Fall 2023 CS307 Project Part I

Contributors:

Topic Design: SUN Kebin

Data Preparation and Documentation: Wang Lishuang, Zhang Haoming, Sun Kebin

Other Contributors: Leng Ziyang, Tang Yulei

Review: WANG Weiyu

This project description is extended from the one of Fall 2022 CS307

一般要求

- 一个只有 2 到 3 名在**同一实验课**中的队友的团队项目。每个团队应独立完成项目并只提交**一份报告**。当一个团队由不同成员提交多份报告时，我们将随机选择其中一份。当一个团队由同一成员提交多份报告时，我们将评判最新的那份。
- 你选择的项目 I 队友也将成为你的项目 II 队友。一旦配对，不允许更换队友。
- 你应该在截止日期之前提交报告。**所有迟交的报告将得零分。**
- **不要从互联网和你的同学那里复制任何代码和图表。**严禁抄袭。
- 你的报告页数应**在 8 到 20 页之间**。少于 8 页的报告将受到扣分处罚，但超过 20 页的报告不会为你赚取更高的分数。

数据库管理系统（DBMS）在处理大量数据时非常有用。它可以帮助我们以便捷的方式管理数据，提高检索和修改的效率。因此，您在项目 I 上的工作包括以下几个部分：

1. 找出所提供数据的内在关系，然后根据您找到的关系设计一个 E-R 图。
2. 使用 PostgreSQL 根据提供的数据文件设计一个关系型数据库。
3. 将所有数据导入数据库。
4. 比较数据库和原始文件 I/O 在数据检索和修改方面的性能表现。**仅允许使用 Java 作为编程语言[1]。**

第 1 节 背景

本项目涉及一个虚构的弹幕视频网站的结构 - Synchronized User-generated Subtitle Technology Company（SUSTech）。经过数十年的发展，它已经建立了一个

[1] 完成这个项目的第一部分使用其他常见的编程语言是可以接受的。然而，由于这门课程项目第二部分将涉及自动基准测试，到那时你将需要将你的代码翻译成 Java 来通过测试。

围绕用户和创作者的生态系统，他们不断产生高质量的内容，包括视频和弹幕评论。通常情况下，视频由被称为“UP 主”的用户提交。然后，这些视频将被审核，以检查是否符合“审核员”的标准。审核成功后，UP 主可以在其指定的时间发布这个视频。用户随后可以观看视频。在观看期间，他们可以通过弹幕跨越时间和空间与彼此互动。当然，除了发送弹幕消息之外，用户还可以进行一系列操作，例如点赞、投币和收藏。

1.1 数据描述

提供了三个数据文件：user.csv、videos.csv 和 danmu.csv。

1.1.1 user.csv

- mid：用户的唯一标识号。
- name：用户创建的名称。
- sex：包括但不限于生物性别。
- birthday：用户的生日。
- level：根据系统决策标准评估的用户参与度。
- sign：用户创建的个人描述。
- following：包含该用户关注的所有其他用户的唯一标识号（mid）的列表。
- identity：取值为 {"user", "superuser"}，指示用户的角色。

1.1.2 videos.csv

- BV：视频的唯一标识字符串。
- title：视频所有者创建的视频名称。
- owner mid：视频所有者的唯一标识号（mid）。
- commit time：所有者提交此视频的时间。
- review time：视频被其审查者检查的时间。
- public time：视频对所有用户公开的时间。
- duration：视频时长。
- description：上传者提供的简短文本介绍。
- reviewer：视频审查者的唯一标识号（mid）。
- like：包含喜欢此视频的用户唯一标识号（mid）的列表。
- coin：包含给予此视频硬币的用户唯一标识号（mid）的列表。
- favorite：包含将此视频标记为收藏的用户唯一标识号（mid）的列表。
- view：包含观看此视频的用户及其上次观看的时长的列表。

1.1.3 danmu.csv

- BV：发送弹幕的视频的 BV。
- mid：发送弹幕的用户的 mid。
- time：弹幕出现的视频时间。
- content：弹幕的内容。

第 2 节 报告结构和任务要求

2.1 团队基本信息和工作量

您需要记录团队**成员信息、具体贡献内容**以及每位团队成员的**百分比**。

2.2 任务 1：E-R 图（15%）

使用任何图形软件设计你的数据库并绘制 E-R 图。不接受手绘的结果。请遵循 E-R 图的标准。在报告中，你必须提供 E-R 图的截图或嵌入的矢量图。另外，请说明你用于绘制图形的软件或在线服务的名称[2]。

2.3 任务 2：数据库设计（25%）

根据上面提供的背景设计表格和列。首先，您应通过 DataGrip 的“显示可视化”功能生成数据库图表，并将快照或矢量图形嵌入到您的报告中。然后，简要描述表格和列的设计，包括但不限于表格和列的含义。

此外，请将一个 SQL 文件作为附件提交，其中包含您创建的所有表格的 DDL（创建表语句）。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。

一些注意事项：

1. 使用 **PostgreSQL** 设计一个数据库，以管理上述提到的所有信息。
2. 您的设计需要遵循三范式的要求。
3. 使用主键和外键来指示关于您的数据的重要属性和关系。
4. 每个表格上的每一行都应该通过其主键唯一标识（可以使用简单的或复合主键）。
5. 每个表格都应该包含在一个链接中，不应该存在孤立的表格。
6. 您的设计不应包含循环链接，即表格之间的外键方向不应形成循环。
7. 每个表格始终应该至少有一个强制非空（“Not Null”）列（包括主键，但不包括 id 列）。
8. 对不同数据字段使用适当的数据类型。
9. 您的设计应尽可能容易扩展（特别适用于三人小组）。

2.4 任务 3：数据导入（28% + 3%）

在这个任务中，您应该**编写一个脚本**将数据导入您设计的数据库中。在导入数据之后，您还应确保**所有数据成功导入**。

在报告中，您需要完成基本要求（28%中的 18%）：

1. 编写一个脚本来导入数据文件。
2. 在报告中描述您的脚本如何导入数据。您应清晰地陈述运行脚本和正确导入数据所需的步骤、必要的前提条件和注意事项。显示每个实体表中的记录数。（特别适用于三人小组）

您还可以完成以下高级要求来获得剩余的分数（28%中的 10%）（三人小组应表现更出色）：

1. 找到多种导入数据的方法，并提供这些方法之间的计算效率的比较分析。

[2] 无论您选择哪种 E-R 图表方法，都请清晰完整地标记基本要素。

2. 尝试优化您的脚本。描述您如何对其进行优化，并分析与原始脚本相比有多快。

对于高级任务，请确保描述您的测试环境、流程和实际时间成本。您需要写一两段话来分析实验结果，可以参考任务 4 中关于报告实验结果的要求以获取详细信息。

当您在数据导入过程中进行深入探索并获得特别高的效率时，您可以获得奖励分数（1%~3%）。

2.5 任务 4：比较 DBMS 与文件 I/O（30% + 7%）

在报告中，您需要完成以下**基本要求**（30%中的 20%）：

1. 您需要描述测试环境，包括（但不限于）：
   a) 硬件规格，包括 CPU 型号和内存大小。
   b) 软件规格，包括您的 DBMS 版本和操作系统版本，您选择的编程语言以及开发环境（语言的版本、编译器和库的具体版本等）。
   c) 在报告环境时，您可以思考这个问题：如果其他人要复制您的实验，应该为他/她提供哪些必要信息？
2. 您需要说明如何在 DBMS 和数据文件中组织测试数据，包括如何生成测试 SQL 语句以及文件的数据格式/结构是什么样的。
3. 您需要描述您的测试 SQL 脚本和程序的源代码。请不要将整个脚本和程序复制粘贴到报告中，而是将源代码作为附件提交。
4. 您需要对相应语句/操作的运行时间进行比较研究。鼓励您使用数据可视化来呈现结果。确保在图形、表格等方面使用一致的样式。除了运行时间的列表/图表外，您还需要描述关于运行性能的主要差异，您在结果中发现的有趣之处，您可以在实验中向其他人展示的见解等。

除了基本要求之外，您还可以考虑以下高级任务（但不限于以下任务），以挑战自己并获得额外的分数（30%中的 10%）（三人小组应在这一部分表现更出色）：

1. 您的数据库是否能处理高并发？您可以尝试在更高数量级上进行上述基准测试，例如数十万次选择操作。
2. 您是否可以比较不同的数据库软件（例如 MySQL、MariaDB、SQLite）、文件系统、磁盘性能和类型、编程语言、库或操作系统的性能？

当您在实验课之外进行深入探索并获得特别高的效率时，您可以获得额外的奖励分数（1%~7%）。

第 3 节 提交（2%）

在北京时间（UTC+8）**2023 年 11 月 5 日的 23:00 之前**，通过 Blackboard 提交一个名为“Report_sid1_sid2_sid3.pdf”的 **PDF 格式**报告，以及包含所有项目文件的**.zip 压缩文件**（例如 SQL 脚本和源代码文件）。在报告中，请将 sid1/sid2/sid3 替换为您的团队成员的学号。对于附件，请将它们放入项目的单独目录中，并将它们压缩成.zip 档案。

第 4 节 免责声明

本项目背景中的角色、企业和事件纯属虚构。文件中的内容是随机生成的虚假数据。任何与实际事件、实体或个人的相似之处都是纯属巧合，不应被解释为 CS307 教学组的观点或含义。

本文件为英文原版内容的翻译，仅供参考之用。如有任何与英文原版内容不一致之处，概以英文原版内容为准。本人对于因使用本翻译文件而引起的任何后果不承担任何责任。

附录 A

在比较数据库 API 和文件 API 之间的数据检索和操作性能时，我们建议您按照以下步骤进行比较分析：

1. 使用数据库 API 进行基准测试：根据您创建的数据库，您需要编写一个 Java 程序，通过数据库 API 访问数据库，并包含一系列 INSERT、DELETE、UPDATE 和 SELECT 语句。您可以自行指定每种类型的语句数量，并决定要修改和读取哪些数据。但是，每种语句类型的操作不能太小，以便能够清楚地展示数据库 API 相对于文件 I/O 的优势和劣势（根据编程语言的不同，通常每个语句类型的操作数都应足够多，比如数万条记录）。最后，您需要记录每种语句类型或每个语句的运行时间。在这里，我们提供了一些您可以参考的数据库中的典型测试描述：

    a) INSERT：首先，随机删除 csv 文件的一些行，并将其导入到您的数据库中。然后评估导入 csv 文件其余行的时间成本。

    b) DELETE：首先，将 csv 文件的所有数据导入到您的数据库中。然后，评估删除您数据库的交付记录表中任意选择的行的时间成本。

    c) UPDATE：首先，将 csv 文件的所有数据导入到您的数据库中。然后，评估将您数据库的交付记录表中所有空值更新为任意值的时间成本。

    d) SELECT：首先，将 csv 文件的所有数据导入到您的数据库中。然后，评估查找所有未完成的交付记录或查找已由任意容器包装的所有交付记录等的时间成本。您可以更多关注 SELECT 语句的测试，因为它们在许多实际场景中比其他语句更常用。

2. 使用文件 API 进行基准测试：此步骤旨在复制第一步中的所有操作，但使用通用编程语言的标准文件 API。首先，创建存储与 DBMS 中表中相同数据的文件。然后，编写一个程序，以与 SQL 语句和查询中相同的方式插入、删除、更新和查找数据项。确保文件操作（以及操作数量）与 SQL 操作（以及语句数量）相同。最后，记录与数据库 API 基准测试中一样的每个操作（类型）的运行时间。

3. 比较分析：比较来自 DBMS 和文件的相同操作/语句的记录运行时间。您可以从多个层面进行比较，例如比较相应操作的语句（语句级别）或比较特定类型中所有语句的总时间与相应操作类型（类型级别）。

附录 B

关于如何更好地完成这个项目的一些注意事项：

1. 您可以使用不同数量级的语句/操作执行上述基准测试（例如，从几百到几千到数万个）。
2. 您可以选择或设计任何您想要在文件中存储数据的格式，例如纯文本格式（CSV、JSON、XML 等）或自定义的二进制格式。
3. 请仅坚持使用标准文件 API，即 Java 中的 java.io。唯一的例外是，如果您选择使用 JSON 和 XML，如果标准库没有提供，您可以使用第三方 JSON/XML 库，例如 Gson[3]。
4. 我们知道有许多库可以简化数据操作工作，甚至可以显著加速插入和选择的性能（例如 Python 中的 pandas）。我们鼓励您还比较这些库与 DBMS 的性能。然而，在此之前，您应该先进行 DBMS 与标准文件 API 的分析。
5. 一些有用的资源：
   a) Advantages of Database Management System over file system
   b) Advantages of Database Management System
   c) Characteristics and benefits of a database

[3] 如果您坚持使用另一种语言，这条规则仍然适用。例如，您应该 在 C/C++中使用 iostream 和 fstream，在 Python 中使用 file 对象，或者在 C#中使用 System.IO。在处理 JSON 和 XML 时，您可以使用 Python 的 json 包；但对于像 C#这样的语言，它的标准库提供了 System.Text.Json 和 System.Xml，您不应该使用替代品。