

2、ALTER TABLE 用来创建普通索引、UNIQUE 索引或 PRIMARY KEY 索引。

3、ALTER TABLE table\_name ADD INDEX index\_name (column\_list)

4、ALTER TABLE table\_name ADD UNIQUE (column\_list)

5、ALTER TABLE table\_name ADD PRIMARY KEY (column\_list)

6、CREATE INDEX

7、CREATE INDEX 可对表增加普通索引或 UNIQUE 索引。

8、CREATE INDEX index\_name ON table\_name (column\_list)

9、CREATE UNIQUE INDEX index\_name ON table\_name (column\_list)

## 简述 SQL 注入原理，以及在代码层面房子 sql 注入

通俗点讲，SQL 注入的根本原因是：“用户输入数据”意外变成了代码被执行。

“用户输入数据”我这里可以指 Web 前端\$\_POST,\$\_GET 获取的数据，也可以指从数据库获取的数据，当然也不排除程序猿无意中使用的特殊字符串。

在 SQL 语句的拼接中，一些含特殊字符的变量在拼接时破坏了 SQL 语句的结构，导致“用户输入数据”意外变成了代码被执行。

规避方法：

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

1、 理语句法（解析协议层面上完全规避 SQL 注入）

2、 字符串转义（不要在 sql 中拼接字符）

# query 作为 sql 模板，args 为将要传入的参数  
execute(query, args=None)

**使用 python 将数据库的 student 表中的数据写入 db.txt**

```
import pymysql
connect=pymysql.Connect(
host="",
port=,
user="",
passwd="",
db="",
charset="",
)

cursor=connect.cursor()
sql='select from student'
cursor.execute(sql)
students=cursor.fetchall()

with open('db.txt','w') as f:
for student in students:
f.write(student)

cursor.close()
```

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

```
connect.close()
```

## 简述 left join 和 right join 的区别

- 1、左连接和右连接很相似，只是左右表位置的不同罢了。
- 2、left join(左连接) 返回包括左表中的所有记录和右表中连接字段相等的记录
- 3、right join(右连接) 返回包括右表中的所有记录和左表中连接字段相等的记录

## 索引有什么作用，有哪些分类，有什么好处和坏处？

作用：为了增加查询速度，提高系统性能

分类：

- 1、唯一索引：不允许其中任何两行具有相同索引值的索引。
- 2、非唯一索引：允许其中任何两行具有相同索引值的索引。
- 3、主键索引：有一列或列组合，其值唯一标识表中的每一行。
- 4、聚集索引：表中行的物理顺序与键值的逻辑（索引）顺序相同。一个表只能包含一个聚集索引。

好处：

- 1、帮助用户提高查询速度

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

2、 利用索引的唯一性来控制记录的唯一性

3、 可以加速表与表之间的连接

4、 降低查询中分组和排序的时间

坏处：

1、 存储索引占用磁盘空间

2、 执行数据修改操作(INSERT、UPDATE、DELETE)产生索引维护

## MySQL 慢日志

MySQL 的慢查询日志是 MySQL 提供了一种日志记录，它用来记录在 MySQL 中响应时间超过阈值的语句，具体指运行时间超过 long\_query\_time 值的 SQL，则会被记录到慢查询日志中。

## Redis 和 Memcached 的区别

1、 存储方式：

Memecache 把数据全部存在内存之中，断电后会挂掉，数据不能超过内存大小。Redis 有部份存在硬盘上，这样能保证数据的持久性。

2、 数据支持类型

Memcache 对数据类型支持相对简单。Redis 有复杂的数据类型。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

### 3、使用底层模型不同

它们之间底层实现方式 以及与客户端之间通信的应用协议不一样。Redis 直接自己构建了 VM 机制，因为一般的系统调用系统函数的话，会浪费一定的时间去移动和请求。

### 4、value 大小

Redis 最大可以达到 1GB，而 memcache 只有 1MB

## 如何高效的找到 Redis 中所有以 felix 开头的 key

1、`scan 0 match felixcount 5`

2、表示从游标 0 开始查询 felix 开头的 key，每次返回 5 条，但是这个 5 条不一定

## 什么是一致性哈希

### [参考链接](#)

一致性哈希简称 DHT,是麻省理工学院提出的一种算法，目前主要应用于分布式缓存当中。一致性哈希可以有效地解决分布式存储结构下动态增加和删除节点所带来的问题。

## Redis 是单进程单线程的吗?

Redis 采用的是基于内存的采用的是单进程单线程模型的 KV 数据库

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

## Redis 默认多少个 db

默认有 16 个数据库

## 如果 Redis 中的某个列表中的数据量非常大，如何实现循环显示每一个值？

使用生成器一个一个取

## Redis 如何实现主从复制？以及数据同步机制？

Redis 主从结构可以采用一主多从或者级联结构，主从复制可以根据是否全量分为全量同步和增量同步。

**全量同步：**

Redis 全量复制一般发生在 slave 初始化阶段，这时 slave 需要将 master 上的所有数据都复制一份步骤如下：

- 1、从服务器连接主服务器，发送 SYNC 命令；
- 2、主服务器接收到 SYNC 命令后，开始执行 BGSAVE 命令生成 RDB 文件并使用缓冲区记录此后执行的所有写命令；
- 3、主服务器 BGSAVE 执行完后，向所有从服务器发送快照文件，并在发送期间继续记录被执行的写命令；

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题





微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

- 4、从服务器收到快照文件后丢弃所有旧数据，载入收到的快照；
- 5、主服务器快照发送完毕后开始向从服务器发送缓冲区中的写命令；
- 6、从服务器完成对快照的载入，开始接收命令请求，并执行来自主服务器缓冲区的写命令；

增量同步：

- 1、Redis 增量复制是指 Slave 初始化后开始正常工作时主服务器发生的写操作同步到从服务器的过程。增量复制的过程主要是主服务器每执行一个写命令就会向从服务器发送相同的写命令，从服务器接收并执行收到的写命令。
- 2、Redis 主从同步策略
- 3、主从刚刚连接的时候，进行全量同步；全同步结束后，进行增量同步。当然，如果有需要，slave 在任何时候都可以发起全量同步。Redis 策略是，无论如何，首先会尝试进行增量同步，如不成功，要求从机进行全量同步

## Redis 中 sentinel 的作用

Redis Sentinel 是一个用来监控 Redis 集群中节点的状态，不用来存储数据。当集群中的某个节点有故障时，可以自动的进行故障转移的操作。通常为了保证 sentinel 的高可用，sentinel 也会部署多个

## 如何实现 Redis 集群

- 1、Twitter 开发的 twemproxy

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

2、 豌豆荚开发的 codis

3、 Redis 官方的 Redis-cluster

## Redis 中默认有多少个哈希槽

1、  $2^{14}$  个

2、 Redis 集群没有使用一致性 hash，而是引入了哈希槽的概念。

## 简述 Redis 的有几种持久化策略以及比较？

1、 RDB 持久化可以在指定的时间间隔内生成数据集的时间点快照。

2、 AOF 持久化记录服务器执行的所有写操作命令，并在服务器启动时，通过重新执行这些命令来还原数据集。AOF 文件中的命令全部以 Redis 协议的格式来保存，新命令会被追加到文件的末尾。Redis 还可以在后台对 AOF 文件进行重写(rewrite)，使得 AOF 文件的体积不会超出保存数据集状态所需的实际大小。

区别：

1、 RDB 持久化是指在指定的时间间隔内将内存中的数据集快照写入磁盘，实际操作过程是 fork 一个子进程，先将数据集写入临时文件，写入成功后，再替换之前的文件，用二进制压缩存储。

2、 AOF 持久化以日志的形式记录服务器所处理的每一个写、删除操作，查询操作不会记录，以文本的方式记录，可以打开文件看到详细的操作记录。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

## 列举 Redis 支持的过期策略

### 定时删除

在设置 key 的过期时间的同时，为该 key 创建一个定时器，让定时器在 key 的过期时间来临时，对 key 进行删除

### 惰性删除

key 过期的时候不删除，每次从数据库获取 key 的时候去检查是否过期，若过期，则删除，返回 null。

### 定期删除

每隔一段时间执行一次删除(在 Redis.conf 配置文件设置 hz, 1s 刷新的频率)过期 key 操作

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题