

项目名称：2023年春季CS307课程项目（第一部分）

主要贡献者：

项目负责人及总体设计：朱跃明

数据准备与文档编写：王丽爽、张超祖

评审：马裕欣

扩展自2022年春季项目

一般要求：

本项目为小组项目，每组只有两名在同一实验课上的成员。每组应独立完成项目，并提交由组员共同完成的一份报告。项目1的队友也将是项目2的队友。一旦配对，不允许更换队友。报告应在截止日期前提交。截止日期后的所有迟交作品将获得零分。严禁从互联网和同学处复制任何句子和图片。本课程严禁抄袭。文本描述应严谨，整体设计应有逻辑，报告结构和图表布局应清晰易读，否则评分阶段将受到惩罚。报告页数应在4到8页之间。只有或少于3页且多于8页的报告将在评分阶段受到惩罚。

数据库管理系统（DBMS）可帮助我们以方便的方式管理数据并提高数据检索效率。项目1的工作主要分为以下三个部分：

1. 根据提供的数据文件和数据关系设计E-R图。
2. 根据提供的数据文件使用PostgreSQL设计关系数据库。
3. 将所有数据导入数据库。

背景

数据描述 posts.json文件包括以下字段：Post ID、Title、Category、Content、Posting Time、Posting City、Author、Author Registration Time、Author's ID、Author's Phone、**Authors Followed By**、Authors Who Favorited the Post、Authors Who Shared the Post、Authors Who Liked the Post。

replies.json文件包括以下字段：Post ID、Reply Content、Reply Stars、Reply Author、Secondary Reply Content、Secondary Reply Stars、Secondary Reply Author。

在本项目中提到的两个JSON文件（posts.json和replies.json）是包含文章和回复数据的文件。JSON（JavaScript Object Notation）是一种轻量级的数据交换格式，易于阅读和编写，同时也易于机器解析和生成。这两个文件的结构大致如下：

1. posts.json 包含文章相关信息，例如文章ID、标题、类别、内容、发布时间、发布城市、作者、作者注册时间、作者ID、作者电话以及与作者相关的其他信息（关注的作者、喜欢该文章的作者、分享该文章的作者等）。
2. replies.json 包含回复相关信息，例如文章ID（与posts.json中的文章ID相对应）、回复内容、回复星级、回复作者、二级回复内容、二级回复星级、二级回复作者等。

这两个JSON文件的数据将用于设计和创建关系数据库，以满足项目需求。

报告及任务

1. 小组基本信息，包括成员姓名、学号、实验课时。
2. 记录每位小组成员的贡献及贡献百分比，明确指出任务/任务部分是由哪位组员完成的。

任务1：E-R图（占总分30%）

使用任何图形软件制作数据库设计的E-R图。不接受手绘结果。请遵循E-R图的标准。报告中需要提供E-R图的截图，并注明用于绘制图表的软件/在线服务名称。

在根据本项目提供的数据文件（posts.json和replies.json）绘制ER（实体-关系）图时，可以遵循以下步骤：

1. 识别实体：首先，需要从数据文件中识别出不同的实体。在这个项目中，我们可以识别出以下实体：
 - Post（文章）
 - Author（作者）
 - Reply（回复）
 - Secondary Reply（二级回复）
2. 为实体定义属性：根据数据文件中的信息，为每个实体定义相应的属性。例如：
 - Post：Post ID、Title、Category、Content、Posting Time、Posting City等。
 - Author：Author ID、Author Registration Time、Author's Phone等。
 - Reply：Post ID、Reply Content、Reply Stars、Reply Author等。
 - Secondary Reply：Post ID、Secondary Reply Content、Secondary Reply Stars、Secondary Reply Author等。
3. 确定主键：为每个实体选择一个能唯一标识其实例的属性作为主键。在这个项目中，可以选择以下主键：
 - Post：Post ID
 - Author：Author ID
 - Reply：Reply ID（可能需要创建一个新的属性作为主键）
 - Secondary Reply：Secondary Reply ID（可能需要创建一个新的属性作为主键）
4. 确定实体间的关系：根据数据文件中的信息，确定实体之间的关系。例如：
 - 一个作者（Author）可以发布多篇文章（Post），一个文章（Post）只能有一个作者（Author）。这是一个一对多（1:N）关系。
 - 一个文章（Post）可以有多个回复（Reply），一个回复（Reply）只能属于一个文章（Post）。这是一个一对多（1:N）关系。
 - 一个回复（Reply）可以有多个二级回复（Secondary Reply），一个二级回复（Secondary Reply）只能属于一个回复（Reply）。这是一个一对多（1:N）关系。
5. 添加外键：根据实体间的关系，为相关实体添加外键。例如：
 - 在Reply实体中添加一个名为“Post ID”的外键，该外键引用Post实体中的主键“Post ID”。
 - 在Secondary Reply实体中添加一个名为“Reply ID”的外键，该外键引用Reply实体中的主键“Reply ID”。
6. 使用绘图工具创建ER图：将识别出的实体、属性、主键、外键和关系添加到ER图中。可以使用诸如Lucidchart、Draw.io、Microsoft Visio等绘图工具来创建ER图。

完成以上步骤后，您将获得一个清晰、合理的ER图，为后续的关系数据库设计和实现奠定基础。

初步ER图构建思路

1. 作者 (Author) 表:

- Author's ID: 主键、唯一、非空
- Author: 非空
- Author Registration Time: 非空
- Author's Phone: 唯一、非空

2. 文章 (Post) 表:

- Post ID: 主键、唯一、非空
- Title: 非空
- Category: 非空
- Content: 非空
- Posting Time: 非空
- Posting City: 非空
- Author: 外键 (引用作者表的Author's ID) , 非空

3. 评论 (Reply) 表:

- Reply ID: 主键、唯一、非空
- Reply Content: 非空
- Reply Stars: 非空
- Reply Author: 非空, 如果不在作者表中, 需要为其生成ID及合理注册时间并添加至作者表
- Post ID: 外键 (引用文章表的Post ID) , 非空

4. 次级评论 (Secondary Reply) 表:

- Secondary Reply ID: 主键、唯一、非空
- Secondary Reply Content: 非空
- Secondary Reply Stars: 非空
- Secondary Reply Author: 非空, 如果不在作者表中, 需要为其生成ID及合理注册时间并添加至作者表
- Reply ID: 外键 (引用评论表的Reply ID) , 非空

5. 关注 (Followed_Authors) 表

- Author ID: 主键、非空
- Follower_Author_ID: 非空 (可以有不在Author中的ID)

6. 收藏 (Favorited_Posts) 表

- Post ID: 主键、外键 (引用文章表的Post ID) 、非空
- Favorited_Author_ID: 非空 (可以有不在Author中的ID)

7. 分享 (Shared_Posts) 表

- Post ID: 主键、外键（引用文章表的Post ID）、非空
- Sharing_Author_ID: 非空（可以有不在Author中的ID）

8. 点赞 (Liked_Posts) 表

- Post ID: 主键、外键（引用文章表的Post ID）、非空
- Liking_Author_ID: 非空（可以有不在Author中的ID）

任务2：关系数据库设计（占总分40%）

根据上述背景设计表格和列。通过“显示可视化”功能生成E-R图。简要描述表格和列的设计，包括（但不限于）表格和列的含义。报告中需要提供以下内容：

1. 附上由DataGrip生成的E-R图的快照。
2. 简要描述表格设计及每个表格和列的含义。

此外，请提交一个包含所有创建表的DDL（创建表语句）的SQL文件作为附件。请将其制作成一个单独的文件，而不是将语句复制粘贴到报告中。

数据库设计注意事项：

1. 所有数据项应基于两个文件posts.json和replies.json。
2. 设计需遵循三个范式的要求。
3. 使用主键和外键表示关于数据的重要属性和关系。
4. 每个表中的每行应通过其主键唯一标识（可以使用简单或复合主键）。
5. 每个表都应涉及外键。不允许孤立表。
6. 设计中不能有循环外键链接。
7. 每个表应至少包含一个强制性（“Not Null”）列（包括主键但不包括id列）。
8. 除系统生成的自增ID列外，应至少有一个带有“唯一”约束的列。
9. 应为不同字段使用适当的数据类型。
10. 设计应在需求发生变化时易于扩展。

任务3：数据导入（占总分30%）

在此任务中，您应编写脚本将这两个json文件的内容导入到先前设计的数据库中。导入数据后，还应确保所有数据已成功导入。

任务3.1基本要求：10%

1. 编写的用于导入数据文件的脚本。
2. 关于如何使用脚本导入数据的描述。应清楚说明运行脚本并正确导入数据所需的步骤、必要的先决条件和注意事项。

任务3.2高级要求：10%

为获得剩余分数（10%），您可能还需要完成以下高级要求：

1. 尝试优化脚本，找到多种导入数据的方法，并对这些方法的计算效率进行比较分析。

对于高级分数，请务必描述测试环境、程序和实际时间成本。要求撰写一两段分析实验结果的文字。

任务3.3数据准确性检查：10%

我们将在4月25日和27日的实验课上检查此部分。

1. 根据posts.json和replies.json中的数据，我们将在实验课上提出一些问题，以检查是否所有数据都已正确导入到数据库中。
2. 如果作者名称出现在以下字段中，但未出现在Author字段中，需要为作者帐户生成作者ID和合理的注册时间，并将其添加到数据库中。

报告提交方式 请在2023年4月24日23:30（北京时间，UTC+8）前将报告（PDF格式）及必要的附件（如SQL脚本和源代码文件）提交至Sakai网站。对于附件，请根据任务将它们放入单独的目录中，并将它们压缩成.zip文件。