

沙威玛卷饼商店顾客评价系统

背景：

沙威玛卷饼商店不仅注重食品的质量，还非常注重顾客的服务体验和商店环境。为了更好地提升服务质量，商店决定引入一个顾客评价系统，顾客可以对食品、服务、环境等多个维度进行评分。商店希望能使用一个**链表数据结构**来管理这些评价，以便灵活地进行插入、删除、修改以及查询操作。

评价内容：

每个顾客在评分时，必须对以下三个维度进行评分：

- 食品评分：**顾客对沙威玛卷饼食品质量的评分（1.0到5.0分，精确到小数点后一位）。
- 服务评分：**顾客对商店服务质量的评分（1.0到5.0分，精确到小数点后一位）。
- 环境评分：**顾客对商店环境的评分（1.0到5.0分，精确到小数点后一位）。
- 注：**以下所有需要输出所有评分的场景按照默认按照这三个评分的顺序输出

每个评价包含以下内容：

- 顾客ID：**唯一标识顾客的ID。为一个正整数，保证每个顾客的ID唯一。
- 评分：**顾客对每个维度的评分（1.0-5.0，精确到小数点后一位）。
- 时间戳：**顾客给出评价的时间，用于排序和版本管理。时间戳格式为 `yyyy-mm-dd hh:mm`，其中 `yyyy` 为四位数年份，`mm` 为两位数月份，`dd` 为两位数日期，`hh` 为两位数小时，`mm` 为两位数分钟。例如，`2024-12-03 15:45` 表示2024年12月3日下午3点45分。
- 版本号：**每次修改评分时，生成一个新的版本号，用于标识评分的历史版本。版本号从1开始，每次修改时版本号加1。

要实现的操作：

商店需要实现以下操作来管理顾客的多维度评分：

1. 插入新的评价：

- 插入评价时，顾客必须插入一个包括**所有维度**的新的评价。
- 每次评价时，记录时间戳，并为每次修改（操作2）生成一个新的版本。

2. 修改评价历史：

- 顾客可以**修改某个维度**的评分。每次修改会生成一个新版本的评分，并保留旧版本记录。
- 新版本的评分包括新的评分分数，新的时间戳和新的版本号。

3. 清空评价：

- 顾客可以清空某评分的所有记录。清空时，**删除评分的所有版本**。

4. 查询某个顾客的评价历史：

- 根据顾客ID和维度查询顾客的所有历史评分记录，显示**每个版本**的评分和时间戳。按照**版本号降序**排列历史评分记录。**如果不指定维度，则显示所有维度**。

5. 显示所有评价：

- 按照**时间戳降序**排序，显示**所有顾客在某个维度上的评价**（每个顾客只输出最新版本），输出顾客ID、评分、时间戳。**不指定维度则输出所有维度**。

6. 按评分范围查询顾客：

- 根据顾客的评分区间查询评分符合条件的顾客(只考虑最新版本)。可以指定多个维度的评分区间，如查询食品评分在4.0到5.0之间，服务评分在3.0到5.0之间的顾客。**按照顾客ID降序排列**。

输入格式：

- 第一行输入为操作命令的总数（整数 `n`），表示接下来有 `n` 个操作命令需要解析。
- 根据操作命令的不同，后续输入相应的参数：

- **insert:**

- 格式：

```
1 insert {customer_id} {food_rating} {service_rating} {environment_rating} {timestamp}
```

- `customer_id`：顾客ID，正整数，保证唯一。
 - `food_rating`、`service_rating`、`environment_rating`：评分值，范围是1.0到5.0，精确到小数点后一位。
 - `timestamp`：一个时间戳，格式为 `yyyy-mm-dd hh:mm`，表示评价的时间。
 - 示例输入：

```
1 insert 101 4.5 5.0 3.5 2024-12-03 15:45
```

- **modify:**

- 格式：

```
1 modify {customer_id} {dimension} {new_rating} {timestamp}
```

- `customer_id`：顾客ID，正整数，保证唯一。
- `dimension`：可以是 `food`、`service`、`environment`
- `new_rating`：新的评分，范围是1.0到5.0，精确到小数点后一位。
- `timestamp`：一个时间戳，格式为 `yyyy-mm-dd hh:mm`，表示修改的时间。
- 示例输入：

```
1 modify 101 food 5.0 2024-12-03 16:00
```

◦ delete:

- 格式：

```
1 delete customer_id
```

- `customer_id`：顾客ID，正整数，保证唯一。
- 示例输入：

```
1 delete 101
```

◦ history:

- 格式：

```
1 history customer_id [dimension]
```

- `customer_id`：顾客ID，正整数，保证唯一。
- `dimension`：可以是 `food`、`service`、`environment`，也可以不指定维度以显示所有维度。
- 示例输入：

```
1 history 101 food
```

- **display:**

- 格式:

```
1 display [dimension]
```

- 示例输入:

```
1 display
```

- **range_query:**

- 格式:

```
1 range_query {food_rating_lower} {food_rating_upper}  
  {service_rating_lower} {service_rating_upper} {environment_rating_lower}  
  {environment_rating_upper}
```

- 查询的评分区间是给定的最小值和最大值（1.0到5.0，精确到小数点后一位）。如果对于某个维度没有约束，直接给1.0-5.0即可。

- 示例输入:

```
1 range_query 4.0 5.0 3.0 5.0 2.0 5.0
```

输出格式:

- 对于 `insert` 操作, 输出 `Review inserted successfully`。
- 对于 `modify` 操作, 输出:
 - `Modification successful` 或
 - `Customer ID not found, modification failed`
- 对于 `delete` 操作, 输出:
 - `Deletion successful` 或
 - `Customer ID not found, deletion failed`
- 对于 `history` 操作, 输出顾客的所有历史评价版本, 包括版本号、评分、时间戳:

- 首先输出 `History:`
- 每个历史记录以 `Version <version_number>: <dimension> Rating <rating> Timestamp <timestamp>` 的格式输出，按**版本号降序**排列。如果不指定维度，则显示所有维度。**维度顺序别忘记按照指定顺序排列（见评价内容部分）。**
- 输出示例：

```
1 History:
2 Version 2: Food Rating 5.0 Service Rating 5.0 Environment Rating 4.5
   Timestamp 2024-12-03 16:00
3 Version 1: Food Rating 4.5 Service Rating 4.0 Environment Rating 3.5
   Timestamp 2024-12-03 15:45
```

- 对于 `display` 操作：
 - **如果指定维度，按照时间戳降序排序**，显示所有顾客在该维度上的**最新评价**，显示顾客ID、评分、时间戳。
 - **如果不指定维度，显示所有顾客在所有维度上的最新评价**，显示顾客ID、评分、时间戳。
 - 输出示例：

```
1 Customer ID 101 Food Rating 5.0 Service Rating 5.0 Environment Rating 4.5
   Timestamp 2024-12-03 16:30
2 Customer ID 102 Food Rating 4.0 Service Rating 4.5 Environment Rating 4.0
   Timestamp 2024-12-03 16:15
```

- 对于 `range_query` 操作，输出所有符合评分区间的顾客信息，包括顾客ID、评分、时间戳：
 - 每条记录以 `Customer ID <customer_id> <dimension> Rating <rating> Timestamp <timestamp>` 的格式输出。
 - 按照**顾客id降序输出**
 - 输出示例：

```
1 Customer ID 103 Food Rating 4.0 Service Rating 5.0 Environment Rating 4.0
   Timestamp 2024-12-03 16:20
2 Customer ID 101 Food Rating 4.5 Service Rating 4.0 Environment Rating 4.5
   Timestamp 2024-12-03 16:30
```

其他注意事项：

- 不考虑异常维度的情况
- 只需要在modify和delete时考虑顾客id异常的情况
- 不需要考虑无评价记录等极端边界情况
- 时间戳降序不等同于用户ID降序，请注意（助教wa）

示例：

Input:

```
1 11
2 insert 101 4.5 4.0 3.5 2024-12-01 14:30
3 insert 102 3.5 4.5 5.0 2024-12-01 15:00
4 modify 101 food 4.8 2024-12-01 16:00
5 modify 101 service 4.9 2024-12-01 16:30
6 history 101
7 delete 102
8 insert 102 4.0 4.2 4.5 2024-12-01 17:00
9 display
10 range_query 4.0 5.0 4.0 5.0 4.0 5.0
11 delete 103
12 display food
```

Output:

```
1 Review inserted successfully
2 Review inserted successfully
3 Modification successful
4 Modification successful
5 History:
6 Version 3: Food Rating 4.8 Service Rating 4.9 Environment Rating 3.5 Timestamp
  2024-12-01 16:30
7 Version 2: Food Rating 4.8 Service Rating 4 Environment Rating 3.5 Timestamp
  2024-12-01 16:00
8 Version 1: Food Rating 4.5 Service Rating 4 Environment Rating 3.5 Timestamp
  2024-12-01 14:30
9 Deletion successful
10 Review inserted successfully
11 Customer ID 102 Food Rating 4 Service Rating 4.2 Environment Rating 4.5
  Timestamp 2024-12-01 17:00
12 Customer ID 101 Food Rating 4.8 Service Rating 4.9 Environment Rating 3.5
  Timestamp 2024-12-01 16:30
```

13 Customer ID 102 Food Rating 4 Service Rating 4.2 Environment Rating 4.5
Timestamp 2024-12-01 17:00

14 Customer ID not found, deletion failed

15 Customer ID 102 Food Rating 4 Timestamp 2024-12-01 17:00

16 Customer ID 101 Food Rating 4.8 Timestamp 2024-12-01 16:30