**Insights on Chlamydia and Gonorrhea data at Howard Brown Health**

Creation Date: 04-November-2017

Last Updated Date: 06-November-2017

Current Version: 2

**Data Analysis**

**On Patients with Chlamydia and/or Gonorrhea**

**Madhumitha Ravi**

Table of Contents

[ETL JOB TO CREATE AN INDICATOR TO EXTRACT PATIENTS WITH CHLAMYDIA AND/OR GONORRHEA WITH THEIR TREATMENT STATUS 3](#_Toc497776216)

[ETL Job Logic (HBH\_Indicator) 3](#_Toc497776217)

[Purpose 3](#_Toc497776218)

[Logic 4](#_Toc497776219)

[Subjob 1 4](#_Toc497776220)

[Subjob 2 6](#_Toc497776221)

[Subjob 3 9](#_Toc497776222)

[Subjob 4 10](#_Toc497776223)

[Subjob 5 11](#_Toc497776224)

[Subjob 6 12](#_Toc497776225)

[Subjob 7 13](#_Toc497776226)

[File Handling 14](#_Toc497776227)

[Data Dictionary 15](#_Toc497776228)

[Sample Files 16](#_Toc497776229)

[Files for execution 16](#_Toc497776230)

[VISUALIZATION OF THE TRANSFORMED DATA 17](#_Toc497776231)

[TROUBLESHOOTING THE ETL WORKFLOW 18](#_Toc497776232)

# ETL JOB TO CREATE AN INDICATOR TO EXTRACT PATIENTS WITH CHLAMYDIA AND/OR GONORRHEA WITH THEIR TREATMENT STATUS

# ETL Job Logic (HBH\_Indicator)

## Purpose

* To create a dataset that has an indicator for whether a patient received appropriate treatment for gonorrhea and/or chlamydia.

1. **Input: Testing dataset**

One dataset contains the **visit** **date** and results of **gonorrhea** and **chlamydia** testing at our clinics in **2016**.

1. **Input: Drug dataset**

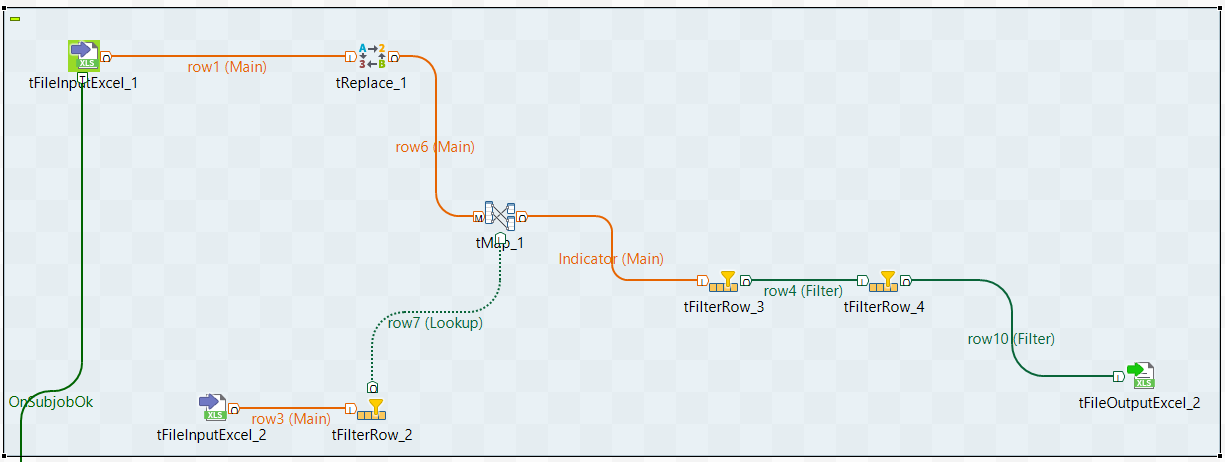
The drug dataset contains **prescriptions** that were given to **patients** with the **prescribed date**.

1. **Output:**

* **Output**: **indicator\_treatment\_positive**
* **Output**: **date\_of\_treatment (drug\_date)**
* **Output**: **patientid (Unique Identifier)**
* **Output**: **gonorrhea\_test\_result**
* **Output**: **chlamydia\_test\_result**
* **Output**: **date\_of\_testing\_gonorrhea\_chlamydia (visit\_date)**
* **Condition:** **Chlamydia\_result = “Positive”** **or** **Gonorrhea\_result = “Positive”** **and treatment\_date = visit\_date or greater than visit\_date**

## Logic

### Subjob 1



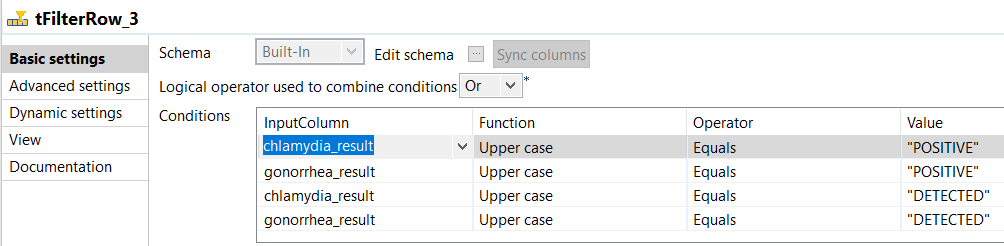
* 1. **tFileInputExcel\_1** points to the location of Testing\_data.xlsx (C:/Users/MadhumithaRavi/Desktop/HBH/Input/Testing\_data.xlsx)
  2. **tFileInputExcel\_2** points to the location of Drug\_data.xlsx (C:/Users/MadhumithaRavi/Desktop/HBH/Input/Drug\_data.xlsx)
  3. **tReplace\_1** will look for **null** values in **chlamydia\_result** and **gonorrhea\_result** and replace with **“N/A”**
  4. **tFilterRow\_2** will filter the drug dates that don’t have a valid date (For example: Patientid = 120168, has a drug\_date of “.” In the drug\_datatset sheet)

**!input\_row.drug\_date.equals(".")**

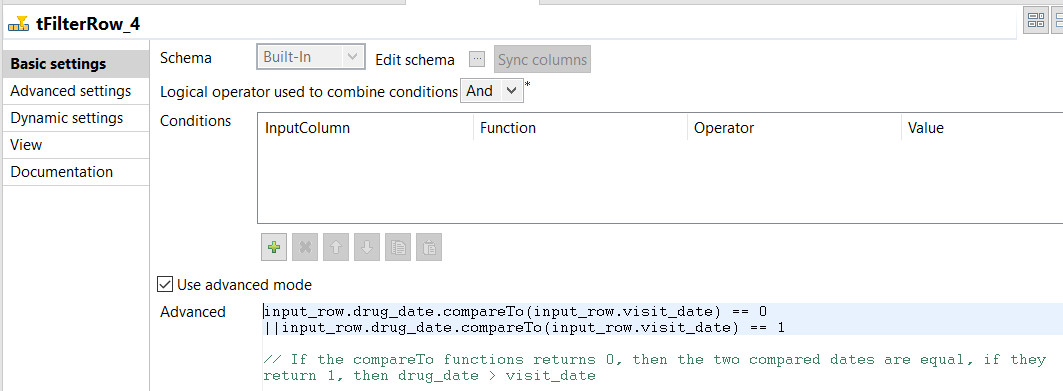
* 1. **tMap\_1** will perform **Inner join** based on the **PatientID** column to fetch matching results from both the sources and send it to the next transformation **tFilterRow\_3**.
  2. Further, in **tMap\_1**, the following operations are performed.
     1. chlamydia\_result = StringHandling.UPCASE(row6.chlamydia\_result)
     2. gonorrhea\_result = StringHandling.UPCASE(row6.gonorrhea\_result)
     3. drug\_description = StringHandling.UPCASE(row7.drug\_description)

This will make sure that all the String comparisons are done against Upper case strings and that there are no variations based on cases.

* 1. **tFilterRow\_3** will filter the patients who are tested “POSITIVE” or “DETECTED” for either gonorrhea and/or chlamydia.

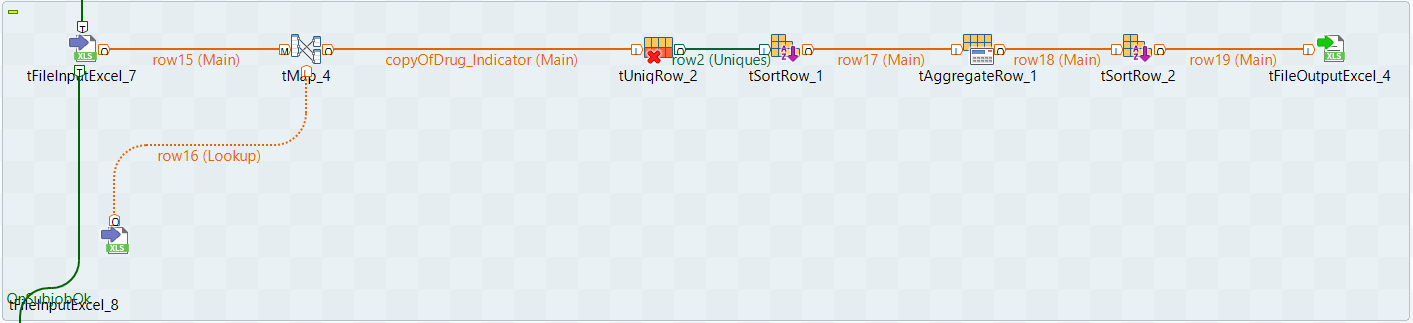


* 1. **tFilterRow\_4** will filter the patients (with “POSITIVE” or “DETECTED” status for either gonorrhea and/or chlamydia) where the drug\_date >= visit\_date. I have assumed that the visit\_date is the testing\_date where the issue has been diagnosed and the corresponding drug\_date is the treatment date.



* 1. **tFileOutputExcel\_2** points to the location of a temp file created for validations (C:/Users/MadhumithaRavi/Desktop/HBH/Temp/Temp\_Post\_Filter\_Positive\_Indicator.xlsx)

### Subjob 2



1. **tFileInputExcel\_7** points to the location of the previously written temp file (C:/Users/MadhumithaRavi/Desktop/HBH/Temp/Temp\_Post\_Filter\_Positive\_Indicator.xlsx)
2. **tFileInputExcel\_8** points to the location of the Drug\_Lookup sheet which holds the list of drugs prescribed for gonorrhea and/or chlamydia

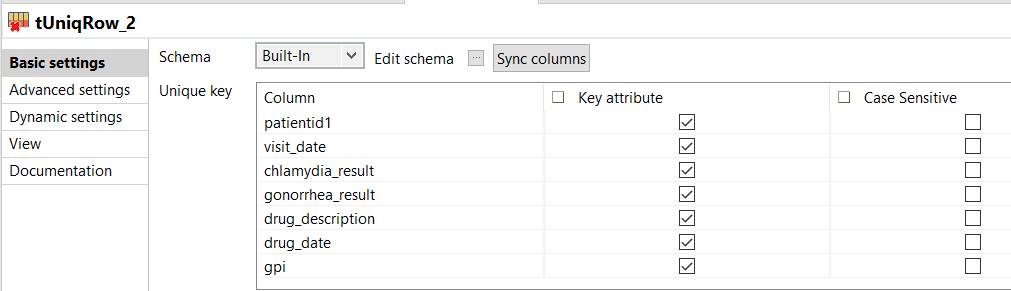
(C:/Users/MadhumithaRavi/Desktop/HBH/Input/Drug\_lookup.xlsx). This step is crucial as there are some medications which are general and are not specific to treat gonorrhea and/or chlamydia. For example: PatientID = 101574, has been tested positive for chlamydia and was treated with TIVICAY 50 MG TABS , TRUVADA 200-300 MG TABS. This could be because the patient has not started the treatment with HBH and had consulted for a disease other than gonorrhea and/or chlamydia even after being tested positive for chlamydia. This file will help us in branching the output into two:

* + 1. Patients tested positive for gonorrhea and/or chlamydia and have been treated.
    2. Patients tested positive for gonorrhea and/or chlamydia and have not been treated.

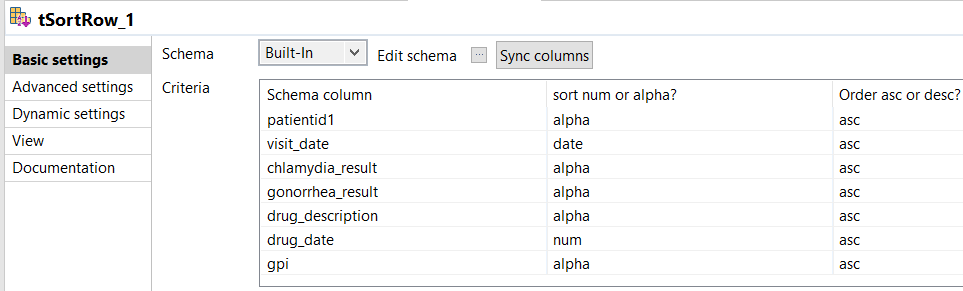
1. **tMap\_4** is used to join the main and lookup links. Here the PatientID along with all other details is filtered based on the lookup in the Drug\_Lookup data. Therefore, only the drugs used for treating gonorrhea and/or chlamydia are looked up and the corresponding patient details are emitted to the next component.

**(row15.drug\_description.contains(row16.Drug\_Names))**

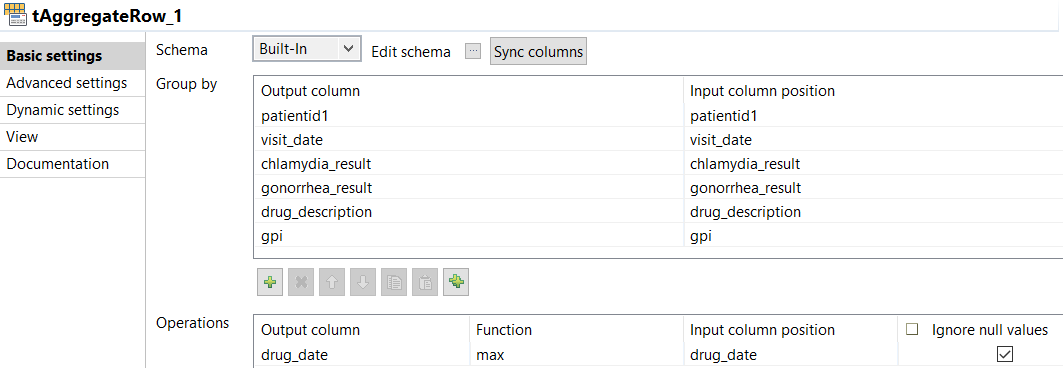
1. **tUniqRow\_2** is used to remove any redundant records based on the conditions to filter such records. I have fetched all the unique records based on the below mentioned columns.

****

1. **tSortRow\_1** is used to sort the data in the order that we want to and send the output to perform aggregate functions.

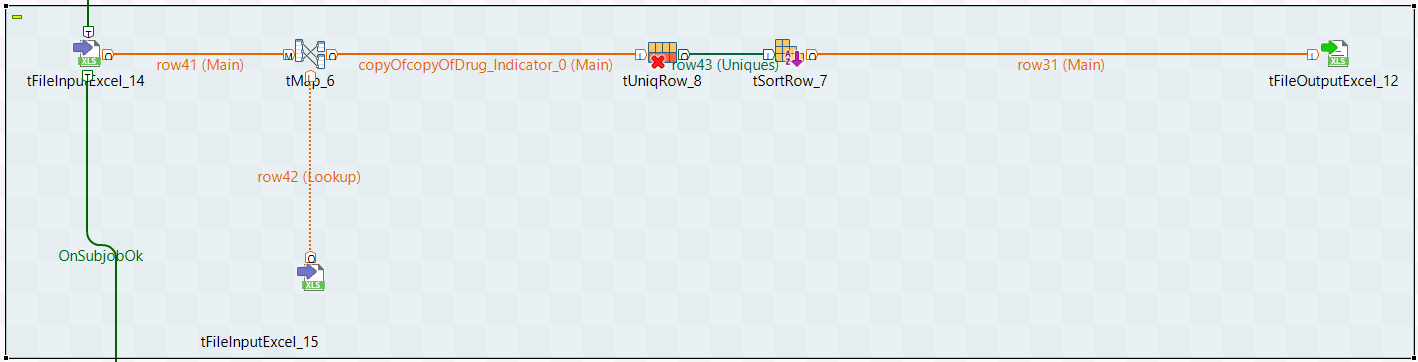
****

1. **tAggregateRow\_1** is used to group the output columns and perform aggregate functions. Here, I have performed a maximum of drug\_date to find the most recent occurrence of treatment for a patient who is tested positive for gonorrhea and/or chlamydia.

****

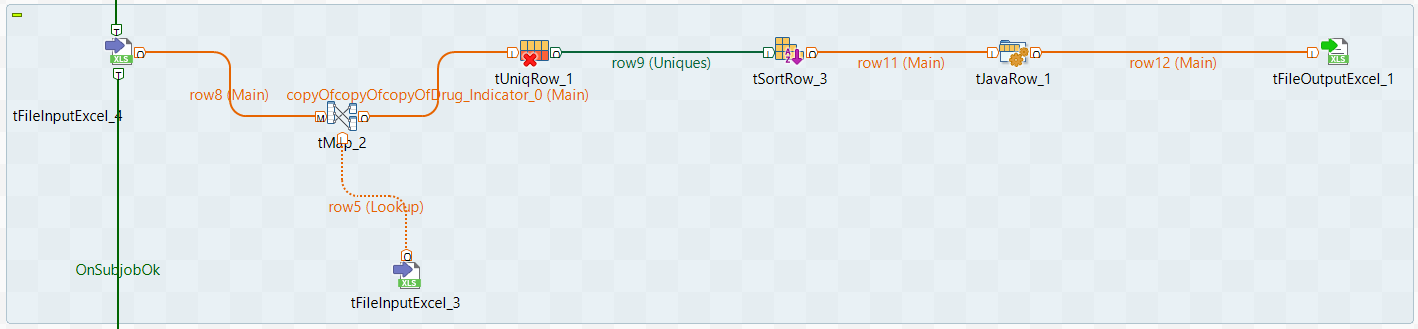
1. Step e is repeated.
2. **tFileOutputExcel\_4** points to the location of a temp file created for validations (C:/Users/MadhumithaRavi/Desktop/HBH/Temp/Lookup\_Drugs\_Present\_Indicator.xlsx)

### Subjob 3

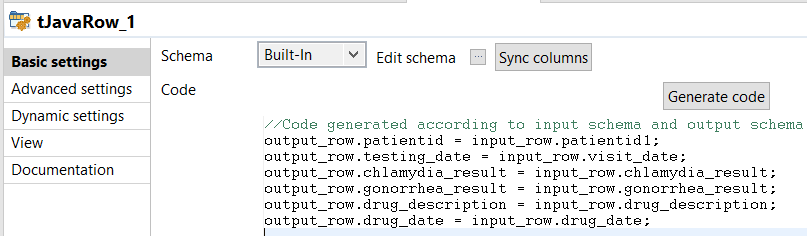


1. **tFileInputExcel\_14** points to the location of temp file which holds all the patients who are tested positive. (C:/Users/MadhumithaRavi/Desktop/HBH/Temp/Temp\_Post\_Filter\_Positive\_Indicator.xlsx)
2. **tFileInputExcel\_15** points to the location of the previously written temp file (C:/Users/MadhumithaRavi/Desktop/HBH/Temp/Lookup\_Drugs\_Present\_Indicator.xlsx).
3. **tMap\_6** is used to perform inner join with keys patientid, visit\_date and drug\_description. However, here the inner join rejects are emitted to the output and hence we would get the list of records which do not match the drugs that are used to treat either gonorrhea and/or chlamydia.
4. **tUniqRow\_1** is used to remove any redundant records based on the conditions to filter such records. I have fetched all the unique records like previous sub jobs.
5. **tSortRow\_1** is used to sort the data in the order that we want to and send the output to perform aggregate functions.
6. **tFileOutputExcel\_12** points to the location of the temp file which holds the patient details with drugs other than those used to treat either gonorrhea and/or chlamydia.

### Subjob 4

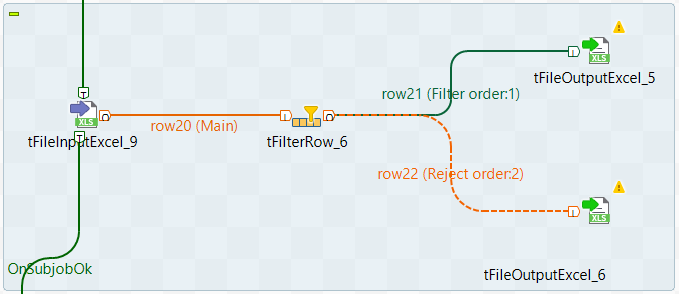


1. **tFileInputExcel\_3** points to the location of temp file which holds all the patients who are treated with drugs from the list of drugs used to treat either gonorrhea and/or chlamydia (C:/Users/MadhumithaRavi/Desktop/HBH/Temp/Lookup\_Drugs\_Present\_Indicator.xlsx)
2. **tFileInputExcel\_4** points to the location of the previously written temp file which hold the list of patients with drugs other than the lookup drugs. (C:/Users/MadhumithaRavi/Desktop/HBH/Temp/Lookup\_Drugs\_Absent\_Indicator.xlsx).
3. **tMap\_2** is used to perform inner join with based on patientid. However, here the inner join rejects are emitted to the output. Here we get the list of
   * 1. Patients tested positive for gonorrhea and/or chlamydia and have been treated.
     2. Patients tested positive for gonorrhea and/or chlamydia and have not been treated.
4. **tUniqRow\_1** is used to remove any redundant records based on the conditions to filter such records. I have fetched all the unique records like previous sub jobs.
5. **tSortRow\_3** is used to sort the data in the order that we want to and send the output to perform aggregate functions.
6. **tJavaRow\_1** is used to map the input and output schema, thereby letting us rename fields to our convenience.



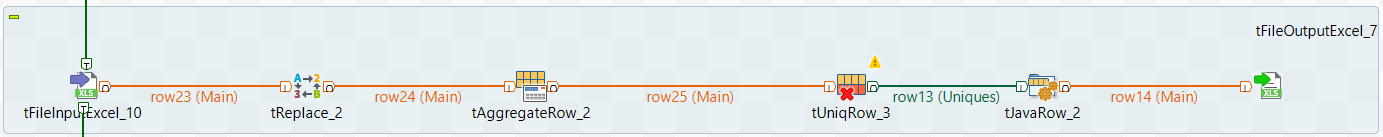
1. **tFileOutputExcel\_1** points to the location of an **Output** which holds the list of untreated patients who have been tested positive for either gonorrhea and/or chlamydia (C:/Users/MadhumithaRavi/Desktop/HBH/Output/Patients\_untreated\_for\_Gonorrhea\_and\_or\_Chlamydia.xlsx)

### Subjob 5

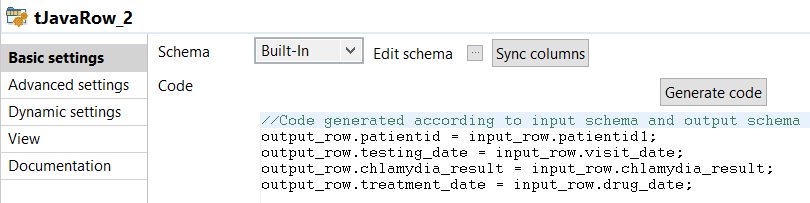


1. **tFileInputExcel\_9** points to the location of temp file which holds all the patients who are treated with drugs from the list of drugs used to treat either gonorrhea and/or chlamydia (C:/Users/MadhumithaRavi/Desktop/HBH/Temp/Lookup\_Drugs\_Present\_Indicator.xlsx)
2. **tFilterRow\_6** is used to filter the records based on the CHLAMYDIA and GONORRHEA results, thus branching into two files.
3. **tFileOutputExcel\_5** points to the location of a temp which holds the list of patients who have been tested positive for chlamydia. (C:/Users/MadhumithaRavi/Desktop/HBH/Output/Patients\_treated\_for\_Chlamydia.xlsx)
4. **tFileOutputExcel\_6** points to the location of a temp which holds the list of patients who have been tested positive for gonorrhea. (C:/Users/MadhumithaRavi/Desktop/HBH/Output/Patients\_treated\_for\_Gonorrhea.xlsx)

### Subjob 6



1. **tFileInputExcel\_10** points to the location of a temp which holds the list of patients who have been tested positive for chlamydia. (C:/Users/MadhumithaRavi/Desktop/HBH/Output/Patients\_treated\_for\_Chlamydia.xlsx)
2. **tReplace\_2** is used to replace all the occurrences of “DETECTED” with “POSITIVE”, thereby letting us have a uniform testing\_status both in database and for visualizations.
3. **tAggregateRow\_2** is used to group the output columns and perform aggregate functions. Here, I have performed a maximum of drug\_date and visit\_date to find the most recent occurrence of treatment for a patient who is tested positive for chlamydia.
4. **tUniqRow\_3** is used to remove any redundant records based on the conditions to filter such records. I have fetched all the unique records like previous sub jobs.
5. **tJavaRow\_2** is used to map the input and output schema, thereby letting us rename fields to our convenience.

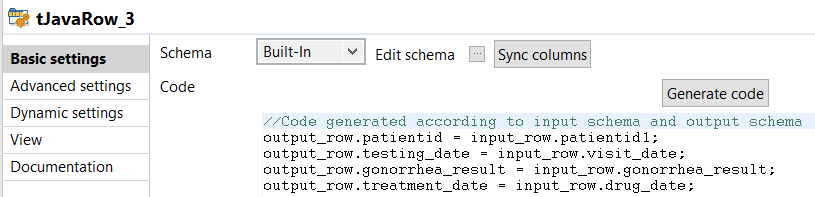
****

1. **tFileOutputExcel\_7** points to the location of an **Output** which holds the list of treated patients who have been tested positive for chlamydia (C:/Users/MadhumithaRavi/Desktop/HBH/Output/Patients\_treated\_for\_Chlamydia.xlsx)

### Subjob 7



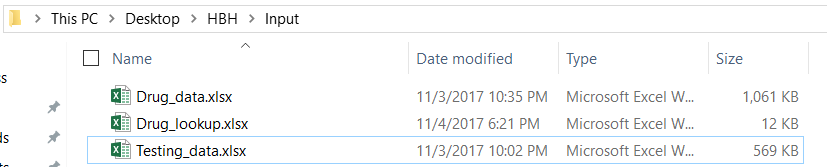
1. **tFileInputExcel\_11** points to the location of a temp which holds the list of patients who have been tested positive for gonorrhea. (C:/Users/MadhumithaRavi/Desktop/HBH/Output/Patients\_treated\_for\_ Gonorrhea.xlsx)
2. **tReplace\_3** is used to replace all the occurrences of “DETECTED” with “POSITIVE”, thereby letting us have a uniform testing\_status both in database and for visualizations.
3. **tAggregateRow\_3** is used to group the output columns and perform aggregate functions. Here, I have performed a maximum of drug\_date and visit\_date to find the most recent occurrence of treatment for a patient who is tested positive for gonorrhea.
4. **tUniqRow\_4** is used to remove any redundant records based on the conditions to filter such records. I have fetched all the unique records like previous sub jobs.
5. **tJavaRow\_3** is used to map the input and output schema, thereby letting us rename fields to our convenience.

****

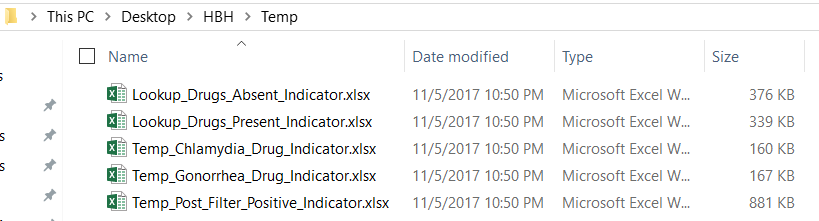
1. **tFileOutputExcel\_8** points to the location of an **Output** which holds the list of treated patients who have been tested positive for gonorrhea (C:/Users/MadhumithaRavi/Desktop/HBH/Output/Patients\_treated\_for\_ Gonorrhea.xlsx)

# File Handling

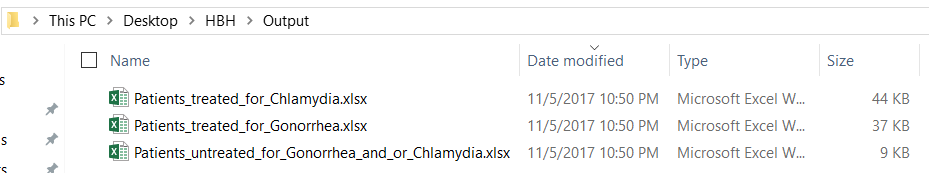
1. Input files reside at the **Input** folder and contains both the input files from HBH – Testing\_data.xlsx and Drug\_data.xlsx along with the lookup file Drug\_lookup.xlsx



1. All the temporary files, using which we derive the output files are placed in the **Temp** folder.



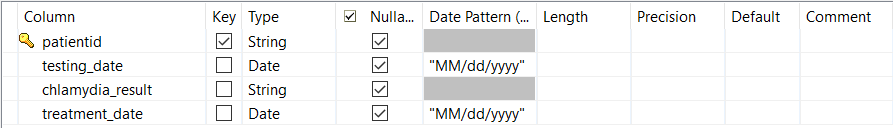
1. All the Output files are placed in the **Output** folder and contain the following files.



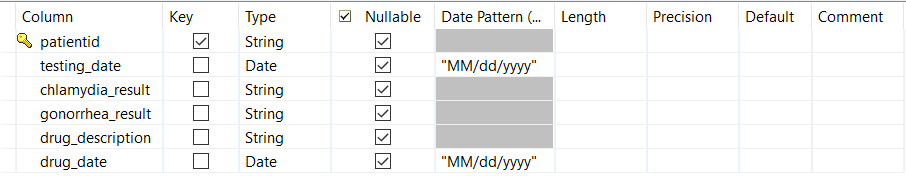
* + 1. **Patients\_treated\_for\_Chlamydia**
       - Contains the PatientID, Testing\_Date, Chlamydia\_Result and Treatment\_Date for Chlamydia.
    2. **Patients\_treated\_for\_Gonorrhea** 
       - Contains the PatientID, Testing\_Date, Gonorrhea\_Result and Treatment\_Date for Gonorrhea.
    3. **Patients\_untreated\_for\_Gonorrhea\_and\_or\_Chlamydia**
       - Contains the PatientID, Testing\_Date, Chlamydia\_Result, Gonorrhea\_Result, Drug\_Description and Drug\_Date for all untreated patients.

# Data Dictionary

* 1. For loading datasets of patients who are tested positive for either chlamydia and/or gonorrhea.



* 1. For loading datasets of patients who are untreated and tested positive for either chlamydia and/or gonorrhea.



# Sample Files

1. All the files that were used for the flow are zipped and provided below as a part of OLE.



# Files for execution

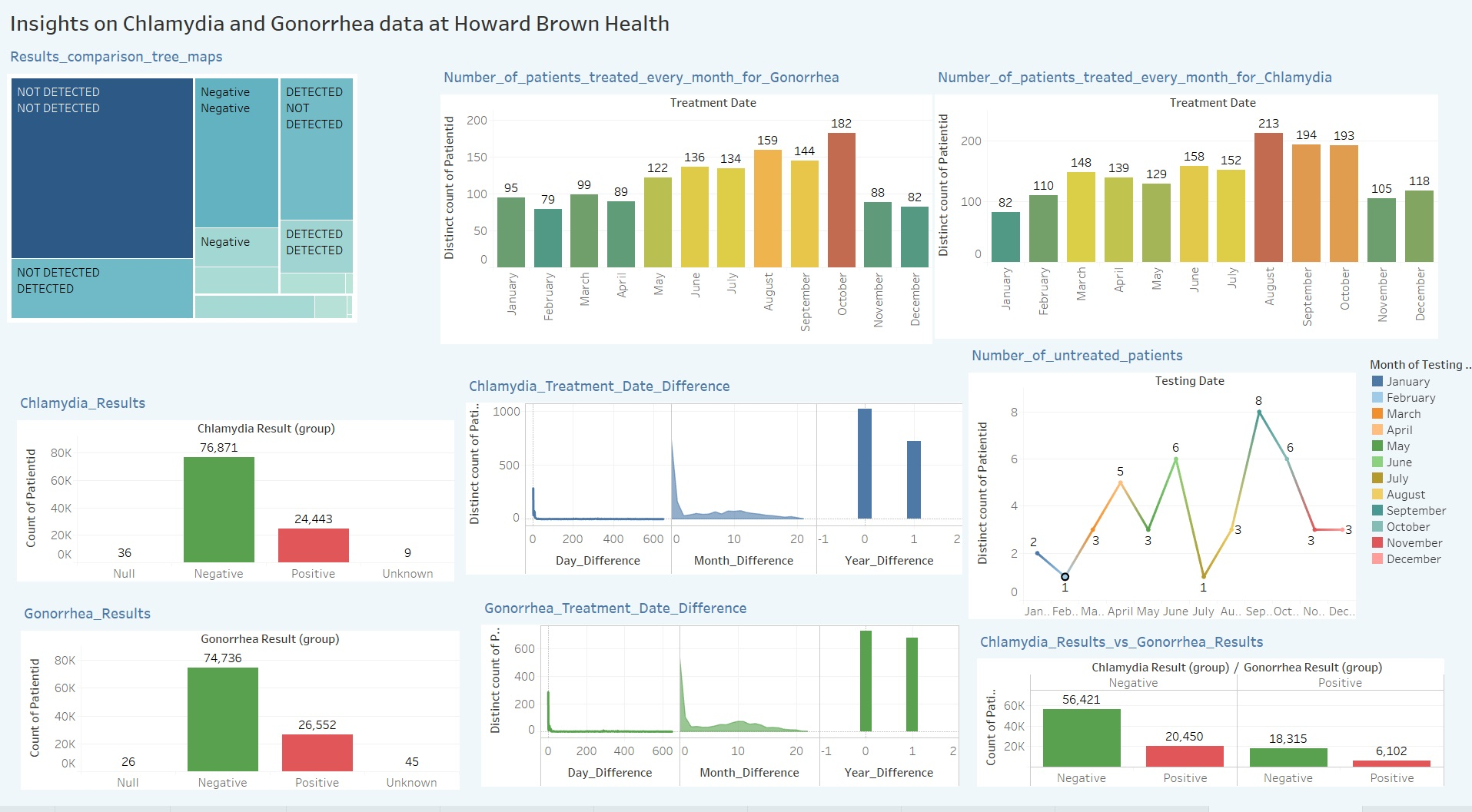
1. By running the .exe file that is attached here, the complete flow will execute even when there is no Talend installed locally. However, JAVA should be installed to aid in the execution.
2. Similarly, since I have used folder structure pertaining to my system, the folder structure should be changed in the below mentioned file.

HBH\_Indicator\src\main\java\local\_project\hbh\_indicator\_0\_1\HBH\_Indicator.java

1. Once all the input, lookup, temp and output locations are changed in the above-mentioned java file, HBH\_Indicator\HBH\_Indicator\_run.bat should be run and the output would be generated in the desired file.



# VISUALIZATION OF THE TRANSFORMED DATA



Link for Dynamic Visualization: <https://public.tableau.com/profile/madhumitha.ravi4251#!/vizhome/Indicator_3/InsightsonChlamydiaandGonorrheadataatHowardBrownHealth?publish=yes>

# TROUBLESHOOTING THE ETL WORKFLOW

* If the job fails due to the following error, “java.lang.OutOfMemoryError: GC overhead limit exceeded”, please make sure the following steps are followed.
* It is caused due to large amount of data, and to cater to the same, we will l have to increase the memory space to allow the execution of the job.
* Please note that not all people will face this error, as the RAM requirements change based on the system used by every individual.
* Click on the job that fails, and go to the run statistics window.
* Click on Advanced settings, and change the memory size. By default -Xms256M, -Xmx256M would be set.
* There are two types of memory allocation:  
  -Xms :--When we will use this memory allocation this means that when we get started a job it is using this much memory.  
  -Xmx:--When we will use this memory allocation this means that the memory is the maximum memory, your job cannot use more than this memory space.
* Below is the list of values that could be accommodated into the fields.

|  |
| --- |
| 1GB = -Xmx1024M -Xms1024M  2GB = -Xmx2048M -Xms2048M  3GB = -Xmx3072M -Xms3072M  4GB = -Xmx4096M -Xms4096M  5GB = -Xmx5120M -Xms5120M  6GB = -Xmx6144M -Xms6144M  8GB = -Xmx8192M -Xms8192M |

* Discretion is advised while choosing the memory value, as it will affect the performance of other applications when the Talend job runs.