

1、ES 和 mongoDB 的区别

mongodb 和 ES 都属于是 NoSql 类型的数据存储媒介，两者相似度高，但是使用方式和场景上有区别

mongodb 的目标是干掉关系型数据库和 RDBMS 是竞争关系

ES 起初就是以检索查询为主要应用场景和 RDBMS 相互协助的关系

一、相同点：

数据存储格式位 json

聚合全文索引

CRUD

分片和复制

简单的 join 操作

适用于大数据量的处理

不支持事物

二、不同点：

开发语言：ES 是 java；mongodb 是 C++

分片方式：ES 是 hash；mongodb 是 range 和 hash

分布式：ES 是主副分片自动组合和配置；mongodb 是需要手动配置集群“路由+服务配置+sharding”

索引：ES 自建倒序索引，检索力度强；mongodb 手动创建索引(B 树)不支持倒序索引，和 RDBMS 类似

检索字段：ES 全文索引，可用的检索插件较多；mongodb 对索引字段个数有限制，全文索引效率低，所以不采用

时效性：ES 非实时，有数据丢失风险；mongodb 实时，理论上无丢失数据风险

总体来说：ES 偏向于检索、查询、和数据分析适合用于 OLAP(on line analytical processing)系统；mongodb 偏向于大数据下的 CRUD 适用于 OLTP 系统(on line transaction processing)

2、mysql 多表查询

连接查询：等值连接查询、非等值连接查询、内连接查询

外连接查询：左外连接、右连接、全外连接

自然连接查询：全自然连接、左自然连接、右自然连接

自身连接：自身连接查询

嵌套查询：子查询

3、mysql 单表查询

在一张表里面通过条件查询数据。

4、怎么使用锁将表锁住

LOCK TABLES tbl_name READ

5、查询数据库的最大字节是多少

mysql varchar 类型最大存储 65533 字节

6、会写 shell 脚本吗

shell 脚本的本质就是 linux 系统命令的集合
语法也挺简单，可以学一下，1 小时上手那种。

7、用过 awk 命令吗

linux 三大文本操作神器,gerp、sed、awk
gerp、sed 作为测试一定要会
awk 运维用得最多，测试会用是加分项

8、Linux 命令说几个

gerp sed tail tee top free-m find stat vim zip tar
chmod chown scp ps

9、会 httprunner 吗

HttpRunner 是一款面向 HTTP(S) 协议的通用测试框架，会用就说会用，不会用就回答替代品，pytest、unittest

10、会 httpclients 吗

HttpClient 是 Apache Jakarta Common 下的子项目，用来提供高效的、最新的、功能丰富的支持 HTTP 协议的客户端编程工具包。
属于 java 开发的一个工具包，用 java 做自动化肯定是会的，用 python 做自动化可以回答不会。

11、压力测试系统响应时间多少

根据 2-5-8 原则去回答即可

2 秒以内得到响应时，会感觉系统的响应很快

2-5 秒之间得到响应时，会感觉系统的响应速度还可以

5-8 秒以内得到响应时，会感觉系统的响应速度很慢，但是还可以接受
超过 8 秒后仍然无法得到响应时，会感觉系统糟透了

12、怎么使用代码实现今天是今年中的第几天

思路：

1、先区分平年和闰年， 闰年： $year \% 4 == 0 \text{ and } year \% 100 != 0 \text{ or } year \% 400 == 0$

2、先将平年和闰年每个月是当前年份的第几天，分别做成 list

3、根据输入的年份，判断等到是平年还是闰年，

4、根据上一步结果通过 2 个 list 得到输入月份的前一个月份是当前年份的多少天

5、根据如数的日期，得到当前月份过了多少天

6、输入日期是当前年份的天数 = 上一个月在当前年的天数 + 本月日期

```

7
8 def countDay():
9     year = int(input("year: "))
10    month = int(input("month: "))
11    day = int(input("day: "))
12    the_month1 = [0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334] #平年
13    the_month2 = [0, 31, 60, 91, 121, 152, 182, 213, 244, 274, 305, 335] #闰年
14    if ((year % 4 == 0) and (year % 100 != 0)) or (year % 400 == 0):
15        sumday = the_month2[month-1] + day
16    else:
17        sumday = the_month1[month-1] + day
18    print("%d.%d.%d是%d年的第 %d 天" % (year, month, day, year, sumday))
19
20 countDay()
21

```

13、前端语言，css、js、vue

前端知识回答了解，能满足 UI 自动化测试需求即可。重点体现学习能力，前端知识不够用会主动去学习。

14、测试方案里面包含什么

测试人员、资源(硬件、软件环境)、测试目标、测试范围、测试完成标准

15、API 测试怎么做

API 测试就是接口测试

参数校验、幂等性、接口性能(响应时间、并发数)、安全性(参数传递安全性、接口鉴权、数据安全性(数据隔离，数据权限))

16、postman 怎么做接口关联的测试

使用 js 提取保存到环境变量里面，其他接口可以直接使用环境变量的值即可。

17、HTTP 和 HTTPS 有什么不同

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，http 使用 80 端口，https 是 443。

4、http 是简单的无状态的连接；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

18、元素的定位方式

八大元素定位方法：

find_element_by_name 通过元素 name 属性查找

find_element_by_partial_link_text 模糊匹配超链接文本(超链接文本必须是唯一)

find_element_by_id 通过元素 id 查找

find_element_by_tag_name tag_name(标签名称)定位

find_element_by_xpath 通过路径查找

`find_element_by_class_name` class 属性的值定位
`find_element_by_link_text` 精确匹配超链接文本(超链接文本必须是唯一)

`find_element_by_css_selector` 通过 css 选择器定位

19、简述下测试流程

根据自己公司的测试流程回答。

完整测试流程: 需求评审->测试任务分配->测试用例->冒烟测试->冒烟通过后正式测试(2 轮)-> 回归测试(主流程或者全量回归, 根据公司要求) ->预发布环境测试 -> 写测试报告 -> 找产品经理验收测试->上线验证测试

20、接口关联用代码怎么实现

通过面向对象思维做参数化。

21、如果和开发意见有分歧怎么办

找对应产品经理确认分歧, 直到问题解决

22、如果负责的系统和其他测试负责的系统测试范围有歧义怎么办?

牵头开会讨论(拉上双方的测试领导和当事人), 确认各自的测试范围, 避免漏测。

23、用 classname 来定位的情况多吗

classname 定位的场景不多, classname 不唯一。

24、用相对定位还是绝对定位

都可以, 首选相对定位, 因为绝对定位稳定性不如相对定位, 前端随便加一个元素绝对路径必然改变, 而相对路径不一定会改变。

25、自动化测试脚本修改频率高吗, 每次版本都需要去修改自动化脚本?

自动化脚本目的是用来做主流程的回归测试, 或者是前置测试参数的构造, 所以只要主流程稳定, 自动化脚本就稳定。

26、手工和自动化占比

首先: 自动化测试不能替代手工测试

我们公司的情况:

新功能都是手工测试

老功能都是自动化测试(这里是指核心流程, 非核心业务也没人动)

27、有没有出现比较严重的 bug

没有严重 bug, 有的话你也不能说。

28、安全测试会吗

如实说就好, 但是基础的安全项要知道, 比如: 越权、CSRF 攻击、跨域等等, 主要是考你是否有安全意识, 安全测试涉及知识面广而深, 真正能玩得转安全测试的凤毛麟角, 属于加分项。

29、怎么样在不同的测试环境进行高效测试（手工和自动化在同一个环境，不同环境指 UAT、SIT 环境等）

UAT: (User Acceptance Testing) 用户验收测试

SIT: (System Integration Testing) 系统集成测试

自动化测试的时候，使用特定标签的账号，在脚本前置条件处理的问题，保证自动化测试完成后将每次测试产生的数据清理干净即可。

手工测试的时候 一样使用特定标签的账号，测试完成后使用脚本清理测试产生的数据。

30、断言做在什么地方，怎么做断言

测试用例执行完成之后进行断言

怎么做断言分情况：

接口自动化：根据接口返回，与预期结果进行对比（预期结果事先计算好的，或者是直接去数据库查询出来，或者其他方式，总之要保证数据准确性没有问题）

UI 自动化：根据 UI 页面元素返回进行断言，元素是否存在，或者值是否正确

31、项目相关，问的比较细

项目相关的问题，总体思路：

1、注意逻辑性

2、多用专业术语（包括行业专业术语）

3、只讲业务大的流程，不要讲细节（如果你真的了解的很细，知道代码的内部逻辑实现，这种情况可以讲细节，这是加分项）

4、问到不懂的直接说不是自己负责不知道细节

32、多线程了解吗？

多线程又称轻量级进程。一个进程开始便会创建一个线程，称为主线程。一个进程可以创建多个线程，多线程即是同一进程下的不同执行路径，同一进程下的线程共享该进程的数据区。线程以并发的方式执行，线程执行时可以被中断和挂起。（在多核 cpu 中，多线程才可能并行执行）

33、算法题：给一组数据，找出两个数字，得到最大的乘积

思路：

结论：最大的两个数乘积是最大的

1、先排序

2、找到两个最大的数，return 他们的乘积

```
21
22 ▶ def test_demo(list_no):
23     sort_resule=sorted(list_no)
24     return sort_resule[-1]*sort_resule[-2]
25
```

34、Static 的作用

java 里的关键字，static 是一个修饰符，用于修饰类的成员方法、类的成员变量，另外可以编写 static 代码块来优化程序性能。

static 修饰的方法一般称作静态方法

static 修饰的变量也称为静态变量

static 用来形成静态代码块以优化程序性能

35、Try catch 的运行顺序

先运行 try 里面的代码块，如果报错了才会运行 catch 里面的代码块

36、Xpath 轴是什么

轴是相对于当前节点的节点集

37、MongoDB 的模糊查询

`db.user.find({"字段名称":/查询内容/})`

38、MongoDB 怎么查询大于、小于

`db.user.find({"likes":{"$gt":50}}).pretty()` 大于 50

`db.user.find({"likes":{"$lt":50}}).pretty()` 小于 50

39、Mysql 的多表查询有几种连接方式，区别是什么

连接查询：等值连接查询、非等值连接查询、内连接查询

外连接查询：左外连接、右连接、全外连接

自然连接查询：全自然连接、左自然连接、右自然连接

自身连接：自身连接查询

嵌套查询：子查询

区别：

查询结果不一样，时候用场景不一样。

连接查询：不设置查询字段，得到的是多张表的全部数据

外连接查询：查询得到的是以基准表为基础的两张表的数据。

自然连接：去除两个表中重复的列名，左自然连接以左表为基准匹配，右自然连接以右表为基准匹配

自身连接：自身连接就是一张表，通过起别名的方式自己和自己连接

嵌套查询：子查询的结果是为父查询提供数据的查询

40、Web 的定位方式有几种

答案同 18 题

41、元素等待方式有几种，分别有什么不同

`time.sleep(10)` 强制等待 10 秒

`driver.implicitly_wait(10)` 隐式等待 10 秒

`WebDriverWait(driver=self.driver,`

`timeout=10,poll_frequency=1).until(method=lambda`

`x :self.find_elements(drivers=x),message='执行失败')` 显式等待

强制等待：等待时间固定不变，时间解释后执行下一个语句。

隐式等待：隐式等待会在 WebDriver 对象实例的整个生命周期起作用，它不针对某一个元素，是全局元素等待，即在定位元素时，需要等待页面全部元素加载完成，才会执行下一个语句。如果超出了设置时间的则抛出异常

显式等待(很多用法常用的是与 util 结合使用)：timeou 时间内会不断调用 method 查询元素，调用间隔通过 poll_frequency=0.5 设置，等待的元素出现执行下一步，超时抛出异常。

42、设计 case：将物品添加购物车，商场现在有两种促销活动

一、满减活动，限两次

二、买多件送一件，限两次

两个活动可以同时使用，请设计相应的 case

使用等价类、边界值两种测试用例设计方法即可，这个太简单了现场发挥吧。

43、如果一个系统调用另一个系统，出现异常了（网络异常），通过手工测试测不出来，这时候要怎么测

通过脚本，使用 MOCK 模拟故障系统各种类型的返回来测试本系统。

44、需求分析时，有进行非功能性分析吗

有，需求分析时会做易用性分析，目标用户习惯分析，视觉效果分析，竞品分析等等

45、ABtest 时，怎么去实现？比如设置走 A 的概率 60%，走 B 的概率 40%，怎么去开发

ABTest：简单来说，就是为同一个产品目标制定两个方案，让一部分用户使用 A 方案，另一部分用户使用 B 方案。

实现思路：

abtest 对于业务开发来说，最好是独立的，也就是我们需要在用户无感知切入 abtest，同时注意保持业务一致性，通过 nginx+lua 方案，通过在 nginx 中执行嵌入的 lua 脚本，动态计算 upstream，将不同标签的用户导向不同的程序版本，达到 abtest 的目的。

46、怎么定位性能瓶颈，比如线程组是 10 的时候，TPS 是 60，线程组是 20 的时候，TPS 还是 60，怎么去定位问题

10 个线程的时候事物处理能力是 60，20 个线程的时候事务处理能力是 60，根据当前数据暂时无法判断是否存在性能问题，因为并不能证明 60 就是该服务器的最大事物处理能力，还需要进行其他的测试来确定服务器最大事物处理能力峰值。

建议增加测试场景如下：

场景一、超过 20 个线程场景，记录 TPS 值

场景二、10-20 之间线程场景下，记录 TPS 值

当某个场景得到的结果 TPS 到最大值后，随着线程数增加 TPS 值变小，这就说明这个最大是就是该机器的最大 TPS。最终通过其他指标去寻找系统瓶颈即可。

47、Spring MVC 会吗

并不是 java 开发，未使用过，但是大概知道原理 django flask 一致。

MVC 是 Model、View 和 Controller 的缩写，分别代表 Web 应用程序中的 3 种职责

模型：用于存储数据以及处理用户请求的业务逻辑。

视图：向控制器提交数据，显示模型中的数据。

控制器：根据视图提出的请求判断将请求和数据交给哪个模型处理，将处理后的有关结果交给哪个视图更新显示

48、怎么去做兼容性测试

1、web 端做浏览器内核兼容即可

Trident 内核（代表：Internet Explorer）

Gecko 内核（代表：Mozilla Firefox）

WebKit 内核（代表：Safari、Chrome）

Presto 内核（代表：Opera）

2、移动端

android 系统版本

屏幕分辨率

主流手机厂商（苹果、华为、小米、三星、中兴、oppo、vivo）

49、三方接口，第三方出现问题了，你们怎么处理

第三方接口出了问题，首先排查出现问题的原因，是我方导致还是第三方导致。根据排查结果做对接流程和对接测试流程优化。

如果是第三方修改了接口上线了，我方不知道，导致的 bug，这个时候需要变更对接流程，约定版本上线以及变更能及时通知我方，进行联调测试。

如果是我方的问题，优化测试流程，增加测试范围和交叉测试环节，防止漏测。

50、带 cookie 的接口，自动化脚本怎么写

将 cookie 获取作为前置条件即可。

51、从数据库中查出的字段是个字符串，怎么用 sql 将格式修改成 int

```
select CAST(host as SIGNED INTEGER) from t_api;
```

52、怎么样用 sql 将数据库的任意一个字段设置成唯一自增的

```
alter table test_table add id int auto_increment primary key
```

53、在生产中，当 mongoDB 数据库挂了，怎么办

mongoDB 主机挂了会自动切换到从机，程序会尝试重启主机，并且会有生产告警，开发会根据日志定位问题。

54、VUE 会吗

VUE 是一套用于构建用户界面的渐进式框架。

前端必学，测试属于加分项，一般答了解即可。

如果更多软件测试资源
请关注柠檬班软件测试公众号！

