Note

- Instructions have been included for each segment. You do not have to follow them exactly, but they are included to help you think through the steps.

In [1]:

```
# Dependencies and Setup
import pandas as pd

# File to Load (1 directory up in Resources folder where the jupyter notebook is running)
school_data_to_load = "../Resources/schools_complete.csv"
student_data_to_load = "../Resources/students_complete.csv"

# Read School and Student Data File and store into Pandas Data Frames
school_data = pd.read_csv(school_data_to_load)
student_data = pd.read_csv(student_data_to_load)

# Combine the data into a single dataset
school_data_complete = pd.merge(student_data, school_data, how="left", on=["school_name"])

school_data_complete.head()
```

Out[1]:

	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	Dr. 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

District Summary

- Calculate the total number of schools

- Calculate the total number of students
- Calculate the total budget
- Calculate the average math score
- Calculate the average reading score
- Calculate the overall passing rate (overall average score), i.e. (avg. math score + avg. reading score)/2
- Calculate the percentage of students with a passing math score (70 or greater)
- Calculate the percentage of students with a passing reading score (70 or greater)
- Create a dataframe to hold the above results
- Optional: give the displayed data cleaner formatting

In [2]:

```
#Total number of schools
Total_Schools = school_data["school_name"].count()

#Total number of students
Total_Students = student_data["Student ID"].count()

#Total Budget
Total_Budget = school_data["budget"].sum()

#Average Math Score
Average_Math_score = student_data["math_score"].mean()

#Average Reading Score
Average_Reading_score = student_data["reading_score"].mean()

#overall average score
overall_average_score = (Average_Math_score+Average_Reading_score)/2

# %passing Math
Total_Passed_math = student_data.loc[(student_data["math_score"] >= 70)]
Percent_Passing_Math = (Total_Passed_math["Student ID"].count())*100/Total_Students

# %passing reading
Total_Passed_Reading = student_data.loc[(student_data["reading_score"] >= 70)]
Percent_Passing_reading = (Total_Passed_Reading["Student ID"].count())*100/Total_Students

#Data Frame to hold the summary values
District_summary_df = pd.DataFrame({"Total Schools":[Total_Schools],
                                   "Total Students":[Total_Students],
                                   "Total Budget":[Total_Budget],
                                   "Average Math Score":[Average_Math_score],
                                   "Average Reading Score":[Average_Reading_score],
                                    "% Passing Math":[Percent_Passing_Math],
                                    "% Passing Reading":[Percent_Passing_reading],
                                    "% Overall Passing Rate":[overall_average_score]})

#formatting values
District_summary_df["Total Students"] = District_summary_df["Total Students"].map("{:,}".format)

District_summary_df["Total Budget"] = District_summary_df["Total Budget"].astype(float).map("${:,.2f}".format)

District_summary_df
```

Out[2]:

	Total Schools	Total Students	Total Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing Rate
0	15	39,170	$24,649,428.00	78.985371	81.87784	74.980853	85.805463	80.431606

School Summary

- Create an overview table that summarizes key metrics about each school, including:
    - School Name
    - School Type
    - Total Students
    - Total School Budget
    - Per Student Budget
    - Average Math Score
    - Average Reading Score
    - % Passing Math
    - % Passing Reading
    - Overall Passing Rate (Average of the above two)
- Create a dataframe to hold the above results

Top Performing Schools (By Passing Rate)

- Sort and display the top five schools in overall passing rate

In [3]:

```
#School data grouped by School name
school_data_grouped = school_data_complete.groupby("school_name")

#School_Type - get the first value in the group, since one school will have only one type.
School_Type = school_data_grouped["type"].first()

#Total Students
Total_Students_by_school = school_data_grouped["Student ID"].count()

#Total School Budget - get the first value in the group, since one school will have a single value for budget
Total_School_Budget = school_data_grouped["budget"].first()

#Per Student Budget
Per_Student_budget = Total_School_Budget/Total_Students_by_school

#Average Math Score
Average_Math_score = school_data_grouped["math_score"].mean()

#Average Reading Score
Average_Reading_score = school_data_grouped["reading_score"].mean()

# %passing Math
All_Math_Passed = school_data_complete.loc[(school_data_complete["math_score"] >= 70)]
All_Math_Passed_by_school = All_Math_Passed.groupby(['school_name'])
Percent_Passing_Math_by_school = (All_Math_Passed_by_school["Student ID"].count()*100)/Total_Students_by_school

# %passing Reading
All_Reading_Passed = school_data_complete.loc[(school_data_complete["reading_score"] >= 70)]
All_Reading_Passed_by_school = All_Reading_Passed.groupby(['school_name'])
Percent_Passing_Reading_by_school = (All_Reading_Passed_by_school["Student ID"].count()*100)/Total_Students_by_s

# Overall Passing rate
Overall_Passing_rate = (Percent_Passing_Math_by_school+Percent_Passing_Reading_by_school)/2

#Data Frame to hold the School summary values
School_summary_df = pd.DataFrame({"School Type":School_Type,
                                  "Total Students":Total_Students_by_school,
                                  "Total School Budget":Total_School_Budget,
                                  "Per Student Budget":Per_Student_budget,
                                  "Average Math Score":Average_Math_score,
                                  "Average Reading Score":Average_Reading_score,
                                  "% Passing Math":Percent_Passing_Math_by_school,
                                  "% Passing Reading":Percent_Passing_Reading_by_school,
                                  "% Overall Passing Rate":Overall_Passing_rate})


#Creating a Copy of the School summary for formatting changes
formatted_school_summary = School_summary_df.copy()

formatted_school_summary.index.name = None
formatted_school_summary["Per Student Budget"] = formatted_school_summary["Per Student Budget"].astype(float).ma
formatted_school_summary["Total School Budget"] = formatted_school_summary["Total School Budget"].astype(float).

formatted_school_summary.head()

#Sorting by OverAll Passing rate and displaying top 5 performing schools

Top_Performing_schools = formatted_school_summary.sort_values("% Overall Passing Rate", ascending=False)
Top_Performing_schools.head(5)
```

Out[3]:

	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing Rate
Cabrera High School	Charter	1858	$1,081,356.00	$582.00	83.061895	83.975780	94.133477	97.039828	95.586652
Thomas High School	Charter	1635	$1,043,130.00	$638.00	83.418349	83.848930	93.272171	97.308869	95.290520
Pena High School	Charter	962	$585,858.00	$609.00	83.839917	84.044699	94.594595	95.945946	95.270270
Griffin High School	Charter	1468	$917,500.00	$625.00	83.351499	83.816757	93.392371	97.138965	95.265668
Wilson High School	Charter	2283	$1,319,574.00	$578.00	83.274201	83.989488	93.867718	96.539641	95.203679

Bottom Performing Schools (By Passing Rate)

- Sort and display the five worst-performing schools

In [4]:

```
#Sorting by OverAll Passing rate and displaying bottom 5 performing schools
Bottom_Performing_schools = formatted_school_summary.sort_values("% Overall Passing Rate", ascending=True)
Bottom_Performing_schools.head(5)
```

Out[4]:

	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing Rate
Rodriguez High School	District	3999	$2,547,363.00	$637.00	76.842711	80.744686	66.366592	80.220055	73.293323
Figueroa High School	District	2949	$1,884,411.00	$639.00	76.711767	81.158020	65.988471	80.739234	73.363852
Huang High School	District	2917	$1,910,635.00	$655.00	76.629414	81.182722	65.683922	81.316421	73.500171
Johnson High School	District	4761	$3,094,650.00	$650.00	77.072464	80.966394	66.057551	81.222432	73.639992
Ford High School	District	2739	$1,763,916.00	$644.00	77.102592	80.746258	68.309602	79.299014	73.804308

Math Scores by Grade

- Create a table that lists the average Reading Score for students of each grade level (9th, 10th, 11th, 12th) at each school.
  - Create a pandas series for each grade. Hint: use a conditional statement.
  - Group each series by school
  - Combine the series into a dataframe
  - Optional: give the displayed data cleaner formatting

In [5]:

```
# Average Math Score for 9th Grade
School_math_score_9th = school_data_complete.loc[(school_data_complete["grade"] == "9th")]
School_math_score_9th_by_school = School_math_score_9th.groupby(['school_name'])
School_math_score_average_9th = School_math_score_9th_by_school["math_score"].mean()

# Average Math Score for 10th Grade
School_math_score_10th = school_data_complete.loc[(school_data_complete["grade"] == "10th")]
School_math_score_10th_by_school = School_math_score_10th.groupby(['school_name'])
School_math_score_average_10th = School_math_score_10th_by_school["math_score"].mean()

# Average Math Score for 11th Grade
School_math_score_11th = school_data_complete.loc[(school_data_complete["grade"] == "11th")]
School_math_score_11th_by_school = School_math_score_11th.groupby(['school_name'])
School_math_score_average_11th = School_math_score_11th_by_school["math_score"].mean()


# Average Math Score for 11th Grade
School_math_score_12th = school_data_complete.loc[(school_data_complete["grade"] == "12th")]
School_math_score_12th_by_school = School_math_score_12th.groupby(['school_name'])
School_math_score_average_12th = School_math_score_12th_by_school["math_score"].mean()


# Creating a dataframe to hold Math Average score results

Math_Scores_by_Grade_df = pd.DataFrame({"9th":School_math_score_average_9th,
                                        "10th":School_math_score_average_10th,
                                        "11th":School_math_score_average_11th,
                                        "12th":School_math_score_average_12th})
#Formatting
Math_Scores_by_Grade_df.index.name = None
Math_Scores_by_Grade_df
```

Out[5]:

	9th	10th	11th	12th
Bailey High School	77.083676	76.996772	77.515588	76.492218
Cabrera High School	83.094697	83.154506	82.765560	83.277487
Figueroa High School	76.403037	76.539974	76.884344	77.151369
Ford High School	77.361345	77.672316	76.918058	76.179963
Griffin High School	82.044010	84.229064	83.842105	83.356164

	9th	10th	11th	12th
Hernandez High School	77.438495	77.337408	77.136029	77.186567
Holden High School	83.787402	83.429825	85.000000	82.855422
Huang High School	77.027251	75.908735	76.446602	77.225641
Johnson High School	77.187857	76.691117	77.491653	76.863248
Pena High School	83.625455	83.372000	84.328125	84.121547
Rodriguez High School	76.859966	76.612500	76.395626	77.690748
Shelton High School	83.420755	82.917411	83.383495	83.778976
Thomas High School	83.590022	83.087886	83.498795	83.497041
Wilson High School	83.085578	83.724422	83.195326	83.035794
Wright High School	83.264706	84.010288	83.836782	83.644986

Reading Score by Grade

- Perform the same operations as above for reading scores

In [6]:

```
# Average Reading Score for 9th Grade
School_reading_score_9th = school_data_complete.loc[(school_data_complete["grade"] == "9th")]
School_reading_score_9th_by_school = School_reading_score_9th.groupby(['school_name'])
School_reading_score_average_9th = School_reading_score_9th_by_school["reading_score"].mean()


# Average Reading Score for 10th Grade
School_reading_score_10th = school_data_complete.loc[(school_data_complete["grade"] == "10th")]
School_reading_score_10th_by_school = School_reading_score_10th.groupby(['school_name'])
School_reading_score_average_10th = School_reading_score_10th_by_school["reading_score"].mean()

# Average Reading Score for 11th Grade
School_reading_score_11th = school_data_complete.loc[(school_data_complete["grade"] == "11th")]
School_reading_score_11th_by_school = School_reading_score_11th.groupby(['school_name'])
School_reading_score_average_11th = School_reading_score_11th_by_school["reading_score"].mean()


# Average Reading Score for 11th Grade
School_reading_score_12th = school_data_complete.loc[(school_data_complete["grade"] == "12th")]
School_reading_score_12th_by_school = School_reading_score_12th.groupby(['school_name'])
School_reading_score_average_12th = School_reading_score_12th_by_school["reading_score"].mean()


# Creating a dataframe to hold Reading Average score results

reading_Scores_by_Grade_df = pd.DataFrame({"9th":School_reading_score_average_9th,
                                        "10th":School_reading_score_average_10th,
                                        "11th":School_reading_score_average_11th,
                                        "12th":School_reading_score_average_12th})
#Formatting
reading_Scores_by_Grade_df.index.name = None
reading_Scores_by_Grade_df
```

Out[6]:

	9th	10th	11th	12th
Bailey High School	81.303155	80.907183	80.945643	80.912451
Cabrera High School	83.676136	84.253219	83.788382	84.287958
Figueroa High School	81.198598	81.408912	80.640339	81.384863

	9th	10th	11th	12th
Ford High School	80.632653	81.262712	80.403642	80.662338
Griffin High School	83.369193	83.706897	84.288089	84.013699
Hernandez High School	80.866860	80.660147	81.396140	80.857143
Holden High School	83.677165	83.324561	83.815534	84.698795
Huang High School	81.290284	81.512386	81.417476	80.305983
Johnson High School	81.260714	80.773431	80.616027	81.227564
Pena High School	83.807273	83.612000	84.335938	84.591160
Rodriguez High School	80.993127	80.629808	80.864811	80.376426
Shelton High School	84.122642	83.441964	84.373786	82.781671
Thomas High School	83.728850	84.254157	83.585542	83.831361
Wilson High School	83.939778	84.021452	83.764608	84.317673
Wright High School	83.833333	83.812757	84.156322	84.073171

Scores by School Spending

- Create a table that breaks down school performances based on average Spending Ranges (Per Student). Use 4 reasonable bins to group school spending. Include in the table each of the following:
  - Average Math Score
  - Average Reading Score
  - % Passing Math
  - % Passing Reading
  - Overall Passing Rate (Average of the above two)

In [7]:

```python
# Sample bins. Feel free to create your own bins.
spending_bins = [0, 585, 615, 645, 675]
group_names = ["<$585", "$585-615", "$615-645", "$645-675"]
```

In [8]:

```
# categorize the school summary dataframe by adding  Spending Ranges (Per Student) and use bins
School_summary_df["Spending Ranges (Per Student)"]= pd.cut(School_summary_df["Per Student Budget"],spending_bins


# Creating a new data frame to hold Average scores from the School Summary dataFrame
Scores_with_school_spending = School_summary_df.loc[:,["Spending Ranges (Per Student)",
                                                       "Average Math Score",
                                                       "Average Reading Score",
                                                       "% Passing Math",
                                                       "% Passing Reading",
                                                       "% Overall Passing Rate"]]

# Scores_grouped_by_Sepndig_bins
Scores_grouped_by_Spending_range = Scores_with_school_spending.groupby("Spending Ranges (Per Student)")

#school performances based on average Spending Ranges (Per Student)
Scores_grouped_by_Spending_range.mean()
```

Out[8]:

	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing Rate
Spending Ranges (Per Student)					
<$585	83.455399	83.933814	93.460096	96.610877	95.035486
$585-615	83.599686	83.885211	94.230858	95.900287	95.065572
$615-645	79.079225	81.891436	75.668212	86.106569	80.887391
$645-675	76.997210	81.027843	66.164813	81.133951	73.649382

Scores by School Size

- Perform the same operations as above, based on school size.

In [9]:

```python
# Sample bins. Feel free to create your own bins.
size_bins = [0, 1000, 2000, 5000]
group_names = ["Small (<1000)", "Medium (1000-2000)", "Large (2000-5000)"]
```

In [10]:

```python
# categorize the school summary dataframe by adding  School Size Range and use bins,
#based on Total Students which is same as the school size
School_summary_df["School Size"]= pd.cut(School_summary_df["Total Students"],size_bins,labels=group_names)

# Creating a new data frame to hold Average scores from the School Summary dataFrame with School Size
Scores_with_school_size = School_summary_df.loc[:,["School Size",
                                                     "Average Math Score",
                                                     "Average Reading Score",
                                                     "% Passing Math",
                                                     "% Passing Reading",
                                                     "% Overall Passing Rate"]]

#Scores_grouped_by_school_size
Scores_grouped_by_school_size = Scores_with_school_size.groupby("School Size")

#school performances averages based on School Size
Scores_grouped_by_school_size.mean()
```

Out[10]:

```
                    Average Math Score  Average Reading Score  % Passing Math  % Passing Reading  % Overall Passing Rate
        School Size
      Small (<1000)          83.821598              83.929843       93.550225          96.099437               94.824831
 Medium (1000-2000)          83.374684              83.864438       93.599695          96.790680               95.195187
  Large (2000-5000)          77.746417              81.344493       69.963361          82.766634               76.364998
```

Scores by School Type

- Perform the same operations as above, based on school type.

In [11]:

```
# Creating a new data frame to hold Average scores from the School Summary dataFrame with School Type
Scores_with_school_Type= School_summary_df.loc[:,["School Type",
                                                     "Average Math Score",
                                                     "Average Reading Score",
                                                     "% Passing Math",
                                                     "% Passing Reading",
                                                     "% Overall Passing Rate"]]

#Scores_grouped_by_school_size
Scores_grouped_by_school_type = Scores_with_school_Type.groupby("School Type")

#school performances averages based on School Type
Scores_grouped_by_school_type.mean()
```

Out[11]:

	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing Rate
School Type					
Charter	83.473852	83.896421	93.620830	96.586489	95.103660
District	76.956733	80.966636	66.548453	80.799062	73.673757

In [ ]: