

Table of Contents

[Assessment Data Islands 3](#_Toc532205803)

[Assumptions 4](#_Toc532205804)

[Assessments in the Ed-Fi Data Model 5](#_Toc532205805)

[Getting Started 5](#_Toc532205806)

[Inventory the existing Assessments and data sources. 5](#_Toc532205807)

[Analyze the assessments to determine which aspects should be brought into Ed-Fi. 6](#_Toc532205808)

[Getting Data into the Ed-Fi ODS 6](#_Toc532205809)

[Vendor APIs communicating with the Ed-Fi ODS 6](#_Toc532205810)

[Data Flow using APIs 6](#_Toc532205811)

[Bulk loading XML 7](#_Toc532205812)

[Other Approaches 7](#_Toc532205813)

[How do Assessments Interrelate? 7](#_Toc532205814)

[Surfacing Data to an Analytic 9](#_Toc532205815)

[Initial approach 9](#_Toc532205816)

[Revised approach 9](#_Toc532205817)

[Other approaches 10](#_Toc532205818)

[Analytics Middle Tier 10](#_Toc532205819)

[Steps to Deploy the Solution 10](#_Toc532205820)

[Security and Authorization 11](#_Toc532205821)

[Database Structure 12](#_Toc532205822)

[Power Bi Setup and Configuration 17](#_Toc532205823)

[Power BI initial setup and configuration: 17](#_Toc532205824)

[EQ 24 Design and Configuration: 19](#_Toc532205825)

[EQ 24 Detail Design View Breakdown 19](#_Toc532205826)

[Summary View (Default landing page) 20](#_Toc532205827)

[Summary View (Individual data grid and charts breakdown) 20](#_Toc532205828)

[Summary View (Code Behind) 21](#_Toc532205829)

[Student Detail View (Default landing page) 24](#_Toc532205830)

[Student Detail View (Individual data grid and charts breakdown) 24](#_Toc532205831)

[Student Detail View (Code Behind) 27](#_Toc532205832)

[Group Details View 28](#_Toc532205833)

[Group Detail View (Individual data grid and charts breakdown) 28](#_Toc532205834)

[Student Detail View (Code Behind) 31](#_Toc532205835)

[EQ 24 Power BI Security Setup and Configuration 32](#_Toc532205836)

# Assessment Data Islands

Since the 70’s consumers of information technology software solutions have struggled with Data Islands or silos. These data islands are business and education critical information siloed either intentionally or unintentionally by software vendors or imposed by the limits of the underlying infrastructure platforms. Initially in banking and Insurance, this limitation to organizing data has spread to every industry, including K12 Education. How does an education organization gather, compare, and analyze isolated data marooned in proprietary databases of different software products, often delivered by multiple vendors and sometimes implemented on incompatible systems?



From Wikimedia Commons, the free media repository

Our focus is on the Assessment domain. This document is to be consumed by the technical team as a guide to the technical decision-making process involved in delivering the analytic solution described in the EQ 24 Analytics Guide. The purpose is to illustrate how student assessment data that is available to an education organization might be mapped to the Ed-Fi data model, and the different ways that the data might be loaded into the Ed-Fi ODS. An approach to abstracting the relational data in the Ed-Fi ODS to a set of views that the Power BI dashboard can consume in a more efficient manner is also illustrated. A Power BI dashboard has been developed to show how to perform analytics on this data, by which to better understand how to tailor educators’ approach to educating their students.

# Assumptions

The set of skills needed to develop an analytic solution using EQ 24 as a guide are:

* Subject matter expert with through knowledge of the available assessments under consideration and related student data.
* Business Analyst to serve as the bridge between the assessment subject matter expert and the Tech Team, working to create actionable requirements for the technical resources to meet.
* Database administrator to create database objects and to modify views to meet the specific needs of the problem being addressed.
* Azure Power BI System Administrator.
* Office 365 administrator.
* Power BI developer, or a developer with the skills necessary to implement an alternative analytic tool.
* Ed-Fi ODS administrator to work with the Vendor API administrator to configure, administer, and monitor the APIs from the Ed-Fi side.
* Vendor API administrator to configure, administer, and monitor the APIs from the vendor side.
* ETL/SQL developer to assist the Ed-Fi and Vendor API administrators in the implementation and troubleshooting of the data exchanges and to develop alternative data loading mechanism when required.

Depending on the size of the organization, a number of these skills might be covered by a single individual or spread across a larger team.

Software used for the pilot implementation includes:

* Ed-Fi ODS v2.5 running on-premise
* SQL Server 2014 running on-premise
* Microsoft Azure Subscription
* Microsoft Office 365 Subscription
* Power BI Desktop
* Power BI License
* On-Premise data gateway

# Assessments in the Ed-Fi Data Model

In the Ed-Fi data model, an assessment can have one or more scores associated with it at the overall assessment level. Those might include raw scores, scale scores, percentiles, etc. Assessments can also have more granular Domain scores, also known as Strands, Reporting Categories, or Objectives. Assessments can also have subordinate, child level skill or standard scores that roll up to the Parent domains.

**Overall Assessment** (e.g.) Florida Standard Assessment scale score

**Objective Assessment (Domain Parent Level)** (e.g.) Expressions and Equations raw score

**Objective Assessment (Child Level)** (e.g.) Linear Expressions, Equations, and Inequalities raw score

An Assessment that is not subdivided into specific reporting categories may have no Objective Assessments. On the other extreme, assessments that report by skill can have any number of Objective Assessments depending on how granular the data is. Both cases can be accommodated by the Ed-Fi data model.

Objective Assessments can also be mapped to Learning Objectives or Learning Standards, neither of which were in scope for EQ 24, nor were they mapped or loaded.

State and Local assessments are often organized by Academic Subject, such as Mathematics, English Language Arts, etc. They are also typically organized by Grade Level. Essential Question 24 is restricted to a cohort of students in grade level 8 that are currently taking eighth grade math, many of whom also had assessment scores for seventh grade math.

# Getting Started

To achieve the greatest benefit from using Ed-Fi to process their Student Assessments, and to provide an integration with Student, Staff, and other student data, the Education Organization should implement the full Ed-Fi ODS/API and pull the full set of data from their Student Information Systems, using the Ed-Fi APIs. If the Education Organization does not already have an existing populated Ed-Fi ODS, a number of Assessment implementation tasks can be undertaken concurrently with the deployment of an Ed-Fi ODS/API.

If the Education Organization already has an existing populated Ed-Fi ODS, the next steps would be to:

## Inventory the existing Assessments and data sources.

The best way to source data from an assessment vendor would be by using the Ed-Fi APIs to push the assessment data directly from the assessment vendor into the Ed-Fi ODS. There are cases where an assessment vendor does not have an existing API implementation planned and cases where the API implementation is underway, but not complete. In those cases, alternative data sources are usually available. These alternative data sources could include CSV files, Excel Spreadsheets, or XML files. These data sources could come directly from the assessment vendors or from other sources, such as a Student Information System.

For those assessments in the inventory, determine which assessment vendors have already developed APIs that can be used to populate the Ed-Fi ODS.

For those sources of assessment data that don’t have APIs developed, work with the Ed-Fi Alliance or your integrator to influence the vendor to develop an API for the data interchange between their assessment product and Ed-Fi.

## Analyze the assessments to determine which aspects should be brought into Ed-Fi.

If there is an API, the assessment vendor software may be configurable in a manner which would allow the choice of at how granular a level the data should be pulled into the ODS. Regardless of whether the API or an alternative approach is used to pull the data into the ODS, there are tradeoffs to be considered with regard to the granularity of the data. Depending on the size of the district, Item level assessments might contain many millions of rows, and while in general it would be better to have all of the available data loaded, if an organization didn’t have a requirement for the lower level data, the data storage and available processing power might be better used elsewhere.

# Getting Data into the Ed-Fi ODS

There are a number of ways to load the student assessment data into the ODS. These include, in order of desirability:

## Vendor APIs communicating with the Ed-Fi ODS

This is the preferred method by a wide margin. In addition to assessment vendors, some SIS vendors, that have student assessment data loaded into them, also provide APIs to send the assessment data to Ed-Fi. It might be necessary to evaluate the completeness of deployment and work with the vendor if the full set of data isn’t being sent to Ed-Fi and more is required. For example, a vendor might be sending in the overall assessment scores, but not the objective level data. If this lower level data is required, work with the vendor software to configure their software to send the objective level data.

## Data Flow using APIs

Data Flow is a proof-of-concept application designed to transform and load CSV data into an Ed-Fi ODS/API. Data Flow provides an admin panel to map CSV file formats to Ed-Fi API endpoints along with components to extract, transform and load assessment data into the Ed-Fi ODS/API. It is available on the Ed-Fi Exchange. At present, it doesn’t support Ed-Fi version 2.5, which is the recommended level at the time of this writing, but the product will support this release of Ed-Fi in the near future.

## Bulk loading XML

This is a less desirable approach than the use of the API, as there may be a significant amount of work involved converting the data from the source system into the XML format that can be consumed by the Bulk Loader. There are several different bulk loading methods available, and the documentation should be reviewed to determine which would be appropriate for a given implementation. There was also a proposal surfaced at the 2018 Ed-Fi Technical Congress to use JSON as the primary data format for bulk loading and to support XML to a lesser degree. Those considering the use of the Bulk Loading approach should monitor Ed-Fi Tech Docs and the Ed-Fi Technology Roadmap to help inform their choice of which Bulk Load method to use.

## Other Approaches

PHP programs can be developed to send student assessment data into the Ed-Fi ODS using the Ed-Fi APIs, pulling the data from csv files or other data sources and sending it as JSON to the ODS through the Ed-Fi API.

The use of SQL to directly insert assessment data into the ODS is the least desirable of the options, but could be useful should sets of data be required for use in developing reports or analytics while one of the other options is being developed, or prior to the availability of a fully populated ODS. The development and use of SQL scripts could also be helpful to prototype the data in Ed-Fi as part of working with the vendor to develop APIs.

# How do Assessments Interrelate?

Projects implementing analytics such as this one can consume data that was sourced into the Ed-Fi ODS in any number of the ways, as outlined above. For the pilot implementation the data was loaded without the benefit of an API from the vendors. Since the pilot project was unconstrained by any limitations imposed by how the data was to loaded into the ODS, the pilot implementation mapped and loaded the assessment data into the ODS in a consistent manner as described below. This consistency was desirable since the goal of the project was to compare the assessment results to see if the students were improving over time and to map the assessments to the dashboard in a consistent manner. If the assessments were loaded into the ODS via a vendor supplied API, they may not be loaded into the ODS consistently. That is, each assessment vendor might choose to load their data into the ODS differently, as long as their implementation conformed to the Ed-Fi standard.

The heart of the challenge for EQ 24 was to determine how to map the assessment data so that the various aspects could be compared. Specifically, was the performance of the student improving over time. Initially we mapped and loaded the data into the ODS, in a manner that seemed reasonable based on the data, only to discover that we didn’t have comparable data across the Assessments that we loaded. We had Overall and Domain Level scores for the Florida Standard Assessment (FSA) but on the opposite end of the spectrum, we only had Skill Level scores for the iXL formative assessment. Star Math had the broadest range of scores, with Overall scores, including scale scores and national percentile ranks, and Domain level scores, for the various reporting categories, and additionally had scores by skill (standard) level. Using the Ed-Fi model described above and getting direction from our Pilot district, this is how we mapped these different assessments.

**Overall Assessment**

FSA scale score Star Math national percentile rank iXL participation indicator

**Objective Assessment (Domain Parent Level)**

FSA Learning Objective Star Math Domain iXL Percent Proficient

**Objective Assessment (Child Level – roll up to the Parent Domains)**

\*No skill level FSA Star Math Skill Level iXL Skill Level

There was some variation in the scores reported at the Domain Level, but there was sufficient overlap to make meaningful graphs and tabular displays between the assessments. Some of this variation in the objectives appeared to be caused by comparing Domain scores for different grade levels, as it appeared that the assessment vendors made a lot of effort to have their assessments match the Standards promulgated by the State by grade level. (e.g.) Functions are not one of the Learning Objectives (Domains) for Grade 7 Mathematics but they were in the Grade 8 Mathematics for both Star and iXL. Note that we were comparing the assessments that the cohort of students we were tracking across time, from their Grade 7 test in May of 2017 through the iXL exercises in March of 2018. This time range was selected to allow sufficient temporal distance between testing events, to better gauge whether a student was improving over time.

Since there were no Overall or Domain level scores in the iXL data, it was necessary to create meaningful place holder values, so that the more granular level data, didn’t just show up on the detail screens without context. At the Overall level, the thinking was that a certain type of dashboard user might contemplate the data and only want to drill down to see the detailed student data if a student had participated in the iXL training exercises. To that end, the database views generate an iXL participation indicator, which is just a Y/N to show whether the student participated in iXL. There still remained the issue of the Domain though, since iXL only had skill level records. While it wasn’t explicit in the data, there was an implicit link for each skill back to a Domain by way of the standard that the skill was associated with. During the load process, a synthetic Domain Parent Level progress indicator score was created for iXL, which was the percent of skills attempted that the student achieved proficiency on. The skill level data was mapped directly to the domain, despite there being no explicit relationship in the data, instead relying on the implicit relationship between the standards (that the skill was associated with) and the domain that the standards roll up to.

The Skill “Complete a table and graph a linear function” maps explicitly to Standard 8.F.1.2 (Compare properties of two functions each represented in a different way (algebraically, graphically, numerically in tables, or by verbal descriptions)

Standard 8.F.1.2 is in the “Function” domain

Skill “Complete a table and graph a linear function” maps implicitly to the “Function” domain

The Child level data is supported by the database artifacts delivered with this guidebook, but since they weren’t in scope to answer the question posed by EQ 24, they are not displayed on the dashboard.

# Surfacing Data to an Analytic

The Ed-Fi ODS database is a relational database which may require a degree of denormalization to provide sets of data that can be easily and efficiently consumed by an Analytic product, such as Power BI.

## Initial approach

The initial approach used in the development of the Analytic used to illustrate the solution to Essential Question 24 was to develop database views, which provide an abstraction layer over the queries that were used to generate the denormalized sets of Ed-Fi data. This left the “heavy lifting” work of joining the relational data behind the scenes. These views were located in a separate persistent database called EADM\_Staging\_v25. The reason they were stored in a separate database was because in some Ed-Fi implementations the ODS is may be dropped and recreated at different points in time. Maintaining the views in the persistent database prevents them from being lost when an ODS is recreated. This initial approach, reading the Ed-Fi data “on the fly” through non-materialized views worked adequately at the Superintendent level. However, when Row-level security was implemented to restrict the dashboard data to Teachers and Principals, depending on who is logged on, the queries Power BI generated to filter the data proved to be very inefficient and hugely time consuming. At this point the initial approach was abandoned we the revised approach described below was adopted.

## Revised approach

The revised approach is to cache the view data into tables, thereby materializing those views. The logic of the view was moved to a “view to table cache process”, with the view still existing, but rather than the view having executing logic to join the tables, the view reads the table that was produced using the “view to table cache process” described above.

If an implementer required an even greater performance boost, they could replace the views in the Power BI code with the tables. Not using views would allow them to tune the database objects, adding indexes to tables and the like, to get maximum performance out of the database.

## Other approaches

Azure cached in the cloud. This approach would build on the Revised approach by moving the table cache to an Azure SQL instance and allow Power BI to access the tables in the cloud from the Azure SQL instance. This approach would remove the need for a On Premise gateway and allow for the row level security and any other user security to be performed in Azure using Azure Active Directory accounts. The cache ETL load process would then be modified to load the Azure instance instead of the tables in the staging database. Most if not all of the staging database entities could possibly be moved to the Azure instance.

For more information about Power BI deployment alternatives, there is an excellent white paper, the Version 2 of “Planning a Power BI Enterprise Deployment” available at this link: https://aka.ms/pbienterprisedeploy

## Analytics Middle Tier

There is an Analytics Middle Tier Special Interest Group that has delivered views for an “Early Warning System” and documentation of a methodology for developing views to serve other domains.

Their stated goals are to:

“…simplify access to the Ed-Fi ODS through provision of targeted, de-normalized, SQL views. The initial proof-of-concept solution will provision data required for development of Early Warning Systems (EWS), along with a set of sample visualizations. The target audience for this work is local education agencies (LEA) and those supporting LEA’s.”

“In addition to the proof-of-concept development work, the Alliance plans to document further the recommended practices and policy considerations when using these views. This documentation may cover topics including authentication, authorization, object security, ETL scheduling, and architecting for performance.”

A thorough review of their contributions on the Ed-Fi Exchange and their GitHub repository, evaluating the views and techniques employed, and the adoption of the recommended practices is strongly encouraged in conjunction with the views delivered with this guidebook.

# Steps to Deploy the Solution

The example solution for Essential Question 24 is based on a set of sample school district data, the “Glendale” dataset. It contains fake student, parent, school, etc. data that were leveraged to map synthetic assessment data to the Glendale Students and Organizations that are representative of various Assessments that might be used by a Florida school district. The Ed-Fi Version 2.5 instance of this sample data used for this development can be located at this link:

[https://s3.amazonaws.com/Ed-Fi\_ods\_samples/v2.0/Ed-Fi-Glendale-v22.bacpac](https://s3.amazonaws.com/edfi_ods_samples/v2.0/Ed-Fi-Glendale-v22.bacpac)

It has been imported into a SQL Server 2014 instance and takes roughly 11 gig of storage.

The Glendale\_Ods\_v25 was used to load the assessment data described by this guidebook. The assessment data isn’t part of this distribution, as this is a guide to developing an analytic, and a district should load their assessment data, and modify the solution provided to meet their specific needs.

The EADM\_Staging\_v25 database described by this guidebook was deployed on a SQL Server 2014 database. There are scripts provided along with this guide to create the persistent database and to create and populate the two permanent tables, the ID table and the Assessments List table. There are also database scripts that define the views that are used to surface the data and provide for a security implementation to the Power BI solution that is also provided along with this guidebook.

# Security and Authorization

Security and Authorization will likely vary widely from implementation to implementation. A simplified security implementation is described here. The end users ability to log on to Power BI using their [xxxxxorg.onmicrosoft.com](http://nefecorg.onmicrosoft.com/)  Id is treated as a proxy for authentication, standing in for an Active Directory solution that is one of the ways a production solution might be implemented.

Assessment, School, Student, Teacher, etc. views are employed to surface the data. The intent is for Power BI to filter the data returned from the ODS by a Staff to Student ID mapping view to restrict the data externalized to the dashboard to Students the Staff member is authorized to access. The Student and Assessment Data Power BI will display will be limited to the students that are associated with that Superintendent, Principal, Assistant Principal, or Teacher. Teachers see students that they have a classroom relationship with. Principals and Assistant Principals see students that are in the school they are the Principals of, and Superintendents see all of the students in the district.



# 

# Database Structure

The database structure for EQ24 is as follows:

There are 2 configuration tables that drive the process. We created tables and views in a separate database so that when the ODS database is cycled out configurations will still be in place.

The 2 tables are as follows

dbo.PowerBiIdMapping is the table that handles wiring up a powerbi user email to the staff email address of a user in EdFi using there DistrictEmail address

This is an example of rows that might exist in this table.



The PowerBiId field is an identity field that is generated when the row is added to the table. As delivered, all of the fields are required to have values except for the DatabaseUser which can be left null. The LocalEducationAgencyId and PositionTitle, while required, are documentation fields not directly related to the functioning of the table in the security setup. The essential fields are PowerBIEmailAddress and DistrictEmailAddress as those are the fields that make the connection between the Power BI user and the Ed-Fi user.

dbo.EADM\_Assessments contains the keys for any assessment that will be used in the  datasets that will be displayed.  In order for an assessment/test to be considered in the set it needs to be added to this table.  Assessments not included in this table are ignored.

This is an example of the rows that were loaded in this table to control the pilot dashboard



The AssessmentID field is an identity field that is generated when the row is added to the table. The fields AssessmentTitle, Version, AssessmentIdentifier, and Namespace correspond directly to those same fields in EdFi\_Glendale\_v25 and are used to uniquely identify the assessment. The includeinnce field is used to indicate whether the NCE calculation is to be performed for a particular assessment. See the Analytics Guide for more information about how the NCE is used. In this example, the IXLMathMar2018 assessment is not included in the NCE calculations.

Views were created that are then reused within the displayed datasets for different dimensional and fact data.  These views are all filtered using the above mentioned EADM\_Assessments table.

The key view is a wrapper for the edfi.StudentAssessment table with a join to the EADM\_Assessments table this filters out assessments that we are not interested in.

All of the datasets are returned in views.

The main views/procedures are

v\_EADM\_SummaryView

v\_EADM\_Fact\_DomainLevelScores

v\_EADM\_Fact\_SkillevelScore

v\_EADM\_Fact\_StudentAssessmentPerfomanceLevel

There are lower level views/procs used for returning School, Student demographics, Student to Staff mappings all of which rely on the v\_EADM\_StudentAssessment view/function.

Security is applied to the front end dashboard using PowerBi’s row level security feature. The view dbo.v\_EADM\_DIM\_Usersecurity is used by the front end to fliter out only the student that a given PowerBI account can see by using the EDFI district email address and the dbo.powerbimapping table to make that association to a given staffUSI from the dbo.v\_EADM\_DIM\_StudentStaff view.

The [v\_EADM\_DIM\_StudentStaff\_BuildCache] returns all of the student staff mappings using the students classes as well as the principals and assistant principals that are at the students school. Users that are at the superintendent level will see all students in the district that they have access to.

To create the views, tables, and procs to be used in the persistent data store database, EADM\_Staging\_v25:

1. Create an empty database EADM\_Staging\_v25
2. Run the script CreateEADM\_withCache.sql

The following is a list of all the database tables and views that exist in the EADM\_Staging database

|  |  |
| --- | --- |
| Table/View | Description/Usage |
| Tables |  |
| [dbo].[EADM\_Assessments] | Table used to filter out only the Assessments that the dashboard is interested in |
| [dbo].[PowerBiIdMapping] | This table is used to map the powerbi userid/email address to the correct Staff district email address and StaffUSI in the EdFi ODS |
| Cached Data Tables | **These tables hold the data that is cached from the corresponding BuildCache Views** |
| [dbo].[EADM\_DIM\_GradeLevel\_Cache] |  |
| [dbo].[EADM\_DIM\_Schools\_Cache] |  |
| [dbo].[EADM\_DIM\_Student\_Cache] |  |
| [dbo].[EADM\_DIM\_StudentChronicAbsence\_Cache] |  |
| [dbo].[EADM\_DIM\_StudentRace\_Cache] |  |
| [dbo].[EADM\_DIM\_StudentSpecialEd\_Cache] |  |
| [dbo].[EADM\_DIM\_StudentStaff\_Cache] |  |
| [dbo].[EADM\_DIM\_Subject\_Cache] |  |
| [dbo].[EADM\_DIM\_UserSecurity\_Cache] |  |
| [dbo].[EADM\_FACT\_Assessment\_NCE\_Cache] |  |
| [dbo].[EADM\_FACT\_AssessmentNumbersPivot\_Cache] |  |
| [dbo].[EADM\_FACT\_DomainLevelScores\_Cache] |  |
| [dbo].[EADM\_FACT\_SkillLevelScores\_Cache] |  |
| [dbo].[EADM\_FACT\_StudentAssessmentPerformanceLevel\_Cache] |  |
| [dbo].[EADM\_student\_ELL\_Cache] |  |
| [dbo].[EADM\_student\_gifted\_Cache] |  |
| [dbo].[EADM\_StudentAssessment\_Cache] |  |
| [dbo].[EADM\_StudentSchoolYears\_Cache] |  |
| [dbo].[EADM\_SummaryView\_Cache] |  |
| Views |  |
| Views used to build Cache |  |
| [dbo].[v\_EADM\_DIM\_UserSecurity\_BuildCache\_BuildCache] | This view joins the dbo.PowerBiIdMapping table to the dbo.v\_EADM\_DIM\_StudentStaff\_BuildCache view to give the fronend a source for row level security. |
| [dbo].[v\_EADM\_DIM\_StudentStaff\_BuildCache\_BuildCache] | This view associates all the students that have an Assessment in the dbo.EADM\_Assessments table to every EDFI StaffUSI that has access to that student. |
| [dbo\_BuildCache].[v\_EADM\_DIM\_Schools\_BuildCache\_BuildCache] | This view returns a list of the schools that have students with assessments from the EducationOrganization table and the Associated tables for the frontend to use if needed. |
| [dbo\_BuildCache].[v\_EADM\_DIM\_Student\_BuildCache\_BuildCache] | Returns all of the demographic information about a student as well as associated gifted and Ell flags |
| [dbo\_BuildCache].[v\_EADM\_StudentAssessment\_BuildCache\_BuildCache] | Returns all StudentAssessment rows for students that have an assessment in the dbo.EADM\_Assessment table. |
| [dbo\_BuildCache].[v\_EADM\_student\_ELL\_BuildCache] | Returns and ELL Flag for each student in dbo.v\_EADM\_StudentAssessment |
| [dbo].[v\_EADM\_student\_gifted\_BuildCache] | Returns a gifted flag for each student in dbo.v\_EADM\_StudentAssessment |
| [dbo].[v\_EADM\_StudentSchoolYears\_BuildCache] | Returns all the SchoolYears that each student is associated with |
| [dbo].[v\_EADM\_SummaryView\_BuildCache] | Returns the bulk of the data for the Main dashboard page |
| [dbo].[v\_EADM\_FACT\_AssessmentNumbersPivot\_BuildCache] | Pivots the NCE and metric data to be consumed by the EADM\_SummaryView\_BuildCache |
| [dbo].[v\_EADM\_DIM\_StudentChronicAbsence\_BuildCache] | Returns aggregated absence information for students that have an Assessment |
| [dbo].[v\_EADM\_DIM\_StudentRace\_BuildCache] | Retuns the associated race(s) for students in the selection set |
| [dbo].[v\_EADM\_FACT\_Assessment\_NCE\_BuildCache] | Returns the NCE (Normal Curve Equivalent) values for the assessment scores for each assessment score that a student received |
| [dbo].[v\_EADM\_FACT\_StudentAssessmentPerformanceLevel\_BuildCache] | Returns all of the performance level information associated with a student assessment. |
| [dbo].[v\_EADM\_FACT\_DomainLevelScores\_BuildCache] | Returns all Domain level scores for each Assessment for each student |
| [dbo].[v\_EADM\_FACT\_SkillLevelScores\_BuildCache] | Returns all the Skill level scores for each Assessment for each domain for each student |
| Views to Surface Data from Cache or used by Power Bi | **These views are wrappers for the tables with the same name with \_Cache to provide a layer of indirection for the PowerBi Layer should it be necessary.** |
| [dbo].[v\_EADM\_DIM\_UserSecurity] | Return all rows from the EADM\_DIM\_UserSecurity\_Cache table |
| [dbo].[v\_EADM\_DIM\_StudentStaff] | Return all rows from the EADM\_DIM\_StudentStaff\_Cache table |
| [dbo].[v\_EADM\_DIM\_Schools] | Return all rows from the EADM\_DIM\_Schools\_Cache table |
| [dbo].[v\_EADM\_DIM\_Student] | Return all rows from the EADM\_DIM\_Student\_Cache table |
| [dbo].[v\_EADM\_StudentAssessment] | Return all rows from the EADM\_StudentAssessment\_Cache table |
| [dbo].[v\_EADM\_student\_ELL] | Return all rows from the EADM\_student\_ELL\_Cache table |
| [dbo].[v\_EADM\_student\_gifted] | Return all rows from the EADM\_student\_gifted\_Cache table |
| [dbo].[v\_EADM\_StudentSchoolYears] | Return all rows from the EADM\_StudentSchoolYears\_Cache table |
| [dbo].[v\_EADM\_SummaryView] | Return all rows from the EADM\_SummaryView\_Cache table |
| [dbo].[v\_EADM\_FACT\_AssessmentNumbersPivot] | Return all rows from the EADM\_FACT\_AssessmentNumbersPivot\_Cache table |
| [dbo].[v\_EADM\_DIM\_StudentChronicAbsence] | Return all rows from the EADM\_DIM\_StudentChronicAbsence\_Cache table |
| [dbo].[v\_EADM\_DIM\_StudentRace] | Return all rows from the EADM\_DIM\_StudentRace\_Cache table |
| [dbo].[v\_EADM\_FACT\_Assessment\_NCE] | Return all rows from the EADM\_FACT\_Assessment\_NCE\_Cache table |
| [dbo].[v\_EADM\_FACT\_StudentAssessmentPerformanceLevel] | Return all rows from the EADM\_FACT\_StudentAssessmentPerformanceLevel\_Cache table |
| [dbo].[v\_EADM\_FACT\_DomainLevelScores] | Return all rows from the EADM\_FACT\_DomainLevelScores\_Cache table |
| [dbo].[v\_EADM\_FACT\_SkillLevelScores] | Return all rows from the EADM\_FACT\_SkillLevelScores\_Cache table |
| Stored Procedure(s) |  |
| dbo.sp\_EADM\_LoadCacheTables | This procedure clears and loads the cache tables from the \_BuildCache views |

# Power Bi Setup and Configuration

## Power BI initial setup and configuration:

1. **Obtain Power BI license.**
   1. Your organization Office 365 administrator will need to add this license to your Office 365 subscriptions.
      1. There are 3 types of subscription for Power-BI
         1. Power BI (free)
         2. Power BI (Pro)
         3. Power BI (Premium)
      2. The types of subscriptions will really depend on the capacities
      3. of usage that your organization will be leveraging the Power BI.
2. **Setup Power BI Workspace**

Power BI workspace is where you will be sharing your dashboard with your colleagues.

* 1. Create new workspace and assign members to your workspace.

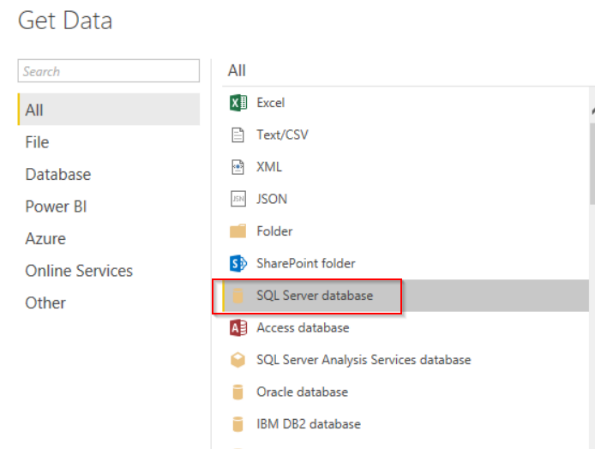
*Note*: Make sure to *set members to view only if you are implementing RLS* on your dashboard.

1. **Setup and configure On-Premise data gateway** 
   1. Download On-Premise data gateway to your local server where you’ll be hosting your on-premise connection.
   2. Install On-premises data gateway on local server and setup/login to Power BI using power bi login
   3. Once you have established connection On-premise gateway with Power BI services. Setup Data Source connection with On-Premise Data Gateway.
   4. From Power BI services create new On-Premise data gateway connection with your local hosted data base server.

*Note*: This step should be performed by local network administrator who has administrative rights the local servers where the data source resides. Also, please be aware that if you have heavy loads of users accessing the reports at the same time will hinder the performance issues of how long the load time will be. I would recommend that the on-premise gateway be installed on a server with good memory/processors.

1. **Install Power BI Desktop**
   1. Download Power BI desktop from PowerBi.com
   2. Install the 64-bit version of the application, unless you still have the 32-bit version of Windows OS.
2. **Data Source Setup and Configuration in Power BI service or Power BI Desktop**

**Note:** *This step will need to be repeated every time there is a new data source added to the dashboard. Setting up data source can be done through the Power BI service or during dashboard develop in Power BI desktop.*

* 1. Setup and configuration using Power BI service
     1. Login to PowerBi.com
     2. *Select Manage Gateways*
        1. *Select provided gateway and “Add Data Source”*
           1. *Specific all required fields and connect to your local SQL database*
           2. *Test to make sure connection was successful*
  2. Setup and configuration using Power BI Desktop
     1. Click Get Data - > SQL Server -> Connect
        1. Specific your server name (local server name) of SQL Server
        2. *Database (optional) – Specific a database* 
           1. (Leave data connectivity mode to “DirectQuery”)
* 
  + 1. *Input servername and data source*
    2. *Once load -> Select required tables-> Load*

## EQ 24 Design and Configuration:

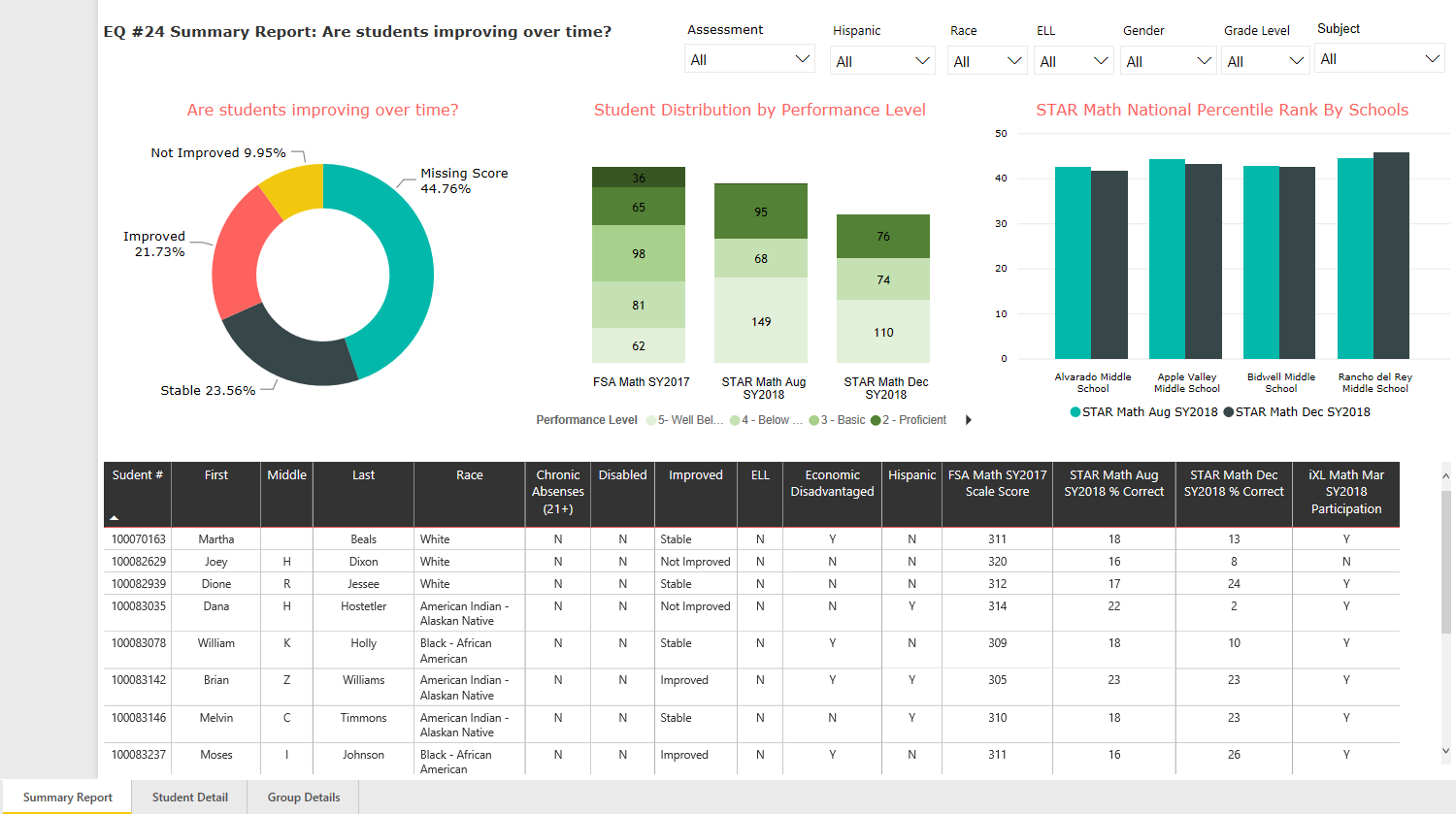
1. **Working with Power BI Desktop**
   1. Launch application and sign-in using your Power Bi user login
   2. Load provided EQ24 .PBIX file
   3. Verify all data source connection are working
2. **Design Views (***Please view Detail Section below for complete breakdown of each view***)** 
   1. Summary View (Default Landing Page)
   2. Student View
   3. Group View
3. **Security Setup and Implementation (Further detail breakdown below)** 
   1. (RLS) Row Level Security Setup to filter current user view of data
      1. Create Roles and Rules
      2. *Note: When publishing new desktop file, dataset permissions are removed from Power Bi services. You will need to re-add it back to make it work again.*
   2. Assign permission (Power BI Service – on cloud)
      1. Assign Power BI user to Roles

# EQ 24 Detail Design View Breakdown

EQ 24 Dashboard is broken down into 3 separate views. The default view is the Summary view, follow by 2 detail views. The 2 detail views are drill-through selection from the summary view. All views are inter-linked to one-another.

## Summary View (Default landing page)

Screen Capture:

**

## Summary View (Individual data grid and charts breakdown)

|  |  |  |
| --- | --- | --- |
| **Visual** | **Chart Type** | **Sources:** |
|  | Donut chart | **Data Used:** v\_EADM\_SummaryView(2)  **Fields Used:** Status, StudentUSI  **-----------------------**  **Legend**: Status  **Value**: Student USI - Counts |
|  | Stacked column chart | **Data Used:** v\_EADM\_FACT\_Student  AssessmentPerformanceLevel(2)  **Fields Used:** Assessment, StudentUSI, - Custom groups ( Advance)  **-----------------------**  **Axis:** Assessment  **Legend**: Performance Level  **Value**: StudentUSI - Counts |
|  | Clustered column chart | **Data Used:** v\_EADM\_SummaryView(2)  **Fields Used:** NameofInstitution, StarMathAug2018\_NationtalPercentile,  StarMathDec2018\_NationalPercentile  **-----------------------**  **Legend**: Status  **Values**: Student USI - Counts |
|  | Table | **Data Used: v\_EADM\_SummaryView(2)**  **Fields Used:** Abscritical, Disabled, EconomicDisadv,ELL,Firstname,FSAMO\_2017\_Scale  Hispanic,IXL,LastSurname,MiddleName,race,  StarMathAug2018\_NationalPercentile, StarMathDec2018\_NationalPercentile,  Status,StudentUSI  **-----------------------** |

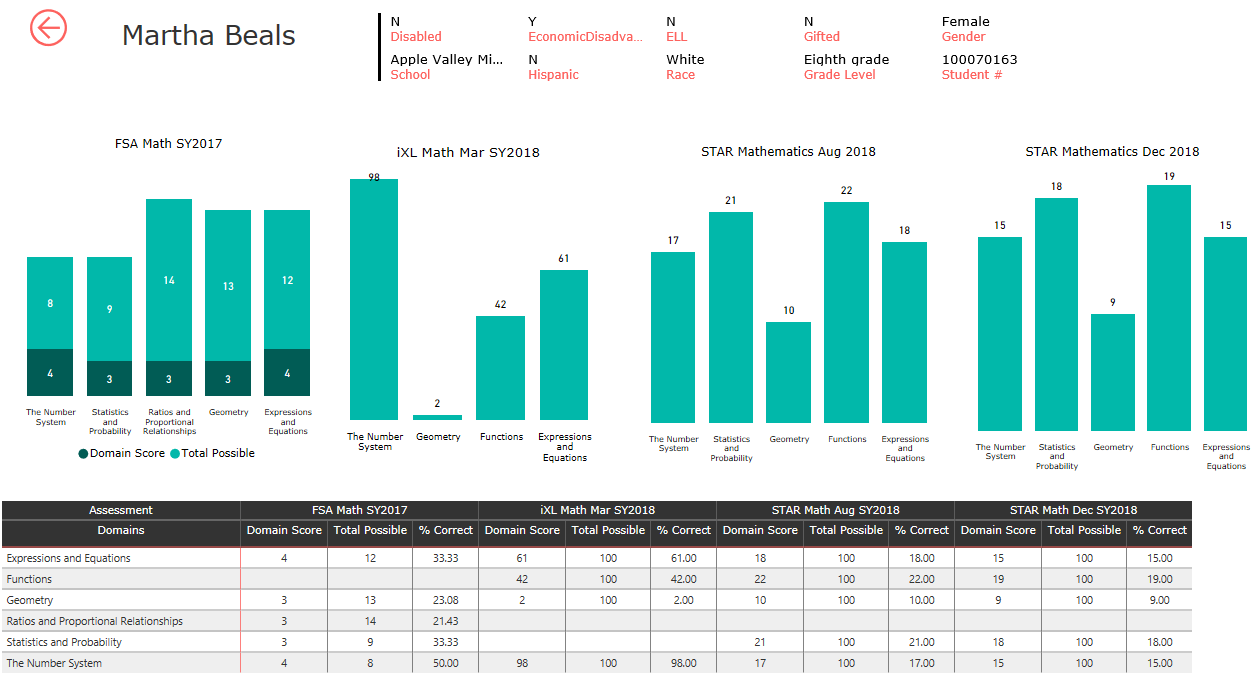
## Summary View (Code Behind)

* *Note: All are created and added using the* ***v\_EADM\_SummaryView(2****) dataset*
* *To view the Quires: Click on the Home -> Edit Queries*

|  |  |  |
| --- | --- | --- |
| **Summary View Custom Columns** | | |
| **Column Name** | **Purpose** | **Formula** |
| Abs Critical | Find Students Absent > 21 times | = Table.AddColumn(#"Nec Gains #", "AbsCritical", each if [Absences] > 21 then "Y" else "N") |
| Disabled | Find all students who are disabled | = Table.AddColumn(#"Abs Critical", "Disabled", each if [DisabilityStatus] = "NonDisabled" then "N" else "Y") |
| HispanicEthnicity | Find all students who are Hispanics | = Table.AddColumn(Disabled, "Hispanic", each if [HispanicLatinoEthnicity] = true then "Y" else "N") |
| Chg12 | Find the student scores difference between StarMathAug2018\_NCE and FSAMO\_2017\_NCE | = Table.AddColumn(HispanicEthnicity, "chg12", each [StarMathAug2018\_NCE] - [FSAMO\_2017\_NCE]) |
| Chg23 | Find the student scores difference between StarMathDEC2018\_NCE and StarMathAug2018\_NCE | = Table.AddColumn(#"Added Custom", "Chg23", each [StarMathDec2018\_NCE]-[StarMathAug2018\_NCE]) |
| Chg13 | Find the student scores difference between StarMathDEC2018\_NCE and FSAMO\_2017\_NCE | = Table.AddColumn(#"Added Custom1", "Chg13", each [StarMathDec2018\_NCE]-[FSAMO\_2017\_NCE]) |
| Results | Find the Decision\_Score of the student | = Table.AddColumn(#"Added Custom2", "Decision\_Score", each (([chg12]\*.8)+([Chg23]\*1.2)+([Chg13]))/3) |
| Status | Determine the STATUS of the students | = Table.AddColumn(#"Added Custom3", "Status", each if [Decision\_Score] > 6.62 then "Improved" else if [Decision\_Score] < -6.62 then "Not Improved" else if [Decision\_Score] = null then "Missing Score" else "Stable") |
| Summary View Measurement | | |
| Name | Purpose | Formula |
| Change 1 to 2 | Determine if student from chg1 to chg 2 improved or not | Change 1 to 2 = if('v\_EADM\_SummaryView (2)'[chg12] >6.62 ,"Improved","Not Improved") |
| Change 1 to 3 | Determine if student from chg 1 to chg 3 improved or not | Change 1 to 3 = if('v\_EADM\_SummaryView (2)'[Chg13] > 6.62,"Improved","Not Improved") |
| Change 2 to 3 | Determine if student from chg 2 to chg 3 improved or not | Change 2 to 3 = if('v\_EADM\_SummaryView (2)'[Chg23] > 6.62,"Improved","Not Improved") |
| Decision\_Score2 |  | Decision\_Score2 = (('v\_EADM\_SummaryView (2)'[chg12]\*.8)+('v\_EADM\_SummaryView (2)'[Chg23]\*1.2)+('v\_EADM\_SummaryView (2)'[Chg13]\*1))/3 |

## Student Detail View (Default landing page)

Screen Capture:



## Student Detail View (Individual data grid and charts breakdown)

|  |  |  |
| --- | --- | --- |
| **Visual** | **Chart Type** | **Sources:** |
|  | Stacked column chart | **Data Used:** V\_EADM\_FACT\_DomanLevelScores  **Fields Used:** description,MaxRawScroe,RawScore  **Filters**: FSA Math SY2017  **-----------------------**  **Axis**: Doman  **Value**: Domain Score, and Total Possible |
|  | Stacked column chart | **Data Used:** V\_EADM\_FACT\_DomanLevelScores  **Fields Used:** Description, RawScore  **Filters**: iXL Math Mar SY2018  **-----------------------**  **Axis:** Domain  **Value**: Domain Score |
|  | Stacked column chart | **Data Used:** V\_EADM\_FACT\_DomanLevelScores  **Fields Used:**  Description, RawScore  **Filters**: STAR Mathematics Aug 2018  **-----------------------**  **Legend**: Domain  **Values**: Domain Score |
|  | Stacked column chart | **Data Used:** V\_EADM\_FACT\_DomanLevelScores  **Fields Used:** Description, RawScore  **Filters**: STAR Mathematics Dec 2018  **-----------------------**  **Legend**: Domain  **Values**: Domain Score |
|  | Table | **Data Used:** V\_EADM\_FACT\_DomanLevelScores  **Fields Used:** Description, RawScore  **Filters**: STAR Mathematics Dec 2018  **-----------------------**  **Legend**: Domain  **Values**: Domain Score |
|  | Card | **Data Used:** V\_EADM\_FACT\_DomanLevelScores  **Fields Used:** FirstLast  **Filters**: StudentUSI |
|  | Multi-Row Card | **Data Used:** V\_EADM\_FACT\_DomanLevelScores  **Fields Used:** Disabled,EconomicDisadvantaged,  ELL,Gifted,Gender,School,Hispanic,Race,Grade Level, StudentUSI  **Filters**: Student USI  **-----------------------**  **Rows**: Domain  **Columns**: Assessment  Values: Domain Scores, Total Possible,  % Correct |

## 

## Student Detail View (Code Behind)

* *Note: All are created and added using the* ***v\_EADM\_FACT\_DomainLevelScores***
* *To view the Quires: Click on the Home -> Edit Queries*

|  |  |  |
| --- | --- | --- |
| **Student Detail View Custom Columns** | | |
| **Column Name** | **Purpose** | **Formula** |
| Abs Critical | Find Students Absent > 21 times | = Table.AddColumn(dbo\_v\_EADM\_FACT\_DomainLevelScores, "AssessmentNameChange", each if [AssessmentTitle] = "FSAMO" then "FSA Math SY2017" else if [AssessmentTitle] = "StarMathAug2018" then "STAR Math Aug SY2018" else if [AssessmentTitle] = "StarMathDec2018" then "STAR Math Dec SY2018" else if [AssessmentTitle] = "IXLMathMar2018" then "iXL Math Mar SY2018" else null) |
| **Student Detail View Measurement** | | |
| Name | Purpose | Formula |
| Assessment |  | Assessment = CONCATENATE(v\_EADM\_FACT\_DomainLevelScores[AssessmentTitle], CONCATENATE(" ",v\_EADM\_FACT\_DomainLevelScores[Version])) |
| FirstLast |  | FirstLast = CONCATENATE(v\_EADM\_FACT\_DomainLevelScores[FirstName], CONCATENATE(" ",v\_EADM\_FACT\_DomainLevelScores[LastSurname])) |
|  |  |  |
|  |  |  |

## Group Details View

Screen Capture:

## Group Detail View (Individual data grid and charts breakdown)

|  |  |  |
| --- | --- | --- |
| **Visual** | **Chart Type** | **Sources:** |
|  | Stacked column chart | **Data Used:** V\_EADM\_FACT\_DomanLevelScores  **Fields Used:** description,MaxRawScroe,RawScore  **Filters**: FSA Math SY2017  **-----------------------**  **Axis**: Doman  **Value**: Domain Score, and Total Possible |
|  | Stacked column chart | **Data Used:** V\_EADM\_FACT\_DomanLevelScores  **Fields Used:** Description, RawScore  **Filters**: iXL Math Mar SY2018  **-----------------------**  **Axis:** Domain  **Value**: Domain Score |
|  | Stacked column chart | **Data Used:** V\_EADM\_FACT\_DomanLevelScores  **Fields Used:**  Description, RawScore  **Filters**: STAR Mathematics Aug 2018  **-----------------------**  **Legend**: Domain  **Values**: Domain Score |
|  | Stacked column chart | **Data Used:** V\_EADM\_FACT\_DomanLevelScores  **Fields Used:** Description, RawScore  **Filters**: STAR Mathematics Dec 2018  **-----------------------**  **Legend**: Domain  **Values**: Domain Score |
|  | Table | **Data Used:** V\_EADM\_FACT\_DomanLevelScores  **Fields Used:** Description, RawScore  **Filters**: STAR Mathematics Dec 2018  **-----------------------**  **Legend**: Domain  **Values**: Domain Score |
|  | Card | **Data Used:** V\_EADM\_FACT\_DomanLevelScores  **Fields Used:** Status  **Filters**: Status |
|  | Card | **Data Used:** V\_EADM\_FACT\_DomanLevelScores  **Fields Used:** LastSurname  **Filters**: Status |
|  | Table | **Data Used:** V\_EADM\_FACT\_DomanLevelScores  **Fields Used:** Description, RawScore  **Filters**: iXL Math Mar SY2018 |

## 

## Student Detail View (Code Behind)

* *Note: All are created and added using the* ***v\_EADM\_FACT\_DomainLevelScores***
* *To view the Quires: Click on the Home -> Edit Queries*

|  |  |  |
| --- | --- | --- |
| **Group Detail View Custom Columns** | | |
| **Column Name** | **Purpose** | **Formula** |
| Abs Critical | Find Students Absent > 21 times | = Table.AddColumn(dbo\_v\_EADM\_FACT\_DomainLevelScores, "AssessmentNameChange", each if [AssessmentTitle] = "FSAMO" then "FSA Math SY2017" else if [AssessmentTitle] = "StarMathAug2018" then "STAR Math Aug SY2018" else if [AssessmentTitle] = "StarMathDec2018" then "STAR Math Dec SY2018" else if [AssessmentTitle] = "IXLMathMar2018" then "iXL Math Mar SY2018" else null) |
| **Student Detail View Measurement** | | |
| Name | Purpose | Formula |
| Assessment |  | Assessment = CONCATENATE(v\_EADM\_FACT\_DomainLevelScores[AssessmentTitle], CONCATENATE(" ",v\_EADM\_FACT\_DomainLevelScores[Version])) |
|  |  |  |
|  |  |  |
|  |  |  |

# EQ 24 Power BI Security Setup and Configuration

RLS (Row Level Security) was implemented to filter the data the current user that is logged onto the dashboard can see. There are 2 areas where filter needs to be set to make this happens

1. In Power BI Desktop
   1. Select “Modeling” Tab
   2. Select Manage Role
      1. There should already be pre-defined role called “Security”
         1. This is the role that is tied to a view in the data source called v\_User\_Security . This view has a column called PBIEmailAddress. This column contains all PowerBI login account that are tied to the Staff and Student joins.
         2. RLS expression:
            1. Security = v\_EADM\_UserSecurity=DAX Expression
2. In Power BI Service
   1. Login to PowerBI.com
   2. Select you Workspace
   3. On the left side menu, right click on the dashboard name below the header DATASETS
      1. Select “SECURITY”
      2. You will see only one folder called Security
      3. Add all Power BI user email address that you would like to apply the RLS too.

After you add the user Power BI email address to the SECURITY folder, you could test this by asking that specific user to login into the dashboard. The specific user should only see his data that pertain to him loaded.