

元能星泰 研发中心

后端实习岗位笔试题

题目 1：考察点（sql、mybatis 的应用）

已知表结构如下:(下划线为主键)

数据库：mysql

student 学生信息表及其字段如下： sid, name, age

course 课程表及其字段如下： cid, courseName,

grade 成绩表及其字段如下: gid, grade, cid, sid

请问如何查询出 courseName 包含'信息'二字，且成绩 60 以上（包含）的，name 包含'张'姓，名字包含'涂'的学生信息及成绩(grade.grade)

请在以下 Mybatis 的 xml 文档中填充 sql 语句

注：为简化题目,请求参数与返回值均采用 map 的形式

```
<mapper>
  <!-- 可自行自定义 sql 片段等 -->

  <select id="selectStudentInfo" parameterType="hashmap"
resultType="hashmap">
    <!-- 请在此处编写相关 sql -->
  </select>
</mapper>
```

题目 2：考察点（表设计、范式、sql、数据结构算法逻辑、java 代码编写能力）

请使用 mysql 或 postgresql 作为数据库设计表结构，实现将下列非结构化数据进行结构化存储和 sql 查询及遍历复原返回给前端的过程。

需提供**(1)**表结构、实体等、**(2)**创建和获取整个节点树的核心逻辑代码、**(3)**实现查询的 sql 语句*

已知返回给前端的数据结构如下所示，请完成节点的创建（单节点添加、批量节点添加）与节点树的查询操作：

```
{
  "id": "1",
  "name": "节点1",
  "parentId": "-1",
  "children": [
    {
      "id": "2",
      "name": "节点2",
      "parentId": "1",
      "children": []
    },
    {
      "id": "3",
      "name": "节点3",
      "parentId": "1",
      "children": [
        {
          "id": "5",
          "name": "节点5",
          "parentId": "3",
          "children": []
        },
        {
          "id": "6",
          "name": "节点6",
          "parentId": "3",
          "children": []
        }
      ]
    }
  ],
  {
    "id": "4",
    "name": "节点4",
    "parentId": "1",
    "children": []
  }
}
```

题目 3：考察点（docker、nginx 相关运用）

请尝试编写一份 Dockerfile 及镜像制作 linux/unix 命令，将 jar 文件部署到 docker 容器中，并对外暴露 8989 端口，并通过 nginx 代理 8989 端口实现服务访问。

已知：主机文件目录结构如下：

```
目前所在目录为/root
--/root
--/app(目录)
--/app/app.jar
--/app/Dockerfile
```

Dockerfile:

#请填写 dockerfile 配置

执行与启动命令：

\$:

#请编辑 linux 命令

nginx.conf 中代理的 server 域中的核心代码配置：

#请填写 nginx 相关配置

题目 4：考察点（数据处理与算法能力、综合编程能力）

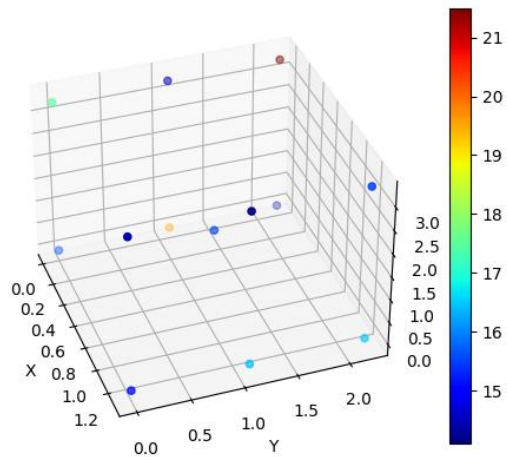
已知某物体燃烧工况 CSV 数据表中存在如下格式数据若干条记录：

```
Node,X,Y,Z,Temperature
1,1.3,2.2,3.3,15.5
2,1.3,2.2,0,16.5
3,1.3,0,0,15.1
4,1.3,0,3.3,14.4
5,0,0,3.3,17.5
6,0,2.2,3.3,21.5
7,0,2.2,0,14.5
8,0,0,0,15.6
9,0,1.1,3.3,14.5
10,0,1.1,0,19.5
```

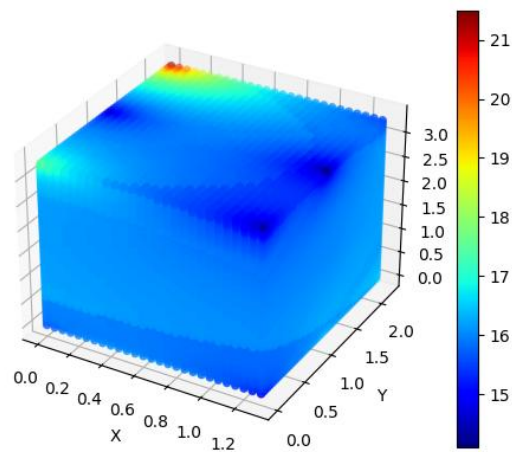
```
11,1.3,1.1,0,16.4
12,1.3,1.1,3.3,14.1
13,0.75,1.1,1.65,15.5
```

csv 中每行数据代表一个三维点位的温度值，整份数据为一个几何立方体，但由于点与点之间数据过于离散，请尝试将上述的数据进行插值运算，使得经过插值运算后的立方体中温度数据得以“相对”连续，效果可参照下图所示。

原始数据温度点位分布如下（过于离散）：



经过某插值运算后，其温度点位分布可能如下（经过插值后）：



请试着编程实现数据插值算法（不限于 python、go、java、c++等实现）