# Java开发面试题

## 1）areas.txt保存了以下数据，行内数据为：地名, 行政级别,上级地名（请留意东莞无区县这一级）

长潭社区居委会, 村,五经富镇,

五经富镇, 镇街,揭西县

揭西县, 区县,揭阳市,

揭阳市,市,广东省

广东省,省

林村居委会, 村,塘厦镇,

塘厦镇, 镇街,东莞市

东莞市,市,广东省

福中居委会, 村,莲花街道

莲花街道, 镇街,福田区

福田区,区县,深圳市

深圳市,市,广东省

## 请写一份代码，实现传入地名，按以下格式输出它本身和所有上级，如传入“福中居委会”，输出：

福中居委会

|-莲花街道

|-福田区

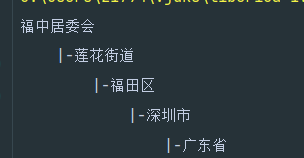
|-深圳市

|-广东省

### **代码**



## 效果



**2）有成绩表数据如下，缺少数据的为缺考：**

+------+---------+-------+

| 姓名 | 科目  | 成绩   |

+------+---------+-------+

| 张三   | 数学       | 90       |

| 张三   | 语文       | 50       |

| 张三   | 地理       | 40       |

| 张三   | 政治       | 65       |

| 李四   | 语文       | 55       |

| 李四   | 数学       | 46       |

| 李四   | 政治       | 55       |

| 王五   | 语文       | 45       |

| 王五   | 政治       | 30       |

+------+---------+-------+

请用mysql语句

**A.查询出各科参加考试人员的平均成绩和及格率**

SELECT

科目,

AVG(成绩) AS 平均成绩,

SUM(CASE WHEN 成绩 >= 60 THEN 1 ELSE 0 END) / COUNT(成绩) \* 100 AS 及格率

FROM 成绩表

WHERE 成绩 IS NOT NULL

GROUP BY 科目;

1. **查询出各科缺考人员名单**

SELECT

DISTINCT c.姓名,

s.科目

FROM (SELECT DISTINCT 姓名 FROM 成绩表) c

CROSS JOIN

(SELECT DISTINCT 科目 FROM 成绩表) s

LEFT JOIN

成绩表 t ON c.姓名 = t.姓名 AND s.科目 = t.科目

WHERE

t.成绩 IS NULL;

**C.查询出每一科都不及格的人员的名单及平均分**

SELECT

姓名,

AVG(成绩) AS 平均分

FROM

成绩表 t1

WHERE

NOT EXISTS (

SELECT 1

FROM 成绩表 t2

WHERE t1.姓名 = t2.姓名 AND t2.成绩 >= 60

)

GROUP BY

姓名

HAVING

COUNT(成绩) > 0; -- 确保学生至少参加了一门考试

**3）有如下表及数据**

user\_list表

+-----------------------------+------------+--------------+

| id | user\_name | birthday |

+-----------------------------+------------+--------------+

| ubcda | 张三 | 1980-11-30 |

| ubcdb | 李四 | 1971-03-10 |

| ubcdc | 王五 | 1983-06-20 |

| ubcdd | 刘二 | 1951-02-22 |

| ubcde | 李七 | 1974-01-30 |

+-----------------------------+------------+--------------+

doctor\_list表

+-----------------------------+------------+--------------+

| id | doct\_name

+-----------------------------+------------+--------------+

| dbcda | 医生三

| dbcdb | 医生四

| dbcdc | 医生五

| dbcdd | 医生二

| dbcde | 医生七

+-----------------------------+------------+--------------+

hm\_body\_check表

+-----------------------------+------------+--------------+

| id | user\_id | check\_date | doctorId

+-----------------------------+------------+--------------+

| c123x | ubcda | 2017-01-03 | dbcda

| c124x | ubcda | 2018-04-10 | dbcdb

| c125x | ubcdb | 2017-01-03 | dbcdc

| c126x | ubcdb | 2018-05-12 | dbcdd

| c127x | ubcdc | 2017-01-03 | dbcde

| c128x | ubcdc | 2018-02-08 | dbcdc

| c129x | ubcdc | 2019-02-05 | dbcda

| c123y | ubcdd | 2017-02-03 | dbcdb

| c124y | ubcdd | 2018-03-02 | dbcda

| c125y | ubcdd | 2019-01-09 | dbcdd

| c126y | ubcde | 2017-11-04 | dbcdb

| c127y | ubcde | 2018-10-03 | dbcdc

+-----------------------------+------------+--------------+

**要求：写一条mysql语句查出hm\_body\_check中各人的体检次数，最后体检日期及最后体检医生**

+-----------------------------+------------+------------------------+

| 用户id | 姓名 | 体检次数 | 最后体检日期 | 最后体检医生

+-----------------------------+------------+------------------------+

| ubcda | 张三 | 2 | 2018-04-10 | 医生四

| ubcdb | 李四 | 2 | 2018-05-12 | 医生二

| ubcdc | 王五 | 3 | 2019-02-05 | 医生三

| ubcdd | 刘二 | 3 | 2019-01-09 | 医生二

| ubcde | 李七 | 2 | 2018-10-03 | 医生五

+-----------------------------+------------+--------------+

SQL语句

SELECT

ul.user\_name AS 用户名,

COUNT(hbc.id) AS 体检次数,

MAX(hbc.check\_date) AS 最后体检日期,

dl.doct\_name AS 最后体检医生

FROM

hm\_body\_check hbc

JOIN

user\_list ul ON hbc.user\_id = ul.id

JOIN

doctor\_list dl ON hbc.doctorId = dl.id

GROUP BY

hbc.user\_id, ul.user\_name;

**4）现有数据库中1000万条从第三方获取数据并保存到title, content字段的任务，假设所要执行的任务为从百度抓取含指定关键词(key字段)的第一条搜索结果的页面内容，执行任务的过程中可能会出错（status为异常），你的代码将部署在3台64G内存的服务器上同时运行。**

**请**

***a描述此需求实现的难点、潜在的问题及你解决此需求的思路；***

我认为有可能存在以下四个难点

性能问题：同时处理1000万条数据，并且可能涉及到网络请求，需要高效的并发处理和请求管理。

错误处理：网络请求、数据处理都可能出错，需要设计完善的错误处理和重试机制。

数据存储：大批量数据写入数据库，需要确保数据库的稳定性和性能。

负载均衡：3台服务器需要合理分配任务，避免某些服务器过载。

潜在的问题

网络不稳定：可能导致请求失败或超时。

数据库瓶颈：大量并发写入可能导致数据库性能下降或锁表。

资源竞争：服务器资源（CPU、内存、网络带宽）可能成为瓶颈。

解决方法

使用消息队列：如RabbitMQ或Kafka，将任务放入队列，由消费者并发处理。

分布式任务调度：如使用Celery或XDL等分布式任务框架，实现任务的分配和调度。

重试机制：对于失败的任务，设置合理的重试策略和重试间隔。

错误日志记录：详细记录每个任务的执行情况，方便排查问题。

数据库优化：使用批量插入、索引优化、读写分离等技术提高数据库性能。

负载均衡：使用Nginx等负载均衡器分配网络请求，或使用Docker Swarm/Kubernetes等容器编排工具实现服务器资源的动态分配。

***b简单说明你的代码结构，包含哪些类，哪些方法，实现什么功能，实现过程要点注释。***

类

TaskQueue: 负责与消息队列交互，提供任务的入队和出队操作。

TaskConsumer: 消费者类，负责从队列中获取任务并执行。

TaskExecutor: 任务执行类，负责从百度抓取搜索结果并保存到数据库。

TaskDatabase: 数据库操作类，提供任务的存储和查询功能。

方法

TaskQueue.enqueue(task): 将任务加入队列。

TaskQueue.dequeue(): 从队列中获取任务。

TaskConsumer.start(): 启动消费者进程，循环从队列中获取任务并执行。TaskExecutor.fetch\_from\_baidu(keyword): 使用指定的关键词从百度抓取搜索结果。

TaskExecutor.save\_to\_database(task): 将抓取到的结果保存到数据库。TaskDatabase.save\_task(task): 将任务保存到数据库。

TaskDatabase.update\_task\_status(task\_id, status): 更新任务的状态。

实现过程要点注释：

在TaskConsumer.start()中，使用循环不断从队列中获取任务，并调用TaskExecutor执行。

在TaskExecutor.fetch\_from\_baidu()中，实现网络请求和结果解析的逻辑。

在TaskExecutor.save\_to\_database()中，实现数据持久化的逻辑，包括错误处理和重试机制。

在TaskDatabase类中，使用ORM框架（如SQLAlchemy）简化数据库操作。

***c预计写完代码所需要的时间；***

1-2周

***d预计程序执行完成这1000万条任务(要求状态全部为已完成)，所需要的时间及依据。***

**task表**

+-----------------------------+------------+----------------------------------+

| id | key | title | content | status | completed\_on

+-----------------------------+------------+----------------------------------+

| 1 | 中国 | NULL | NULL | 未完成 | NULL

| 2 | 新冠 | xxxxx | yyyyyyy | 已完成 | 2021-02-11 10:30

| 3 | 春节 | NULL | NULL | 进行中 | 2021-02-11 10:30

| 4 | 宁夏 | xxxxx | yyyyyyy | 异常 | NULL

…..

| 10000000 | 宁夏 | xxxxx | yyyyyyy | 异常 | NULL

**5）现有一restful接口项目，使用 Spring Mvc实现，为提高开发效率，现需实现根据接口本身代码自动生成调用接口的代码的功能，请阐述一下你实现此功能的思路。（源代码及生成代码如下图，所写代码为kotlin，可当成java 来看，不影响阐述思路）**



### **1. 分析接口定义**

**解析Controller层**：首先，需要扫描项目中的Spring MVC Controller类，特别是带有@RequestMapping、@GetMapping、@PostMapping等注解的方法。

**提取接口信息**：从注解中提取出接口的URL路径、请求方法（GET、POST等）、请求参数（包括路径参数、查询参数、请求体等）以及响应类型等信息。

### **2. 编写代码生成器**

**确定目标语言**：确定需要生成的客户端代码的目标语言，比如Java、Kotlin、TypeScript等。

**设计模板**：为每个目标语言设计代码模板，模板中应包含占位符，用于在生成代码时填充实际的接口信息。

**实现代码生成逻辑**：编写代码以读取接口定义信息，并根据这些信息填充代码模板，生成对应的客户端代码。

**3. 集成到开发流程**

**构建工具集成**：将代码生成器集成到项目的构建工具（如Maven、Gradle）中，以便在构建项目时自动触发代码生成。

**IDE插件**：开发一个IDE（如IntelliJ IDEA、Eclipse）插件，以便开发者可以更方便地在IDE中触发代码生成操作。

### **4. 注意事项**

**错误处理**：在解析接口定义和生成代码时，需要妥善处理各种可能出现的错误和异常情况。

**可配置性**：提供足够的配置选项，允许开发者自定义生成的代码（如输出目录、包名、类名等）。

**文档支持**：生成的代码应该包含必要的注释和文档，以帮助开发者理解如何使用这些代码。

### **5. 技术实现细节**

**注解处理**：使用Java的反射机制或注解处理API来扫描Controller类并提取注解信息。

**模板引擎**：使用模板引擎（如Velocity、FreeMarker、Thymeleaf等）来根据模板生成代码。

**文件操作**：使用Java的文件IO操作来创建和写入生成的代码文件。

### **6. 示例流程**

开发者编写或修改Spring MVC Controller类，定义RESTful接口。

开发者触发代码生成操作（通过构建工具、IDE插件等方式）。

代码生成器扫描Controller类，提取接口定义信息。

代码生成器根据提取的信息和预设的模板生成客户端代码。

生成的代码被写入到指定的输出目录。

开发者在项目中引入生成的客户端代码，并使用这些代码来调用RESTful接口。

### **7. 扩展功能**

**支持多种客户端类型**：除了基本的HTTP客户端外，还可以考虑支持其他类型的客户端，如WebSocket客户端、GraphQL客户端等。

**代码优化和美化**：对生成的代码进行优化和美化，以提高代码的可读性和可维护性。

**集成测试**：为生成的客户端代码编写集成测试，以确保它们能够正确地调用RESTful接口并处理响应。

1. 请阅读以下代码，说明此代码的作用，并描述代码可能存在哪些问题。

public static Properties getProperties(String fileName) {

Properties result = new Properties();

try {

result.load(new InputStreamReader(PropertiesUtil.class.getResourceAsStream(fileName), "UTF-8"));

} catch (Exception e) {

e.printStackTrace();

}

return result;

}

1. 异常处理如果文件加载失败，方法仍然返回一个空的Properties对象，调用者可能无法意识到文件没有正确加载。
2. 返回值不明确当加载失败时，方法返回了一个空的Properties对象。改进可以返回null或者抛出异常
3. 硬编码的字符集：改进方式允许调用者更改字符集
4. 测试: 这个代码没有包含任何测试，这个增加了生产环境出现问题的风险
5. 现有基于spring mvc的应用，提供了将数据库中的内容导出为标准格式的excel文件，供用户检查核对的功能，该应用还提供了restful接口供安卓终端访问。现发现当多人同时使用导出excel文件的功能时，服务器负载会瞬时升高，到一定程度时restful接口的正常访问受到影响，出现超时。请问面对此问题，你会采取何种措施优化应用代码，使得导出excel文件和restful接口均能正常提供服务。

可以通过异步处理excel文件的方式，**使用消息队列，**优化数据库查询的语句，限制并发导出请求来减轻服务器压力，提升速度