南京大学

数据管理基础
实验手册（一）

实验手册一

前言

本实验手册旨在通过实际操作，帮助大家熟悉数据管理的不同方式：文件系统（如 CSV 和 JSON 文件）、关系型数据库（SQL）以及非关系型数据库（NoSQL）。通过对比这三种数据管理方式在数据查询、存储和管理上的异同，将能够了解以下内容：

1. **文件系统的数据管理**：
   - 学习如何通过编程语言（如 Python）直接读取和查询 CSV 和 JSON 文件中的数据。
   - 理解文件系统在数据存储和查询中的优缺点，尤其是在数据规模较小、结构简单时的适用场景。
2. **关系型数据库的数据管理**：
   - 掌握如何使用 SQL 语言进行数据的增删改查（CRUD）操作。
   - 理解关系型数据库在数据一致性、事务支持和复杂查询方面的优势。
3. **非关系型数据库的数据管理**：
   - 学习如何使用 NoSQL 数据库（如 MongoDB）存储和查询非结构化或半结构化数据。
   - 理解 NoSQL 数据库在高并发、分布式场景下的灵活性和扩展性。

实验所用数据下载链接：https://box.nju.edu.cn/d/5b20ed628b684c718681/

一、文件系统的数据管理

1. 实验目的

在不使用数据库的情况下，用编程方式解决对csv文件和json的数据查询和统计任务。

2. 操作环境

强烈建议使用Python环境，Python 的 pandas 库是专门为数据分析设计的，提供了高效的数据结构和操作，支持快速过滤、排序、分组、聚合等操作。

也可选用其他编程语言。

3. 实验内容

3.1. 实现对csv文件的读取和查询

3.1.1. 过滤与排序

要求：

提取所有满足以下条件的日志：

- `logType='custom'`
- `eventId` 为 `"APM-SERVICE-START"` 或 `"APM-SERVICE-END"`

输出字段：

- `_logid`：日志的唯一编号
- `eventId`：事件类型（只保留 `"APM-SERVICE-START"` 或 `"APM-SERVICE-END"`）
- `user_id`：用户 ID
- `logTime`：将原始毫秒时间戳转换为标准字符串时间，格式 `yyyy-MM-dd HH:mm:ss`

排序规则：

- 按 `logTime` 升序排序

输出文件：

- 保存为 `task1.csv`

3.1.2. 会话分析

要求：

对日志按 `trace_id` 分组，统计每个会话的以下信息：

输出字段：

- `trace_id`：会话唯一标识
- `log_count`：该 `trace_id` 下的日志总条数
- `first_event`：该 `trace_id` 下第一条日志的 `eventId`

- `last_event`：该 `trace_id` 下最后一条日志的 `eventId`
- `duration_ms`：该会话的持续时间（最后一条日志的 `logTime` – 第一条日志的 `logTime`，单位毫秒）

输出文件：

- 保存为 `task2.csv`

3.1.3. 条件查询

要求：
找出那些在同一个 `trace_id` 下，`eventId` 同时出现过：

- `"Failed to connect to gameinfoc.biligame.net"`
- `"Unable to resolve host gameinfoc.biligame.net"`

输出字段：

- `trace_id`：满足条件的会话标识
- `log_count`：该 `trace_id` 下日志总条数
- `first_event`：该 `trace_id` 下第一条日志的 `eventId`
- `last_event`：该 `trace_id` 下最后一条日志的 `eventId`
- `duration_ms`：该会话的持续时间（最后一条日志的 `logTime` – 第一条日志的 `logTime`，单位毫秒）

输出文件：

- 保存为 `task3.csv`

3.2. 实现对json文件的读取和查询

3.2.1. 查询

题目描述：

读取 JSON 数据（umas.json），统计所有记录里 **hero_card –> factors** 列表中每个因子 `name` 对应的 `total_rarity` 总和。

要求输出 JSON 文件，格式如下：

```plain text
{
  "速度": 21,
  "力量": 72,
  "草地": 20,
  "英里": 31,
  "中距离": 10,
  ...
}
```

- Key 为因子名称 `name`
- Value 为全局累加的 `total_rarity`

保存结果为 `task4.json` 。

4. 提交内容

结果文件 `task1.csv` 、 `task2.csv` 、 `task3.csv` 、 `task4.json` 。

二、关系型数据库的数据管理

2.1实验目的

1. 理解和掌握 SQL 基本语法。
2. 掌握创建表、插入数据、查询数据、排序数据等基本数据库操作。
3. 掌握如何在数据库中管理和操作学生课程数据。

2.2实验环境

- MySQL v8.2.0

2.3实验内容

（2.3.1–2.3.5自己编写sql语句完成操作并截图）

2.3.1 导入数据

参考：https://cloud.tencent.com/developer/article/2156270
导入 `main.sql` 到数据库里面，数据库名称可自己指定

2.3.2 查询前列跑法的高速度天赋的角色

查询 **高速度天赋** `talent_speed>10` 大于 10 的角色，并且 `running_style=2` （前列跑法）。

2.3.3 修改低稀有度的高速度天赋角色

将**低稀有度** `default_rarity=1` 且 **高速度天赋** `talent_speed>10` 的角色**稀有度** `default_rarity` 提升为 `2` 。

2.3.4 按照天赋排序

对所有角色按 综合**天赋总和**（`talent_speed + talent_stamina + talent_pow + talent_guts + talent_wiz`）进行 **降序排序**，挑选出前 10 名。

2.3.5 删除低天赋角色

删除所有天赋总和小于 30 的角色。

2.4 数据文件

数据文件已包含在实验平台上实验所用数据下载链接：
https://box.nju.edu.cn/d/5b20ed628b684c718681/。

2.5 提交内容

1. 在完成实验后，请在平台上执行上述所有 SQL 语句并截图保存提交 moodle

2. 提交截图包括但不限于以下内容：

创建表的 SQL 语句。

插入数据的 SQL 语句。

查询、排序和更新数据的 SQL 语句及结果。

删除数据的 SQL 语句及结果

三、非关系型数据库的数据管理

3.1 实验目的

1. 体验NoSQL与SQL的区别。
2. 尝试对NoSQL进行简单的操作。

3.2 实验平台

MongoDB在线平台

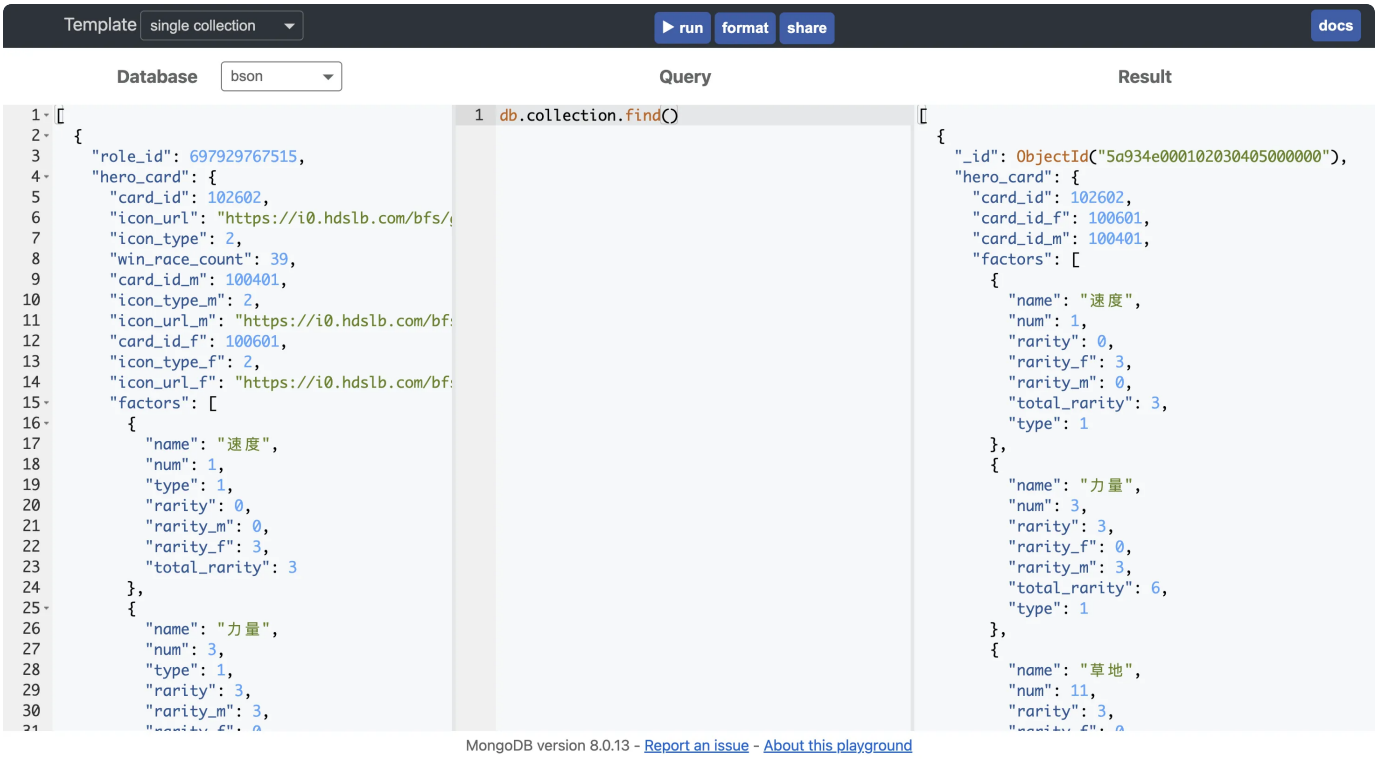
3.3 MongoDB介绍

MongoDB 是一种面向文档的开源 NoSQL 数据库管理系统。它将数据存储为类似 JSON 的文档，具有高度的灵活性和可扩展性。其特点包括支持动态模式，无需预先定义表结构；能处理海量数据和高并发读写操作；通过自动分片实现水平扩展，可将数据分布在多个服务器上；还提供了强大的查询功能和索引机制，能快速查询和处理数据。此外，它拥有丰富的客户端库，便于不同编程语言进行交互，广泛应用于 Web 应用、移动应用、大数据处理等领域。

3.4 实验内容

3.4.1 数据

从 `umas.json` 文件中找到 `data->records` 数组。将该数组内容粘贴至左侧，中间输入具体的命令，结果会在右侧显示。



3.4.2 相关命令

JSON

```
// 查询所有文档
db.collection_name.find()

// 查询符合条件的文档
db.collection_name.find({ "age": { $gt: 18 } })

// 只返回指定字段
db.collection_name.find(
  { "age": { $gt: 18 } },
  { "name": 1, "age": 1, "_id": 0 }
)

// 多条件 AND
db.collection_name.find({ "age": { $gt: 20 }, "name": "Alice" })

// 多条件 OR
db.collection_name.find({ $or: [ { "age": 20 }, { "age": 30 } ] })

```

3.5 提交内容

对所有记录的 `hero_card.factors` 列表进行统计，计算每个因子 `name` 对应的 `total_rarity` 总和。

结果参考如下：

JSON

```
[
  {
    "_id": "力量",
    "total_rarity_sum": 72
  },
  {
    "_id": "耐力",
    "total_rarity_sum": 63
  },
  ......
]
```

请你写出对应的mongodb查询语句并截图提交至moodle。

小提示：

1. `$unwind` 会把数组拆开，每个数组元素都会生成一条单独的中间文档，这样就可以对数组里的每个对象单独处理。 https://www.mongodb.com/zh-cn/docs/manual/reference/operator/aggregation/unwind/
2. `$group` 可以对拆开的文档进行分组操作， `_id` 指定分组字段， `$sum` 、 `$avg` 等聚合函数可以对分组数据计算统计值。https://www.mongodb.com/zh-cn/docs/manual/reference/operator/aggregation/group/

四、提交方式

提交内容包括：

- 实验一： `task1.csv` 、 `task2.csv` 、 `task3.csv` 、 `task4.json`
- 实验二：4张结果截图
- 实验三：1张结果截图

打包为 `[学号+姓名].zip` 提交至moddle。