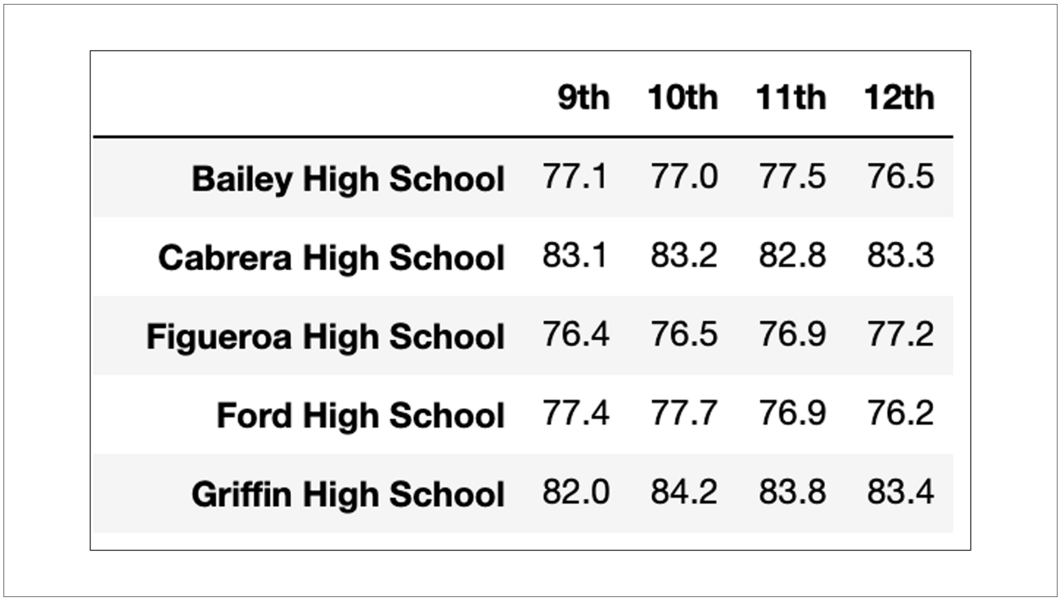
4.10.1 Create Grade-Level DataFrames

There's still some left work to do for the school district data analysis. Maria wants you to dig a little deeper into each school's metadata and get the average math and reading scores for each school. But first, you will need to create DataFrames for each grade level.

You are tasked with creating two new DataFrames: one that will display the average math scores for each grade, and the one that will display the average reading scores for each grade. These will be grouped by the name of the school.

In the final DataFrame, we need to have the index on the school name and the average math or reading score for each grade: 9th, 10th, 11th, and 12th. This is what the DataFrame for the average math scores should look like when we're done:



	9th	10th	11th	12th
Bailey High School	77.1	77.0	77.5	76.5
Cabrera High School	83.1	83.2	82.8	83.3
Figueroa High School	76.4	76.5	76.9	77.2
Ford High School	77.4	77.7	76.9	76.2
Griffin High School	82.0	84.2	83.8	83.4

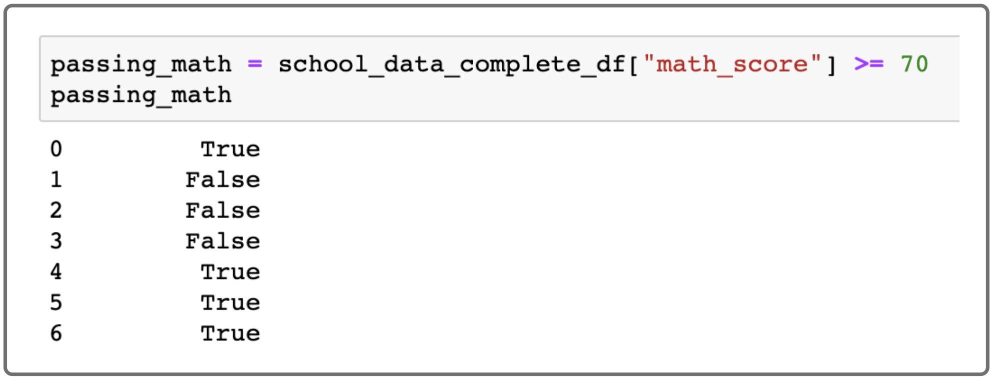
In this DataFrame, each grade—9th, 10th, 11th and 12th—is a Series of data. To get each grade as a Series, we need to filter `school_data_complete_df` for each grade.

REWIND

To get the passing scores for the district and school summary, we used the following code:

```
passing_math = school_data_complete_df["math_score"] >= 70
```

When we executed this code, a Series was returned where rows with a score equal to or greater than 70 were "True," and rows with scores not equal to or greater than 70 were "False." See the following image:



```
passing_math = school_data_complete_df["math_score"] >= 70
passing_math
```

```
0         True
1        False
2        False
3        False
4         True
5         True
6         True
```

We can use this same principle to get the grade level for each school in the `school_data_complete_df` DataFrame.

Let's first get the ninth graders in a Series. Set the variable `ninth_graders` equal to `school_data_complete_df["grade"] == "9th"`. This will retrieve all rows where this statement is "True."

Using the code `school_data_complete_df["grade"] == "9th"`, we can filter `school_data_complete_df` for the "True" cases as follows:

`school_data_complete_df[(school_data_complete_df["grade"] == "9th")]`

Now repurpose this code for each grade by replacing `"9th"` with each grade. Assign a grade-level variable to each grade-level DataFrame; for example, `ninth_graders` is for the ninth grade data.

Add the following code to a new cell and run the cell.

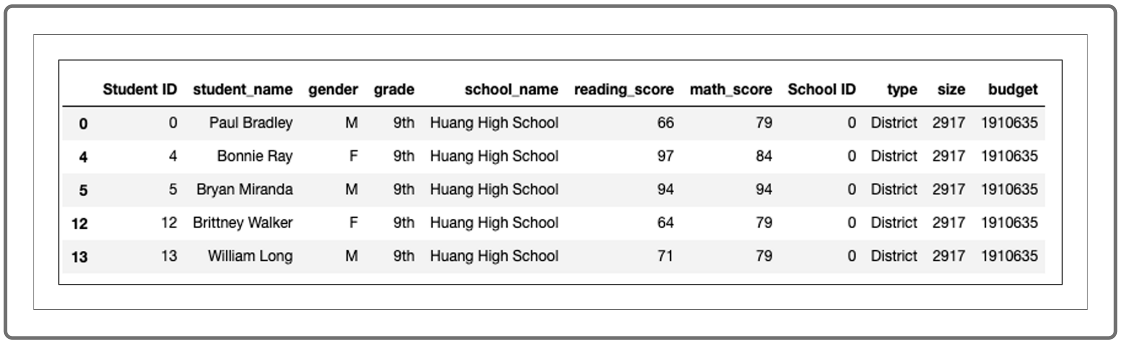
```
# Create a grade level DataFrames.
ninth_graders = school_data_complete_df[(school_data_complete_df["grade"] ==

tenth_graders = school_data_complete_df[(school_data_complete_df["grade"] ==

eleventh_graders = school_data_complete_df[(school_data_complete_df["grade"]

twelfth_graders = school_data_complete_df[(school_data_complete_df["grade"]
```

When we print out the first five rows for the `ninth_graders`, the DataFrame 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
4	4	Bonnie Ray	F	9th	Huang High School	97	84	0	District	2917	1910635
5	5	Bryan Miranda	M	9th	Huang High School	94	94	0	District	2917	1910635
12	12	Brittney Walker	F	9th	Huang High School	64	79	0	District	2917	1910635
13	13	William Long	M	9th	Huang High School	71	79	0	District	2917	1910635

Now that we have the data for each grade level, we need to get the grade averages for each school.