Mechanics of joining/merging two tables

Suman Sahil, Steve Simon

Creation date: 2021-07-24

Overview

- Mechanics of join/merge
- Efficiency issues
- One-to-many merge
- Mismatches

author: Suman Sahil, Steve Simon date created: 2021-07-24 purpose: to show the mechanics of some simple joins license: public domain

This talk will cover the basic mechanics of how you join data. It will use several simple artificial data sets.

Simple example of a join/merge

- Two tables/datasets
  - doctor_list
  - patient_list
- Key/link
  - Defines how to match records

Here is a simple illustration of the mechanics of joining data from two tables.

The volunteers table

```
##   v volunteer srg_yr
## 1 2     House   2004
## 2 7    Howser   1989
## 3 6      John   1979
## 4 3   Kildare   1961
## 5 1     McCoy   1966
## 6 5    Pierce   1972
## 7 8     Quinn   1993
## 8 4     Welby   1969
```

This is a listing of the volunteers for the first surgery by each doctor. The link variable is used to match the volunteers and the doctors.

The doctor table

```
##   d      doctor brth_yr
## 1 5        Alda    1936
## 2 3 Chamberlain    1934
## 3 7      Harris    1973
## 4 1      Kelley    1920
## 5 2      Laurie    1959
## 6 6     Roberts    1928
## 7 8     Seymour    1951
## 8 4       Young    1907
```

This is a listing of doctors in a research database and the year that they performed their first surgery. The data is fictitious, and fans of television and the movies may recognize some of these names.

Simple matching algorithm

- Compare
  - First record of volunteer table with
  - Each record of doctor table
- Repeat for second, third, … records of volunteer table

The simplest matching algorithm tries to link the first record of the doctor with each record of the volunteer table. It keeps only the records that have the same value for link.

It does the same for the second record of the doctor table, the third record, and so forth.

Simple matching algorithm, step 1

```
##   v volunteer srg_yr d      doctor brth_yr
## 1 5                 5        Alda    1936
## 2 3                 3 Chamberlain    1934
## 3 7                 7      Harris    1973
## 4 1                 1      Kelley    1920
## 5 2     House   2004 2      Laurie    1959
## 6 6                 6     Roberts    1928
## 7 8                 8     Seymour    1951
## 8 4                 4       Young    1907
```

The first row of the volunteers table is “House” and has a linking value of 2. The matching value in the doctor table is “Laurie”.

Simple matching algorithm, first row of join

```
##   v volunteer srg_yr d doctor brth_yr
## 1 2     House   2004 2 Laurie    1959
```

This match becomes the first row of join.

Simple matching algorithm, step 2

```
##   v volunteer srg_yr d      doctor brth_yr
## 1 5                  5        Alda    1936
## 2 3                  3 Chamberlain    1934
## 3 7    Howser   1989 7      Harris    1973
## 4 1                  1      Kelley    1920
## 5 2                  2      Laurie    1959
## 6 6                  6     Roberts    1928
## 7 8                  8     Seymour    1951
## 8 4                  4       Young    1907
```

The second row of the volunteer table is “Howser” with a linking value of 7. It matches with the “Harris” row of the doctor table.

Simple matching algorithm, first two rows of join

```
##   v volunteer srg_yr d doctor brth_yr
## 1 2     House   2004 2 Laurie    1959
## 2 7    Howser   1989 7 Harris    1973
```

This match gets added as the second row of the join.

Simple matching algorithm, step 3

```
##   v volunteer srg_yr d      doctor brth_yr
## 1 5                  5        Alda    1936
## 2 3                  3 Chamberlain    1934
## 3 7                  7      Harris    1973
## 4 1                  1      Kelley    1920
## 5 2                  2      Laurie    1959
## 6 6      John   1979 6     Roberts    1928
## 7 8                  8     Seymour    1951
## 8 4                  4       Young    1907
```

The third row of the volunteer table is “John” with a linking variable of 6. This matches with “Roberts” in the doctor table.

Simple matching algorithm, first three rows of join

```
##   v volunteer srg_yr d  doctor brth_yr
## 1 2     House   2004 2  Laurie    1959
## 2 7    Howser   1989 7  Harris    1973
## 3 6      John   1979 6 Roberts    1928
```

This becomes the third row of the join.

Simple matching algorithm, step 4

```
##   v volunteer srg_yr d      doctor brth_yr
## 1 5                 5        Alda    1936
## 2 3   Kildare   1961 3 Chamberlain    1934
## 3 7                 7      Harris    1973
## 4 1                 1      Kelley    1920
## 5 2                 2      Laurie    1959
## 6 6                 6     Roberts    1928
## 7 8                 8     Seymour    1951
## 8 4                 4       Young    1907
```

The fourth row of the volunteer table is “Kildare” with a linking value of 3. It matches “Chamberlain” in the doctor table.

Simple matching algorithm, first four rows of join

```
##   v volunteer srg_yr d      doctor brth_yr
## 1 2     House   2004 2      Laurie    1959
## 2 7    Howser   1989 7      Harris    1973
## 3 6      John   1979 6     Roberts    1928
## 4 3   Kildare   1961 3 Chamberlain    1934
```

This becomes the next row of the join.

Simple matching algorithm, final step

```
##   v volunteer srg_yr d      doctor brth_yr
## 1 5                  5        Alda    1936
## 2 3                  3 Chamberlain    1934
## 3 7                  7      Harris    1973
## 4 1                  1      Kelley    1920
## 5 2                  2      Laurie    1959
## 6 6                  6     Roberts    1928
## 7 8                  8     Seymour    1951
## 8 4     Welby   1969 4       Young    1907
```

Jump to the last row of the volunteer table, “Welby” with a linking value of 4. It matches “Young” in the doctor table.

Simple matching algorithm, the complete join

```
##   v volunteer srg_yr d      doctor brth_yr
## 1 2     House   2004 2      Laurie    1959
## 2 7    Howser   1989 7      Harris    1973
## 3 6      John   1979 6     Roberts    1928
## 4 3   Kildare   1961 3 Chamberlain    1934
## 5 1     McCoy   1966 1      Kelley    1920
## 6 5    Pierce   1972 5        Alda    1936
## 7 8     Quinn   1993 8     Seymour    1951
## 8 4     Welby   1969 4       Young    1907
```

The completes the join.

Break #1

- What you just learned
  - Simple matching algorithm
- What's next
  - Efficiency issues

Efficiency (1/2)

- How much work?
  - Eight steps
  - Eight comparisons within each step
  - 64 total steps
- Complications
  - When one or both tables do not fit in memory
  - NULL values

While efficiency is beyond the scope of this class, I do want to touch on the issue briefly. The algorithm shown here is the easiest to understand, and it is also easy to implement. It can sometimes be terribly slow and inefficient.

In this example, there were eight steps and each step required eight comparisons to find the right match. You can't stop once you find a match, because you don't know if there might be a second match. So there are 64 comparisons needed to join two tables with eight records each.

When tables in a database can have thousands or even millions of records, the number of comparisons can overwhelm even the fastest computer.

There are complications to consider. When one or both tables do not fit in memory, you have an added cost of bringing pieces of each table in and out.

NULL values complicate a join because you can't say what matches what when NULL values are included. Many databases are designed so that NULL values are not allowed on any fields that might be used to link two tables.

Efficiency (2/2)

- Improvements
  - Speed gains can often be substantial, but …
  - Change the order of the join
  - Sort before merging
  - Create a hash

Improvements of the basic join algorithms are often possible. At times this might mean the difference between a join that takes a few seconds verus one that takes a few hours. If your database is really big, then sometimes it might mean the difference between a few hours and a few days.

It is impossible to perfectly predict whether a more complex algorithm will perform better without running both algorithms side-by-side. This defeats the purpose of making an improvement, of course. There are general features of the tables being joined that can often indiciate with pretty good accuracy what approach is best. Nevertheless, picking the most efficient approach can sometimes be more of an art than a science.

Mathematically, joining table A to table B is identical to joining table B to table A. But computationally, one may be much slower than the other. This can happen when there is a large discrepancy in the size of the two tables being joined. Modern databases try to look at features of the join to predict the timing of joining A to B versus B to A and may choose to reverse the order.

Often, but not always, you can sort the two tables and then do the join. It takes time

to sort the table, but matching can go a lot faster.

For very large tables, you can use a hash function on the linking variables. A hash function takes an input value and creates a new value called a hash that has several desirable mathematical properties. The hash is always the same size as the input, often much smaller in size, and it tends to be distributed uniformly across the possible values. Matching hashes rather than the original linking variables can sometimes be more efficient.

As an end user, you don't need to worry too much about these issues, as they are handled behind the scenes by the database programmers.

Break #2

- What you just learned
  - Efficiency issues
- What's next
  - One-to-many join

An updated doctor table, one-to-many join

```
##    d      doctor brth_yr
## 1  5        Alda    1936
## 2  3       Ayres    1908
## 3  3 Chamberlain    1934
## 4  6       Gould    1938
## 5  7      Harris    1973
## 6  3     Jenkins    1943
## 7  1      Kelley    1920
## 8  2      Laurie    1959
## 9  3      McCrea    1905
## 10 6     Roberts    1928
## 11 8     Seymour    1951
## 12 1       Urban    1972
## 13 4       Young    1907
```

This is a listing of doctors in a research database and the year that they performed their first surgery. The data is fictitious, and fans of television and the movies may recognize some of these names.

One-to-many join, step 1

```
##    v volunteer srg_yr d      doctor brth_yr
## 1  5                  5       Alda    1936
## 2  3                  3       Ayres    1908
## 3  3                  3 Chamberlain    1934
## 4  6                  6       Gould    1938
## 5  7                  7      Harris    1973
## 6  3                  3     Jenkins    1943
## 7  1                  1      Kelley    1920
## 8  2     House   2004 2      Laurie    1959
## 9  3                  3      McCrea    1905
## 10 6                  6     Roberts    1928
## 11 8                  8     Seymour    1951
## 12 1                  1       Urban    1972
## 13 4                  4       Young    1907
```

The first step works just like before.

One-to-many join, first row of join

```
##   v volunteer srg_yr d doctor brth_yr
## 1 2     House   2004 2 Laurie    1959
```

Here's the first row of our join.

One-to-many join, step 2

```
##    v volunteer srg_yr d      doctor brth_yr
## 1  5                  5       Alda    1936
## 2  3                  3       Ayres    1908
## 3  3                  3 Chamberlain    1934
## 4  6                  6       Gould    1938
## 5  7    Howser   1989 7      Harris    1973
## 6  3                  3     Jenkins    1943
## 7  1                  1      Kelley    1920
## 8  2                  2      Laurie    1959
## 9  3                  3      McCrea    1905
## 10 6                  6     Roberts    1928
## 11 8                  8     Seymour    1951
## 12 1                  1       Urban    1972
## 13 4                  4       Young    1907
```

The second row also works the same.

One-to-many join, first two rows of join

```
##   v volunteer srg_yr d doctor brth_yr
## 1 2     House   2004 2 Laurie    1959
## 2 7    Howser   1989 7 Harris    1973
```

Here are the first two rows.

One-to-many join, step 3

```
##    v volunteer srg_yr d      doctor brth_yr
## 1  5                  5       Alda    1936
## 2  3                  3       Ayres    1908
## 3  3                  3 Chamberlain    1934
## 4  6       John   1979 6       Gould    1938
## 5  7                  7      Harris    1973
## 6  3                  3     Jenkins    1943
## 7  1                  1      Kelley    1920
## 8  2                  2      Laurie    1959
## 9  3                  3      McCrea    1905
## 10 6       John   1979 6     Roberts    1928
## 11 8                  8     Seymour    1951
## 12 1                  1       Urban    1972
## 13 4                  4       Young    1907
```

The third row is different. The volunteer “John” with a linking value of 6 matches both “Gould” and “Roberts” from the doctor table.

One-to-many join, first four rows of join

```
##   v volunteer srg_yr d  doctor brth_yr
## 1 2     House   2004 2  Laurie    1959
## 2 7    Howser   1989 7  Harris    1973
## 3 6      John   1979 6   Gould    1938
## 4 6      John   1979 6 Roberts    1928
```

This adds two rows to the join table.

One-to-many join, step 4

```
##    v volunteer srg_yr d      doctor brth_yr
## 1  5                  5       Alda    1936
## 2  3   Kildare   1961 3       Ayres    1908
## 3  3   Kildare   1961 3 Chamberlain    1934
## 4  6                  6       Gould    1938
## 5  7                  7      Harris    1973
## 6  3   Kildare   1961 3     Jenkins    1943
## 7  1                  1      Kelley    1920
## 8  2                  2      Laurie    1959
## 9  3   Kildare   1961 3      McCrea    1905
## 10 6                  6     Roberts    1928
## 11 8                  8     Seymour    1951
## 12 1                  1       Urban    1972
## 13 4                  4       Young    1907
```

We hit the jackpot here. The fourth row of the volunteer table, “Kildare” with a linking value of 3, matches “Ayres”, “Chamberlain”, “Jenkins”, and “McCrea” from the doctor table.

One-to-many join, first eight rows of join

```
##   v volunteer srg_yr d      doctor brth_yr
## 1 2     House   2004 2      Laurie    1959
## 2 7    Howser   1989 7      Harris    1973
## 3 6      John   1979 6       Gould    1938
## 4 6      John   1979 6     Roberts    1928
## 5 3   Kildare   1961 3       Ayres    1908
## 6 3   Kildare   1961 3 Chamberlain    1934
## 7 3   Kildare   1961 3     Jenkins    1943
## 8 3   Kildare   1961 3      McCrea    1905
```

This adds four rows to the join.

One-to-many join, the complete join

```
##    v volunteer srg_yr d      doctor brth_yr
## 1  2     House   2004 2      Laurie    1959
## 2  7    Howser   1989 7      Harris    1973
## 3  6      John   1979 6       Gould    1938
## 4  6      John   1979 6     Roberts    1928
## 5  3   Kildare   1961 3       Ayres    1908
## 6  3   Kildare   1961 3 Chamberlain    1934
## 7  3   Kildare   1961 3     Jenkins    1943
## 8  3   Kildare   1961 3      McCrea    1905
## 9  1     McCoy   1966 1      Kelley    1920
## 10 1     McCoy   1966 1       Urban    1972
## 11 5    Pierce   1972 5        Alda    1936
## 12 8     Quinn   1993 8     Seymour    1951
## 13 4     Welby   1969 4       Young    1907
```

Here's what the complete join looks like.

Break #3

- What you just learned
  - One-to-many join
- What's next
  - What to do with mismatches

Another update, what to do with mismatches

```
##   v volunteer srg_yr
## 1 2     House   2004
## 2 7    Howser   1989
## 3 6      John   1979
## 4 3   Kildare   1961
## 5 1     McCoy   1966
## 6 5    Pierce   1972
## 7 8     Quinn   1993
## 8 4     Welby   1969
## 9 9       Who   1999
```

Let's put in a new wrinkle. Suppose there is a volunteer in this database who does not have a matchup with ANY of the doctors. The last row of this dataset is a volunter named "Who" and this "Who" does not have a matching doctor. Let's see what happens.

Mismatched record

```
##    v volunteer srg_yr d      doctor brth_yr
## 1                     5        Alda    1936
## 2                     3       Ayres    1908
## 3                     3 Chamberlain    1934
## 4                     6       Gould    1938
## 5                     7      Harris    1973
## 6                     3     Jenkins    1943
## 7                     1      Kelley    1920
## 8                     2      Laurie    1959
## 9                     3      McCrea    1905
## 10                    6     Roberts    1928
## 11                    8     Seymour    1951
## 12                    1       Urban    1972
## 13                    4       Young    1907
## 14 9       Who   1999
```

Jump straight to the volunteer “who” with a linking value of 9. There is no match at all in the doctor table. This leaves you with two choices.

First solution, discard the mismatch

```
##    v volunteer srg_yr d      doctor brth_yr
## 1  2     House   2004 2      Laurie    1959
## 2  7    Howser   1989 7      Harris    1973
## 3  6      John   1979 6       Gould    1938
## 4  6      John   1979 6     Roberts    1928
## 5  3   Kildare   1961 3       Ayres    1908
## 6  3   Kildare   1961 3 Chamberlain    1934
## 7  3   Kildare   1961 3     Jenkins    1943
## 8  3   Kildare   1961 3      McCrea    1905
## 9  1     McCoy   1966 1      Kelley    1920
## 10 1     McCoy   1966 1       Urban    1972
## 11 5    Pierce   1972 5        Alda    1936
## 12 8     Quinn   1993 8     Seymour    1951
## 13 4     Welby   1969 4       Young    1907
```

The first choice is to discard the mismatch. The volunteers from “House” to “Welby” are included in the join table, but “Who” is not.

Second solution, include the mismatch

```
##    v volunteer srg_yr    d      doctor brth_yr
## 1  2     House   2004    2      Laurie    1959
## 2  7    Howser   1989    7      Harris    1973
## 3  6      John   1979    6       Gould    1938
## 4  6      John   1979    6     Roberts    1928
## 5  3   Kildare   1961    3       Ayres    1908
## 6  3   Kildare   1961    3 Chamberlain    1934
## 7  3   Kildare   1961    3     Jenkins    1943
## 8  3   Kildare   1961    3      McCrea    1905
## 9  1     McCoy   1966    1      Kelley    1920
## 10 1     McCoy   1966    1       Urban    1972
## 11 5    Pierce   1972    5        Alda    1936
## 12 8     Quinn   1993    8     Seymour    1951
## 13 4     Welby   1969    4       Young    1907
## 14 9       Who   1999 NULL        NULL    NULL
```

The second choice is to include the mismatch, and fill the values in the doctor table with NULLs. The choice you make depends largely on the context of the problem.

Summary

- Mechanics of join/merge
- Efficiency issues
- One-to-many merge
- Mismatches

There's a lot more to cover. But you learned the basic algorithm for the one-to-one merge, how it works just as well for the one-to-many join, and the two approaches to handling mismatches. We also snuck in some information about computational efficiency.