Project: Health care Data Analysis

Functional Specification

- High level functional specification requirement from our client **Philomath** Healthcare.
- **Philomath** wants this two R&D Reports to plan and strategize the newly launched health product in the US market.
- They want the U.S. city report and a US Prescriber report.

First Report, US City report

- They want a US City report with number of distinct prescribers assigned for each city.
- A Prescriber in US means
    - a physician or a dentist or a person licensed, registered or otherwise permitted by the US to issue any prescriptions for drugs for human use.
- In the city report, they want a total transaction count prescribed in each city.
- Each prescription prescribed by a physician is calculated as one transaction count.
- They also want the number of zipcodes in each city and they don't want to report a city if no prescriber is assigned to.
- In the final report, they want the File Type as JSON, Compression type as bzip2
- And the number of split files would be one.

Why they want a City Report?

- They want this City report because it is important for them to know which cities produce the maximum number of transactions.
- They want to focus only on the selected cities as of now.

Second report, US Prescriber report

- They want a US prescriber report with top five prescribers with the highest transaction count in each state
- They consider prescribers only with a working experience between 20 and 50 years.

- The final file type should be ORC and the compression type should be snappy.
- The number of split files should be two.

Why they want a prescriber report?

- They want the prescriber report because they want to target the top prescribers from each state.

It is really important to design and build a robust ETL pipeline which will accommodate the current and future changes with little effort

Summary of the reports required

1. USA City report.

- ✓ Number of distinct prescribers assigned for each city.
- ✓ Total TRX_CNT prescribed in each city.
- ✓ Number of zips in each city
- ✓ Do not report a city if no prescriber is assigned to.

File Type :json
Compression Type: bzip2

2. USA Prescriber Report.

Top 5 Prescribers with the highest TRX_CNT in each state. Consider the Prescribers only with working experience from 20 to 50 Years.

File Type :orc
Compression Type: snappy

Input files in HDFS

prescpipeline/staging/city
prescpipeline/staging/prescriber

Output files in HDFS

prescpipeline/output/city
prescpipeline/output/prescriber

Input City Data format/Layout

city	city_ascii	state_id	state_name	county_fips	county_name	lat	lng	population	density	timezone	zips
New York	New York	NY	New York	36061	New York	40.6943	-73.9249	18713220	10715	America/New_York	11229 11226 11225...
Los Angeles	Los Angeles	CA	California	6037	Los Angeles	34.1139	-118.4068	12750807	3276	America/Los_Angeles	90291 90293 90292...
Chicago	Chicago	IL	Illinois	17031	Cook	41.8373	-87.6862	8604203	4574	America/Chicago	60018 60649 60641...
Miami	Miami	FL	Florida	12086	Miami-Dade	25.7839	-80.2102	6445545	5019	America/New_York	33129 33125 33126...
Dallas	Dallas	TX	Texas	48113	Dallas	32.7936	-96.7662	5743938	1526	America/Chicago	75287 75098 75233...

Input Prescriber Data Layout

npi	nppes_provider_last_org_name	nppes_provider_first_name	nppes_provider_city	nppes_provider_state	specialty_description	description_flag	drug_name	generic_name	bene_count	total_claim_count	total_30_day_fill_co
2006000252	ENKESHAFI	ARDALAN	CUMBERLAND	MD	Internal Medicine	S	ATORVASTATIN CALCIUM	ATORVASTATIN CALCIUM	null	13	1
2006000252	ENKESHAFI	ARDALAN	CUMBERLAND	MD	Internal Medicine	S	CIPROFLOXACIN HCL	CIPROFLOXACIN HCL	null	11	1
2006000252	ENKESHAFI	ARDALAN	CUMBERLAND	MD	Internal Medicine	S	DOXYCYCLINE HYCLATE	DOXYCYCLINE HYCLATE	20	20	2
2006000252	ENKESHAFI	ARDALAN	CUMBERLAND	MD	Internal Medicine	S	ELIQUIS	APIXABAN	null	17	1
2006000252	ENKESHAFI	ARDALAN	CUMBERLAND	MD	Internal Medicine	S	FUROSEMIDE	FUROSEMIDE	12	17	1

total_day_supply	total_drug_cost	bene_count_ge65	bene_count_ge65_suppress_flag	total_claim_count_ge65	ge65_suppress_flag	total_30_day_fill_count_ge65	total_day_supply_ge65	total_drug_cost_ge65	years_of_exp
450	139.32	null	*	13	null	15.0	450	139.32	= 45.0
96	80.99	null	*	null	null	null	null	null	= 43.0
199	586.12	null	#	null	null	null	null	null	= 33.0
510	6065.02	null	*	17	null	17.0	510	6065.02	= 44.0
510	6065.02	null	*	17	null	17.0	510	6065.02	= 44.0

Output City Report Layout

No of splits: 1

Output format: JSON

Compression : Bzip2

city	county_name	population	presc_counts	state_name	trx_counts	zip_counts
ANAHEIM	ORANGE	350365	1030	CALIFORNIA	1588424	16
TRAVERSE CITY	GRAND TRAVERSE	50522	566	MICHIGAN	617013	3
HELENA	LEWIS AND CLARK	52936	195	MONTANA	183806	6
PATERSON	PASSAIC	145233	225	NEW JERSEY	345999	15
BRENTWOOD	WILLIAMSON	42783	164	TENNESSEE	135778	2

Output Prescriber report layout

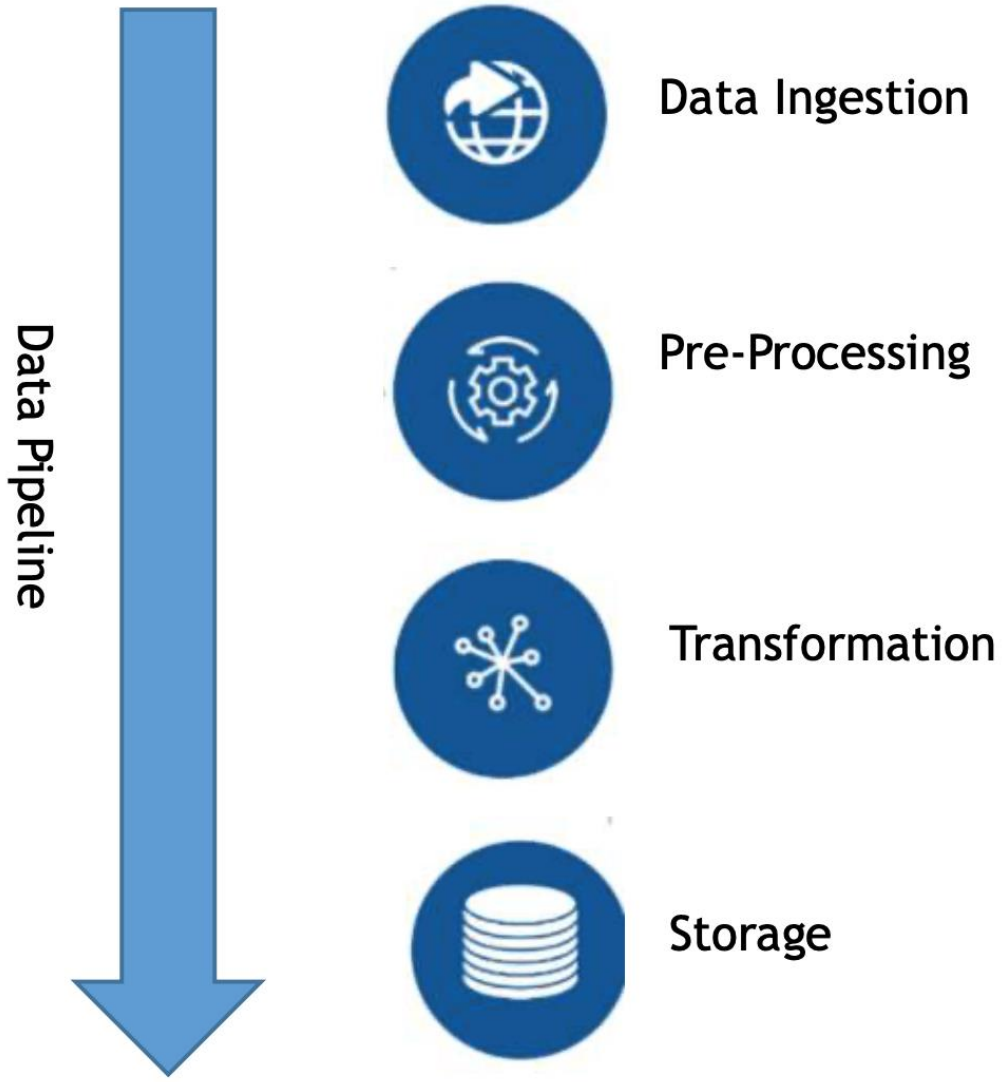
No of splits: 2

Output format: orc

Compression : snappy

presc_id	presc_fullname	presc_state	country_name	years_of_exp	trx_cnt	total_day_supply	total_drug_cost
-1854807747	CARL VANCE	ID	USA	37	1978	121899	41390.65
-1874050584	ADAM REYNOLDS	ID	USA	41	1513	96629	37868.32
-1652843680	JON FISHBURN	ID	USA	34	1388	71699	27881.24
-1359857239	DAVID LILJENQUIST	ID	USA	46	1377	94361	32576.78
-1854807747	CARL VANCE	ID	USA	33	1299	93094	11976.16

Project Flow



- The 1st step in the project is Data Ingestion. Sample data files are provided and they have to be ingested in to HDFS in to the project's location
- The 2nd step is to cleanse the data and use only the data that is required for analysis/reports using pyspark

  Clean City Data:

  - Select only required Columns in city data file like city, state_id,state_name,county_name,population,zips

- Convert city, state and county fields to Upper Case

  Clean Prescriber Data
- Select only required fields such as npi, nppes_provider_last_org_name, nppes_provider_first_name, nppes_provider_city, nppes_provider_state, specialty_description, drug_name, total_claim_count, total_day_supply, total_drug_cost
- Rename the above fields to shorter names
- Add a Country Field 'USA' to the above data
- Clean the “years_of_exp” to extract only the numbers. Hint : use regexp_extract from the package pyspark.sql.functions
- Convert the years_of_exp field to integer
- Combine First Name and Last Name in to a single field and remove the individual columns
- Count the number of null values for each column
  - Hint:Sample code **prescriber_df.select([count(when(isnan(c) | col(c).isNull(),c)).alias(c) for c in prescriber_df.columns]).show()**
- clean all the Null/Nan Values
  - Delete the records where the PRESC_ID and DRUG_NAME is fields are NULL. Use dropna() of dataframe

- The 3rd step is to transform the cleansed data in to the required reports using pyspark

  Transform Logic: City Report
  - Calculate the Number of zips in each city.
  - Calculate the number of distinct Prescribers assigned for each City.

- Calculate total total_claim_count prescribed for each city.
- Do not report a city in the final report if no prescriber is assigned to it.

Output report Layout:

- City Name
- State Name
- County Name
- City Population
- Number of Zips
- Prescriber Counts
- total_claim_counts

Transform logic: Prescriber Report:

- Top 5 Prescribers with highest total_claim_count per each state.
- Consider the prescribers only from 20 to 50 years of experience.

Output report Layout:

- Prescriber ID
- Prescriber Full Name
- Prescriber State
- Prescriber Country
- Prescriber Years of Experience
- Total claim count
- Total Days Supply
- Total Drug Cost

- The 4th step is to store the reports in a suitable storage like HDFS/hive/hbase