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 [152]:
```
# Dependencies and Setup
import pandas as pd
import numpy as np

# File to Load (Remember to Change These)
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", o
n=["school_name", "school_name"])

#school_data_complete.head()
#student_data.head(20)
#school_data_complete.head(20)
```

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 [153]:
```
# District Summary
# --------------------------

# Total Schools
total_schools = school_data['school_name'].count()

# Total Students
total_students = student_data['student_name'].count()
#total_students = "{:,}".format(total_students)

# Total Budget
total_budget = school_data['budget'].sum()
total_budget = "${:,.2f}".format(total_budget)

# Average Math Score
average_math = student_data['math_score'].sum() / student_data['math_sco
re'].count()

# Average Reading Score
average_reading = student_data['reading_score'].sum() / student_data['re
ading_score'].count()

# Students with a passing Math
passing_students_math = student_data.loc[student_data['math_score'] >= 7
0, :]['math_score'].count()
percent_math = (passing_students_math / total_students) * 100

# Students with a passing Reading
passing_students_reading = student_data.loc[student_data['reading_score'
] >= 70, :]['reading_score'].count()
percent_reading = (passing_students_reading / total_students) * 100

# % Overall Passing Rate
overall_passing = (average_math + average_reading) / 2


# Dataframe to hold all the results
district_summary = pd.DataFrame({"Total Schools":total_schools, "Total S
tudents":["{:,}".format(total_students)],
                                 "Total Budget":total_budget, "Average M
ath Score":[average_math],
                                 "Average Reading Score":average_reading
, "% Passing Math":percent_math,
                                 "% Passing Reading":percent_reading,"%
 Overall Passing Rate":overall_passing})

# Display the dataframe
district_summary.head()
```

Out[153]:

	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 [154]:
```
# School Summary
# --------------------

# Group the complete data by School Name
grouped_school = school_data_complete.groupby(["school_name"])

# School Type
school_type = school_data.set_index("school_name")[("type")]

# Total Students in each school
total_students_each = school_data_complete.groupby(["school_name"]).coun
t()["Student ID"]

# Each School Budget
total_budget_each = school_data.groupby(["school_name"]).sum()["budget"]

# Per Student Budget
per_student_budget = round(total_budget_each / total_students_each)

# Average Math Score
avg_math_score = school_data_complete.groupby(["school_name"]).mean()["m
ath_score"]

# Average Reading Score
avg_reading_score = school_data_complete.groupby(["school_name"]).mean()
["reading_score"]

# Percentage Passing Math
passing_math = school_data_complete[(school_data_complete["math_score"]
>= 70)]
math_pass = passing_math.groupby(["school_name"]).count()["Student ID"]
/ total_students_each * 100

# Percentage Passing Reading
passing_reading = school_data_complete[(school_data_complete["reading_sc
ore"] >= 70)]
reading_pass = passing_reading.groupby(["school_name"]).count()["Student
ID"] / total_students_each * 100

#Overall Passing Rate (Average of the above two)
overall_pass = (math_pass + reading_pass) / 2

# Dataframe to hold all the results
school_summary = pd.DataFrame({"School Type":school_type, "Total Student
s":total_students_each,
                               "Per Student Budget":per_student_budget,
"Total School Budget":total_budget_each,
                               "Average Math Score":avg_math_score,"Aver
age Reading Score":avg_reading_score,
                               "% Passing Math":math_pass, "% Passing Re
ading":reading_pass,
                               "% Overall Passing Rate":overall_pass})

# Display the Top 5 Performing schools as per the Overall Score
best_sort = school_summary.sort_values(by='% Overall Passing Rate', asce
```

```
nding=False)
best_sort.head(5)
```

Out[154]:

	School Type	Total Students	Per Student Budget	Total School Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Ov Pas
Cabrera High School	Charter	1858	582.0	1081356	83.061895	83.975780	94.133477	97.039828	95.586
Thomas High School	Charter	1635	638.0	1043130	83.418349	83.848930	93.272171	97.308869	95.290
Pena High School	Charter	962	609.0	585858	83.839917	84.044699	94.594595	95.945946	95.270
Griffin High School	Charter	1468	625.0	917500	83.351499	83.816757	93.392371	97.138965	95.265
Wilson High School	Charter	2283	578.0	1319574	83.274201	83.989488	93.867718	96.539641	95.203

Bottom Performing Schools (By Passing Rate)

- Sort and display the five worst-performing schools

In [155]:
```
# Display the Bottom 5 Performing schools as per the Overall Score

school_summary = pd.DataFrame({"School Type":school_type, "Total Students":total_students_each,
                               "Per Student Budget":per_student_budget, "Total School Budget":total_budget_each,
                               "Average Math Score":avg_math_score,"Average Reading Score":avg_reading_score,
                               "% Passing Math":math_pass, "% Passing Reading":reading_pass,
                               "% Overall Passing Rate":overall_pass})

worst_sort = school_summary.sort_values(by='% Overall Passing Rate', ascending=True)
worst_sort.head(5)
```

Out[155]:

	School Type	Total Students	Per Student Budget	Total School Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% C Pa
Rodriguez High School	District	3999	637.0	2547363	76.842711	80.744686	66.366592	80.220055	73.2
Figueroa High School	District	2949	639.0	1884411	76.711767	81.158020	65.988471	80.739234	73.3
Huang High School	District	2917	655.0	1910635	76.629414	81.182722	65.683922	81.316421	73.5
Johnson High School	District	4761	650.0	3094650	77.072464	80.966394	66.057551	81.222432	73.6
Ford High School	District	2739	644.0	1763916	77.102592	80.746258	68.309602	79.299014	73.8

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 [156]:
```
# Math scores by Grade
# -------------------------

# Using the loc and group by functions to group schools and grades

nineth_math = school_data_complete.loc[(school_data_complete["grade"] ==
"9th")].groupby("school_name")["math_score"].mean()
tenth_math = school_data_complete.loc[(school_data_complete["grade"] ==
"10th")].groupby("school_name")["math_score"].mean()
eleventh_math = school_data_complete.loc[(school_data_complete["grade"]
== "11th")].groupby("school_name")["math_score"].mean()
twelfth_math = school_data_complete.loc[(school_data_complete["grade"] =
= "12th")].groupby("school_name")["math_score"].mean()

# Dataframe to hold all the results
math_scores = pd.DataFrame({"9th Grade": nineth_math, "10th Grade": tent
h_math,
                                        "11th Grade": eleventh_math, "12th
 Grade": twelfth_math})

# Display Math scores by Grade
math_scores.head(20)
```

Out[156]:

school_name	9th Grade	10th Grade	11th Grade	12th Grade
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
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 [157]:
```
# Reading scores by Grade
# --------------------------

# Using the loc and group by functions to group schools and grades

nineth_reading = school_data_complete.loc[(school_data_complete["grade"]
== "9th")].groupby("school_name")["reading_score"].mean()
tenth_reading = school_data_complete.loc[(school_data_complete["grade"]
== "10th")].groupby("school_name")["reading_score"].mean()
eleventh_reading = school_data_complete.loc[(school_data_complete["grad
e"] == "11th")].groupby("school_name")["reading_score"].mean()
twelfth_reading = school_data_complete.loc[(school_data_complete["grade"
] == "12th")].groupby("school_name")["reading_score"].mean()

# Dataframe to hold all the results
reading_scores = pd.DataFrame({"9th Grade": nineth_reading, "10th Grade"
: tenth_reading,
                               "11th Grade": eleventh_reading, "12
th Grade": twelfth_reading})

# Display Reading scores by Grade
reading_scores.head(20)
```

Out[157]:

school_name	9th Grade	10th Grade	11th Grade	12th Grade
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
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 [158]:
```
# 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 [159]:
```
# Scores by School Spending
# ---------------------------------


# Using bins
school_summary["Per Student Budget"] = pd.cut(school_summary["Per Studen
t Budget"], spending_bins, labels=group_names)

# Calculate the ask
scores_spending_math = school_summary.groupby(["Per Student Budget"]).me
an()["Average Math Score"]
scores_spending_reading = school_summary.groupby(["Per Student Budget"])
.mean()["Average Reading Score"]
scores_spending_passing_math = school_summary.groupby(["Per Student Budg
et"]).mean()["% Passing Math"]
scores_spending_passing_reading = school_summary.groupby(["Per Student B
udget"]).mean()["% Passing Reading"]
scores_spending_overall_passing_rate = (scores_spending_passing_math + s
cores_spending_passing_reading) / 2

# Dataframe to hold all the results
scores_by_school_spending_summary = pd.DataFrame({"Average Math Score":
scores_spending_math,
                                                   "Average Reading Score": s
cores_spending_reading,
                                                   "% Passing Math": scores_s
pending_passing_math,
                                                   "% Passing Reading": score
s_spending_passing_reading,
                                                   "Overall Passing Rate": sc
ores_spending_overall_passing_rate})

# Display Scores by School Spending
scores_by_school_spending_summary
```

Out[159]:

Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	Overall Passing Rate
<$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 [160]:
```
# Sample bins. Feel free to create your own bins.
size_bins = [0, 1000, 2000, 5000]
group_names = ["Small (<1000)", "Medium (1000-2000)", "Large (2000-500
0)"]
```

In [161]:
```
# Scores by School Size
# ----------------------------------

# Using bins
school_summary["School Size"] = pd.cut(school_summary["Total Students"],
size_bins, labels=group_names)

# Calculate the ask
scores_size_math = school_summary.groupby(["School Size"]).mean()["Avera
ge Math Score"]
scores_size_reading = school_summary.groupby(["School Size"]).mean()["Av
erage Reading Score"]
scores_size_passing_math = school_summary.groupby(["School Size"]).mean
()["% Passing Math"]
scores_size_passing_reading = school_summary.groupby(["School Size"]).me
an()["% Passing Reading"]
scores_size_overall_passing_rate = (scores_size_passing_math + scores_si
ze_passing_reading) / 2

# Dataframe to hold all the results
scores_by_school_size_summary = pd.DataFrame({"Average Math Score": scor
es_size_math,
                                               "Average Reading Score": s
cores_size_reading,
                                               "% Passing Math": scores_s
ize_passing_math,
                                               "% Passing Reading": score
s_size_passing_reading,
                                               "Overall Passing Rate": sc
ores_size_overall_passing_rate})

# Display Scores by School Spending
scores_by_school_size_summary
```

Out[161]:

School Size	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	Overall Passing Rate
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 [162]:
```
# Calculate the ask
scores_type_avg_math = school_summary.groupby(["School Type"]).mean()["A
verage Math Score"]
scores_type_avg_reading = school_summary.groupby(["School Type"]).mean()
["Average Reading Score"]
scores_type_passing_math = school_summary.groupby(["School Type"]).mean
()["% Passing Math"]
scores_type_passing_reading = school_summary.groupby(["School Type"]).me
an()["% Passing Reading"]
scores_overall_passing_rate = (scores_type_passing_math + scores_type_pa
ssing_reading) / 2

# Dataframe to hold all the results
scores_by_school_type_summary = pd.DataFrame({"Average Math Score": scor
es_type_avg_math,
                                              "Average Reading Score": s
cores_type_avg_reading,
                                              "% Passing Math": scores_t
ype_passing_math,
                                              "% Passing Reading": score
s_type_passing_reading,
                                              "Overall Passing Rate": sc
ores_overall_passing_rate})
# Display Scores by School Spending
scores_by_school_type_summary
```

Out[162]:

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

PyCitySchools Conclusions

- Charter schools outperformed District schools.
- Medium-sized schools (1000-2000) performed better than small and large schools
- Higher per student budget doesn't really seem to have much impact on the overall performance. In fact, the lower the budget, the better the schools performed.
- Best Performing schools have higher percentage passing in math than in reading
- Worst Performing schools have higher percentage passing in reading than in math.
- There's no significant change in math or reading scores per school and they are almost the same throughout all grades (9th, 10th, 11th, and 12th)