

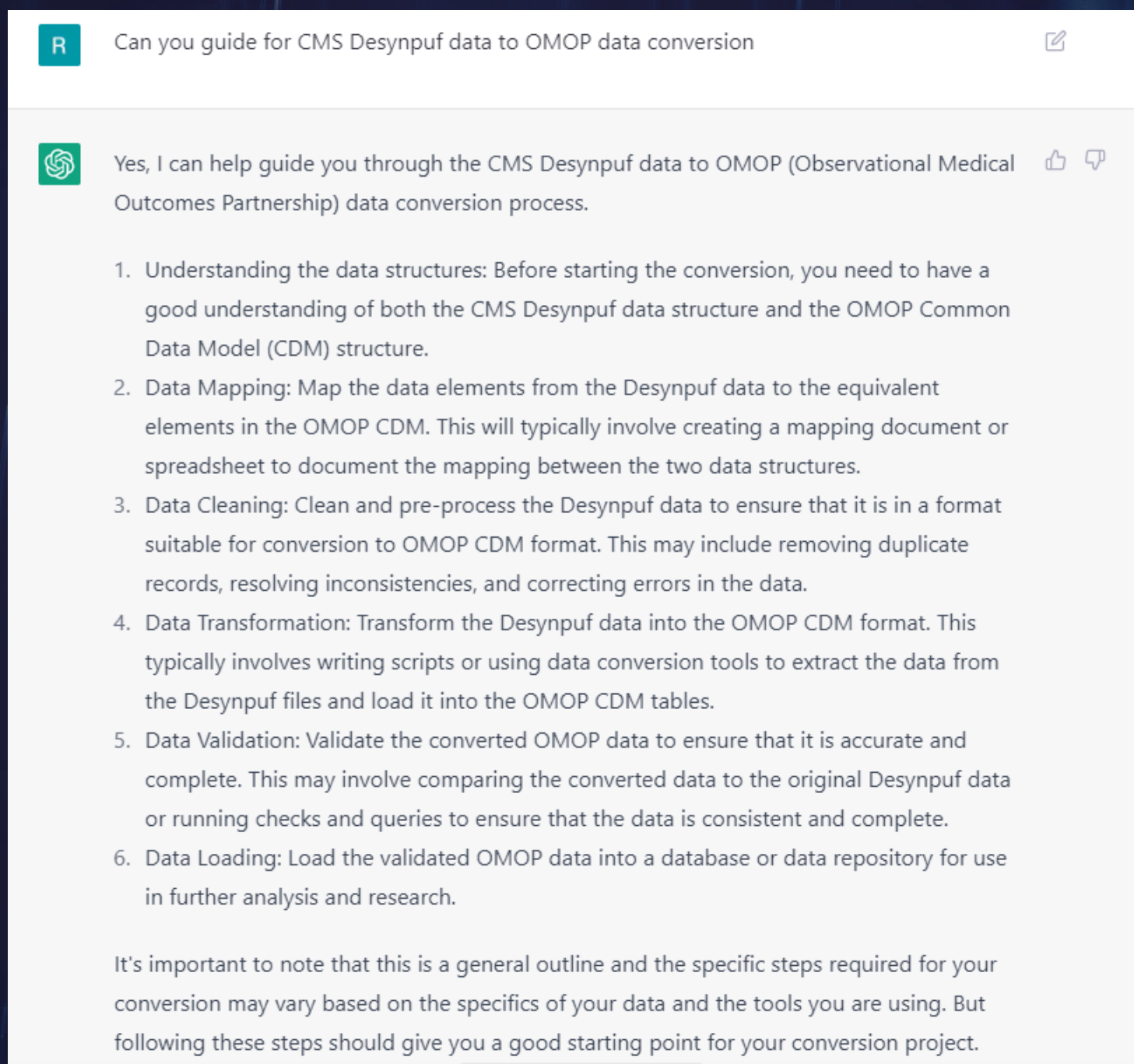


# **Alteryx-based CMS DE-SynPUF to OMOP data conversion using ChatGPT**

# Step: 1

Problem Statement: Converting CMS DE-SynPUF data to OMOP data using the Alteryx tool with the help of ChatGPT

## 1. High-level Steps: Converting CMS DE-SynPUF data to OMOP data



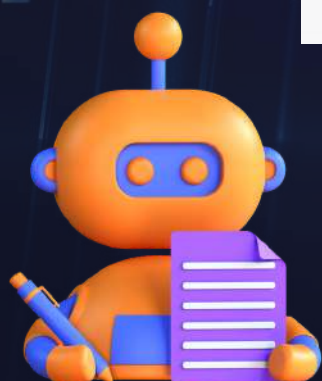
R Can you guide for CMS Desynpuf data to OMOP data conversion

Yes, I can help guide you through the CMS Desynpuf data to OMOP (Observational Medical Outcomes Partnership) data conversion process.

1. Understanding the data structures: Before starting the conversion, you need to have a good understanding of both the CMS Desynpuf data structure and the OMOP Common Data Model (CDM) structure.
2. Data Mapping: Map the data elements from the Desynpuf data to the equivalent elements in the OMOP CDM. This will typically involve creating a mapping document or spreadsheet to document the mapping between the two data structures.
3. Data Cleaning: Clean and pre-process the Desynpuf data to ensure that it is in a format suitable for conversion to OMOP CDM format. This may include removing duplicate records, resolving inconsistencies, and correcting errors in the data.
4. Data Transformation: Transform the Desynpuf data into the OMOP CDM format. This typically involves writing scripts or using data conversion tools to extract the data from the Desynpuf files and load it into the OMOP CDM tables.
5. Data Validation: Validate the converted OMOP data to ensure that it is accurate and complete. This may involve comparing the converted data to the original Desynpuf data or running checks and queries to ensure that the data is consistent and complete.
6. Data Loading: Load the validated OMOP data into a database or data repository for use in further analysis and research.

It's important to note that this is a general outline and the specific steps required for your conversion may vary based on the specifics of your data and the tools you are using. But following these steps should give you a good starting point for your conversion project.

Step Implementation Review: ChatGPT gives an accurate response in terms of specifying high-level steps to convert CMS DE-SynPUF data to OMOP data.






# Step: 2.1

## 2. Table Mapping: CMS DE-SynPUF data to OMOP data

**R** Can you show me field mapping for CMS Desynpuf file's to OMOP

 Here is an example field mapping for CMS Desynpuf data to the OMOP Common Data Model:

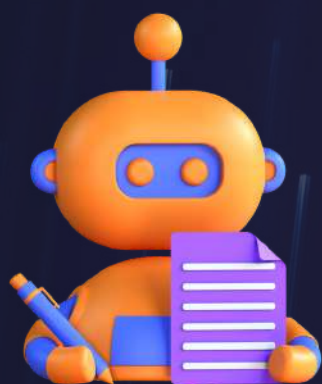
CMS Desynpuf File	OMOP Table
Beneficiary	person
Inpatient Claims	visit_occurrence
Outpatient Claims	visit_occurrence
Carrier Claims	N/A
Prescription Drug Events	drug_exposure
Home Health Agency	N/A
Outpatient Claims	N/A
DME Claims	N/A
HHA Claims	N/A
SNF Claims	N/A
Hospice Claims	N/A

Please note that this mapping is a general guideline and may need to be adjusted based on your specific requirements and the OMOP implementation you are using.

# Step: 2.2

Please note that this mapping is a general guideline and may need to be adjusted based on your specific requirements and the OMOP implementation you are using.

For each file, you may also need to map specific fields within the file to fields within the OMOP table. For example, the DESYNPUF\_ID field in the Beneficiary file can be mapped to the person\_id field in the person table. The mapping of individual fields can be more complex and may require some data transformations to ensure that the data is in the correct format for the OMOP table.



Step Implementation Review: ChatGPT gives an accurate response in terms of mapping tables from CMS DE-SynPUF data to OMOP data using Alteryx .



# Step: 3.1

## 3. Field Mapping: Beneficiary table (CMS DE-SynPUF data) to Person table (OMOP data)

R

Can you show me field mapping for CMS Desynpuf file's Beneficiary table to OMOP



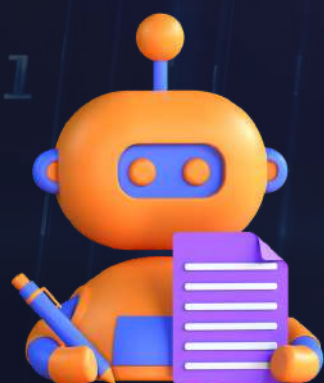
Sure! Here is an example field mapping for the CMS Desynpuf file's Beneficiary table to the OMOP Common Data Model:

CMS Desynpuf Field	OMOP Field
DESYNPUF_ID	person_id
BENEFICIARY_NPI	N/A
CLAIM_ID	N/A
SEQ_NUM	N/A
INDEPENDENT_LABEL	N/A
SEX	gender_concept_id
RACE	race_concept_id
BENE_BIRTH_DT	birth_datetime
BENE_DEATH_DT	death_datetime
BENE_SSN	N/A
SP_STATE_CODE	location_id (linked to location table)
BENE_COUNTY_CD	location_id (linked to location table)

# Step: 3.2

BENE_HI_CVRAGE_TOT_MONS	N/A
BENE_SMI_CVRAGE_TOT_MONS	N/A
BENE_HMO_CVRAGE_TOT_MONS	N/A
SP_ALZHDMTA	condition_occurrence.condition_concept_id
SP_CHF	condition_occurrence.condition_concept_id
SP_CHRNKIDN	condition_occurrence.condition_concept_id
SP_CNCR	condition_occurrence.condition_concept_id
SP_COPD	condition_occurrence.condition_concept_id
SP_DEPRESSN	condition_occurrence.condition_concept_id
SP_DIABETES	condition_occurrence.condition_concept_id
SP_ISCHMCHT	condition_occurrence.condition_concept_id
SP_OSTEOPRS	condition_occurrence.condition_concept_id
SP_RA_OA	condition_occurrence.condition_concept_id
SP_STRKETIA	condition_occurrence.condition_concept_id
HIB_FLAG	N/A
DUAL_FLAG	N/A
INST_FLAG	care_site.location_id
NUM_HC_PROVS	N/A
NUM_PRIMARYCARE_PROVS	N/A

Note that some fields in the CMS Desynpuf data may not have an equivalent in the OMOP Common Data Model and therefore may not be mapped. Also, the mapping may need to be adjusted based on the specific requirements of your project and the specific OMOP implementation you are using.



Step Implementation Review: ChatGPT gives a partially correct response in terms of mapping fields from the Beneficiary table (CMS DE-SynPUF data) to the Person table (OMOP data). It does not give information regarding, where in the Person table, the original field values of Beneficiary table be mapped. For example, the DE-SYNPUF\_ID in the Beneficiary table should flow to Person\_Source\_Value in the Person table.



# Step: 4

## 4. Alteryx implementation (High level steps): Mapping CMS DE-SynPUF data to OMOP data

R Can you show me alteryx workflow to implement it

Here is a high-level workflow in Alteryx to implement the mapping from CMS Desynpuf data to the OMOP Common Data Model:

1. Input the Desynpuf data: The first step is to input the CMS Desynpuf data into Alteryx. This can be done using the Input Data tool.
2. Clean and transform the data: Once the data is inputted, the next step is to clean and transform the data. This may involve removing any irrelevant or unnecessary columns, renaming columns to match the OMOP field names, and converting data types. Alteryx has a variety of tools such as Select, Formula, and Type Cast that can be used for this purpose.
3. Map the data to OMOP fields: The next step is to map the Desynpuf data to the OMOP fields. This can be done using the formula tool to create new fields that match the OMOP fields, and then mapping the Desynpuf fields to the new fields. For example, you could create a new field "person\_id" and use a formula to populate it with the value of the "DESYNPUF\_ID" field.
4. Load the data into OMOP tables: The final step is to load the transformed data into the OMOP tables. This can be done using the Output Data tool and specifying the OMOP database as the output destination.
5. Repeat for each file: The above workflow should be repeated for each of the CMS Desynpuf files, such as Inpatient Claims, Outpatient Claims, and Prescription Drug Events, mapping the fields in each file to the appropriate OMOP table.




Step Implementation Review: ChatGPT gives an accurate response in terms of specifying high level steps to convert CMS DE-SynPUF data to OMOP data using Alteryx .

# Step: 5.1

## 5. Alteryx workflow/implementation to map DE-SYNPUF\_ID (CMS data) to person\_id (OMOP data)

 Can you show me alteryx workflow to generate OMOP person\_id from CMS DESYNPUF\_ID

 Yes, here is an example Alteryx workflow to generate the OMOP person\_id from the CMS DESYNPUF\_ID:

1. Connect to the CMS data source and retrieve the data containing the DESYNPUF\_ID column.
2. Use the Formula tool to create a unique identifier for each DESYNPUF\_ID, if necessary.
3. Use the Output tool to write the newly generated OMOP person\_id to a new table or file.

This workflow should give you the newly generated OMOP person\_id based on the CMS DESYNPUF\_ID. Of course, you may need to make adjustments depending on the specific requirements of your data and the configuration of your Alteryx environment. Additionally, you may need to consider how this generated person\_id will be used in the OMOP database and ensure that it is a unique and valid identifier for each record.



# Step: 5.2

R

Which formula tool should I use, simple or multi-row. How to configure it?



You would use the Simple Formula tool in Alteryx to generate the OMOP person\_id from the CMS DESYNPUF\_ID. Here's how you can configure it:

1. Drag the Simple Formula tool from the Alteryx Designer tool palette to the canvas.
2. Connect the output from the CMS data source to the input of the Simple Formula tool.
3. In the Simple Formula tool, select the DESYNPUF\_ID column as the input field.
4. In the formula box, enter the expression for generating the OMOP person\_id. For example, you can concatenate a prefix and the DESYNPUF\_ID to create a unique identifier:

```
python
```

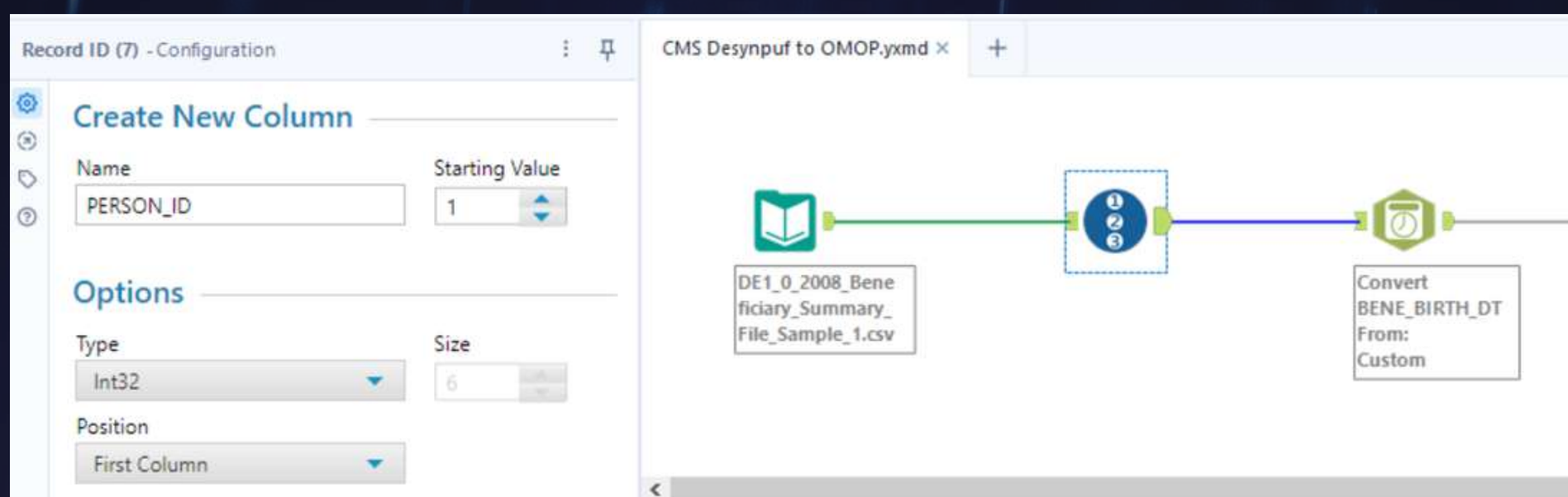
[Copy code](#)

```
"OMOP_" + [DESYNPUF_ID]
```

5. Give the output field a name, such as "person\_id".
6. Ensure that the data type for the output field is set to the desired data type, such as string or text.
7. Run the workflow to generate the OMOP person\_id from the CMS DESYNPUF\_ID.

Keep in mind that this is just an example, and you may need to make adjustments depending on the specific requirements of your data and the OMOP database. It is also important to make sure that the generated person\_id is unique for each record and meets any requirements for the OMOP database.

# Step: 5.3



## Step Implementation Review:

1. ChatGPT gives a partially correct response in terms of specifying Alteryx workflow to map CMS DE-SYNPUF\_ID to OMOP person\_id.
2. ChatGPT suggests to attach the prefix "OMOP\_" to DE-SYNPUF\_ID (using formula tool) in order to generate person\_id, whereas actually, person\_id is of type "Integer", which can be generated using Record ID tool.
3. Kindly refer to below screenshot for actual workflow to generate person\_id using Record ID tool:



# Step: 6.1

## 6. Alteryx workflow/implementation to map BENE\_BIRTH\_DT (CMS data) to day\_of\_birth, month\_of\_birth, year\_of\_birth, birth\_datetime (OMOP data)

R

Can you show me alteryx workflow to convert CMS BENE\_BIRTH\_DT to OMOP day\_of\_birth, month\_of\_birth, year\_of\_birth, birth\_datetime



Yes, here's an example Alteryx workflow to convert the CMS BENE\_BIRTH\_DT to the OMOP day\_of\_birth, month\_of\_birth, year\_of\_birth, and birth\_datetime fields:

1. Connect to the CMS data source and retrieve the data containing the BENE\_BIRTH\_DT column.
2. Use the Formula tool to extract the day, month, and year from the BENE\_BIRTH\_DT column. For example, you can use the following expressions:
  - For day\_of\_birth: ``DayOfWeek([BENE_BIRTH_DT])``
  - For month\_of\_birth: ``Month([BENE_BIRTH_DT])``
  - For year\_of\_birth: ``Year([BENE_BIRTH_DT])``
3. Give the output fields descriptive names, such as "day\_of\_birth", "month\_of\_birth", and "year\_of\_birth", respectively.
4. Ensure that the data type for each output field is set to the appropriate type, such as "Integer" for the day, month, and year fields.
5. Use another Formula tool to convert the BENE\_BIRTH\_DT column to a date-time format that is compatible with the OMOP database. For example, you can use the following expression:

css

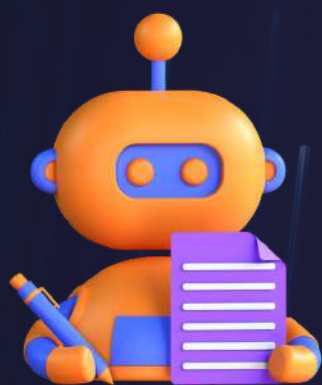
Copy code

```
DateTimeFormat([BENE_BIRTH_DT], "yyyy-MM-dd hh:mm:ss")
```

# Step: 6.2

6. Give the output field a name, such as "birth\_datetime".
7. Ensure that the data type for the output field is set to "Date/Time".
8. Use the Output tool to write the newly extracted and formatted fields to a new table or file.

This workflow should give you the OMOP-compatible day\_of\_birth, month\_of\_birth, year\_of\_birth, and birth\_datetime fields based on the CMS BENE\_BIRTH\_DT. As always, you may need to make adjustments depending on the specific requirements of your data and the configuration of your Alteryx environment. Additionally, you may need to ensure that the extracted and formatted fields are correct for the OMOP database and meet any requirements for that database.



1.ChatGPT gives partially correct response in terms of specifying Alteryx workflow to map BENE\_BIRTH\_DT (CMS data) to day\_of\_birth, month\_of\_birth, year\_of\_birth, birth\_datetime (OMOP data)

2.ChatGPT suggests using formula tool for converting BENE\_BIRTH\_DT (CMS data) to day\_of\_birth, month\_of\_birth, year\_of\_birth, birth\_datetime (OMOP data) which is accurate.

3.ChatGPT suggests using fucntions DayOfWeek(), Month(), Year() functions for converting BENE\_BIRTH\_DT to day\_of\_birth, month\_of\_birth, year\_of\_birth which is inaccurate. Actual functions for conversion are DateTimeDay(), DateTimeMonth(), DateTimeYear()



# Step: 6.3



4. ChatGPT suggests using function `DateTimeFormat("[BENE_BIRTH_DT]", "yyyy-MM-dd hh:mm:ss")` for converting `BENE_BIRTH_DT` to `birth_datetime` which is inaccurate. The actual function for conversion is `DateTimeFormat("[BENE_BIRTH_DT]", "%Y-%M-%d %H:%M:%S")`

5. Kindly refer to below screenshot for actual workflow to map `BENE_BIRTH_DT` (CMS data) to `day_of_birth`, `month_of_birth`, `year_of_birth`, `birth_datetime` (OMOP data):

The screenshot displays a data integration tool interface with two main panels:

- Formula (8) - Configuration:**
  - Output Column:** DAY\_OF\_BIRTH
  - Data Preview:**

```
DateTimeDay([BENE_BIRTH_DT])
```
  - Data type:** Int16, **Size:** 2
  - Output Column:** MONTH\_OF\_BIRTH
  - Output Column:** YEAR\_OF\_BIRTH
  - Output Column:** BIRTH\_DATETIME
  - Data Preview:**

```
DateTimeFormat([BENE_BIRTH_DT], "%Y-%M-%d %H:%M:%S")
```
  - Data type:** V\_WString, **Size:** 1073741823
- CMS Desynpuf to OMOP.yxdm:**
  - Workflow steps:
    - DE1\_0\_2008\_Bene Facility\_Summary\_File\_Sample\_1.csv
    - Convert BENE\_BIRTH\_DT (From: Custom)
    - DAY\_OF\_BIRTH = DateTimeDay([BENE\_BIRTH\_DT])
    - MONTH\_OF\_BIRTH = DateTimeMonth([BENE...])
    - OMOPData Query= Sheet1
- Results - Formula (8) - Output:**
  - 0 of 0 Fields
  - Cell Viewer
  - Data Metadata Actions

# Conclusion:

ChatGPT is reliable/accurate in terms of recommending high-level steps/ Alteryx workflow design for CMS DE-SynPUF to OMOP data conversion. But when it comes to field mapping, it does not mention detail regarding where the original field value from the CMS DE-SynPUF data model should flow/mapped to the OMOP data model. When it comes to configuring a specific Alteryx tool, be it either using a function or regex pattern in the formula tool, the results might not be accurate.

## ChatGPT efficiency in Alteryx:



### Efficient

- High level steps for CMS DE-SynPUF to OMOP data conversion
- High level Alteryx workflow implementation for CMS DE-SynPUF to OMOP data conversion
- CMS DE-SynPUF to OMOP table mapping



### Partially Efficient

- CMS DE-SynPUF to OMOP field mapping
- Individual tool configuration in terms of formula, regex pattern, etc.



# Authors



**Ani Kannal**  
**Head of Data Science and Analytics**  
**Healthark Insights**



**Rushi Shah**  
**Consultant**  
**Healthark Insights**

**CONTACT US**

821, Sun Avenue One  
Manekbag, Ambawadi,  
Amedabad, India  
[contact@healtharkinsights.com](mailto:contact@healtharkinsights.com)  
[healtharkinsights.com](https://healtharkinsights.com)