Dimension tables:

* + Dim\_Store ( Table for Store related information )
  + Dim\_Region (Table for Geo related Information)
  + Dim\_Item ( Table for Items related Information)
  + Dim\_Calendar ( Date Dimension)

Facts :

* + Fact\_instk\_dly ( Item in stocks related information at date level)

Information which we can derive from this table :

What is the

onhand\_each\_qty (which is relatable to ‘in stocks’ in general terms)

of an item(mds\_fam\_id)

in a store (store\_nbr) of a geo (geo\_region\_cd)

on a given date (bus\_dt)

* + fact\_scanv\_dly (Items getting scanned and sold every visit day of customers)

Information which we can derive from this table :

What is the

Total sales amount, sales unit, return amount

On a given customer visit dt

This table is a sales transaction table and not a aggregated table. Table will have details for each sale transaction done from store POS.

Basic EDA details :

* If any table has geo\_region\_cd column then geo\_region\_cd = ‘US’ are eligible records to work with. Filter out all other records. Dimension might have values for countries other than ‘US’ with column values in different languages which are not eligible for our use case.
* If any table has op\_cmpny\_cd column then op\_cmpny\_cd = ‘WMT-US’ are our eligible records to work with. Filter out all other records.
* If any table has both the columns then we can filter on any of them with values mentioned above.
* If Metric columns (e.g. onhand\_each\_qty,sales\_unit\_qty etc.) have negative values then we can ignore them as bad/corrupt data.
* Below are list of columns in each table which are important for our use case. Other columns are either audit columns or columns which are not of our concern.

Dim\_region :

region\_nbr

region\_nm

region\_mgr\_nm

geo\_region\_cd

op\_cmpny\_cd

dim\_store:

store\_nbr

geo\_region\_cd

store\_nm

region\_nbr

market\_nm

city\_nm

op\_cmpny\_cd

dim\_item:

mds\_fam\_id

item\_nbr

item\_desc\_1

upc\_nbr

dept\_desc

brand\_nm

geo\_region\_cd

dim\_Calendar:

wm\_yr\_wk\_nbr

wm\_qtr\_nm

cal\_dt

geo\_region\_cd

fact\_instk\_dly:

store\_nbr

geo\_region\_cd

mds\_fam\_id

onhand\_each\_qty

fcst\_dmand\_each\_qty

bus\_dt

op\_cmpny\_cd

fact\_scanv\_dly:

store\_nbr

geo\_region\_cd

mds\_fam\_id

upc\_nbr

sales\_unit\_qty

sales\_amt

rtn\_amt

visit\_dt

op\_cmpny\_cd

**Problem Statement :**

We need to find stores which are going out of stock on a day due to less number of in stocks maintained and more number of sales.

if

fact\_instk\_dly.onhand\_each\_qty < total (fact\_scanv\_dly.sales\_unit\_qty)

for an item

in a store

on a day then item got out of stock in that store on that day.

**Prerequisites :**

All the target tables in Hive must be partitioned on some columns with ORC storage format. All the tables should have additional row\_insertion\_dttm column which will store timestamp of a record when it was loaded to the table. You can create stage tables in hive if required.

**ETL flow :**

A screenshot of a computer

Description automatically generated