4.7.1 Merge DataFrames

As you sit down to get started on your work, Maria presents your task for today. She tells you that the first report you need to generate is the school district summary, which will be based on several key metrics.

The school district summary will be a high-level snapshot of the district's key metrics:

- Total number of students
- Total number of schools
- Total budget
- Average math score
- Average reading score
- Percentage of students who passed math
- Percentage of students who passed reading
- Overall passing percentage

We'll find this information and visualize the data with a table like the following:

	Total Schools	Total Students	Total Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing
0	15	39,170	$24,649,428.00	79.0	81.9	75	86	80

In order to get all of this data organized in one table, we'll need to merge the two DataFrames and perform analysis on the single, merged DataFrame. Although we'll be merging the DataFrames, it may be more efficient to use either `school_data_df` or `student_data_df` when performing certain calculations.

IMPORTANT

> Even though we can perform calculations on both DataFrames to get the information we need, it's a best practice to create a new DataFrame to do calculations. This way, the original data is not affected.

To merge two DataFrames, there must be a column in each of the DataFrames with the same name. So before we merge, let's review the column names in each DataFrame.

The columns in `school_data_df` are:

- School ID
- school_name
- type
- size
- budget

The columns in the `student_data_df` are:

- Student ID
- student_name
- gender

- grade
- school_name
- reading_score
- math_score

We'll merge `school_data_df` and `student_data_df` on a shared column using the `merge()` method. The column that these DataFrames have in common is `school_name`. Inside the parentheses of the `merge()` method, we'll do the following:

- Add the DataFrames to be merged.
- Add the shared column to each DataFrame so that the merge can occur.
- Define how the DataFrames should be merged: left, right, inner, or outer. The default is inner. (You will learn more about merging later in this course.)

NOTE

> For more information, see the following documentation:
>
> - **Merging Pandas DataFrames (https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.merge.html)**
> - **Merging, joining, and concatenating Pandas DataFrames (https://pandas.pydata.org/pandas-docs/stable/user_guide/merging.html)**

In a new cell, add the following code and run the cell.

```
# Combine the data into a single dataset.
school_data_complete_df = pd.merge(student_data_df, school_data_df, on=["sch
school_data_complete_df.head()
```

Let's break down how the DataFrames are being merged with this code.

- We create a new DataFrame for the merged DataFrames, called `school_data_complete_df`.
- The new DataFrame is created as a result of merging DataFrame #2 (`school_data_df`), which is the "right" DataFrame, into DataFrame #1 (`student_data_df`), which is the "left" DataFrame. We refer to the DataFrames as "left" and "right" to reflect the order they appear inside the parentheses.
- We use the parameter "on," which is equal to a list of the columns that are identical from each DataFrame, in this case, "school_name." We can also use the column name like this: `on="school_name"`.

IMPORTANT

> There may be cases in which you want to merge on a column that has similar information in two separate DataFrames, but is named differently in each—for example, "school_name" in one DataFrame and "high_school" in the second. In these cases, you should rename the columns so that they match. This will help avoid duplicate columns or merging issues.

After you run the cell, the output should look like this:

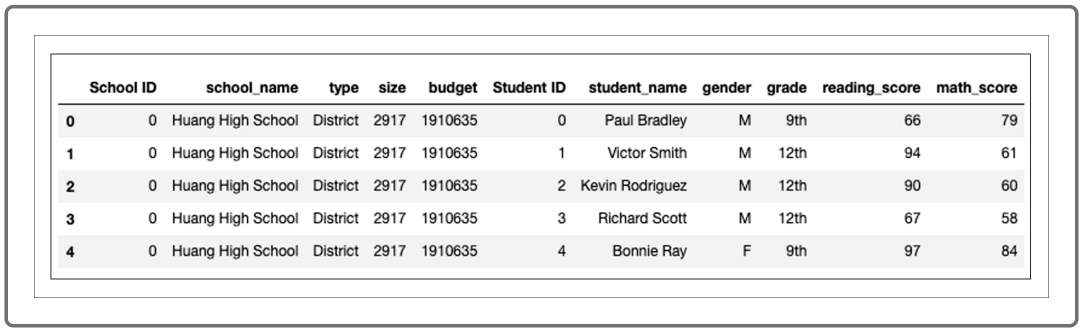
	Student ID	student_name	gender	grade	school_name	reading_score	math_score	School ID	type	size	budget
0	0	Paul Bradley	M	9th	Huang High School	66	79	0	District	2917	1910635
1	1	Victor Smith	M	12th	Huang High School	94	61	0	District	2917	1910635
2	2	Kevin Rodriguez	M	12th	Huang High School	90	60	0	District	2917	1910635
3	3	Richard Scott	M	12th	Huang High School	67	58	0	District	2917	1910635
4	4	Bonnie Ray	F	9th	Huang High School	97	84	0	District	2917	1910635

The data from `student_data_df` takes up the first seven columns, and the four columns from the `school_data_df` are added at the end: School ID,

type, size, and budget.

CAUTION

Did you get output that looks like this?



	School ID	school_name	type	size	budget	Student ID	student_name	gender	grade	reading_score	math_score
0	0	Huang High School	District	2917	1910635	0	Paul Bradley	M	9th	66	79
1	0	Huang High School	District	2917	1910635	1	Victor Smith	M	12th	94	61
2	0	Huang High School	District	2917	1910635	2	Kevin Rodriguez	M	12th	90	60
3	0	Huang High School	District	2917	1910635	3	Richard Scott	M	12th	67	58
4	0	Huang High School	District	2917	1910635	4	Bonnie Ray	F	9th	97	84

If so, this is probably because the "left" DataFrame was `school_data_df` and the "right" DataFrame was `student_data_df` inside the `merge()` method. You'll need to reverse the order of the DataFrames and run your code again to avoid errors later.

Now that we have our merged DataFrame, we can start getting some of the school district's key metrics.