Data Profiling and Analysis Tool Tutorial

Setup

Start by running the two MySQL database files in the database folder. Before anything can be automated, the customized queries script needs to be run so that any sources and connections needed to be loaded in can be. From there, the automatic query builder and analyzer can be used.

In addition to the scripts, xmlstarlet must be installed. This is to work around an issue with DataCleaner 2.5.1. The tool will be upgraded for DataCleaner for 3.x, which resolves several issues.

The conf_template.xml needs to be stored in the DataCleaner install folder.

PDI kettle.properties properties

There are some added kettle properties to add to your system:

Property	Description	Value
data_cleaner_filename_par	Default filename used for DataCleaner files	${etl_file_extract_location_par}/quality/${profile_source_name_par}/${profile_source_table_name_par}/${profile_source_table_name_par}_${profile_sampling_query_pk_par}.csv
etl_file_extract_location_par	Location where extracts are to be stored.	
data_profile_settings_location_par	Place where the profile_customization folder is stored	
data_cleaner_location_par	Place where DataCleaner is installed.	
etl_source_location_par	Place where etl code is stored.	

Building Source Table Queries Automatically

There is a process in place in which to build simple SELECT queries automatically. To run the process, the following command should be run:

sh /<code_location_parent_directory>/etl_code/quality/generic/profile_sample_generator_start.sh process_auto

If queries are needing to be built for a new source, please be sure to add the source before running (see under Loading Customized Queries and Sources).

Once the process is complete, the new queries should be available to the data profiling and analysis tool.

Loading Customized Queries and Sources

Adding new sources and customized queries to the data profiling tool has never been easier! There are two Excel files. One file, the data_profile_source_list.xls allows for quickly adding sources. The second file, data_profile_custom_queries_list.xls allows for quickly adding customized queries.

These files are stored under the profile_customization folder on the profile box. Copy them to your local computer, update them, and copy them back to the profile box. The following command can then be run to process the new sources and queries:

sh /<code_location_parent_directory>/etl_code/quality/generic/profile_sample_generator_start.sh process_custom

The fields of the two Excel files are as follows.

data_profile_source_list.xls fields

Field Name	Description
source_name	The unique name of the source
source_type	The type of database/file.
source_schema	The schema name of the database (if needed)
source_connection_name	The unique name of the source connection.
source_tunnel_script	For using the tool locally, a tunneling script may be required.

data_profile_custom_queries_list.xls fields

Field Name	Description
source_name	The name of the source that the query is intended for.
table_name	The name of the table the query is for.
table_query_number	The unique number for the custom query. Number is manually incremented in the file.

custom_query	The custom query to be used by the profiling tool.
query_type	The two valid types of queries for custom use is ‘profiling’ and ‘sanity’.

Running Profile and Analysis Jobs

Running a profile or analysis job has been made simpler with a shell script that takes three parameters. By running this command, the profile job will start based on the given parameters:

sh /<code_location_parent_directory>/etl_code/quality/generic/profile_sample_generator_start.sh analyze <source_name> <table_name> <query_type>

The parameters are the actual names of the entities and are **not** required for the tool to run. For example, if I just want to run all the queries for subscription_proxy my command would be:

sh /<code_location_parent_directory>/etl_code/quality/generic/profile_sample_generator_start.sh analyze subscription_proxy

I can then reduce my scope by declaring a table:

sh /<code_location_parent_directory>/etl_code/quality/generic/profile_sample_generator_start.sh analyze subscription_proxy xxrh_subscription

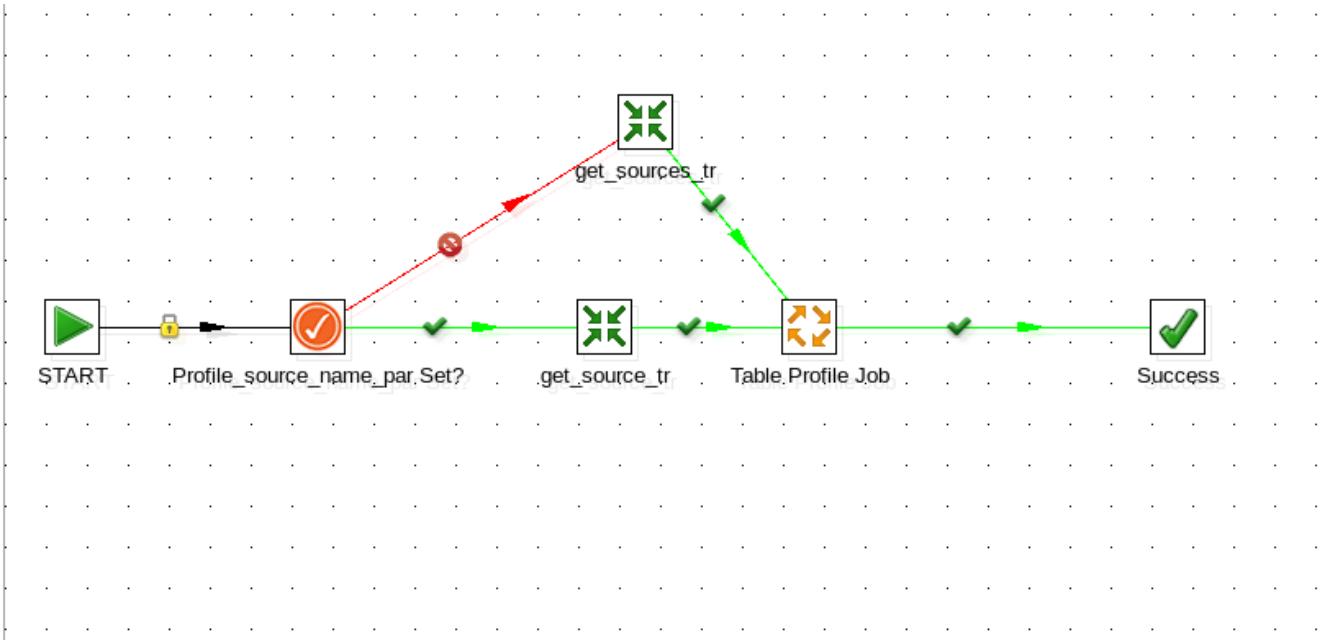
I can reduce my scope even further by declaring a query_type (i.e. Profiling):

sh /<code_location_parent_directory>/etl_code/quality/generic/profile_sample_generator_start.sh analyze subscription_proxy xxrh_subscription profiling

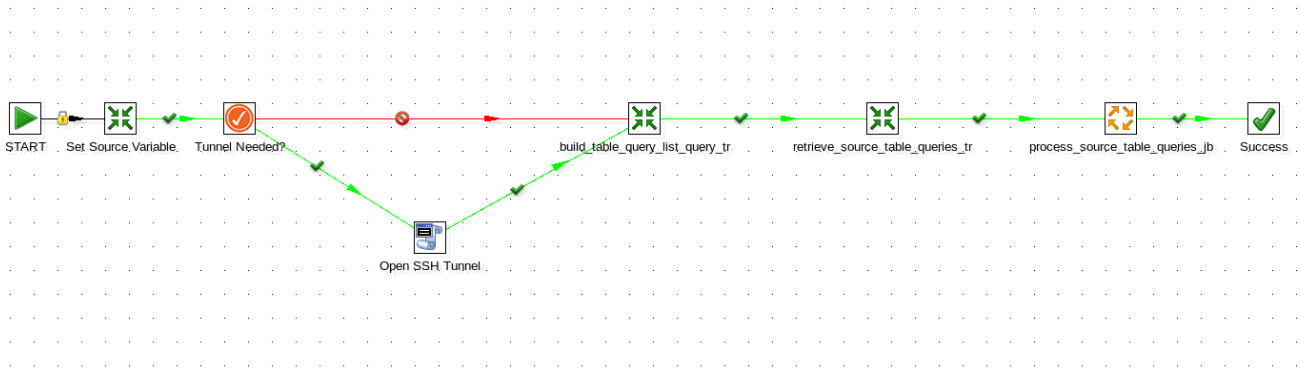
For advanced customization of the tool, please refer to the code in the profile_sample_generator_start.sh for ideas.

Detailed Description of the Analyzer Code

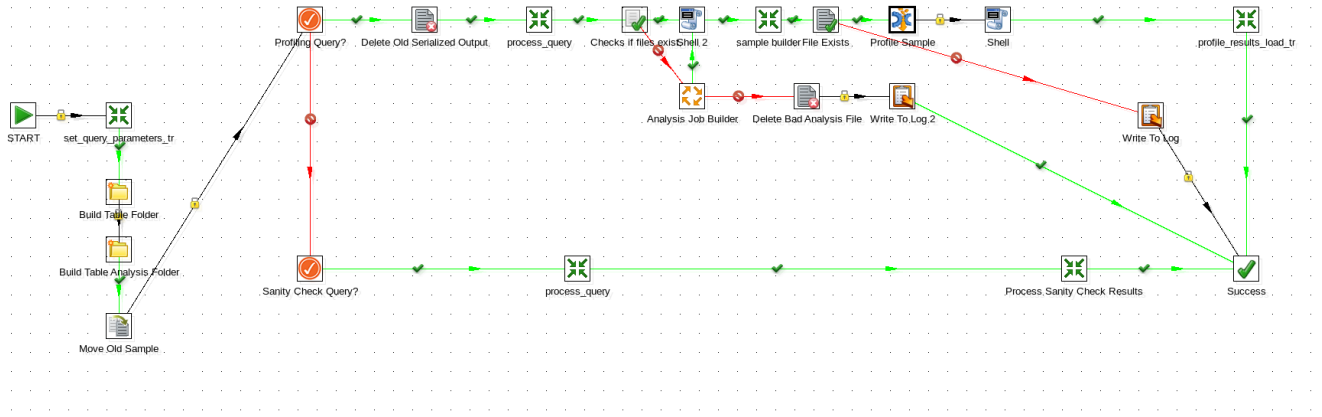
The parent job for the profiling tool is profile_sample_generator_jb.kjb. As of this writing, it appears as:



Upon starting the job, the script checks to see if the profile_source_name_par parameter is set. If it is not, the script will begin to run in automated mode and pull all sources and being to process them (as processed with get_sources_tr.ktr). If the source name is set, connection information for that given source is retrieved (as processed with get_source_tr.ktr). In either case, the source(s) are processed by table_profile_jb.kjb to obtain the list of tables and their queries.



The table_profile_jb.kjb runs any tunnel scripts that may be required (as shown via the Open SSH Tunnel step), finds all the queries for the given table that is currently being processed (as built by the build_table_query_list_query_tr.ktr script and run by the retrieve_source_table_queries_tr.ktr script), and then proceeds to run the queries in the process_source_table_queries_jb.kjb script. The most complex component of this job is the build_table_query_list_query_tr.kjb because it has to build the query based off given parameters passed by the user of the tool.



The process_source_table_queries_jb.kjb is the heart of the profiling tool. It is here that the different types of analysis are performed and their results stored in the profiling database.

The process begins by ensuring that the necessary folder structures are in place (in the event that new tables have been added) as well as renaming the existing table query sample. Then, based on the type of query will process it accordingly. Currently the profiling process is the most fleshed out part of this script. Other query types will include sanity check queries.

Profiling Query Process

The first step removes the last serialized output for the query (which can be used for troubleshooting in between runs). The query is then processed (via the retrieve_<source_name>_table_sample_tr.ktr) by a custom transformation for a given source. The data is stored in a serialized output for further consumption. If a DataCleaner analysis file already exists for a given table/query then the data is analyzed by DataCleaner. If a DataCleaner analysis file does not exist, the the analysis_job_builder_jb.kjb will process the table's metadata information and build an analysis job.

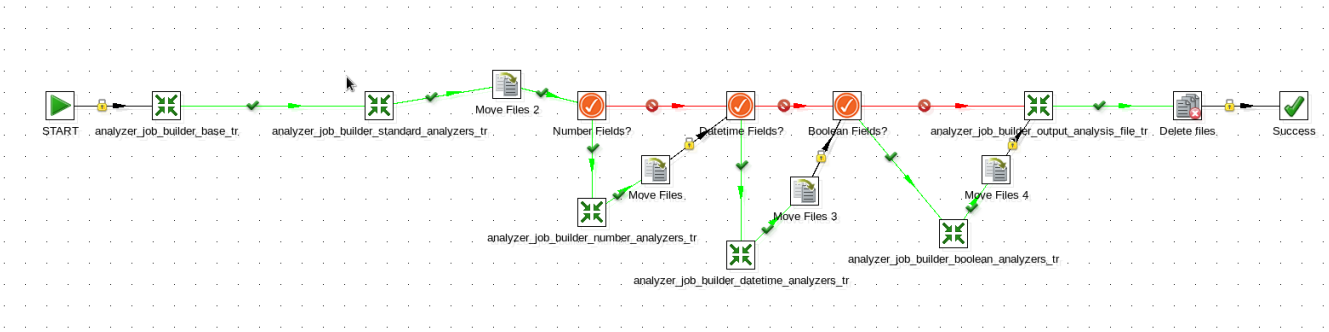
If table changes have occurred and the analysis file is out of date, just delete the analysis file and a new one will be created by the profiling tool.

Once the data is processed, the results are then broken down and stored in the profile tool's database. This will allow for long-term trending on:

- number analysis
- date/time analysis
- weekday distribution analysis
- character set distribution analysis
- string analysis
- value distribution analysis
- pattern finding analysis

There are two safeguards in place for the profiling process. One makes sure that if the analysis file builder fails to build correctly, then the file is deleted and an error message is written to the log. The other makes sure that the sample pulled from the source table is not empty. If it is, then the profiling

process is aborted and an error message is written to the log.



The analysis_job_builder_jb.kjb job file uses the table's metadata information (obtained by analyzing the sample data) and automatically produces an analysis file for DataCleaner to know how to process the sample retrieved from the source table. There is a transformation for each type of data structure that is available in DataCleaner.

Sanity Check Query Process

The intent of the sanity check process is to validate results between source tables and their target tables, where ever they may be. This is not a fully fleshed out component of the profiling/data analysis tool and will be developed as the tool continues to see use.