Creating and Loading a TRIM3 “OutOfModel” Database Table

Created by JWM Dec 18, 2017

Last Edited: Mar 8, 2019

Most of the results tables in a TRIM3 database are created by running simulations. However, increasingly, analysts are creating tables outside of the TRIM3 model and are requesting that they be uploaded to the TRIM3 database. We refer to these database tables as out-of-model tables. This documentation describes two different processes for creating and loading the tables.

* A [manual](#manual) process that can be used only by TRIM3 programmers
* An [automated](#automatic) process that can be used by TRIM3 programmers and analysts

# Manual Process

The manual process loads much faster than the automated process. The steps, which are described in detail below, are:

1. Create the table structure on the applicable database server—TRIMDB01 (the ASPE-TRIM database server), TRIMDB02 (the public TRIM3 database server) or TRIMDB03 (for ATTIS data).
2. Load the data.
3. Create the ctd.runs and ctd.rundetails records that are required to enable user access.
4. Using the microdata tool, ascertain the data are accessible.

## Create the Table Structure

NOTE: It is a good practice for all out-of-model table names to begin with the name of the input schema with which they are associated.

In SQLyog, use the following command to see how an existing table is structured:

SHOW CREATE TABLE schema.tablename;

If using SQLyog, you can then highlight the “Create Table” cell, select “Copy cell data to clipboard” from the pull-down box, paste to your query window, then edit for the new table you want to create.

Here is one example that will create an annual table:

CREATE TABLE `cpsi2016\_aca\_medicaid\_blank` (

 `HouseholdID` DECIMAL(10,0) NOT NULL DEFAULT '0',

 `PersonID` DECIMAL(10,0) NOT NULL DEFAULT '0',

 `ACAPctOfPov` FLOAT DEFAULT '0',

 `AnnualEligibilityTypeMD` FLOAT DEFAULT '0',

 `AnnualEnrollmentTypeMD` FLOAT DEFAULT '0',

 `AnnualEnrollmentTypeMR` FLOAT DEFAULT '0',

 `CitizenshipEligibility` FLOAT DEFAULT '0',

 PRIMARY KEY (`HouseholdID`,`PersonID`)

) ENGINE=INNODB DEFAULT CHARSET=latin1;

Notice the difference in the PRIMARY KEY in the following example for a monthly table:

CREATE TABLE `ac\_cs\_test1\_monthly` (

 `HouseholdID` FLOAT NOT NULL DEFAULT '0',

 `PersonID` FLOAT NOT NULL DEFAULT '0',

 `Month` FLOAT NOT NULL DEFAULT '0',

 `AssetsOfUnit` FLOAT DEFAULT NULL,

 …..

 `Test1Monthly` FLOAT DEFAULT NULL,

 PRIMARY KEY (`HouseholdID`,`PersonID`,`Month`)

) ENGINE=INNODB DEFAULT CHARSET=latin1;

## Load the Data

All data that are uploaded must first be sorted by HouseholdID and PersonID. If it is monrhly data, it should be sorted by HouseholdID, PersonID and Month. There are multiple formats and methods that may be used to load data. The following is one that I am currently using.

Request that the data be provided in CSV format. You can then use the SQLyog import function to import the data from a local drive such as the S: drive or one on your own PC.

To use the SQLyog import function, highlight the empty table you have created that you want to load with data. Right click and select “Import.” Select the CSV data option and select the file containing the data you want to import. Click the “Change” button and select “Fill with Excel Friendly Values.” Then import the data.

NOTE: I find that using this method always adds an empty row. You need to delete that (or find a way that does not create it). I have been using this SQL command to delete the empty row: “Delete from results.tablename where HouseholdID=0.”

Check to ascertain you have the correct number of records in the table—e.g., the same number of records as in the input schema person table.

## Create the Required CTD Records

In order for the microdata tool and simulation setup pages to gain access to the newly loaded data, you have to create CTD.RUNS and CTD.RUNDETAILS records for the table you have created and loaded.

If there are no monthly data, RUNID and SIMULATIONID may be the same for out-of-model tables, but you may also want to follow the practice for in-model tables—i.e., the final character of both the SimulationID and annual table name is an underscore character. Use “OutOfModel” for the simulation name and ask the analyst who requested the table who the “owner” should be. Using SQLyog, you may search for existing records in the CTD.RUNS and CTD.RUNDETAILS tables to serve as a “template” for the new records by using a command such as “Select \* from ctd.runs where runid like ‘CPSI2016%’.

## Ascertain the Data are Accessible

Ascertain that the new table shows up in the microdata tool before you alert analysts that the table is ready. A typo in table name, runid or simulationid can result in a failure of the table to be accessible. If you have to correct an error, refresh the affected schemas by highlighting the schema name and clicking on the SQLyog refresh icon. If you need to correct an error and refresh, refresh the CTD schema first, then the MICRODATA schema.

# Automated Process

The use of the automated process may be unacceptably slow, especially if the dataset is very large. However, using the procedure eliminates the need to create the table structure, load the data using SQL commands, and manually insert the ctd.runs and ctd.rundetails records.

The automated procedure is accessible from the TRIM3 website on the ASPE-TRIM3 server. From the TRIM3 Navigator, select “Internal scripts and procedures,” then “Import out-of-model results.”

The data file must be in CSV format, must have a header record with the names of all fields, must contain the unique TRIM3 idenfiers HouseholdID and PersonID, and must be sorted by those identifiers. The data file should be uploaded to the scratch drive on Cybele in the “outofmodelCSVs” folder.

Additional instructions are provided on the out-of-model website page.