

Задача 3. sql join

Источник:	базовая
Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Многие приложения используют базы данных для постоянного хранения информации.

На сегодняшний день наиболее популярны реляционные базы данных. В реляционной базе данные представляются в виде таблиц. Каждая таблица состоит из произвольного набора записей, каждая запись занимает одну строку таблицы. Запись можно воспринимать как структуру языка C: в ней есть фиксированный набор полей фиксированного типа. Каждый столбец таблицы содержит значение одного конкретного поля для всех записей. Таким образом, в таблице строки задают отдельные записи, а столбцы — поля этих записей.

Для извлечения и фильтрации данных из реляционной базы данных чаще всего используют язык SQL. При этом нередко пользователю базы данных нужно составить сборный отчёт по нескольким таблицам. В таком случае можно использовать операции соединения (join) таблиц. В данной задаче предлагается реализовать операцию Inner Join для двух конкретных таблиц.

Первая таблица содержит биографии известных актёров кино. Она была создана следующей командой SQL:

```
CREATE TABLE ActorBio (  
    Name varchar(30),  
    BirthYear int,  
    Country varchar(10)  
);
```

Первый столбец называется `Name` и хранит имя актёра. Во втором записан год рождения как целое число. А третьем столбце записана страна, в которой жил актёр.

Вторая таблица содержит информацию о том, какой актёр в каких фильмах играл. Она была создана командой:

```
CREATE TABLE ActorInMovie (  
    ActorName varchar(30),  
    MovieName varchar(20)  
);
```

Первый столбец содержит имя актёра `ActorName`, а второй — название кино, в котором этот актёр играл роль.

От вас требуется реализовать следующий SQL-запрос:

```
SELECT *  
FROM ActorBio INNER JOIN ActorInMovie  
ON ActorBio.Name = ActorInMovie.ActorName
```

Результатом этой операции является одна таблица, в которой пять полей: первые три поля взяты из таблицы `ActorBio`, а последние два — из `ActorInMovie`.

Механизм выполнения операции следующий:

1. Перебираем все пары записей A и B , где A взята из таблицы `ActorBio`, а B — из таблицы `ActorInMovie`.

2. Для каждой пары проверяем условие соединения: что имя актёра **Name** в записи *A* совпадает с именем актёра **ActorName** в записи *B*.
3. Если условие выполнено, то конкатенируем записи *A* и *B* и полученную запись с пятью полями добавляем в таблицу-результат.

Чтобы было проще понять, как работает операция, крайне рекомендуется посмотреть первый пример к задаче.

Формат входных данных

В первой строке входного файла записано целое число N — количество записей в таблице **ActorBio** ($1 \leq N \leq 10^5$). Далее идёт N строк, в которых записаны записи таблицы **ActorBio**. Затем записано целое число M — количество записей в таблице **ActorInMovie** ($1 \leq M \leq 10^5$). Далее идёт M строк, в которых записаны записи таблицы **ActorInMovie**.

Для каждой записи в файле записываются все её поля через пробел, в том порядке, в котором они определены. Все записи кроме года рождения строковые: они окружены символом двойной кавычки (ASCII 34) с обеих сторон. Каждое строковое значение непустое и может содержать в себе любые печатные символы ASCII (коды от 32 до 126 включительно) кроме символа двойной кавычки. Год рождения записан как целое число. Имена актёров не длиннее 30 символов, названия стран не длиннее 10 символов, названия фильмов не длиннее 20 символов.

Формат выходных данных

Требуется вывести ровно K строк: записи, получившиеся в таблице-результате после соединения. Каждая строка должна описывать одну запись с пятью полями, в том же формате, в котором эти записи записаны во входных данных. Порядок записей в выходном файле может быть любой.

Гарантируется, что $N \cdot M \leq 10^5$.

Пример

input.txt
8 "Peter Falk" 1927 "USA" "Oleg Tabakov" 1935 "USSR" "Andrei Mironov" 1941 "USSR" "Arnold Schwarzenegger" 1947 "USA" "Jean Reno" 1948 "France" "Sharon Stone" 1958 "USA" "Tom Cruise" 1962 "USA" "Ryoko Hirosue" 1980 "Japan" 12 "Sharon Stone" "Basic Instinct" "Jean Reno" "Mission: Impossible" "Arnold Schwarzenegger" "Total Recall" "Tom Cruise" "Mission: Impossible" "Andrei Mironov" "Twelve Chairs" "Sharon Stone" "Total Recall" "Ryoko Hirosue" "Wasabi" "Arnold Schwarzenegger" "Terminator" "Jean Reno" "Wasabi" "Peter Falk" "Colombo" "Anatoli Papanov" "Twelve Chairs" "Jean Reno" "Leon"
output.txt
"Andrei Mironov" 1941 "USSR" "Andrei Mironov" "Twelve Chairs" "Arnold Schwarzenegger" 1947 "USA" "Arnold Schwarzenegger" "Total Recall" "Arnold Schwarzenegger" 1947 "USA" "Arnold Schwarzenegger" "Terminator" "Jean Reno" 1948 "France" "Jean Reno" "Leon" "Jean Reno" 1948 "France" "Jean Reno" "Mission: Impossible" "Jean Reno" 1948 "France" "Jean Reno" "Wasabi" "Peter Falk" 1927 "USA" "Peter Falk" "Colombo" "Ryoko Hirosue" 1980 "Japan" "Ryoko Hirosue" "Wasabi" "Sharon Stone" 1958 "USA" "Sharon Stone" "Basic Instinct" "Sharon Stone" 1958 "USA" "Sharon Stone" "Total Recall" "Tom Cruise" 1962 "USA" "Tom Cruise" "Mission: Impossible"

input.txt

```

5
"0]V| c -(SZ9mY ~'/{8" 1950 "bo"
" Q G*u4 T:Eqy'd" 1979 "9"
"0" 2005 "jMsB"
" w cQ" 1982 "&"
" Q G*u4 T:Eqy'd" 2003 " J|hL"
7
" Q G*u4 T:Eqy'd" "6q*EDh!"
"0" "s"
" Q G*u4 T:Eqy'd" "CQMG::dw{"
":D h%$ W~cr" "%'De!Si"
" Q G*u4 T:Eqy'd" "]"Zo"
" Q G*u4 T:Eqy'd" "t"
"0" "Uxb/.& "

```

output.txt

```

" Q G*u4 T:Eqy'd" 1979 "9" " Q G*u4 T:Eqy'd" "t"
" Q G*u4 T:Eqy'd" 1979 "9" " Q G*u4 T:Eqy'd" "CQMG::dw{"
" Q G*u4 T:Eqy'd" 1979 "9" " Q G*u4 T:Eqy'd" "]"Zo"
" Q G*u4 T:Eqy'd" 1979 "9" " Q G*u4 T:Eqy'd" "6q*EDh!"
" Q G*u4 T:Eqy'd" 2003 " J|hL" " Q G*u4 T:Eqy'd" "t"
" Q G*u4 T:Eqy'd" 2003 " J|hL" " Q G*u4 T:Eqy'd" "CQMG::dw{"
" Q G*u4 T:Eqy'd" 2003 " J|hL" " Q G*u4 T:Eqy'd" "]"Zo"
" Q G*u4 T:Eqy'd" 2003 " J|hL" " Q G*u4 T:Eqy'd" "6q*EDh!"
"0" 2005 "jMsB" "0" "Uxb/.& "
"0" 2005 "jMsB" "0" "s"

```