VDW Validation Scripts

# Overview

The VDW Validation scripts, VDW\_DataValidation.sql and VDW\_Schema\_Validation.sql, are provided to give a high level of checking for how your VDW is structured as well as validate certain fields against an expected list of values. They were introduced starting in Summer 2019. When running these scripts it's important to note that they are written according to the Data Model Manual. This means that customizations like additional fields, tables, and additional values not found in the Data Model Manual are not validated or may be flagged as something that is out of spec.

When reviewing the output, check any flagged items against the Data Model Manual to determine if this would cause a problem querying data.

An example of a flagged item would be if the VITAL\_SIGNS table was named VITAL\_SIGN. This incorrect table name would cause a normal query to be unable to find the table because a CHORDS query would be looking for VITAL\_SIGNS not VITAL\_SIGN. This would be an item that should be corrected.

Another example of a flagged item would be if the scale on a field in the DDL was DECIMAL(20,6) but in your VDW it was DECIMAL(20,7). This would be flagged by the code as an inconsistency because the scale is greater than expected. You might determine that while this is different, no action is going to be taken because a value should still be able to be queried. In this example the scale might be higher because your organization determined a higher scale was required. This might be an opportunity to share your findings with the CHORDS data team to potentially update the DDL. If this situation were reversed and your data type was DECIMAL(20,5) that may be a problem because we would be expecting more data to be stored in your VDW and this may be a problem for your ETL process because some implicit rounding or truncation may be occurring when you load your VDW.

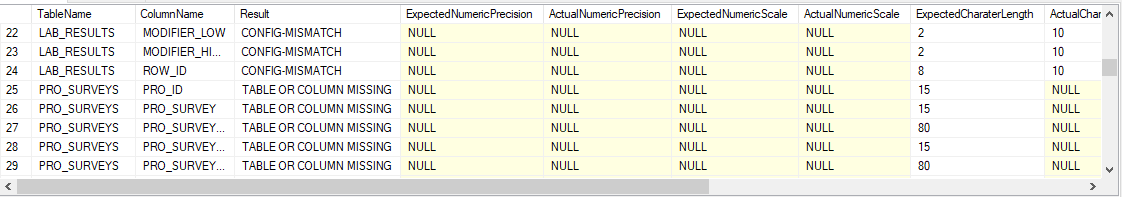
The scripts are best run from SQL Server Management Studio.

# VDW\_Schema\_Validation.sql Script

This script validates table names, column names, column character length, column numeric length, column precision, and if a column allows null values. Because this script just validates structure, it should run fairly quickly. It should also run against Views.

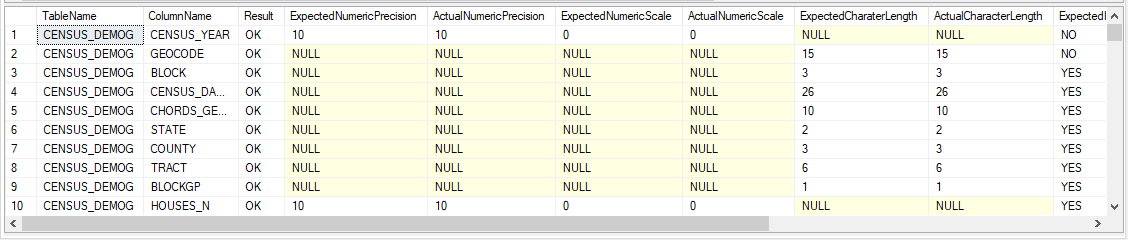
To run the script:

1. Open Management Studio and either open the VDW\_Schema\_Validation.sql file or paste the text of the file into an open query window.
2. Ensure the selected database is your VDW.
3. Run the script.
4. Two results will be created.
   1. The first set of results shown is any tables or columns that were either not found or are out of spec. The Result column will give you some clues about what the problem might be.



A CONFIG-MISMATCH means the column was found but the length, numeric scale, numeric precisions, or null-ability of the column did not match the DDL. A TABLE OR COLUMN MISSING value means either the Table could not be found or the Column in the table could not be found.

* 1. The second set of results is the complete result which shows all of the fields that were checked. If the Result Column is OK, then the column was found in the correct table and the data properties of the column match to the specs in the DDL.



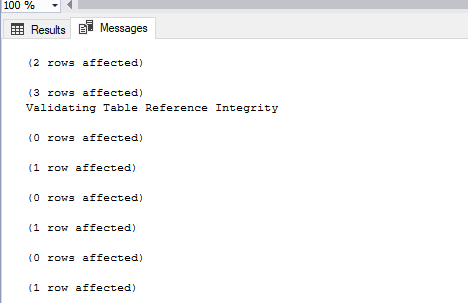
You can save your results to a spreadsheet to reference later. You may need to close and reopen the script to clear any temporary tables created by the validation script.

# VDW\_DataValidation.sql Script

This script tries to validate the values in your VDW for many of the columns. Columns like Address\_Type are validated against the expected list and any unexpected values are flagged for review. This script also validates tables that reference each other. For example, validating the PERSON\_IDs in the ENCOUNTERS table exist in the DEMOGRAPHICS table. This data validation and integrity check is important to ensure data can be queried effectively from your VDW.

To run the script:

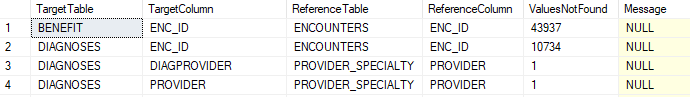
1. Open Management Studio and either open the VDW\_DataValidation.sql file or paste the text of the file into an open query window.
2. Ensure the selected database is your VDW.
3. Run the script.
   1. Because of all of the validation checks, this script may take some time to run. If the script runs for more than 30 minutes, check the Messages tab in Management Studio. You may be able to see some hints about the progress



* 1. Some of the messages you may see are **Validating Table Values Integrity**, which is validating the values in certain fields (like BENEFIT\_CAT) and **Validating Table Reference Integrity**, which is checking that values in certain columns can be referenced back to a source table (like the PERSON\_IDs in ENCOUNTERS can be found in DEMOGRAPHICS).

1. Two results will be created.
   1. The first set of results any tables where values in specific columns contain values that cannot be referenced back to a source table

The target table values in the target column were validated against what exists in the reference table in the reference column. The ValuesNotFound column is a count of the number of values from a column in the target table not found the specific column of the reference table.



Only summary data is displayed here. You can run a separate query on your own to find individual values to check. A simple example of a query to find the actual rows values would be:

SELECT \*

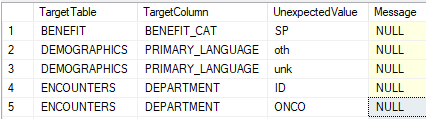
FROM [TargetTable] a

LEFT JOIN [ReferenceTable] b ON b.[ReferenceColumn] = a.[TargetColumn]

WHERE b.[ReferenceColumn] IS NULL

Replace the bracketed items with the specific tables and columns you want to see the full row values. If you have thousands of results, take care in running a query.

* 1. The second set of results shows any values found in certain columns that do not exist in the Data Model Manual for that column.



Only unexpected values are displayed here. You can run a separate query on your own to see the full row. A simple example of a query to find the actual rows values would be:

SELECT \*

FROM [TargetTable]

WHERE [TargetColumn] = '[UnexpectedValue]'

Replace the bracketed items with the specific tables, columns and values you want to see the full row values.