

prototype3628800

## Школа бэкенд-разработки 2021 (осень)

11 сен 2021, 15:11:33

старт: 11 сен 2021, 10:55:06

финиш: 11 сен 2021, 16:55:06

до финиша: 01:43:30

длительность: 06:00:00

## D. Join между файлами

Ограничение времени	6 секунд
Ограничение памяти	128.0 Мб
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Во время проектирования распределенной системы планировалось использовать две различные базы данных. Разработчики не предусмотрели, что им может понадобится выполнять операцию сопоставления значений между таблицами этих баз данных. Помогите им придумать, как выполнить операцию [inner join](#).

База данных **market** и таблица **market.shop** представляет собой таблицу магазинов из двух колонок

- `shop_id` - идентификатор магазина
- `shop_name` - название магазина

База данных **billing** и таблица **billing.order** представляет собой таблицу заказов из трех колонок

- `order_id` - идентификатор заказа
- `shop_id` - идентификатор магазина для, который выполнил заказ
- `cost` - общая стоимость товаров в заказе

Необходимо получить новую таблицу в которой будут следующие колонки:

- `order_id` - номер заказа
- `shop_name` - название магазина
- `shop_id` - идентификатор магазина
- `cost` - общая стоимость товаров в заказе

### Формат ввода

Для удобства базы данных представлены в виде файлов в формате **csv**, каждый файл отсортирован по возрастанию поля **shop\_id**.

Содержимое файлов, используемых в тестах условия, представлено **в примечании**.

На вход программе подаются названия файлов в одной строке, разделенные пробелом, в которых содержатся таблицы, описанные ниже. Строка заканчивается переводом строки (символом `\n`).

`market.shop.filename billing.order.filename`

Пример содержимого файла таблицы `market.shop`:

```
shop_id,shop_name
1,MyShop
2,Pleer.ru
...
```

Гарантируется, что все **shop\_id** в таблице `market.shop` **уникальны**.

Пример содержимого файла таблицы `billing.order`:

```
order_id,shop_id,cost
16423,1,23040
242,2,1680000
...
```

Гарантируется, что все **order\_id** в таблице `billing.order` **уникальны**.

Обозначим за  $N$ ,  $M$  количество строк с данными в файлах `market.shop.filename` и `billing.order.filename` соответственно.

Гарантируются следующие ограничения:

$$0 \leq N, M \leq 10^6.$$

$$0 \leq \text{shop\_id}, \text{order\_id}, \text{cost} \leq 10^{18}$$

$$1 \leq |\text{shop\_name}| \leq 30, \text{ значения в колонке shop\_name состоят из латинских букв, цифр и знаков подчеркивания.}$$

### Формат вывода

Таблица в виде

```
order_id,shop_name,shop_id,cost
16423,MyShop,1,23040
242,Pleer.ru,2,1680000
...
```

## Пример 1

Ввод 

market\_1.csv billing\_1.csv

Вывод 

```
order_id,shop_name,shop_id,cost
1,shop_name_1,1,100
2,shop_name_2,2,200
3,shop_name_3,3,300
4,shop_name_4,4,400
5,shop_name_5,5,500
```

## Пример 2

Ввод 

market\_1.csv billing\_2.csv

Вывод 

```
order_id,shop_name,shop_id,cost
1,shop_name_1,1,100
2,shop_name_2,2,200
3,shop_name_2,2,200
4,shop_name_2,2,200
5,shop_name_5,5,500
```

## Пример 3

Ввод 

market\_2.csv billing\_1.csv

Вывод 

```
order_id,shop_name,shop_id,cost
2,shop_name_2,2,200
```

## Пример 4

Ввод 

market\_2.csv billing\_2.csv

Вывод 

```
order_id,shop_name,shop_id,cost
2,shop_name_2,2,200
3,shop_name_2,2,200
4,shop_name_2,2,200
```

## Примечания

```
market_1.csv
shop_id,shop_name
1,shop_name_1
2,shop_name_2
3,shop_name_3
```

[Отправить](#)[Предыдущая](#)[Следующая](#)

4, 2, 200

5, 5, 500

© 2013–2021 ООО «Яндекс»

Язык Python 3.7 (PyPy 7.3.3)

[Набрать здесь](#)[Отправить файл](#)

```
1 import csv
2
3 f1, f2 = input().split()
4
5 market_file = open(f1, 'r')
6 billing_file = open(f2, 'r')
7 market_csv = csv.reader(market_file, delimiter=',')
8 billing_csv = csv.reader(billing_file, delimiter=',')
9
10 market_data = dict()
11
12 print('order_id,shop_name,shop_id,cost')
13 try:
14     market_csv.__next__()
15     billing_csv.__next__()
16     sm = market_csv.__next__()
17 except StopIteration:
18     exit(0)
19 for b in billing_csv:
20     shop_id = int(b[1])
21     while int(sm[0]) < shop_id:
22         try:
23             sm = market_csv.__next__()
24         except StopIteration:
25             exit(0)
26     if int(sm[0]) > shop_id:
27         continue
28     shop_name = sm[1]
29     print(','.join((b[0], shop_name, str(shop_id), b[-1])))
30
```