初始设置

创建一个名为 Data Processing.js 的 Javascript 文档。在该文档中创建一个名为Data P rocessing”的类。

在该类中编写一个名为 load CSV 的函数，该函数接受 csv 文档的文档名作为输入，例如

加载 CSV（“用户详细信息”）。结果数据应作为全局变量保存在类的本地

称为原始用户数据。编写一个名为format\_data的函数，该函数将没有变量作为参数。

格式化数据

编写一个format\_data函数，在函数格式数据内，应处理原始用户数据中存储的数据并将其输出到称为格式化用户数据的全局变量。数据最初以 CSV 格式提供，其中分隔符是“,”字符。数据的第一列是用户的标题和全名。第二第三列分别是用户的出生日期和年龄。最后，第四列是用户的电子邮件。确保数据集转换为表 1 中概述的适当格式。

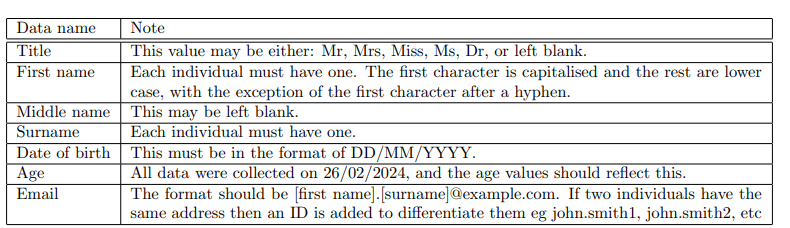
数据应以 JSON 格式保存. 转换为小写字母下划线而不是空格字符，例如“first\_name”。

表1

数据清理

编写一个clean\_data函数，在 clean data 函数中，应该执行数据清理任务，加载存储在中的数据

格式化的用户数据。所有这些代码都可以编写在清理数据函数中，或者可以由

在此类中调用的一系列函数。后一种选择通常被认为是更好的做法。

检查数据以确定哪些值的格式不正确或哪些值可能丢失。

如果值的格式不正确，则必须将其转换为正确的格式。如果某个值缺失或

不正确，则应尝试在给定其他值的情况下填写该数据。清理后的数据应该

保存到全局变量清理后的用户数据

查询

使用下列中给出的名称编写一个函数，

执行相应“查询描述”中给出的查询。答案应该由函数返回，而不是本地或全局存储。

