

Power BI Report: Cumulative GPA by Athletics Roster

This document describes how to create a Power BI report on cumulative GPA for the athletics roster.

Contents

Introduction	1
Pre-requisites.....	2
Data Connection.....	2
Data Model.....	4
Data Exploration	7
Report Build	12
Tips for Consistently Beautiful Reports	18
Bring it all together	22
Bonus: Save a Template.....	23
Bonus: Update Power BI Desktop to use SharePoint Folder data source	24

Introduction

A GPA and Athletics Roster report enables schools to analyze student athletes by athletic team and cumulative GPA, making it easier to recognize academic achievement across athletics programs. With cumulative GPA and athletics roster data stored in CSV files on SharePoint, Power BI efficiently supports monitoring academic achievements by team and athlete.

This document builds on the Blackbaud Education Management list-based data extraction process by demonstrating how to use those CSV files in Power BI—from initial data connection and transformation through semantic modeling, measure creation, and report design. By following these instructions, readers will have a functional, end-to-end Power BI report that analyzes cumulative GPA by athletic team and can be extended or automated further based on the school's reporting practices.

Pre-requisites

Save the two CSV files locally:

1. CumulativeGPA.csv
2. AthleticsRoster.csv

The document *Education Management Lists: Cumulative GPA by Athletics Roster* describes how to create the lists behind the CSV files required for this report.

Data Connection

Connect to local CSV files.

1. Open Power BI Desktop and start a new, **Blank report**
2. Click the drop-down arrow under the icon for **Get data** and select **More...**
3. From the **Get data** menu, select **Text/CSV**
4. Browse to your local copies of the CSV files and select the file **CumulativeGPA.csv**
5. Select **Transform Data** from the data preview screen.

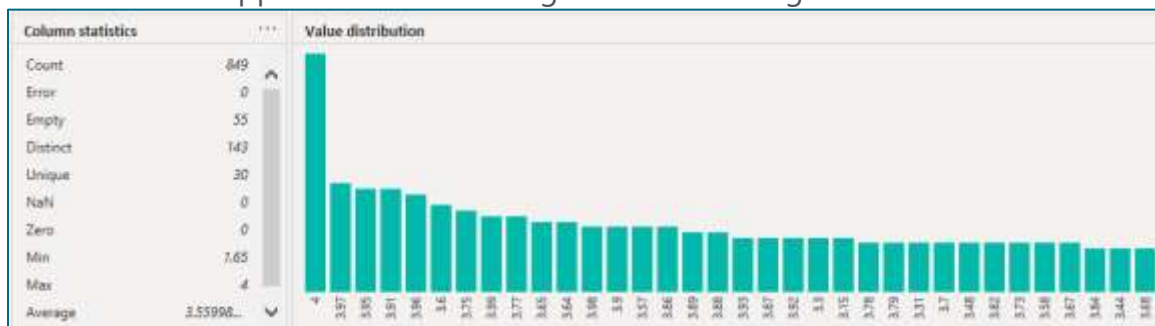
Power Query Editor opens next. Note your first query is automatically named after the file **CumulativeGPA**. Some steps were automatically applied to your csv: Source, Promoted Headers, and Changed Type.

6. Review and update column names and data types.
- If your columns are not in the order pictured, you can click and drag to rearrange the column order. The order does not matter to any of our next steps in Power BI, so do not worry about making the column order match exactly.
 - You can also use this opportunity to rename fields. If you created your own CSV file, you may be dealing with slightly different field names than what is in the file provided. Watch out for spaces that might have been removed. *Note:* you will have different names for the unweighted and weighted GPA values at mid-year and year-end than what is pictured here, as those field names are determined by configuration in your school's environment.

List field name (original)	Power Query field name (new)	Data type
Student ID	Student ID	Text
First name	First name	Text
Last name	Last name	Text
Preferred name	Preferred name	Text
Grad year	Grad year	Whole number
Student grade level	Grade	Whole number
Midyear Transcript GPA Unweighted – Cumulative GPA – Previous	UW GPA, Mid-Year	Decimal
Midyear Transcript GPA Weighted – Cumulative GPA – Previous	WTD GPA, Mid-Year	Decimal
Final Transcript GPA Unweighted – Cumulative GPA – Previous	UW GPA, Final	Decimal
Final Transcript GPA Weighted – Cumulative GPA – Previous	WTD GPA, Final	Decimal

Before moving on, check data quality with Power Query's Column Profile view.

7. Click on the header for the first column, **Student ID**. Then open the **View** ribbon and select the check box next to **Column profile**. View the **Column statistics** and **Value distribution** displayed at the bottom of the screen. Check for Error, Empty, NaN and Zero values.
8. Review the column statistics and value distribution for GPA fields. Note that there are some Empty values. We do not expect Freshmen to have a cumulative GPA yet for the Upper School. When we apply an average to GPA fields, the blank values will be skipped and will not drag down the average.



Next, load the file **AthleticsRoster.csv**


9. From the **New Source** menu, select **Text/CSV** and click **OK**
10. Review columns for data accuracy and data type

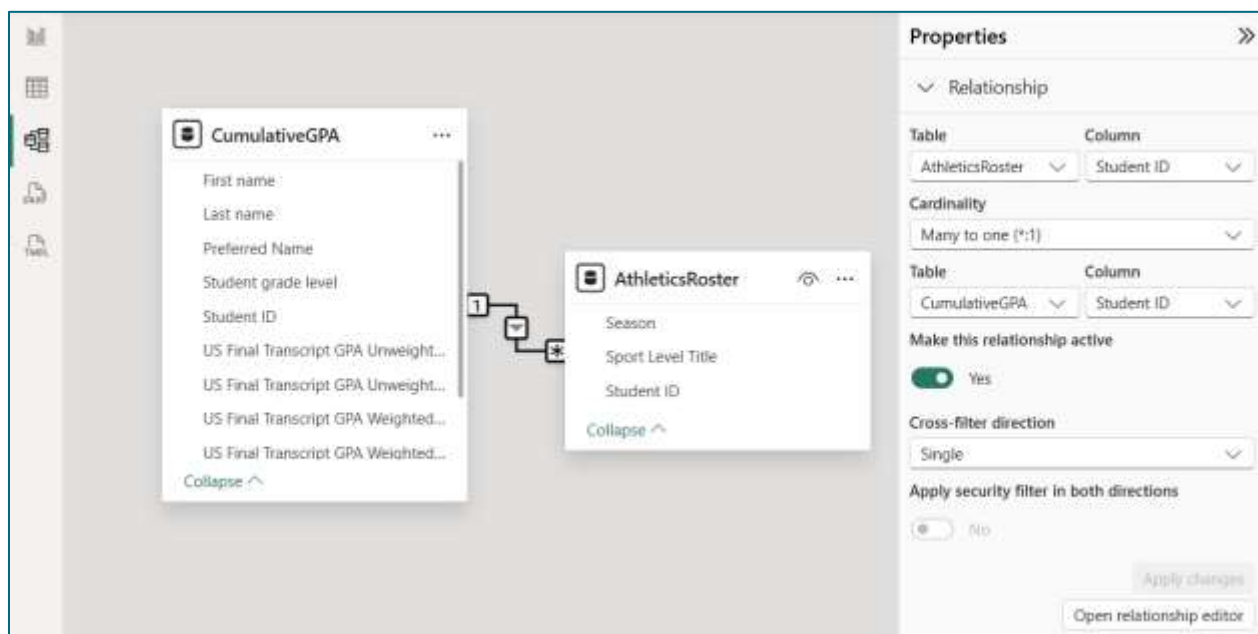
List field name (original)	Power Query field name (new)	Data type
Student ID	Student ID	Text
Sport Level Title	Team name	Text
Season	Season	Text

11. After you are satisfied with the two queries, click **Close & Apply** from the Home ribbon.

Data Model

Power Query Editor closes and you are working in **Power BI Desktop** again. Monitor the Load message while data is loaded from the two SharePoint files. After the data is loaded, you will see two tables listed in the Data pane, representing the results from the two queries you defined in Power Query Editor. **Before you can start building with this data, the table relationship needs to be defined.**

12. Open the **Model view**  from the icons on the left-side of the screen.
13. Check that the relationship between the tables was added automatically. If not, then drag the field "Student ID" from the **CumulativeGPA** table and drop it on top of the "Student ID" field on the **AthleticsRoster** table.
14. View the properties of the relationship by double clicking the relationship line connecting the two tables. The properties should be defined as pictured below.
 - c. From table: AthleticsRoster, Column: Student ID
 - d. To table: CumulativeGPA, Column: Student ID
 - e. Cardinality: Many to one (*:1)
 - f. Make this relationship active: Yes
 - g. Cross-filter direction: Single




Hint: If the Student ID relationship did not appear automatically, check that Student ID is a text data type in both tables and that the field name's spelling is identical across the two tables.

Next, take a moment to explore the **Properties** pane in the **Model view**.

15. Click on **CumulativeGPA** to view properties of the table.
16. Expand **CumulativeGPA** and click on the column **Student grade level** to view properties of this column.

17. Scroll through the **Student grade level** properties and set **Summarize by** = **None**, if not already set. This property will make sure that when Student grade level is added to a view, it will be treated as a dimension and not a measure to aggregate.

With the model relationship defined, we can start to explore the data. Return to the **Report view** . Next, we will create some measures that will help with data exploration and validation.

To make the measures easy to find, let's create a table to hold all measures.

18. From the **Modeling** menu, select **New table**.
19. In the expression bar, name the table and leave the right side of the equation blank, i.e. ".Measures = ".
Hint: placing the dot "." In front of the table title causes it to sort at the top of the data list. Also note, "Measures" is not a valid table name on its own.
20. In the **Data** pane, right click on **.Measures** and choose **New measure**. Add each of the following measures by copying the data analysis expression into the expression bar.

Count AthleticsRoster Records = <code>COUNTROWS(AthleticsRoster)</code>
Count Distinct Students on Roster = <code>DISTINCTCOUNT(AthleticsRoster[Student ID])</code>
Count CumulativeGPA Records = <code>COUNTROWS(CumulativeGPA)</code>
Count Distinct CumulativeGPA Students = <code>DISTINCTCOUNT(CumulativeGPA[Student ID])</code>
Avg UW GPA MidYr = <code>AVERAGE (CumulativeGPA[UW GPA, Mid-Year])</code>

We will create additional average GPA measures later.

With the one:many relationship, it is important to understand which students are represented in each table. There may be some students in the CumulativeGPA table who are not assigned to any team. Since this is an *Athletics* report, viewers are only interested in the Cumulative GPA of students who belong to at least one team. Identify which students belong to at least one team by adding a calculated column to the CumulativeGPA table.


21. Select the **CumulativeGPA** table header in the **Data** pane and right click to select **New column**. Enter the following expression to record each student's record count from the Athletics Roster on the student's CumulativeGPA record.

```
Count StudentTeams = [Count AthleticsRoster Records]
```

Note: You have created a **calculated column** with a DAX expression that is evaluated row-by-row, i.e. for each student in CumulativeGPA. The calculated result is now stored in the data model. It is calculated during the data load/refresh and will be recalculated with each refresh. It is not calculated dynamically as users interact with the report.

Data Exploration

Now you are ready to explore the data relationships and measures. **Always take time to evaluate data integrity and familiarize yourself with the data model.** In this case, we need to understand how many students are in CumulativeGPA and not AthleticsRoster and vice versa.

22. Add a **Card visual**  to **Page 1**.
23. Add **Count AthleticsRoster Records**, **Count Distinct Students on Roster**, **Count CumulativeGPA Records**, **Count Distinct CumulativeGPA Students**, and **Avg UW GPA MidYr** to the card visual.

Example:



Interpret the results: Though there are **845** records in the AthleticsRoster, some students are on multiple teams, so there are only **575** *distinct* students on the roster. The next question is whether all these Roster students have a record in the CumulativeGPA table too. There are **849** students in the CumulativeGPA table. Since the distinct count of Student IDs in the CumulativeGPA table is also 849, we know that Student ID is a unique value and is populated for every CumulativeGPA record.

Next, let's see how many teams students belong to.

24. Add a **Table visual**  with fields **Count StudentTeams**, **Count AthleticsRoster Records**, **Count Distinct CumulativeGPA Students**, and **Avg UW GPA MidYr**.

Count StudentTeams	Count AthleticsRoster Records	Count Distinct CumulativeGPA Students	Avg UW GPA MidYr
	15	285	3.59
1	328	328	3.58
2	412	206	3.49
3	90	30	3.53
Total	845	849	3.56

Interpret the results: Note how many students have no value for **Count StudentTeams**. For 285 students on the **CumulativeGPA** table, this is because they have no *AthleticsRoster* record. For 15 students on the **AthleticsRoster**, this is because they have no *CumulativeGPA* record. Perhaps some of our student athletes are star performers from lower grades not included in our CumulativeGPA list. Let's exclude the students with no value for **Count StudentTeams**.

25. With the table active on the report canvas, review the **Filters** pane. Under Filters on this visual, expand **Count StudentTeams**. Select Filter type **Advanced filtering** and Show items when the value **is not blank**. Click **Apply filter**.

Interpret the results: Now we can see that only 564 distinct students are on at least one team and have a CumulativeGPA. This is out of 575 student athletes.

Also, note that **Count AthleticsRoster Records** shows the total records from the AthleticsRoster for students on more than one team. The Count of AthleticsRoster records is exactly double the distinct student count for students on two teams and exactly triple for students on three teams. This is validation that our table relationship and calculated column are behaving as expected.

Now that we understand that not all students on teams have a CumulativeGPA record and not all students with CumulativeGPA are athletes, we want to define filters that apply to the entire report so that we focus only on **student athletes in grades 10, 11, and 12**.


26. To filter by grade level: in the **Filter** pane, locate the section **Filters on all pages**. Drag **Student grade level** into the space below **Filters on all pages**. Select **Basic filtering** and check the boxes next to 10, 11, and 12.
27. To filter for student athletes: drag **Count StudentTeams** into the space below **Filters on all pages**. Select Filter type **Advanced filtering** and Show items when the value **is not blank**. Click **Apply filter**.

Note the new values in the Card and Table visuals agree on the total **Count Distinct Cumulative GPA Students and Avg UW GPA MidYr**.

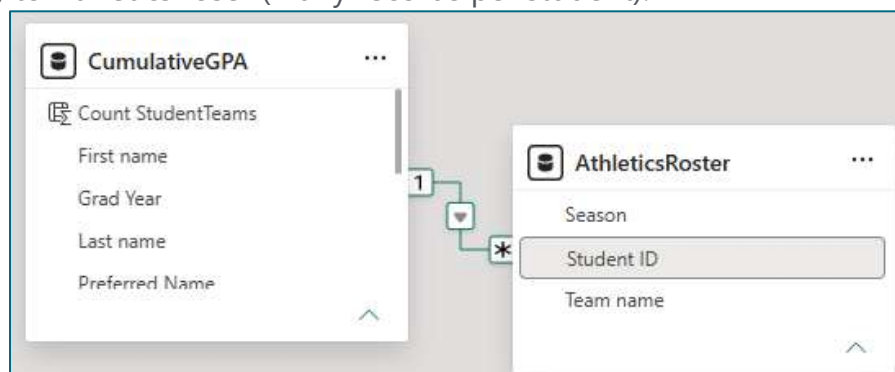
633 Count AthleticsRoster Records	429 Count Distinct Students on R...	429 Count CumulativeGPA Records	429 Count Distinct CumulativeGP...	3.53 Avg UW GPA MidYr
---------------------------------------------	-----------------------------------------------	-------------------------------------------	----------------------------------------------	---------------------------------

Count StudentTeams	Count AthleticsRoster Records	Count Distinct CumulativeGPA Students	Avg UW GPA MidYr
1	250	250	3.57
2	308	154	3.49
3	75	25	3.50
Total	633	429	3.53

Next, let's explore by Team name.

28. Add a new **Table** visual with **Team name** from **AthleticsRoster** and add measures **Count AthleticsRosterRecords**, **Count Distinct Students on Roster**, **Count CumulativeGPA Records**, and **Avg UW GPA MidYr**.
29. Navigate to  **Format your visual** > **General** > **Title** > **On** and provide title text "Teams".

Note that the same **Count CumulativeGPA Records** and **Avg UW GPA MidYr** values are repeated for every team. This is due to the **Single** cross-filter direction of our table relationship where there is a one to many relationship from CumulativeGPA (one record per student) to AthleticsRoser (many records per student).



We can filter the **AthleticsRoster** by **CumulativeGPA** dimensions such as Grade level or Name, but we cannot filter “upstream” from the **AthleticsRoster** table’s Team Name to the specific students on the **CumulativeGPA** table.

To display the average GPA per team for the specific students on that team, it is necessary to identify the Student IDs from **CumulativeGPA** that are in the set of Student IDs from **AthleticsRoster** for the selected Team name. The DAX expression that allows for this is **TREATAS()**. In this case, we need to treat the **AthleticsRoster[Student ID]** values as the list of **CumulativeGPA[Student ID]** values of interest.

30. Create a new measure **Mid-Year GPA (Unweighted)** with this definition and add the measure to the Teams table visual:

```
Mid-Year GPA (Unweighted) =
CALCULATE (
    AVERAGE ( CumulativeGPA[UW GPA, Mid-Year] ),
    TREATAS (
        VALUES ( AthleticsRoster[Student ID] ),
        CumulativeGPA[Student ID]
    )
)
```

Notice that the average GPA is unique per team name now. To validate the average GPA, create a third table listing GPA by Student.

31. First, clean up the Team table view so that it only includes fields **Team name**, **Count Distinct Students on Roster**, and **Mid-Year GPA (Unweighted)**.
32. Then, add another Table visual. From the CumulativeGPA table, add **Student ID**. Then add measures and **Count Distinct Students on Roster** and **Mid-Year GPA (Unweighted)**.
33. Add a title to the table visual: “**Student Roster.**”
34. Click on a Team name from the **Teams** table and see the **Student Roster** table respond to focus on students within the selected team. Validate the **Mid-Year GPA (Unweighted)** totals across students listed and each selected team name.

Notice the interactivity as you click on Team names and the Student Roster filters to focus on student athletes. This is because the TreatAs() function is filtering which students qualify for the GPA measures.

Explore: Note that if you add **Count CumulativeGPA Records** to the Student Roster, additional students will be listed, students not on the selected Team. By default, the only dimension values shown on a visual are those with a value in the displayed measures. For our use case, this is exactly the behavior we want!

Use what you learned about **TREATAS()** to create additional average cumulative GPA measures for weighted and unweighted GPAs at mid-year and final at year end.

```
Mid-Year GPA (Weighted)=
CALCULATE (
    AVERAGE ( CumulativeGPA[WTD GPA, Mid-Year] ),
    TREATAS (
        VALUES ( AthleticsRoster[Student ID] ),
        CumulativeGPA[Student ID]
    )
)
```

```
Final GPA (Unweighted) =
CALCULATE (
    AVERAGE ( CumulativeGPA[UW GPA, Final] ),
    TREATAS (
        VALUES ( AthleticsRoster[Student ID] ),
        CumulativeGPA[Student ID]
    )
)
```

```
Final GPA (Weighted) =
CALCULATE (
    AVERAGE ( CumulativeGPA[WTD GPA, Final] ),
    TREATAS (
        VALUES ( AthleticsRoster[Student ID] ),
        CumulativeGPA[Student ID]
    )
)
```

Continue exploring until you feel comfortable with the data relationships and measures.

35. You can delete the measure **Avg UW GPA MidYr** from your model. Where Avg UW GPA MidYr was on visuals, add **Mid-Year GPA (Unweighted)**, **MidYear GPA (Weighted)**, **Final GPA (Unweighted)**, and **Final GPA (Weighted)** instead.

Report Build

Now we are ready to build the report layout.


36. Click on the "+" tab to add a new tab. Name this tab **GPA's by Team**.
 37. Right click on the **Page 1** tab and choose **Rename**. Name the tab **Exploration**.
 38. Right click again and choose **Hide**.

During your exploration, you already created a Teams table and Student Roster table. You can copy those visuals and paste them onto the GPAs by Team tab to begin building your report.

39. While you are on the **Exploration** tab, select the **Teams** table and hold down CTRL so that when you click on the **Student Roster** table you have both table visuals active. Right click on either active visual and select **Copy > copy visual**.
 40. Navigate to the **GPA's by Team** tab and **paste** the visuals onto the empty report canvas.

Note that the visuals are pasted in the same location as they appeared on the Exploration tab. You can adjust the size and location of the tables.

Slicer

41. Add a **Button Slicer** visual  to the page.
- Add **Season** from **AthleticsRoster** to the slicer field list.
 - Consider display options under **Format visual**.
 - Adjust the shape of the visual to fit.

Explore the filter interactions across the three visuals. Notice that **Season** filters both the Team visual and the Student Roster visual. Use the Season to locate a specific team, then select the team to view the team's roster.

Report Field Parameters

Now we need to address how the school year progresses and the fact that cumulative GPA is updated mid-term and at year-end. We created four measures to allow for weighted and unweighted cumulative GPAs at mid-term and year-end, but based on our report date, only mid-term *or* year-end values are relevant at one time.

Field parameters are an easy way to select which measures are displayed on the visuals. A field parameter behaves like a slicer, but the object being “sliced” or “selected” is a set of fields rather than a set of values in a field.

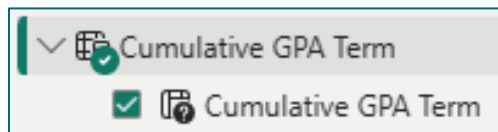
Microsoft Learn:

<https://learn.microsoft.com/en-us/power-bi/create-reports/power-bi-field-parameters>

- 14

Create

Now, let's review the objects that were added to your Data pane and to the report canvas. The Data pane includes a new table named for your parameter **Cumulative GPA Term**. This table includes one field with the same name.



This is a calculated table created for you by the expression:

```
1 Cumulative GPA Term = {
2   ("Final GPA (Unweighted)", NAMEOF('.Measures'[Final GPA (Unweighted)]), 0),
3   ("Final GPA (Weighted)", NAMEOF('.Measures'[Final GPA (Weighted)]), 1),
4   ("Mid-Year GPA (Unweighted)", NAMEOF('.Measures'[Mid-Year GPA (Unweighted)]), 2),
5   ("Mid-Year GPA (Weighted)", NAMEOF('.Measures'[Mid-Year GPA (Weighted)]), 3)
6 }
```

The result is that you can use the field parameter Cumulative GPA Term on visuals, but instead of adding just one field to the visual, it can add all four fields that you defined in the parameter.

43. Clean up your **Teams** table by removing all GPA fields.
44. With the **Teams** table active, check the box next to your field parameter **Cumulative GPA Term** in the Data pane. Note that all four GPA fields are added to the visual.

Team name	Count Distinct Students on Roster	Final GPA (Unweighted)	Final GPA (Weighted)	Mid-Year GPA (Unweighted)	Mid-Year GPA (Weighted)
Boys Junior Varsity Basketball	12	3.58	3.73	3.28	3.68
Boys Junior Varsity Cross Country	6	3.68	3.73	3.58	3.68
Boys Junior Varsity Golf	7	3.58	3.73	3.40	3.68
Boys Junior Varsity Ice Hockey	12	3.58	3.73	3.27	3.68
Boys Junior Varsity Lacrosse	4	3.58	3.73	3.21	3.68
Boys Junior Varsity Rowing	2	3.58	3.73	3.39	3.68
Boys Junior Varsity Skiing	3	3.58	3.73	3.27	3.68
Boys Junior Varsity Soccer	14	3.58	3.73	3.85	3.68
Boys Junior Varsity Tennis	19	3.58	3.73	3.39	3.68
Boys Junior Varsity Track and Field	13	3.58	3.73	3.81	3.68
Boys Varsity Basketball	17	3.58	3.73	3.35	3.68
Boys Varsity Cross Country	5	3.58	3.73	3.52	3.68
Boys Varsity Golf	17	3.58	3.73	3.44	3.68
Boys Varsity Ice Hockey	23	3.58	3.73	3.47	3.68
Boys Varsity Lacrosse	23	3.58	3.73	3.23	3.68
Boys Varsity Rowing	27	3.58	3.73	3.28	3.68
Boys Varsity Skiing	2	3.58	3.73	3.87	3.68
Boys Varsity Soccer	20	3.58	3.73	3.61	3.68
Total	429	3.58	3.73	3.53	3.68

Now you can explore the slicer that was added to this page. When you select a field name from the **Cumulative GPA Terms** slicer, the fields displayed on the **Teams** table respond to show only your selection. You will update your **Teams** and **Student Roster** tables to respond to this slicer, but you do not want end users to have access to the slicer. The slicer will be controlled by you, the report developer. When mid-term grades are complete, you will update the slicer in Power BI Desktop and re-publish the report. After year-end grades are complete, you will update the report again.

45. Update your **Student Roster** table to use the field parameter instead of the original measures, as you did for the **Teams** table.
46. In the **Cumulative GPA Term** slicer, select weighted and unweighted GPAs for the Final term by holding down CTRL to check the second box.
 - a. This behavior, requiring CTRL for multi-select, is optional. Explore your slicer selection settings under the visual's formatting options **Slicer settings > Selection**. Let's turn *off* **Multi-select with CTRL**.

Next, we want to hide the slicer from end users. We can do this in Power BI Desktop's **Selection Pane**.


47. From the **View** ribbon, click **Selection** within the **Show panes** group of icons.



In the selection pane you find a list of your visuals. You can choose which visuals are visible vs. hidden, group visuals, and reorder the layering as they appear on your report canvas. In our case, we have four visuals and two of them are titled "Slicer." This is because we did not name the **Season** slicer or the **Cumulative GPA Term** slicer. The names that appear in the Selection tab are based on the visual titles.

48. Click on the slicer visual **Cumulative GPA Term** to activate it. Then navigate to **Format visual > General > Title**. You must turn the title **On** in order to enter a value. Name your slicer **Cumulative GPA Term Slicer** and then turn the title **Off** because it is redundant with the slicer header already displayed on this visual.
49. In the **Selection** pane, you will find the visual named **Cumulative GPA Term Slicer** now. Click the eye icon to **hide** the slicer from report viewers.

A hidden element introduces risk. If another report developer needs to update the GPA term while you are on vacation, the steps are not obvious. To make the semi-annual update easy, you will create another, hidden page where you will control the visible slicer and provide documentation. This step requires slicer synchronization.

50. Create a new page called "**Edit This!**"
51. Add a **Slicer**  visual for **Cumulative GPA Term**.
52. From the **View** report ribbon, click **Sync slicers** within the **Show panes** group of icons.

When your Cumulative GPA Term slicer is active on your report canvas, the Sync slicer pane lists the other report pages, identifies which ones have the Cumulative GPA Term slicer visible and which pages have that slicer synchronized with each other. Note that you can jump to the other pages that have this slicer using hyperlinks from the **Sync slicers** page names list.

53. In the **Sync slicers** pane, check the **Synchronize** box next to the page names **GPA's by Team** and **Edit This!**
54. Test the slicer synchronization by changing your selection in the Cumulative GPA Term slicer on the **Edit This!** page. You should see the update on the **GPA's by Team** page.
55. Add a **text box** with some documentation for the next report developer responsible for updating the GPA term.
56. Hide the **Edit This!** page. Only report developers editing the report will be able to view this page.

From this point on, you can use the **Cumulative GPA Term** field parameter on visuals. Remember to add a hidden slicer to any new pages using this field parameter and then synchronize the slicer with the **Edit This!** slicer.

57. Return to the GPA's by Team page and edit the layout of your visuals.
58. Alternate between Final and Mid-Year GPA's (using the slicer on the Edit This! page) and note that when you set the column width for each selected measure, it is remembered for the next time that measure is selected.

Tips for Consistently Beautiful Reports

Now that the data model and report layout are complete, apply consistent design standards to improve readability and usability. Use the following practices to create reports that feel polished and cohesive.

Theme

Define a report theme to ensure consistent colors, fonts, and visual styles across all pages.

59. Open the **View** ribbon and click **Themes**.
60. Review available built-in themes.
61. Create or customize a theme:
 - a. Select **Customize** current theme.
62. Define: data colors, background and foreground colors, text styles and titles, and visual header formatting.
63. Preview changes directly on the report canvas.
64. Select **Apply** to update the report.
65. Save the theme for reuse: **Select View > Themes > Save current theme**. This creates a .json file that can be imported into other Power BI Desktop files.

Additional reading:

<https://learn.microsoft.com/en-us/power-bi/create-reports/report-themes-create-custom>

Page Template

Create a reusable page template to maintain consistent layout and navigation across report pages.


66. Create a new page named **Template**. Use this page as the foundation for all report pages.

The following instructions show how to add these elements to your page template:

- Background page color
- Header
- Footer
- Refresh date
- Clear all slicers button

BACKGROUND PAGE COLOR

Set a consistent background for all pages.

67. Click on an empty area of the report canvas. In the **Visualizations** pane, open **Format** .
68. Expand **Canvas background** and select a color or upload an image. Adjust transparency as needed.

Validate: Confirm the background is visible (not fully transparent). Confirm alignment with your theme colors.

HEADER

Add a consistent page title area.

69. Insert a **Text box** at the top of the page.
70. Enter placeholder text for the page title such as "Page Title."
71. Format title font and size.
72. Optionally, apply a background color for contrast.

FOOTER

Add supporting information for context and trust.

73. Insert one or more **Text boxes** at the bottom of the page.
74. Optionally include:
 - a. Confidentiality notice

- b. Definitions or context
 - c. Refresh date
75. Improve layout by inserting a line shape to separate the footer area from the report area. Find shapes on the **Insert** ribbon.

REFRESH DATE

Capture and display the report refresh date.

- 76. Go to **Home > Get Data > Blank Query**
- 77. Select **Connect**

This opens Power Query and is ready for you to supply the M Code that will return the refresh date.

- 78. Name the query **Refresh Stats**.
- 79. Note the first step in **APPLIED STEPS** is named **Source** and is empty.
- 80. Place your cursor into the formula bar and enter the following exactly:

```
= DateTimeZone.UtcNow()
```

A datetime value is returned, representing the coordinated universal time (UTC) date and time.

Note: When you publish your report to the Power BI Service and manage refreshes online, the "local" time zone for Microsoft Fabric is UTC. Consistently using the expression `DateTime.UtcNow()` instead of `DateTime.LocalNow()` makes it very clear which time zone is reported.

Documentation:

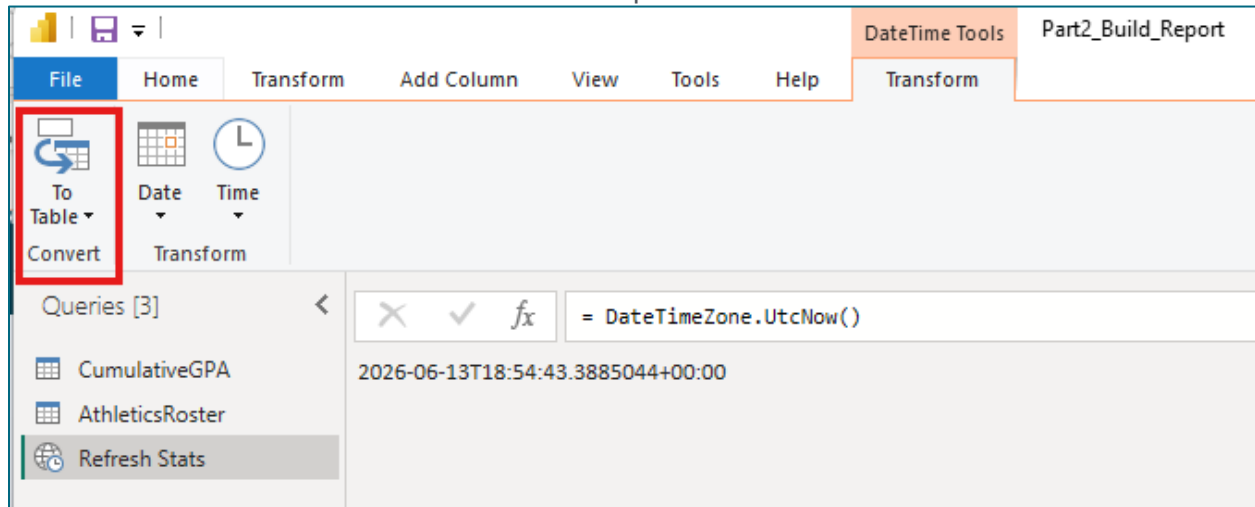
<https://learn.microsoft.com/en-us/powerquery-m/m-local-fixed-utc-variants>

Also see DateTimeZone functions:

<https://learn.microsoft.com/en-sg/powerquery-m/datetimetimezone-functions>

You cannot use the datetime value as-is in Power BI Desktop. You must convert this value into a table so that the **Refresh Stats** query shows up as a table in Power BI Desktop.

81. Since this is a datetime value, a new menu is available on the ribbon bar. Select the **DateTime Tools > Transform** menu, which is different from the standard Transform menu. With the **Source** step selected for **Refresh Stats**, click **To Table**.



82. Rename the Column to **RefreshDateTimeUTC**.

83. Change the data type of the Refresh DateTime field to **Date/Time/Timezone**.

Handling Daylight Saving Time is beyond the scope of this guide, but let's add a column that assumes our local time is Eastern Standard Time. This will get us closer to our local date value. You can convert the UTC time to local time by adding or subtracting hours to fit your time zone with the `DateTimeZone.SwitchZone()` expression.

84. From Power Query's **Add Column** ribbon, select **Custom column**. Paste the following formula into the custom formula window. This converts the original column `RefreshDateTimeUTC` to Eastern Standard Time.

```
= DateTimeZone.SwitchZone([RefreshDateTimeUTC], -5)
```

Note: Daylight Saving Time presents complications. You will find rich discussion and various DIY solutions online, including in the Blackbaud Community.

Discussion: <https://community.blackbaud.com/discussion/72715/power-bi-refresh-date-time>

85. Name the new column **RefreshDateTimeUTC-5_EST**. Change the column's data type to **Date/Time/Timezone**.
86. From Power Query's **Home** ribbon, select **Close & Apply**.

Back in Power BI Desktop, the **Data** pane now includes a table called **Refresh Stats** with two refresh datetime columns for the two time zones.

87. On the Template page, click into the text box in the footer where you want to display the refresh date.
88. Click + **Value** and type "Max RefreshDateTimeUTC-5_EST" as a response to **How would you calculate this value**. Review the sample result. Select a format and click **Save**.
89. You can add text in the text box to specify that this time is in "EST."

CLEAR ALL SLICERS

Add a slicer reset option for report users.

90. Go to **Insert > Buttons > Clear all slicers**
91. Place the button on the page
92. Explore your formatting options in the pane called **Format button**.

Bring it all together

Let's use the template now by placing the **GPA's by Team** visuals on to a copy of the **Template** page. The idea is to copy/paste the visuals from GPA's by Team on to the new page that starts with a copy of the template. But remember there is a hidden slicer on the **GPA's by Team** page that needs to come over too.

93. Right click on the **Template** page and select **Duplicate**. Rename the new tab **Student GPA's by Team**.
94. Navigate to the **GPA's by Team** page and lasso all three visuals by dragging a box around them. When all three visuals are surrounded by lines indicating they

can be reshaped, then right click on any one of them and select **Copy > copy visual**. This will copy all active visuals.

95. Navigate to your new page and **paste** the visuals.
96. You will be asked if you want to sync the slicers. You have seen what it means to sync slicers across pages. This question is referring to the Season slicer that you copied from GPAs by Team and are pasting on the new page. You can select **Don't sync** because we will be deleting GPAs by Team soon.
97. Adjust the visual size and locations to fit within your page template.
98. Return to **GPAs by Team** page and unhide the **Cumulative GPA Term Slicer** visual.
99. Copy and paste this visual to the new, formatted page.
100. When asked whether to sync visuals, select **Sync**.
101. With the **Cumulative GPA Term Slicer** active on your report canvas, open the **Sync Slicers** pane and review the synchronization and hidden/visible properties of this slicer across pages.
102. **Hide** the slicer **Cumulative GPA Term Slicer**.
103. Once satisfied, delete the original, unformatted view **GPAs by Team** tab.
104. You may have additional adjustments to the formatting on the table visualizations. Experiment with the background color and transparency under **Format visual > General > Background**.

Bonus: Save a Template

Now that you have tested your page template and theme, you can save a copy of your workbook to use as the starting template for future reports.

1. Save your workbook as-is
2. Save your workbook with a new name such as **School_Template.pbix**
3. Unhide the **Template** page
4. Delete all pages except for your **Template** page
5. Open Power Query from **Home > Transform Data**
6. Delete all queries except for **Refresh Stats**
7. **Close & Appy**
8. Save your workbook as a template file:
 - a. In the Save As dialog, select the file extension type **.pbix**
 - b. Provide a description of your template and click **Save**.
9. Use the **.pbix** file as your template for future reports.

Bonus: Update Power BI Desktop to use SharePoint Folder data source

Now that you have a Power BI report designed from local **csv** copies of your Education Management list results, you may want to swap out the data source to use **csv** files stored on SharePoint and updated on a schedule. The following steps explore a SharePoint connection and then show how to swap out an existing data source for the new connection. This example will use parameters to refer to the SharePoint Site URL and the folder paths within the SharePoint Documents Library.

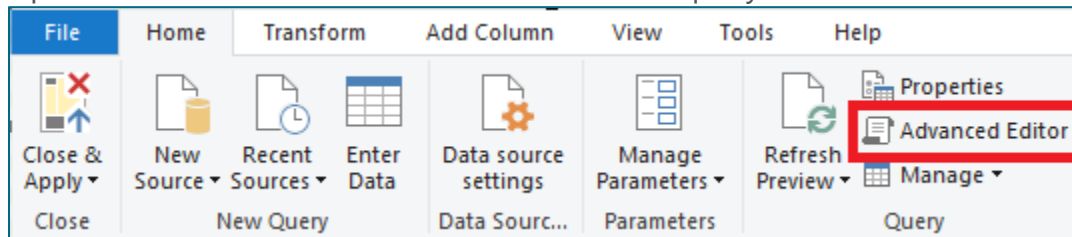
1. Select **Get data > More...**
2. Select **SharePoint folder**
3. Enter the URL for the SharePoint site where your csv files are stored in a document library. Click **OK**.
4. Authenticate with your Microsoft account.
5. You will see a list of all the files in the Site's Documents folder and sub folders. Select **Transform Data**.
6. In Power Query Editor, if you have a lot of folders in your Documents Library, you can filter by folder path to find the csv file you are looking for.
7. Locate your file name, e.g. **CumulativeGPA.csv**, and click **Binary** in the **Content** field to access the file.

	Content	^A _C Name	^A _C Extension
1	Binary	AthleticsRoster.csv	.csv
2	Binary	CumulativeGPA.csv	.csv

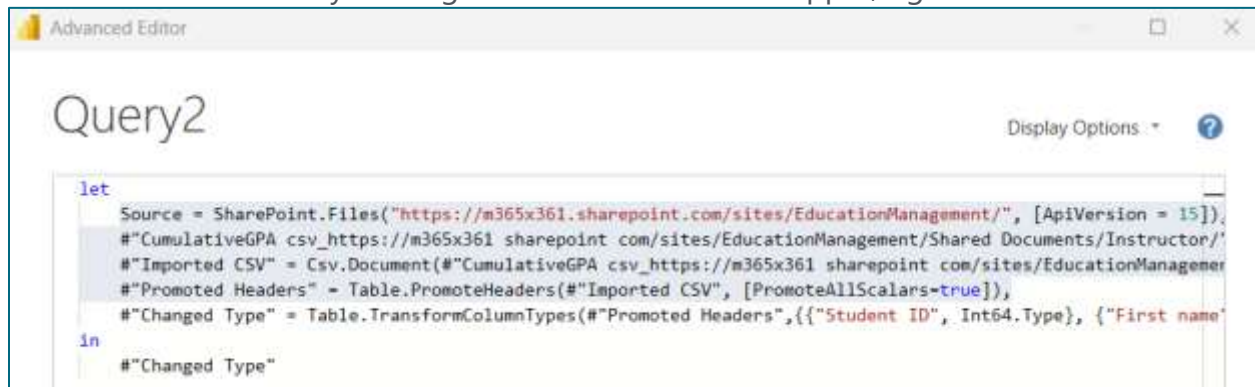
8. New steps are automatically applied to the transformation to promote headers and estimate data types.

At this point, the result should look familiar. You have been here before in your original **CumulativeGPA** query. Now that you know how to access the data with a SharePoint Folder connection, you can replace the Source step in your original Constituent query with the new applied steps.

9. Open **Advanced Editor** for the new SharePoint query.



10. Copy the steps from **Source** to **#"Promoted Headers"** to your clipboard. Close **Advanced Editor** by clicking **Cancel** or the **X** in the upper, right corner.



11. Navigate to the original **CumulativeGPA** query and open **Advanced Editor** for your original **Constituents** query. Replace the first two lines with the lines from your clipboard. Click **Done**.

Note: Be sure to include the comma at the end of the **#"Promote Headers"** line!

The **CumulativeGPA** query should be pulling the file from SharePoint now. Subsequent applied steps should continue to work smoothly. You can delete the new query that you used to test your SharePoint connection.

You can repeat these steps to update the source for **AthleticsRoster**.