IBM® InfoSphere® Optim™ Test Data Management Solution

Masking Data Using a Column Map Procedure Part 3

Welcome

Welcome to the IBM® InfoSphere® Optim™ Test Data Management Solution video series about masking data using a column map procedure.

This video is the third in a series about masking sensitive data to privatize that data. The objective of this video is to show how to mask a person's first name using a gender-specific lookup table. The Optim random lookup data privacy provider is used in this script.

Scenario Overview

This video shows how to use a random lookup data privacy provider to mask a salesperson's first name, based on the salesperson's gender. The script uses two separate lookup tables to mask male and female names. One lookup table is used to obtain gender-appropriate replacement names for males, while the other is used to obtain gender-appropriate names for females.

In the video, we show how to mask the FIRST_NAME column entry in the OPTIM_SALES table, based on the salesperson's gender in the SEX column:

- If the value in the SEX column is 'F' for female, the OPTIM_US_FIRSTNAME_F lookup table is used to obtain an appropriate female name.
- If the value in the SEX column is 'M' for male, the OPTIM_US_FIRSTNAME_M lookup table is used to obtain an appropriate male name.

This masking technique can be applied to any column for which an appropriate lookup table exists.

Masking sensitive data using a random lookup data privacy provider

This portion of the video shows how to mask the FIRST_NAME column entry in the OPTIM_SALES table, based on the person's gender. After starting Optim Designer, do the following:

1. Go to the Repository Explorer view and expand Optim Repository, and then expand the objects folder in which your column map is located. In our example, the objects folder is named **Masking on Demand**.

2. Next, expand the Column Maps folder to display a list of available column maps.

3. In this example, the column map that will be used to mask the FIRST_NAME column in the OPTIM_SALES table is named MASK.FRSTNAME.

4. Right-click the MASK.FRSTNAME column map and select "Open" to display that column map in the Column Map editor.

5. When the Column Map editor opens, scroll to the column that you want to mask and select that column by clicking on its name in the Target Column. In this example, we selected the FIRST_NAME column in the Target Column, and then clicked "Add Procedure" to open the script editor.

6. When the script editor opens, a sample script will be displayed. You can use this sample to assist you in developing your own script. If you already created a script and stored it in a text file, open that file and copy the script, and then paste it in the Column Map editor, as we have done in the next slide. You also can manually create a script by typing the appropriate information in the Column Map editor.

7. This is the script we pasted into the editor. Notice that the script uses a "cm_load ()" function to set up the two parameter strings that will be used in all Optim masking calls. The two parameters are named parms1 and parms2:

    - The parms1 parameter indicates that a DB2 RANDOM_LOOKUP table will be used to obtain replacement values for the FIRST_NAME column in the OPTIM_SALES table.

- The parms2 parameter consists of two lines that are concatenated together. The first line shows the connection information for the lookup table. The second line indicates that the FIRSTNAME column in the lookup table will be used to obtain the required replacement names for the masking process. Column attributes, such as data type and length, are also defined in this string.

The script used in this video is available at the end of the printed transcript of this narrative. That sample script is provided for you to copy and edit, as required, to meet your needs.

8. As we scrolled further down in the script, notice the information in the “cm_transform ()” area. This portion tells the script to check the SEX column entry in the OPTIM_SALES table to determine each salesperson’s gender.

   - If the person’s gender is M for male, the script will use the OPTIM_US_FIRSTNAME_M lookup table to obtain an appropriate male name for masking purposes.

   - If the person’s gender is F for female, the script will use the OPTIM_US_FIRSTNAME_F lookup table to obtain an appropriate female name for masking purposes.

   The rest of the “cm_transform ()” function instructs the script to set the name to be masked in the destination table, which is the OPTIM_SALES table, to the name obtained from the appropriate lookup table.

9. To save both the column map and the script, click File > Save All. Then close the script editor and the column map.

10. Now, we will execute the insert service that will invoke the script that we added to the column map. First, expand the “Services” folder to display a list of valid services.

11. Next, right-click the appropriate insert service, which in this case is MASK.FRSTNAME, and then select “Run Service.”

12. When Optim Manager opens, click “Run” on the Run Service popup to execute the insert service.

13. Click the “Service Monitoring” tab on Optim Manager to view the status of the request, as shown here. Notice that the Status for the service is “Ended” and the number of Rows Updated is shown in the Output section in the Process Summary area. After you’re finished viewing the output report, close Optim Manager.

14. After running the insert service, we ran an Optim Compare definition to compare the "before" and “after” images of the data that was masked. The details of the comparison are shown here and in the next slide. In the first slide, the FIRST_NAME “Signe” was changed to “BRENNA.” This name was obtained from the OPTIM_US_FIRSTNAME_F lookup table because an F for female was listed in the SEX column for Signe.

15. The second slide highlights how a male name was changed from “Dylan” to “BRYAN.” This name was obtained from the OPTIM_US_FIRSTNAME_M lookup table because an M for male was listed in the SEX column for Dylan.

    This concludes our example of how to mask a person’s first name in a table using a random lookup data privacy provider.

Sample column map procedure script used in this video

```
-- This script will mask the OPTIM_SALES.FIRST_NAME column using a random lookup.
-- The lookup table to be used is decided by the value in the (gender)
-- OPTIM_SALES.SEX column.
--
-- If the table you use for the demo has a different name for
-- the SEX column, then you will need to change the script accordingly.
--
-- This script also will need changes based on the names of your
-- lookup tables, database type, credentials, etc.  In this example,
-- optim is the Userid; optimadmin is the Password, and
-- db2 is the connect string; DB2LUW is the database type; FIRSTNAME is
-- the column name in the lookup table for replacement data; and its
-- datatype is VARCHAR with a length of 60 -- must code WVARCHAR_SZ for Lua
-- with a length of 15 to match FIRST_NAME column length in OPTIM_SALES table.
-- The lookup tables are DB2.OPTIM.OPTIM_US_FIRSTNAME_M (for male names)
-- and DB2.OPTIM.OPTIM_US_FIRSTNAME_F (for female names).
--
-- In the video, it's worth mentioning that this script demonstrates
-- the use of CM_LOAD(), a function that is called once at the start
-- of the process to set up fields that are used later.



parms1 = string.char()
parms2 = string.char()


function cm_load()
   -- Set up parm strings to be used in all optimmask calls
   -- Note: Since the parms are string.char datatypes, the parser will automatically
   -- return a value in quotes.  Thus you do not need to add the starting/ending quotes
   -- around the Provider string in the optimmask statement


   parms1 = 'PRO=RANDOM_LOOKUP,lib=DB2LUW,ID="'
```

```
   parms2 = '",REPLACE="FIRSTNAME",CONN="db2",user=optim,pass="optimadmin",'
   parms2 = parms2 .. 'FLDDEF1=(NAME="FIRSTNAME",DATATYPE=WVARCHAR_SZ,LEN=15)'
end

function cm_transform()
    -- Obtain gender of individual for row being processed
        gender = source.column.getvalue('SEX')

        -- Determine which Lookup table to use based on gender
        if gender == 'M' then
               datasource = 'OPTIM.OPTIM_US_FIRSTNAME_M'
        else
            datasource = 'OPTIM.OPTIM_US_FIRSTNAME_F'
        end

        -- Obtain current ("old") value of the first name for row being processed
        oldvalue = source.column.getvalue()

    -- Perform random lookup passing the current value to be masked and
    -- concatenating the contents of parms1 with the appropriate lookup table
    -- concatenated to the contents of parms2
        newvalue = optimmask(oldvalue,parms1 .. datasource .. parms2)

        -- Set the destination first name to the replacement value from the random lookup table
        target.column.setvalue(newvalue)
end
```