Before we merge our data, we'll need to decide on the merge strategy we want to use. We'll be using the pandas [pandas.DataFrame.merge()](http://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.merge.html" \t "_blank) function, which supports four types of joins -- left, right, inner, and outer. Each of these join types dictates how pandas combines the rows.

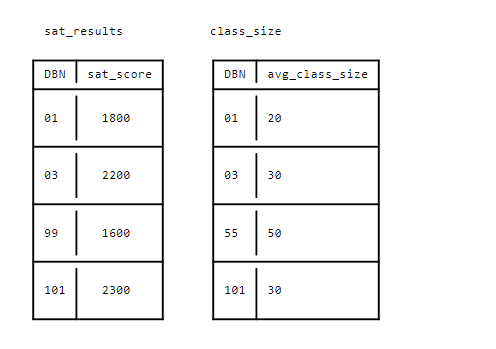
We'll be using the DBN column to identify matching rows across data sets. In other words, the values in that column will help us know which row from the first data set to combine with which row in the second data set.

There may be DBN values that exist in one data set but not in another. This is partly because the data is from different years. Each data set also has inconsistencies in terms of how it was gathered. Human error (and other types of errors) may also play a role. Therefore, we may not find matches for the DBN values in sat\_results in all of the other data sets, and other data sets may have DBN values that don't exist in sat\_results.

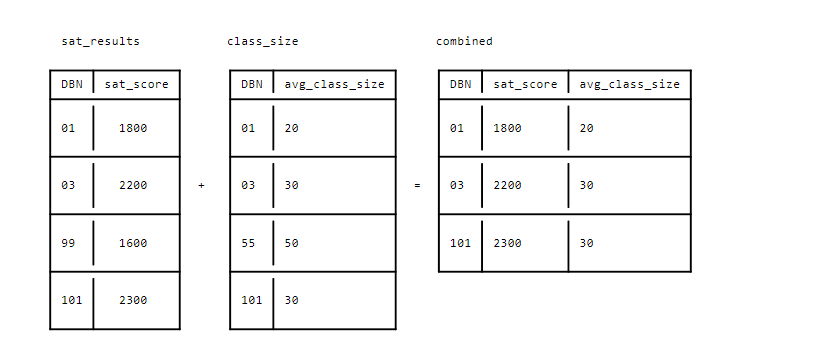
We'll merge two data sets at a time. For example, we'll merge sat\_results and hs\_directory, then merge the result with ap\_2010, then merge the result of that with class\_size. We'll continue combining data sets in this way until we've merged all of them. Afterwards, we'll have roughly the same number of rows, but each row will have columns from all of the data sets.

The merge strategy we pick will affect the number of rows we end up with. Let's take a look at each strategy.

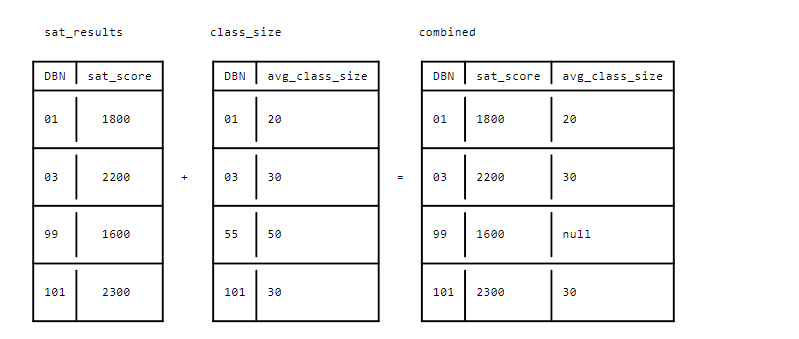
Let's say we're merging the following two data sets:



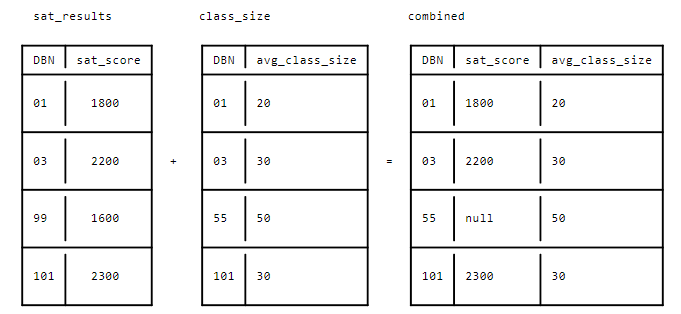
With an inner merge, we'd only combine rows where the same DBN exists in both data sets. We'd end up with this result:



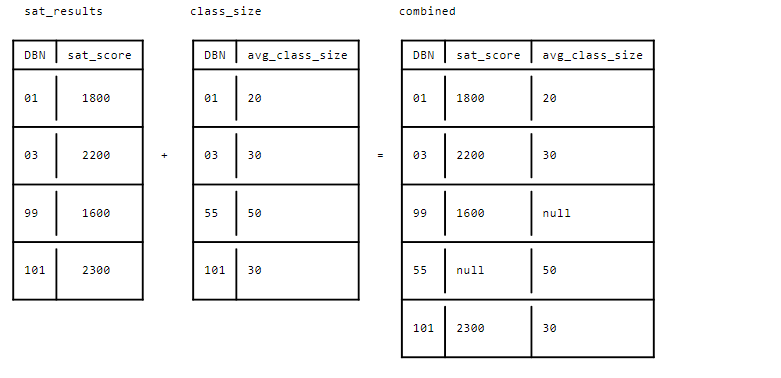
With a left merge, we'd only use DBN values from the dataframe on the "left" of the merge. In this case, sat\_resultsis on the left. Some of the DBNs in sat\_results don't exist in class\_size, though. The merge will handle this by assiging null values to the columns in sat\_results that don't have corresponding data in class\_size.



With a right merge, we'll only use DBN values from the dataframe on the "right" of the merge. In this case, class\_size is on the right:



With an outer merge, we'll take any DBN values from either sat\_results or class\_size:



As you can see, each merge strategy has its advantages. Depending on the strategy we choose, we may preserve rows at the expense of having more missing column data, or minimize missing data at the expense of having fewer rows. Choosing a merge strategy is an important decision; it's worth thinking about your data carefully, and what trade-offs you're willing to make.

Because this project is concerned with determing demographic factors that correlate with SAT score, we'll want to preserve as many rows as possible from sat\_results while minimizing null values.

This means that we may need to use different merge strategies with different data sets. Some of the data sets have a lot of missing DBN values. This makes a left join more appropriate, because we don't want to lose too many rows when we merge. If we did an inner join, we would lose the data for many high schools.

Some data sets have DBN values that are almost identical to those in sat\_results. Those data sets also have information we need to keep. Most of our analysis would be impossible if a significant number of rows was missing from demographics, for example. Therefore, we'll do an inner join to avoid missing data in these columns.