WA STEM H2P Data Pipeline Documentation

[Overview 2](#_Toc193194077)

[Maintaining Codebooks/Reference Tables 2](#_Toc193194078)

[Survey Codebook: 2](#_Toc193194079)

[Data Processing 3](#_Toc193194080)

[Overview 3](#_Toc193194081)

[Processing Raw Data 4](#_Toc193194082)

[process\_student\_survey.R 4](#_Toc193194083)

[process\_staff\_survey.R 5](#_Toc193194084)

[Creating Dashboard Tables for Tableau 6](#_Toc193194085)

[Overview 6](#_Toc193194086)

[Suppression 7](#_Toc193194087)

[create\_student\_survey\_dashboard.R 7](#_Toc193194088)

[create\_staff\_survey\_dashboard.R 8](#_Toc193194089)

[Dashboard Creation 9](#_Toc193194090)

[Report Card Enrollment 10](#_Toc193194091)

[Calculated Fields 10](#_Toc193194092)

# 

# 

# Overview

This document is intended to detail out the data pipeline that takes Raw Survey Data collected from QuestionPro and uses a combination of reference sheets and R scripts to process the data to be loaded into Tableau for dashboard creation.

# Maintaining Codebooks/Reference Tables

## Survey Codebook:

**If any changes are made to the survey, there will very likely need to be related changes made to the survey codebook.**

*Example: If we add a new Dual Credit Type option to the question “Has anyone from your school helped you explore the following dual credit opportunities?”, then we will need to add a new option to that normalized\_question\_id “group” (in this case q17x). It is vital to maintain consistent normalized\_question\_id because it makes it significantly easier to identify if a survey item is associated with a broader set of questions.*

If there are changes made to the wording of the question that does not change the way that the question would be responded to or if there were no changes made to the question, then you should not change/create a new question ID. This is important to do because that prevents the need to make unnecessary changes to the subsequent processing scripts, as well providing the ability for questions that have been asked each survey administration to be analyzed longitudinally.

After making any additions, deletions, and updates to the Survey Codebook, you will want to note/track the changes that you have made because you will need to make related changes to the processing scripts.

For each survey iteration, copy the prior year’s rows on the dim\_question\_student or dim\_question\_staff sheet, updating the new rows to the current year.

* If question language is updated in a way that does not affect the analysis, update the question\_header in the new year’s row
  + *No change to code*
* If a response option is updated in a way that does not affect the analysis, update the question\_subheader in the new year’s row
  + *No change to code*
* If there is a new response option, add a new row under the question\_header with the new option as question\_subheader in the new year’s row
  + *Update processing script*
    - *Define new sub\_header in #Defined Functions to convert values*
* If there is a new question, add all fields
  + *Select all that apply questions should put No for mutually\_exclusive. For all others, put yes.*

1. If the question is a new answer\_type, update the processing script with steps a-c & 2a-b/c. If it’s an existing answer\_type, add normalized\_question\_id to filtering in step b only.
   1. Define new function in #Defined Functions to convert values
   2. Add new df\_longfile for the new function, filtered by the relevant question(s). Remove questions from old functions where no longer relevant.
   3. Union new function’s longfile to df\_longfile\_final
2. Update create dashboard script
3. Add to df\_detail$anwser\_type ==
4. Add to # Code Denominator Flag
5. Staff codebook only: add new construct under #All fields for Tableau

# Data Processing

## Overview

Data Processing for H2P is done using R Scripts, and is used to do the following processes:

* Process Raw Survey Data
* Calculate Survey Completion Rates
* Creating Dashboard Tables for Tableau

These processing scripts need to be built/maintained in conjunction with the Survey Codebook as they are built with references to normalized question IDs that **need to be consistent across survey administrations in order for this data pipeline to work properly**. The reason why we do this codebook maintenance is because it affords us the ability to make changes to the actual survey, while allowing us to reuse the R scripts and ultimately allowing us to generate consistent datasets that can be combined with prior administrations, so data can be analyzed longitudinally.

**If any changes are made to the survey codebook, there will very likely need to be related changes made to the processing scripts.**

*Example: If we add a new Dual Credit Type option to the question “Has anyone from your school helped you explore the following dual credit opportunities?”, then we will need to add that new normalized\_question\_id to run through all of the functions like coding the numerical value into the appropriate text value and ensuring that it is identified as one of the question ids.*

If there are changes made to the wording of the question that does not change the way that the question would be responded to or if there were no changes made to the question, then you should not make any related changes to the processing script.

Additional Notes:

* Outputs are written as .csv with | delimiter because other delimiters may be used in text responses.
* Certain chunks of code are left inactive either for future use or because it originated in SQL where deselection is an easier method.

## Processing Raw Data

There are 2 scripts that are in the `H2P/R` folder that are used to process the Raw Excel Output file from Question Pro:

* process\_student\_survey.R
* process\_staff\_survey.R

A quick overview of the two scripts is included below. Note that the more granular line by line notes will be included as comments within the two scripts. Only the most current survey data should be read into these scripts since prior year data is joined back later in the create dashboard scripts.

### process\_student\_survey.R

This script uses 3 packages:

* **readxl**: This package is used to read in the raw Excel files.
* **dplyr**: This package is used for the majority of the data transformations.
* **tidyr**: This package is used to pivot the data from the wide format to the long format.

Next, the script reads in all raw files from all versions of the Student Survey. In the sample 2024 data, there are 2 versions (Spanish & English) of the student survey files.

These different versions are then normalized and unioned together by deleting extra columns & rows, adding placeholder columns, and consistently renaming columns, so that all different versions of the survey have the exact same column structure.

Once the files are combined, we begin to sanitize the responses by clearing out line breaks that may have been entered in the open ended responses as well as flagging invalid responses to demographic questions.

* The general sanitization stuff is tied to the gibberish responses. The invalid responses are tied to people that select every single race/ethnicity, gender, and/or tribe option in the multi-select. Rather than grouping them as a part of each of those selected demographic groupings, we just replace those instances with an "Unknown" response.

After this, we pull the final set of columns that we will be using for our final file and pivot the data from wide form to long form. This pivot essentially changes the structure of the data from being 1 row per respondent with a column for each survey question to 1 row per student/survey question where each row has all demographic characteristics associated with the student.

Next, we define all of the different numerical value to text value mappings that are needed for each Likert scale question that is asked.

* In QuestionPro, if you export the answer values, you don't have the numerical values associated with the values. Those numerical values are useful to have because you can sort by those numerical values and you are also able to simplify certain aspects of measure development by allowing you to evaluate an inequality rather than explicitly naming every possible value.

Note that there is a need to code for both decimal/floating point & integer versions of the numerical value because there were inconsistencies in how the numerical values showed up across questions and values are exported as text values in the Excel Output file. **If there is a new Likert scale that gets added, then there will need to be a new function defined to code that data. It is important to note that if there is even a small variation like the addition of a “N/A” option, then there needs to be a new function defined. Additionally, some new functions, may need to output to a new data source for the viz.**

Once these functions are all defined, we need to run all of these functions on the appropriate subset of questions. The subsets should be easily identified in the dim\_question\_student tab of the SurveyCodebook\_H2P under the `answer\_type` column. Each question type will write the processed version into a separate data frame that will need to be unioned together at the end. **A check that can and should be done at this stage is to validate this resulting unioned version (`df\_longfile\_final`) against the original version (`df\_longfile`) of the file.** The row counts of these two files should be identical if everything is done correctly. Some common reasons for discrepancies include: a question id was missing, the union was missing a dataframe, a data frame was accidentally used twice.

Once you have validated that the resulting longfile is correct, then the final step is writing the wide & long versions of the data into a csv file that can be used in Tableau.

### process\_staff\_survey.R

The staff survey script parallels the student survey script but varies in that there is only a single version of the survey and there are generally less responses. This script uses 3 packages:

* **readxl**: This package is used to read in the raw Excel files.
* **dplyr**: This package is used for the majority of the data transformations.
* **tidyr**: This package is used to pivot the data from the wide format to the long format.

Next, the script reads in the raw Excel files for the staff survey.

We then begin to sanitize the responses by clearing out line breaks that may have been entered in the open ended responses as well as flagging invalid responses to demographic questions. After this, we pull the final set of columns that we will be using for our final file and pivot the data from wide form to long form. This pivot essentially changes the structure of the data from being 1 row per respondent with a column for each survey question to 1 row per respondent/survey question where each row has all demographic characteristics associated with the respondent.

The next set of steps will be identical to that of student survey, where we start by defining all of the different numerical value to text value mappings that are needed for each Likert scale question that is asked. Note that there is a need to code for both decimal/floating point & integer versions of the numerical value because there were inconsistencies in how the numerical values showed up across questions and values are exported as text values in the Excel Output file. **If there is a new Likert scale that gets added, then there will need to be a new function defined to code that data. It is important to note that if there is even a small variation like the addition of a “N/A” option, then there needs to be a new function defined.**

Once these functions are all defined, we need to run all of these functions on the appropriate subset of questions. The subsets should be easily identified in the dim\_question\_staff tab of the \Documents\GitHub\H2P\data\1\_raw\SurveyCodebook\_H2P under the `answer\_type` column. Each question type will write the processed version into a separate data frame that will need to be unioned together at the end. **A check that can and should be done at this stage is to validate this resulting unioned version (`df\_longfile\_final`) against the original version (`df\_longfile`) of the file.** The row counts of these two files should be identical if everything is done correctly. Some common reasons for discrepancies include: a question id was missing, the union was missing a dataframe, a data frame was accidentally used twice.

Once you have validated that the resulting longfile is correct, then the final step is writing the wide & long versions of the data into a csv file that can be used in Tableau.

## Creating Dashboard Tables for Tableau

## Overview

The output files from `process\_student\_survey.R` and `process\_staff\_survey.R` are already structured in a way that can be used for analysis and dashboarding purposes. However, there is a key limitation to using these files, which is that the individual (student/staff) level data needs to be passed into whatever dashboarding/analysis software that is being used. This increases the risk that personally identifiable information (PII) becomes exposed and limits our ability to apply suppression logic. Additionally, pre-aggregating the data provides additional advantages which includes defining the filter groups that can be used as well as creating the ability to link state/regional reference values.

To address this, there are 2 scripts that are in the `H2P/R` folder that are used to process the output files from the `process\_student\_survey.R` and `process\_staff\_survey.R` scripts that further process and pre-aggregate the data to prep it for use in Tableau.

* create\_student\_survey\_dashboard.R
* create\_staff\_survey\_dashboard.R

Note: These scripts had previously been done using stored procedures within a SQL environment and have been adapted to work within a csv/flat file environment.

## Suppression

Data is suppressed in alignment with OSPI requirements, found [here](https://ospi.k12.wa.us/sites/default/files/2022-12/SuppressionRulesforPublicReporting.pdf). Suppression of student & staff counts < 10 are calculated in the respective R scripts. Additional comments for potential future suppression needs are listed in the script as well.

## create\_student\_survey\_dashboard.R

This script uses 3 packages

* **dplyr**: This package is used for the majority of the data transformations.
* **readr**: This package is used to read in the pipe (“|”) delimited output files.
* **readxl**: This package is used to read in the raw Excel files.

This script starts by doing some cleanup related to removing extraneous columns, joining the survey data with the `dim\_question\_student` (student survey question dimension file with details like the actual question text) and the `School Reference (includes details related to school name and region) data.

Once the files are all joined, we begin to do the processing needed to build filters and measures for the dashboard including:

* Race/Ethnicity Reporting Groups
* Gender Reporting Groups
* Met Conditions for Survey Items/Questions
* Denominator Flags

Once those elements have been coded, we begin the process of pre-aggregating the data. While this section takes up several rows, this is more due to needing to repeat multiple similar steps multiple times.

There are 3 different pre-aggregations that are done for dashboarding purposes:

* **df\_dashboard\_stacked**: This pre-aggregation just aggregates the data across all dimensions and derives the count of students across all of those intersections. This is not used in the dashboards that are shared with regional leads/school teams but can be used to generate more granular analysis dashboards.
* **df\_dashboard\_pct\_met**: This pre-aggregation is used to generate the percent met value for all survey questions. This is used for the “Disaggregated by Student Group” dashboards and is also used for the “Student Respondent Demographics” dashboards.
* **df\_dashboard\_stacked\_rpt**: This pre-aggregation is used to create the counts of students for each response to each survey item aggregated across every reporting group. This is used for the “Stacked Bar Chart View” of the dashboard.

For the `df\_dashboard\_pct\_met` and `df\_dashboard\_stacked\_rpt` pre-aggregations, the aggregation process followed the same steps. The data would be aggregated across the following student groups, and for each of those groups the data would be aggregated at the State, Region, and School Levels.

* All Students
* Grade Level
* Race/Ethnicity
* All Students of Color
* Gender
* First Generation

Once all the data has been aggregated at each level, the data is all normalized, unioned together, suppressed, joined to prior year data (as needed), and then ultimately written into a csv file.

## create\_staff\_survey\_dashboard.R

The staff dashboard script is modeled after the student survey script and follows nearly all the same steps with a couple of modifications. Of note, there is an additional pre-aggregation to generate the rings that compare student & staff data and there are no grade level or first generation aggregations.

This script uses 3 packages

* **dplyr**: This package is used for the majority of the data transformations.
* **readr**: This package is used to read in the pipe (“|”) delimited output files.
* **readxl**: This package is used to read in the raw Excel files.

This script starts by doing some cleanup related to removing extraneous columns, joining the survey data with the `dim\_question\_staff` (student survey question dimension file with details like the actual question text) and the `School Reference (includes details related to school name and region) data.

Once the files are all joined, we begin to do the processing needed to build filters and measures for the dashboard including:

* Race/Ethnicity Reporting Groups
* Gender Reporting Groups
* Met Conditions for Survey Items/Questions
* Denominator Flags

Once those elements have been coded, we begin the process of pre-aggregating the data. While this section takes up a number of rows, this is more due to needing to repeat multiple similar steps multiple times.

There are 5 different pre-aggregations that are done for dashboarding purposes:

* **df\_dashboard\_stacked**: This pre-aggregation just aggregates the data across all dimensions, and derives the count of staff across all of those intersections. This is not used in the dashboards that are shared with regional leads/school teams but can be used to generate more granular analysis dashboards.
* **df\_dashboard\_pct\_met**: This pre-aggregation is used to generate the percent met value for all survey questions. This is used for the “Disaggregated by Staff Group” dashboards and is also used for the “Staff Respondent Demographics” dashboards.
* **df\_dashboard\_stacked\_rpt**: This pre-aggregation is used to create the counts of staff for each response to each survey item aggregated across every reporting group. This is used for the “Stacked Bar Chart View” dashboard.
* **df\_rings\_pct\_rpt**: This pre-aggregation is used to create the Staff ring charts that are used in the “Staff Perception vs Student” dashboard. This is needed because in order to create a ring/donut chart, we need to have the complementary value to the part of the ring that we want to show, so the percentage can sum up to 100%. Additionally, this needs to be calculated differently because the relevant questions all get responses on a continuous scale, so the values need to be averaged.
* **df\_rings\_student**: This pre-aggregation is used to build the student complement to the staff survey responses about staff perception of students at their school. This table is built by lightly processing the relevant questions from the student files.

For the `df\_dashboard\_pct\_met`, `df\_dashboard\_stacked\_rpt`, and `df\_rings\_pct\_rpt` pre-aggregations, the aggregation process followed the same steps. The data would be aggregated across the following staff groups, and for each of those groups the data would be aggregated at the State, Region, and School Levels.

* All Staff
* Race/Ethnicity
* All Staff of Color
* Gender

Once all of the data has been aggregated at each level, the data is all normalized, unioned together, suppressed, joined to prior year data (as needed), and then ultimately written into a csv file.

# Dashboard Creation

Animations

Animations are used to further highlight corresponding data between multiple sheets on dashboards.

* Data Model
* Dashboards
* Survey Respondent Demographics
* Student Survey Completion Dashboards

## Report Card Enrollment

Each dashboard contains a Report Card Demographics file (manually created using available OSPI report card data (Student Enrollment and Teacher Demographics), used as a viz-in-tooltip on the Survey Respondent Demographics tab to compare how the survey sample compares to the overall school and region demographics.

## Calculated Fields

Year fields on h2p\_staff\_dashboard\_rings\_pct\_rpt may need to be adjusted to date field type as Tableau may read it in as string or numeric. Update in the data source pane.

Whenever possible, write formulas in R since there is access to version control, unlike with Tableau calculated fields.

|  |  |  |  |
| --- | --- | --- | --- |
| **Formula** | **Notes** | **Data Sources** | **Formula Code** |
| **Staff\_group (filter limiter)** | **aggregates staff groups since specific demographic info has limited N** | **h2p\_staff\_dashboard\_stacked** | **[staff\_group]** |
| **report\_group (filter limiter)** | **Not used currently- could be used in the future to prevent disaggregating by various report groups** | **h2p\_staff\_dashboard\_stacked** | **[report\_group]** |
| **Combined Question Filter** |  | **h2p\_student\_dashboard\_pct\_met, h2p\_staff\_dashboard\_pctmet** | **if isnull([Question Subheader]) then [Construct\_rpt] + ' - ' + [Question Header] else [Construct\_rpt] + ' - ' + [Question Header] + ' ' + [Question Subheader] end** |
| **Met Condition (title)** |  | **h2p\_student\_dashboard\_pct\_met, h2p\_staff\_dashboard\_pctmet** | **if [Answer Type] = 'Education Level' then if [Question Header] = 'What is the highest level of education you want to obtain?' then 'that they want ' elseif [Question Header] = 'What is the highest level of education you expect to obtain?' then 'that they expect ' end else '' END** |
| **Met Condition (title)** |  | **H2p\_student\_dashboard\_stacked (detail)** | **if [Answer Type] = 'Education Level' then if [Question Header] = 'What is the highest level of education you want to obtain?' then 'who want ' elseif [Question Header] = 'What is the highest level of education you expect to obtain?' then 'who expect ' end else '' END** |
| **Student & Family Aspirations Aliases** |  | **h2p\_student\_dashboard\_pct\_met** | **if [Question Header] = 'What is the highest level of education you want to obtain?' then 'Student Aspirations' elseif [Question Header] = 'What is the highest level of education you expect to obtain?' then 'Student Expectations' elseif [Question Header] = 'Does your family expect you to pursue some form of post-high school education (apprenticeship, 1 or 2 year certificate, 2 or 4 year college degree)?' then 'Family Expectations' end** |
| **Student & Family Aspirations Tooltips** |  | **h2p\_student\_dashboard\_pct\_met** | **if [Question Header] = 'What is the highest level of education you want to obtain?' then 'students want' elseif [Question Header] = 'What is the highest level of education you expect to obtain?' then 'students expect' elseif [Question Header] = 'Does your family expect you to pursue some form of post-high school education (apprenticeship, 1 or 2 year certificate, 2 or 4 year college degree)?' then 'families expect' end** |
| **First Gen Count Met** |  | **h2p\_staff\_dashboard\_pct\_met** | **[Count Staff]-[Count Met]** |
| **First Gen Count Met** |  | **h2p\_student\_dashboard\_pct\_met** | **[Count Students]-[Count Met]** |
| **First Gen Pct Met** |  | **h2p\_student\_dashboard\_pct\_met** | **1-[Pct Met]** |
| **Index** |  | **h2p\_student\_dashboard\_pct\_met** | **INDEX()** |
| **want/expect** |  | **h2p\_student\_dashboard\_stacked** | **IF [question] = "q08a" then "want" elseif [question] = "q22a" then "expect" end** |
| **Pct Met (State Reference)** |  | **h2p\_staff\_dashboard\_stacked** | **Will need to manually be updated based on number provided by WASTEM after all responses are collected after December 13th** |
| **Pct Met (State Reference)** |  | **h2p\_student\_dashboard\_stacked** | **Will need to manually be updated based on number provided by WASTEM after all responses are collected after December 13th** |
| **Construct\_Rpt** |  | **h2p\_student\_dashboard\_pct\_met, h2p\_student\_dashboard\_stacked, h2p\_student\_dashboard\_stacked (detail)** | **if [Construct] = 'DC' then 'Dual Credit' elseif [Construct] = 'FA' then 'Financial Aid' elseif [Construct] = 'HSBP' then 'High School & Beyond Plan' elseif [Construct] = 'PS' then 'Postsecondary' elseif [Construct] = 'SS' then 'School Supports' else 'REVIEW' end** |
| **Pct Met (label)** |  | **h2p\_student\_dashboard\_stacked, h2p\_student\_dashboard\_stacked (detail)** | **IF [Question Header] = 'What is the highest level of education you want to obtain?' then 'Percent want ' + [Met Condition] elseif [Question Header] = 'What is the highest level of education you expect to obtain?' then 'Percent expect ' + [Met Condition] else 'Percent ' + [Met Condition] end** |
| **Pct Met** |  | **h2p\_student\_dashboard\_stacked, h2p\_student\_dashboard\_stacked\_detail** | **sum([Met Flag]\*[Count Students])/sum([Denominator Flag]\*[Count Students])** |
| **Pct Met** |  | **h2p\_staff\_dashboard\_stacked, h2p\_staff\_dashboard\_stacked\_detail** | **sum([Met Flag]\*[Count Staffs])/sum([Denominator Flag]\*[Count Staffs])** |
| **Color Coding** | **Allows for a continuous single color coding scale** | **h2p\_student\_dashboard\_stacked, h2p\_student\_dashboard\_stacked (detail)** | **if [Answer Type] = 'Yes/No' then 2-[Value] //positive for yes and neutral for no**  **elseif [Answer Type] = 'Multiselect' then 2-[Value] //positive for yes and neutral for no**  **elseif [Answer Type] = 'Knowledgeable' then [Value] - 2 //positive for 3 & 4, neutral for 2, negative for 1**  **elseif [Answer Type] = 'Helpfulness' then [Value] - 3 //positive for 4 & 5, neutral for 3, negative for 2, non-answer for 1**  **elseif [Answer Type] = 'Likeliness' then [Value] +10 //plus 10 to move to a scale with no neutral**  **elseif [Answer Type] = 'Likeliness w/ NA' then [Value] + 10 //plus 10 to move to a scale with no neutral**  **elseif [Answer Type] = 'Agree/Disagree' then [Value] + 10 //plus 10 to move to a scale with no neutral**  **elseif [Answer Type] = 'Education Level' then [Value] + 20 //plus 20 to go to a color gradient**  **elseif [Answer Type] = 'Rank' then**  **if [Value] = 1 then**  **100 //specific color for 1st place**  **elseif [Value] = 2 then**  **90 //specific color for 2nd place**  **elseif [Value] = 3 then**  **80 //specific color for 3rd place**  **elseif [Value] >= 4 and [Value] <= 10 then**  **50 //shared color for 4th-10th place**  **end**  **end** |
| **Color Coding** | **Allows for a continuous single color coding scale** | **h2p\_staff\_dashboard\_stacked, h2p\_staff\_dashboard\_stacked\_detail** | **if [Answer Type] = 'Yes/No' then 2-[Value] //positive for yes and neutral for no**  **//elseif [Answer Type] = 'Multiselect' and isnull([Value]) then 0 //positive for yes and neutral for no**  **elseif [Answer Type] = 'Multiselect' then 2-[Value] //positive for yes and neutral for no**  **elseif [Answer Type] = 'Knowledgeable' then [Value] - 2 //positive for 3 & 4, neutral for 2, negative for 1**  **elseif [Answer Type] = 'Helpfulness' then [Value] - 3 //positive for 4 & 5, neutral for 3, negative for 2, non-answer for 1**  **elseif [Answer Type] = 'Likeliness' then [Value] +10 //plus 10 to move to a scale with no neutral**  **elseif [Answer Type] = 'Likeliness w/ NA' then [Value] + 10 //plus 10 to move to a scale with no neutral**  **elseif [Answer Type] = 'Agree/Disagree' then [Value] + 10 //plus 10 to move to a scale with no neutral**  **elseif [Answer Type] = 'Education Level' then [Value] + 20 //plus 20 to go to a color gradient**  **elseif [Answer Type] = 'Dual Credit' then [Value] + 30 //plus 30 to go to a color gradient**  **elseif [Answer Type] = 'Not Applicable/Yes/No/Not Offered' then [Value] + 40 //plus 40 to go to a color gradient**  **elseif [Answer Type] = 'Continuous' then [Value] + 1000 //plus 1000 to go to a color gradient**  **end** |
| **Combined Question** |  | **h2p\_student\_dashboard\_pct\_met, h2p\_student\_dashboard\_stacked, h2p\_staff\_dashboard\_pctmet, h2p\_staff\_dashboard\_stacked** | **if isnull([Question Subheader]) then [Question Header] else [Question Header] + ' ' + [Question Subheader] End** |
| **Survey Question Filter** |  | **h2p\_student\_dashboard\_pct\_met, h2p\_student\_dashboard\_stacked** | **[Construct\_rpt] + ' - ' + [Question Header]** |
| **Disaggregation Group** |  | **h2p\_student\_dashboard\_stacked (detail), h2p\_staff\_dashboard\_stacked\_detail** | **if [Disaggregation Group Parameter] = 'First Generation Status' then [First Gen] elseif [Disaggregation Group Parameter] = 'Gender' then [Gender Rptgroup] elseif [Disaggregation Group Parameter] = 'Grade Level' then [Grade Level] elseif [Disaggregation Group Parameter] = 'Race/Ethnicity' then [Race Ethnicity Rptgroup] elseif [Disaggregation Group Parameter] = 'Region' then [Region] elseif [Disaggregation Group Parameter] = 'School' then [School] end** |
| **Percentage Bins** |  | **h2p\_staff\_dashboard\_stacked** | **if isnull([Value]) then NULL elseif [Value] < 0 then 'N/A - Below 0%' elseif [Value] <= 10 then '0%- 10%' elseif [Value] <= 20 then '11%- 20%' elseif [Value] <= 30 then '21%- 30%' elseif [Value] <= 40 then '31%- 40%' elseif [Value] <= 50 then '41%- 50%' elseif [Value] <= 60 then '51%- 60%' elseif [Value] <= 70 then '61%- 70%' elseif [Value] <= 80 then '71%- 80%' elseif [Value] <= 90 then '81%- 90%' elseif [Value] <= 100 then '91%- 100%' elseif [Value] > 100 then 'N/A - Above 100%' end** |
| **staff\_perception\_title** |  | **h2p\_staff\_dashboard\_stacked** | **if [question] = 'q08a' then 'can afford to attend a 4-year college using financial aid, scholarships, and family resources?'**  **elseif question = 'q03a' then 'WANT to obtain some level of post-high school education, including apprenticeship, 1-year and 2-year certificate, and 2-year and 4-year college?'**  **elseif question = 'q21a' then 'EXPECT to obtain some level of post-high school education, including apprenticeship, 1-year and 2-year certificate, and 2-year and 4-year college?'**  **end** |
| **Avg Staff Perception** |  | **h2p\_staff\_dashboard\_stacked** | **sum([Value]\*[Count Staff])/sum([Count Staff])** |
| **Value X Count Staff** |  | **h2p\_staff\_dashboard\_stacked** | **[Value]\*[Count Staff]** |
| **Ring Group (copy)** |  | **h2p\_staff\_dashboard\_rings\_pct\_rpt** | **[Ring Group]** |
| **staff\_perception\_text** |  | **h2p\_staff\_dashboard\_rings\_pct\_rpt** | **if [Question] = 'q08a' then 'think they can afford to attend a 4-year college'**  **elseif [Question] = 'q03a' then 'WANT to obtain postsecondary education'**  **elseif [Question] = 'q21a' then 'EXPECT to obtain postsecondary education'**  **end** |
| **Pct Ring Label (staff)** |  | **h2p\_staff\_dashboard\_rings\_pct\_rpt** | **if [Ring Group] = "Ring" then [Avg Staff Response] else null end** |
| **Pct Ring Label (student)** |  | **h2p\_staff\_dashboard\_rings\_pct\_rpt** | **if [Ring Group] = "Ring" then [Pct Met] else null end** |
| **Rank filter** |  | **H2p\_student\_dashboard\_stacked, h2p\_student\_dashboard\_stacked (detail)** | **if [Answer Type] = 'Rank' and (ISNULL([Color Coding]) or [Color Coding] = 50 ) then**  **false**  **else**  **true**  **end** |