

MedicalSideFx

A Technical Guide to Adverse Events Analytics using Apache Hive

By
Sanjay Subramanian

First Edition May 2018

Table of Contents

About the Author	4
How necessity fueled an invention for me	5
Pre-requisites	5
Motivation	6
Where do we get the adverse events data ?	11
FDA Adverse Event Reporting System (FAERS): Latest Quarterly Data Files	11
The Adverse Event Reporting System (AERS): Older Quarterly Data Files	11
Anatomy of a quarterly data zip file	13
Scope of this book	14
Some quick counts	14
Metadata changes over the years	14
Demographic Dataset Columns variations	14
Demographic quarterly dataset to variations mapping	15
Drug Dataset Column variations	18
Drug quarterly dataset to variations mapping	18
Reaction Dataset Column variations	21
Reactions quarterly dataset to variations mapping	21
Hadoop ecosystem	23
Loading the data into HDFS	24
Before Hive came along	24
Creating Hive tables	25

Creating Demographic tables	25
Partitioning Demographic tables	30
Creating a Single Demographic View using subset of columns	35
Creating Drug Tables	38
Partitioning Drug Tables	42
Creating a Single Drug View using subset of columns	46
Creating Reaction Tables	49
Partitioning Reaction Tables	50
Creating a Single Reaction View using subset of columns	55
Joining Demographic, Drug and Reaction tables	57
Creating detail views of Demographic, Drug, Reaction(all column variations included)	61
Demographic Merged View (all column variations included)	61
Drug Merged View (all column variations included)	69
Reac Merged View (all column variations included)	74
Now we are ready for some analysis	76
AgeGroup Query	76
Reactions Query	77
Side effects reported by Year Query	79
Find out which Drugs have a specific side effect	81
Compare the side effects of two medicines	82
Compare the symptom counts for two medicines	82
Compare the age groups counts reported for two medicines	85
Compare the side effect counts reported by year for two medicines	87



Medical Side Fx

MedicalSideFx is intended as an aid for discussion with your doctor. This information is not skewed by any commercial interest and reaches you advertisement free. We incorporate the most recent data available from www.fda.org. Please do not stop or start taking any medication without consulting your doctor.

OK

© 2011 - 2017 Sanjay Subramanian

About the Author



Sanjay Subramanian is currently the founder and CTO of ExplainData (www.explaindata.org) a consulting firm that believes in making AI and ML accessible to all via customized solutions. Sanjay has a Bachelors in Technology (Indian Institute of Technology Madras, India) has been a successful data analytics wrangler for over 10 years executing hadoop, hive and nosql projects at Coremetrics (now IBM), SmartZip Analytics and Nextag. Sanjay designed and dedicated an IOS app MedicalSideFx to the community that allows users to search and compare adverse events of medicines. An interesting trivia about Sanjay is that back in 1986 while being a senior in undergrad school and guided by his professor, he built alternate fuel methanol/ethanol based internal combustion engines. When not doing analytics, Sanjay spends his time on his two passions - photography and playing the guitar.

How necessity fueled an invention for me

Year 2008. My father was in intensive care unit battling many health issues and was required to ingest a cocktail of medicines. There were many side effects we observed during this time and when I asked the doctors if there were alternate medicines that would have lesser of these side effects, I got no answer. I queried many search-engines with the hope of finding some quantified analysis of reported side effects. No luck. All I could find is the adverse reaction notes published from a pharmaceutical company point of view.

We were blessed with good fortune. Many months passed. Dad recovered and came back home. We all wept and rejoiced. But the fact that I could not find a quantified list of side effects for medicines still bothered me.

I decided to use my programming skills to do something about it.

Year 2009. I had managed to find reliable adverse events datasets. I started working furiously late nights and weekends and eventually built the entire application stack and released it as an IOS app (iPhone and iPad) called MedicalSideFx (<https://itunes.apple.com/us/app/medicalsidesfx/id827493563?mt=8>). I decided to dedicate this App to the community. You can download it for free. All data updates are free for a lifetime. In 2015, the app got accepted by FBStart (a very generous program from Facebook meant to help startups).

The Programming Guide to Adverse Events Analytics details the design, architecture and code that runs at the backend of the MedicalSideFx App.

Pre-requisites

This is a technical book and the readers are expected to have worked with Linux, HDFS, SQL, NOSQL, Hive and the Hadoop ecosystem.

Motivation

Before I started work on analyzing side effects, there were questions in my mind that needed answers. There were 4 basic questions

- What are the top side effects for a specific medicine and more importantly their counts - for example how many hepatic failures were reported for acetamophen ?



Levaquin	
To compare two medicines type "motrin,alev..."	
By Symptom	By Year
By AgeGroup	
Levaquin	
Cases reported (top 25)	
All symptoms	9822
Arthralgia	1277
Tendon rupture	1178
Tendonitis	1062
Tendon injury	969
Pain in extremity	818
Rotator cuff syndrome	615
Insomnia	553

(MedicalSideFx IOS App screenshot)

- What are the counts of side effects by year for a specific medicine - for example were more renal failures reported for a specific medicine in a specific year?



Cases reported (top 25)	
All years	9822
2016	402
2015	1761
2014	260
2013	227
2012	402
2011	908
2010	1211

(MedicalSideFx IOS App screenshot)

- What are the counts of side effects by age-group for a specific medicine - for example if the patient was in the age group 25-29 , what were the side effects reported for a specific medicine?



The screenshot shows the MedicalSideFx iOS App interface. At the top, there is a search bar with 'Levaquin' entered. Below the search bar, there is a hint: 'To compare two medicines type "motrin,alev...'. There are three tabs: 'By Symptom', 'By Year', and 'By AgeGroup', with 'By AgeGroup' being the selected tab. Below the tabs, there is a blue envelope icon. The main heading is 'Levaquin'. Below this, there is a table titled 'Cases reported (top 25)'. The table has two columns: the first column lists age groups, and the second column shows the number of cases reported. The data is as follows:

Cases reported (top 25)	
All groups	9822
50-54 years	620
55-59 years	585
45-49 years	542
40-44 years	536
60-64 years	493
65-69 years	403
35-39 years	389

(MedicalSideFx IOS App screenshot)

- Based on the analysis of the above I wanted to be derive a comparison between two medicines

Levaquin,Cipro

To compare two medicines type "motrin,alev..."

By Symptom By Year By AgeGroup

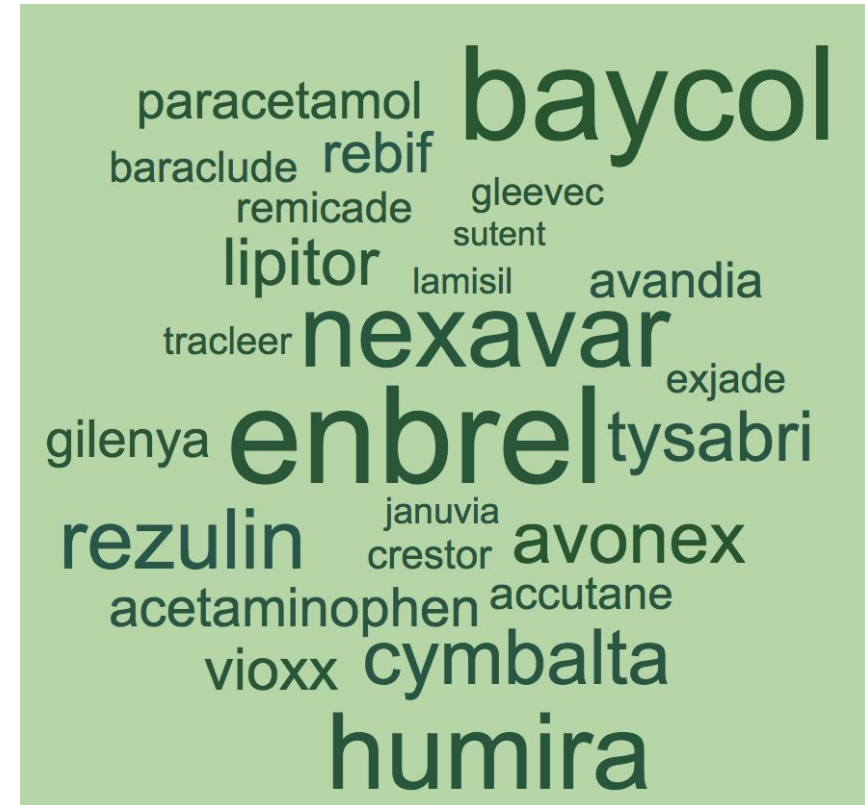
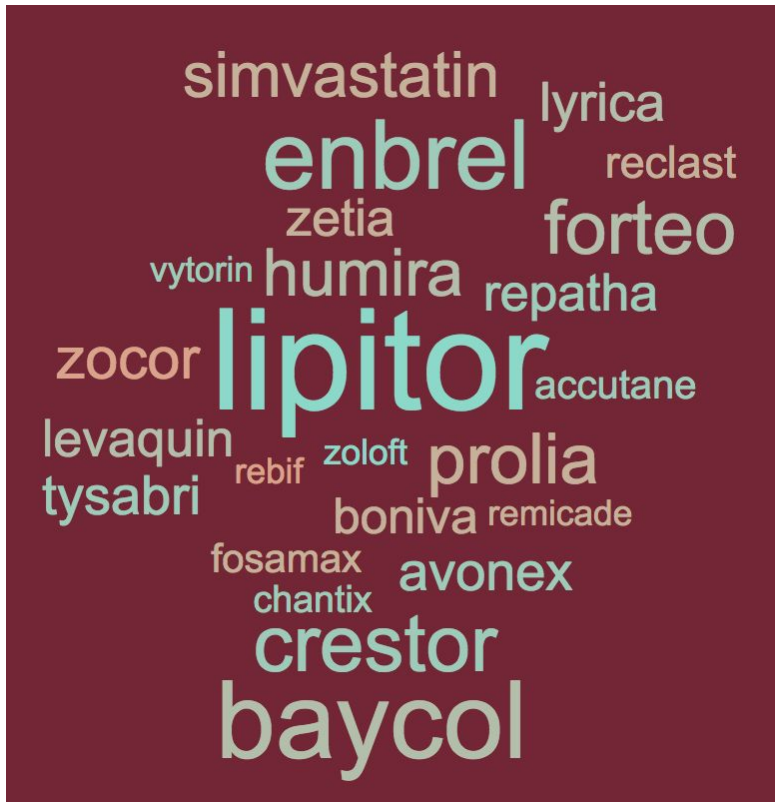
Levaquin ↔ Cipro

Cases reported (top 25)

All symptoms	9822	9645
Arthralgia	1277	1131
Tendon rupture	1178	410
Tendonitis	1062	558
Tendon injury	969	87
Pain in extremity	818	736
Rotator cuff syndrome	615	46
Insomnia	553	456

(MedicalSideFx IOS App screenshot)

- What are the medicines that result in a specific side effect ?



(wordcloud generated using R on results from hive queries)

On the left we have the top 25 medicines that have reported myalgia(muscular pain) as a side effect

On the right we have the top 25 medicines that have reported some kind of a hepatic (liver) dysfunction or failure

Where do we get the adverse events data ?

FDA Adverse Event Reporting System (FAERS): Latest Quarterly Data Files

<https://www.fda.gov/Drugs/GuidanceComplianceRegulatoryInformation/Surveillance/AdverseDrugEffects/ucm082193.htm>

FAERS_ASCII_2017q2 (ZIP - 41.3MB)

FAERS_ASCII_2017q1 (ZIP - 42.6MB)

FAERS ASCII 2016q4 (ZIP - 39.2MB)

FAERS ASCII 2016q3 (ZIP - 40.6MB)

FAERS ASCII 2016q2 (ZIP - 42.3MB)

FAERS ASCII 2016q1 (ZIP - 43.7MB)

FAERS ASCII 2015q4 (ZIP - 39.7MB)

FAERS ASCII 2015q3 (ZIP - 44.7MB)

FAERS ASCII 2015q2.zip (ZIP - 36.4MB)

FAERS ASCII 2015q1.zip (ZIP - 37MB)

FAERS ASCII 2014q4.zip (ZIP - 26.8MB)

FAERS_ASCII_2014q3.zip (ZIP - 27MB)

FAERS_ASCII_2014q2.zip (ZIP - 23.7MB)

FAERS_ASCII_2014q1.zip (ZIP - 27.9MB)

FAERS_ASCII_2013q4.zip (ZIP - 24.4MB)

FAERS_ASCII_2013q3.zip (ZIP - 21MB)

FAERS_ASCII_2013q2.zip (ZIP - 19.9MB)

FAERS_ASCII_2013q1.zip (ZIP - 23.7MB)

FAERS_ASCII_2012q4.zip (ZIP - 26.3MB)

The Adverse Event Reporting System (AERS): Older Quarterly Data Files

<https://wayback.archive-it.org/7993/20170404211700/https://www.fda.gov/Drugs/GuidanceComplianceRegulatoryInformation/Surveillance/AdverseDrugEffects/ucm083765.htm>

AERS_ASCII_2012q3.zip (ZIP - 15.3MB)

AERS_ASCII_2012q2.ZIP (ZIP - 24.7MB)

AERS_ASCII_2012q1.ZIP (ZIP - 25.4MB)

AERS_ASCII_2011q4.ZIP (22.5 MB)
AERS_ASCII_2011q3.ZIP (22.3 MB)
AERS_ASCII_2011q2.ZIP (22.4 MB)
AERS_ASCII_2011q1.ZIP (20.3 MB)
AERS_ASCII_2010q4.ZIP (19.2 MB)
AERS_ASCII_2010q3.ZIP (21.8 MB)
AERS_ASCII_2010q2.ZIP (17.3 MB)
AERS_ASCII_2010q1.ZIP (15.5 MB)
AERS_ASCII_2009q4.ZIP (15.5 MB)
AERS_ASCII_2009q3.ZIP (15.0 MB)
AERS_ASCII_2009q2.ZIP (13.6 MB)
AERS_ASCII_2009q1.ZIP (12.4 MB)
AERS_ASCII_2008q4.ZIP (12.8 MB)
AERS_ASCII_2008q3.ZIP (11.3 MB)
AERS_ASCII_2008q2.ZIP (11.3 MB)
AERS_ASCII_2008q1.ZIP (11.3 MB)
AERS_ASCII_2007q4.ZIP (11.3 MB)
AERS_ASCII_2007q3.ZIP (9.7 MB)
AERS_ASCII_2007q2.ZIP (9.5 MB)
AERS_ASCII_2007q1.ZIP (9.5 MB)
AERS_ASCII_2006q4.ZIP (9 MB)
AERS_ASCII_2006q3.ZIP (8.5 MB)
AERS_ASCII_2006q2.ZIP (10.4 MB)
AERS_ASCII_2006q1.ZIP (10.4 MB)
AERS_ASCII_2005q4.ZIP (9.7 MB)
AERS_ASCII_2005q3.ZIP (9.3 MB)
AERS_ASCII_2005q2.ZIP (9.5 MB)
AERS_ASCII_2005q1.ZIP (8.5 MB)
AERS_ASCII_2004q4.zip (8.1 MB)
AERS_ASCII_2004q3.zip (8.1 MB)
AERS_ASCII_2004q2.zip (7.1 MB)

AERS_ASCII_2004q1.zip (7.2 MB)

Anatomy of a quarterly data zip file

Broadly FDA publishes seven categories of data in each quarterly data update

DEMO → Demographic

DRUG → Drug

INDI → Indication

OUTC → Outcome

REAC → Reaction

RPSR → Report Sources

THER → Therapy

A listing of the files for the latest quarter 2017Q1 (at the time of writing) looks like the following (ascii datasets

faers_ascii_2017q1/ascii/ASC_NTS.pdf

faers_ascii_2017q1/ascii/demo17q1.pdf

[faers_ascii_2017q1/ascii/DEMO17Q1.txt](#)

faers_ascii_2017q1/ascii/drug17q1.pdf

[faers_ascii_2017q1/ascii/DRUG17Q1.txt](#)

faers_ascii_2017q1/ascii/indi17q1.pdf

[faers_ascii_2017q1/ascii/INDI17Q1.txt](#)

faers_ascii_2017q1/ascii/outc17q1.pdf

[faers_ascii_2017q1/ascii/OUTC17Q1.txt](#)

faers_ascii_2017q1/ascii/reac17q1.pdf

[faers_ascii_2017q1/ascii/REAC17Q1.txt](#)

faers_ascii_2017q1/ascii/rpsr17q1.pdf

[faers_ascii_2017q1/ascii/RPSR17Q1.txt](#)

faers_ascii_2017q1/ascii/ther17q1.pdf

[faers_ascii_2017q1/ascii/THER17Q1.txt](#)

faers_ascii_2017q1/FAQs.pdf

faers_ascii_2017q1/Readme.pdf

Scope of this book

In this book we will focus on three categories of datasets

DEMO → Demographic

DRUG → Drug

REAC → Reaction

Some quick counts

At the time of writing here are the counts from the datasets 1999-2017Q1

Demographics = 10,223,368 records

Drug = 36,270,581 records

Reaction = 34,244,838 records

Metadata changes over the years

Demographic Dataset Columns variations

demoHeaderVersion1 =

"ISR\$CASE\$I_F_COD\$FOLL_SEQ\$IMAGE\$EVENT_DT\$MFR_DT\$FDA_DT\$REPT_COD\$MFR_NUM\$MFR_SNDR\$AGE\$AGE_C
OD\$GNDR_COD"

demoHeaderVersion2 =

"ISR\$CASE\$I_F_COD\$FOLL_SEQ\$IMAGE\$EVENT_DT\$MFR_DT\$FDA_DT\$REPT_COD\$MFR_NUM\$MFR_SNDR\$AGE\$AGE_C
OD\$GNDR_COD\$BEST_ISR\$E_SUB\$WT\$WT_COD\$REPT_DT\$OCCP_COD\$DEATH_DT\$TO_MFR\$CONFID"

demoHeaderVersion3 =

"ISR\$CASE\$I_F_COD\$FOLL_SEQ\$IMAGE\$EVENT_DT\$MFR_DT\$FDA_DT\$REPT_COD\$MFR_NUM\$MFR_SNDR\$AGE\$AGE_C
OD\$GNDR_COD\$E_SUB\$WT\$WT_COD\$REPT_DT\$OCCP_COD\$DEATH_DT\$TO_MFR\$CONFID"

demoHeaderVersion4 =

"ISR\$CASE\$I_F_COD\$FOLL_SEQ\$IMAGE\$EVENT_DT\$MFR_DT\$FDA_DT\$REPT_COD\$MFR_NUM\$MFR_SNDR\$AGE\$AGE_C
OD\$GNDR_COD\$E_SUB\$WT\$WT_COD\$REPT_DT\$OCCP_COD\$DEATH_DT\$TO_MFR\$CONFID\$REPORTER_COUNTRY"

demoHeaderVersion5 =

"primaryid\$caseid\$caseversion\$i_f_code\$event_dt\$mfr_dt\$init_fda_dt\$fda_dt\$rept_cod\$mfr_num\$mfr_sndr\$age\$age_cod\$gndr_co
d\$e_sub\$wt\$wt_cod\$rept_dt\$to_mfr\$occp_cod\$reporter_country\$occr_country"

demoHeaderVersion6 =

"primaryid\$caseid\$caseversion\$i_f_code\$event_dt\$mfr_dt\$init_fda_dt\$fda_dt\$rept_cod\$auth_num\$mfr_num\$mfr_sndr\$lit_ref\$age\$
age_cod\$age_grp\$sex\$e_sub\$wt\$wt_cod\$rept_dt\$to_mfr\$occp_cod\$reporter_country\$occr_country"

Demographic quarterly dataset to variations mapping

1999Q1,demoHeaderVersion1

1999Q2,demoHeaderVersion1

1999Q3,demoHeaderVersion1

1999Q4,demoHeaderVersion1

2000Q1,demoHeaderVersion1

2000Q2,demoHeaderVersion1

2000Q3,demoHeaderVersion1

2000Q4,demoHeaderVersion1

2001Q1,demoHeaderVersion1

2001Q2,demoHeaderVersion1

2001Q3,demoHeaderVersion1

2001Q4,demoHeaderVersion1

2002Q1,demoHeaderVersion1

2002Q2,demoHeaderVersion2
2002Q3,demoHeaderVersion3
2002Q4,demoHeaderVersion3
2003Q1,demoHeaderVersion3
2003Q2,demoHeaderVersion3
2003Q3,demoHeaderVersion3
2003Q4,demoHeaderVersion3
2004Q1,demoHeaderVersion3
2004Q2,demoHeaderVersion3
2004Q3,demoHeaderVersion3
2004Q4,demoHeaderVersion3
2005Q1,demoHeaderVersion3
2005Q2,demoHeaderVersion3
2005Q3,demoHeaderVersion4
2005Q4,demoHeaderVersion4
2006Q1,demoHeaderVersion4
2006Q2,demoHeaderVersion4
2006Q3,demoHeaderVersion4
2006Q4,demoHeaderVersion4
2007Q1,demoHeaderVersion4
2007Q2,demoHeaderVersion4
2007Q3,demoHeaderVersion4
2007Q4,demoHeaderVersion4
2008Q1,demoHeaderVersion4
2008Q2,demoHeaderVersion4
2008Q3,demoHeaderVersion4
2008Q4,demoHeaderVersion4
2009Q1,demoHeaderVersion4
2009Q2,demoHeaderVersion4
2009Q3,demoHeaderVersion4
2009Q4,demoHeaderVersion4

2010Q1,demoHeaderVersion4
2010Q2,demoHeaderVersion4
2010Q3,demoHeaderVersion4
2010Q4,demoHeaderVersion4
2011Q1,demoHeaderVersion4
2011Q2,demoHeaderVersion4
2011Q3,demoHeaderVersion4
2011Q4,demoHeaderVersion4
2012Q1,demoHeaderVersion4
2012Q2,demoHeaderVersion4
2012Q3,demoHeaderVersion4
2012Q4,demoHeaderVersion5
2013Q1,demoHeaderVersion5
2013Q2,demoHeaderVersion5
2013Q3,demoHeaderVersion5
2013Q4,demoHeaderVersion5
2014Q1,demoHeaderVersion5
2014Q2,demoHeaderVersion5
2014Q3,demoHeaderVersion6
2014Q4,demoHeaderVersion6
2015Q1,demoHeaderVersion6
2015Q2,demoHeaderVersion6
2015Q3,demoHeaderVersion6
2015Q4,demoHeaderVersion6
2016Q1,demoHeaderVersion6
2016Q2,demoHeaderVersion6
2016Q3,demoHeaderVersion6
2016Q4,demoHeaderVersion6
2017Q1,demoHeaderVersion6
2017Q2,demoHeaderVersion6

Drug Dataset Column variations

drugHeaderVersion1 = "ISR\$DRUG_SEQ\$ROLE_COD\$DRUGNAME\$VAL_VBM\$ROUTE\$DOSE_VBM\$DECHAL\$RECHAL"

drugHeaderVersion2 =

"ISR\$DRUG_SEQ\$ROLE_COD\$DRUGNAME\$VAL_VBM\$ROUTE\$DOSE_VBM\$DECHAL\$RECHAL\$LOT_NUM\$EXP_DT\$NDA_NUM"

drugheaderVersion3 =

"primaryid\$caseid\$drug_seq\$role_cod\$drugname\$val_vbm\$route\$dose_vbm\$cum_dose_chr\$cum_dose_unit\$dechal\$rechal\$lot_nr\$exp_dt\$nda_num\$dose_amt\$dose_unit\$dose_form\$dose_freq"

drugheaderVersion4 =

"primaryid\$caseid\$drug_seq\$role_cod\$drugname\$val_vbm\$route\$dose_vbm\$cum_dose_chr\$cum_dose_unit\$dechal\$rechal\$lot_num\$exp_dt\$nda_num\$dose_amt\$dose_unit\$dose_form\$dose_freq"

drugheaderVersion5 =

"primaryid\$caseid\$drug_seq\$role_cod\$drugname\$prod_ai\$val_vbm\$route\$dose_vbm\$cum_dose_chr\$cum_dose_unit\$dechal\$rechal\$lot_num\$exp_dt\$nda_num\$dose_amt\$dose_unit\$dose_form\$dose_freq"

Drug quarterly dataset to variations mapping

1999Q1,drugHeaderVersion1

1999Q2,drugHeaderVersion1

1999Q3,drugHeaderVersion1

1999Q4,drugHeaderVersion1

2000Q1,drugHeaderVersion1

2000Q2,drugHeaderVersion1

2000Q3,drugHeaderVersion1

2000Q4,drugHeaderVersion1

2001Q1,drugHeaderVersion1

2001Q2,drugHeaderVersion1

2001Q3,drugHeaderVersion1

2001Q4,drugHeaderVersion1

2002Q1,drugHeaderVersion1
2002Q2,drugHeaderVersion2
2002Q3,drugHeaderVersion2
2002Q4,drugHeaderVersion2
2003Q1,drugHeaderVersion2
2003Q2,drugHeaderVersion2
2003Q3,drugHeaderVersion2
2003Q4,drugHeaderVersion2
2004Q1,drugHeaderVersion2
2004Q2,drugHeaderVersion2
2004Q3,drugHeaderVersion2
2004Q4,drugHeaderVersion2
2005Q1,drugHeaderVersion2
2005Q2,drugHeaderVersion2
2005Q3,drugHeaderVersion2
2005Q4,drugHeaderVersion2
2006Q1,drugHeaderVersion2
2006Q2,drugHeaderVersion2
2006Q3,drugHeaderVersion2
2006Q4,drugHeaderVersion2
2007Q1,drugHeaderVersion2
2007Q2,drugHeaderVersion2
2007Q3,drugHeaderVersion2
2007Q4,drugHeaderVersion2
2008Q1,drugHeaderVersion2
2008Q2,drugHeaderVersion2
2008Q3,drugHeaderVersion2
2008Q4,drugHeaderVersion2
2009Q1,drugHeaderVersion2
2009Q2,drugHeaderVersion2
2009Q3,drugHeaderVersion2

2009Q4,drugHeaderVersion2
2010Q1,drugHeaderVersion2
2010Q2,drugHeaderVersion2
2010Q3,drugHeaderVersion2
2010Q4,drugHeaderVersion2
2011Q1,drugHeaderVersion2
2011Q2,drugHeaderVersion2
2011Q3,drugHeaderVersion2
2011Q4,drugHeaderVersion2
2012Q1,drugHeaderVersion2
2012Q2,drugHeaderVersion2
2012Q3,drugHeaderVersion2
2012Q4,drugheaderVersion3
2013Q1,drugheaderVersion3
2013Q2,drugheaderVersion3
2013Q3,drugheaderVersion4
2013Q4,drugheaderVersion4
2014Q1,drugheaderVersion4
2014Q2,drugheaderVersion4
2014Q3,drugheaderVersion5
2014Q4,drugheaderVersion5
2015Q1,drugheaderVersion5
2015Q2,drugheaderVersion5
2015Q3,drugheaderVersion5
2015Q4,drugheaderVersion5
2016Q1,drugheaderVersion5
2016Q2,drugheaderVersion5
2016Q3,drugheaderVersion5
2016Q4,drugheaderVersion5
2017Q1,drugheaderVersion5
2017Q2,drugheaderVersion5

Reaction Dataset Column variations

reachHeaderVersion1 = "ISR\$PT

reachHeaderVersion2 = "primaryid\$caseid\$pt

reachHeaderVersion3 = "primaryid\$caseid\$pt\$drug_rec_act

Reactions quarterly dataset to variations mapping

1999Q1,reachHeaderVersion1

1999Q2,reachHeaderVersion1

1999Q3,reachHeaderVersion1

1999Q4,reachHeaderVersion1

2000Q1,reachHeaderVersion1

2000Q2,reachHeaderVersion1

2000Q3,reachHeaderVersion1

2000Q4,reachHeaderVersion1

2001Q1,reachHeaderVersion1

2001Q2,reachHeaderVersion1

2001Q3,reachHeaderVersion1

2001Q4,reachHeaderVersion1

2002Q1,reachHeaderVersion1

2002Q2,reachHeaderVersion1

2002Q3,reachHeaderVersion1

2002Q4,reachHeaderVersion1

2003Q1,reachHeaderVersion1

2003Q2,reachHeaderVersion1

2003Q3,reachHeaderVersion1

2003Q4,reachHeaderVersion1

2004Q1,reachHeaderVersion1

2004Q2,reachHeaderVersion1
2004Q3,reachHeaderVersion1
2004Q4,reachHeaderVersion1
2005Q1,reachHeaderVersion1
2005Q2,reachHeaderVersion1
2005Q3,reachHeaderVersion1
2005Q4,reachHeaderVersion1
2006Q1,reachHeaderVersion1
2006Q2,reachHeaderVersion1
2006Q3,reachHeaderVersion1
2006Q4,reachHeaderVersion1
2007Q1,reachHeaderVersion1
2007Q2,reachHeaderVersion1
2007Q3,reachHeaderVersion1
2007Q4,reachHeaderVersion1
2008Q1,reachHeaderVersion1
2008Q2,reachHeaderVersion1
2008Q3,reachHeaderVersion1
2008Q4,reachHeaderVersion1
2009Q1,reachHeaderVersion1
2009Q2,reachHeaderVersion1
2009Q3,reachHeaderVersion1
2009Q4,reachHeaderVersion1
2010Q1,reachHeaderVersion1
2010Q2,reachHeaderVersion1
2010Q3,reachHeaderVersion1
2010Q4,reachHeaderVersion1
2011Q1,reachHeaderVersion1
2011Q2,reachHeaderVersion1
2011Q3,reachHeaderVersion1
2011Q4,reachHeaderVersion1

2012Q1,reachHeaderVersion1
2012Q2,reachHeaderVersion1
2012Q3,reachHeaderVersion1
2012Q4,reachHeaderVersion2
2013Q1,reachHeaderVersion2
2013Q2,reachHeaderVersion2
2013Q3,reachHeaderVersion2
2013Q4,reachHeaderVersion2
2014Q1,reachHeaderVersion2
2014Q2,reachHeaderVersion2
2014Q3,reachHeaderVersion3
2014Q4,reachHeaderVersion3
2015Q1,reachHeaderVersion3
2015Q2,reachHeaderVersion3
2015Q3,reachHeaderVersion3
2015Q4,reachHeaderVersion3
2016Q1,reachHeaderVersion3
2016Q2,reachHeaderVersion3
2016Q3,reachHeaderVersion3
2016Q4,reachHeaderVersion3
2017Q1,reachHeaderVersion3
2017Q2,reachHeaderVersion3

Hadoop ecosystem

I used Cloudera Standard (configured using Cloudera Manager) on a on-premises hadoop cluster with Hive configured. The version used is CDH 5.10.0 at the time of writing this book.

Loading the data into HDFS

First we remove the headers from all datasets before loading then into HDFS.

For one quarter my HDFS directory structure looks as follows

```
/data/aers/quarterly_files/faers_ascii_2017q1/ascii/demo/DEMO17Q1.txt.nohead  
/data/aers/quarterly_files/faers_ascii_2017q1/ascii/drug/DRUG17Q1.txt.nohead  
/data/aers/quarterly_files/faers_ascii_2017q1/ascii/indi/INDI17Q1.txt.nohead  
/data/aers/quarterly_files/faers_ascii_2017q1/ascii/outc/OUTC17Q1.txt.nohead  
/data/aers/quarterly_files/faers_ascii_2017q1/ascii/reac/REAC17Q1.txt.nohead  
/data/aers/quarterly_files/faers_ascii_2017q1/ascii/rpsr/RPSR17Q1.txt.nohead  
/data/aers/quarterly_files/faers_ascii_2017q1/ascii/ther/THER17Q1.txt.nohead
```

Before Hive came along

I had started this initiative in 2011 and Hive was yet not firmly established in the Hadoop ecosystem. Hence a large part of the analytics was done using custom map reduce code in Java.

The clear advantage of Hive (given its SQL like syntax) is that it allows programmers as well as technical analysts to run queries and analyze the data as long as they have access to a hadoop ecosystem.

Creating Hive tables

Creating Demographic tables

```
CREATE DATABASE IF NOT EXISTS
  aers
;

USE aers;

DROP TABLE IF EXISTS aers_demo_v1
;
CREATE EXTERNAL TABLE aers_demo_v1 (
  ISR BIGINT,
  CASE_ BIGINT,
  I_F_COD STRING,
  FOLL_SEQ BIGINT,
  IMAGE STRING,
  EVENT_DT BIGINT,
  MFR_DT BIGINT,
  FDA_DT BIGINT,
  REPT_COD STRING,
  MFR_NUM STRING,
  MFR_SNDR STRING,
  AGE DOUBLE,
  AGE_COD STRING,
  GNDR_COD STRING )
PARTITIONED BY (YEAR INT, QUARTER INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY '$'
;

DROP TABLE IF EXISTS aers_demo_v2
```

```

;
CREATE EXTERNAL TABLE aers_demo_v2 (
  ISR BIGINT,
  CASE_ BIGINT,
  I_F_COD STRING,
  FOLL_SEQ BIGINT,
  IMAGE STRING,
  EVENT_DT BIGINT,
  MFR_DT BIGINT,
  FDA_DT BIGINT,
  REPT_COD STRING,
  MFR_NUM STRING,
  MFR_SNR STRING,
  AGE DOUBLE,
  AGE_COD STRING,
  GNDR_COD STRING,
  BEST_ISR STRING,
  E_SUB STRING,
  WT DOUBLE,
  WT_COD STRING,
  REPT_DT BIGINT,
  OCCP_COD STRING,
  DEATH_DT BIGINT,
  TO_MFR STRING,
  CONFID STRING )
PARTITIONED BY (YEAR INT, QUARTER INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY '$'
;

DROP TABLE IF EXISTS aers_demo_v3
;
CREATE EXTERNAL TABLE aers_demo_v3 (
  ISR BIGINT,

```

```

CASE_ BIGINT,
I_F_COD STRING,
FOLL_SEQ BIGINT,
IMAGE STRING,
EVENT_DT BIGINT,
MFR_DT BIGINT,
FDA_DT BIGINT,
REPT_COD STRING,
MFR_NUM STRING,
MFR_SNR STRING,
AGE DOUBLE,
AGE_COD STRING,
GNDR_COD STRING,
E_SUB STRING,
WT DOUBLE,
WT_COD STRING,
REPT_DT BIGINT,
OCCP_COD STRING,
DEATH_DT BIGINT,
TO_MFR STRING,
CONFID STRING )
PARTITIONED BY (YEAR INT, QUARTER INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY '$'
;

```

```

DROP TABLE IF EXISTS aers_demo_v4
;
CREATE EXTERNAL TABLE aers_demo_v4 (
ISR BIGINT,
CASE_ BIGINT,
I_F_COD STRING,
FOLL_SEQ BIGINT,
IMAGE STRING,

```

```

EVENT_DT BIGINT,
MFR_DT BIGINT,
FDA_DT BIGINT,
REPT_COD STRING,
MFR_NUM STRING,
MFR_SNR STRING,
AGE DOUBLE,
AGE_COD STRING,
GNDR_COD STRING,
E_SUB STRING,
WT DOUBLE,
WT_COD DOUBLE,
REPT_DT BIGINT,
OCCP_COD STRING,
DEATH_DT BIGINT,
TO_MFR STRING,
CONFID STRING,
REPORTER_COUNTRY STRING )
PARTITIONED BY (YEAR INT, QUARTER INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY '$'
;

DROP TABLE IF EXISTS aers_demo_v5
;
CREATE EXTERNAL TABLE aers_demo_v5 (
primaryid BIGINT,
caseid BIGINT,
caseversion INT,
i_f_code STRING,
event_dt BIGINT,
mfr_dt BIGINT,
init_fda_dt BIGINT,
fda_dt BIGINT,

```

```

rept_cod STRING,
mfr_num STRING,
mfr_sndr STRING,
age DOUBLE,
age_cod STRING,
gndr_cod STRING,
e_sub STRING,
wt DOUBLE,
wt_cod STRING,
rept_dt BIGINT,
to_mfr STRING,
occp_cod STRING,
reporter_country STRING,
occr_country STRING )
PARTITIONED BY (YEAR INT, QUARTER INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY '$'
;

```

```

DROP TABLE IF EXISTS aers_demo_v6
;

```

```

CREATE EXTERNAL TABLE aers_demo_v6 (
primaryid BIGINT,
caseid BIGINT,
caseversion INT,
i_f_code STRING,
event_dt BIGINT,
mfr_dt BIGINT,
init_fda_dt BIGINT,
fda_dt BIGINT,
rept_cod STRING,
auth_num STRING,
mfr_num STRING,
mfr_sndr STRING,

```

```
lit_ref STRING,  
age DOUBLE,  
age_cod STRING,  
age_grp STRING,  
sex STRING,  
e_sub STRING,  
wt DOUBLE,  
wt_cod STRING,  
rept_dt BIGINT,  
to_mfr STRING,  
occp_cod STRING,  
reporter_country STRING,  
occr_country STRING )  
PARTITIONED BY (YEAR INT, QUARTER INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY '$'  
;
```

Partitioning Demographic tables

```
USE AERS;  
  
ALTER TABLE aers_demo_v1 ADD PARTITION (YEAR=1999, QUARTER=1) LOCATION  
'/data/aers/quarterly_files/aers_ascii_1999q1/ascii/demo';  
ALTER TABLE aers_demo_v1 ADD PARTITION (YEAR=1999, QUARTER=2) LOCATION  
'/data/aers/quarterly_files/aers_ascii_1999q2/ascii/demo';  
ALTER TABLE aers_demo_v1 ADD PARTITION (YEAR=1999, QUARTER=3) LOCATION  
'/data/aers/quarterly_files/aers_ascii_1999q3/ascii/demo';  
ALTER TABLE aers_demo_v1 ADD PARTITION (YEAR=1999, QUARTER=4) LOCATION  
'/data/aers/quarterly_files/aers_ascii_1999q4/ascii/demo';  
ALTER TABLE aers_demo_v1 ADD PARTITION (YEAR=2000, QUARTER=1) LOCATION
```

```

'/data/aers/quarterly_files/aers_ascii_2000q1/ascii/demo';
ALTER TABLE aers_demo_v1 ADD PARTITION (YEAR=2000, QUARTER=2) LOCATION
'/data/aers/quarterly_files/aers_ascii_2000q2/ascii/demo';
ALTER TABLE aers_demo_v1 ADD PARTITION (YEAR=2000, QUARTER=3) LOCATION
'/data/aers/quarterly_files/aers_ascii_2000q3/ascii/demo';
ALTER TABLE aers_demo_v1 ADD PARTITION (YEAR=2000, QUARTER=4) LOCATION
'/data/aers/quarterly_files/aers_ascii_2000q4/ascii/demo';
ALTER TABLE aers_demo_v1 ADD PARTITION (YEAR=2001, QUARTER=1) LOCATION
'/data/aers/quarterly_files/aers_ascii_2001q1/ascii/demo';
ALTER TABLE aers_demo_v1 ADD PARTITION (YEAR=2001, QUARTER=2) LOCATION
'/data/aers/quarterly_files/aers_ascii_2001q2/ascii/demo';
ALTER TABLE aers_demo_v1 ADD PARTITION (YEAR=2001, QUARTER=3) LOCATION
'/data/aers/quarterly_files/aers_ascii_2001q3/ascii/demo';
ALTER TABLE aers_demo_v1 ADD PARTITION (YEAR=2001, QUARTER=4) LOCATION
'/data/aers/quarterly_files/aers_ascii_2001q4/ascii/demo';
ALTER TABLE aers_demo_v1 ADD PARTITION (YEAR=2002, QUARTER=1) LOCATION
'/data/aers/quarterly_files/aers_ascii_2002q1/ascii/demo';

ALTER TABLE aers_demo_v2 ADD PARTITION (YEAR=2002, QUARTER=2) LOCATION
'/data/aers/quarterly_files/aers_ascii_2002q2/ascii/demo';

ALTER TABLE aers_demo_v3 ADD PARTITION (YEAR=2002, QUARTER=3) LOCATION
'/data/aers/quarterly_files/aers_ascii_2002q3/ascii/demo';
ALTER TABLE aers_demo_v3 ADD PARTITION (YEAR=2002, QUARTER=4) LOCATION
'/data/aers/quarterly_files/aers_ascii_2002q4/ascii/demo';
ALTER TABLE aers_demo_v3 ADD PARTITION (YEAR=2003, QUARTER=1) LOCATION
'/data/aers/quarterly_files/aers_ascii_2003q1/ascii/demo';
ALTER TABLE aers_demo_v3 ADD PARTITION (YEAR=2003, QUARTER=2) LOCATION
'/data/aers/quarterly_files/aers_ascii_2003q2/ascii/demo';
ALTER TABLE aers_demo_v3 ADD PARTITION (YEAR=2003, QUARTER=3) LOCATION
'/data/aers/quarterly_files/aers_ascii_2003q3/ascii/demo';
ALTER TABLE aers_demo_v3 ADD PARTITION (YEAR=2003, QUARTER=4) LOCATION
'/data/aers/quarterly_files/aers_ascii_2003q4/ascii/demo';

```



```

ALTER TABLE aers_demo_v3 ADD PARTITION (YEAR=2004, QUARTER=1) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2004q1/ascii/demo';
ALTER TABLE aers_demo_v3 ADD PARTITION (YEAR=2004, QUARTER=2) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2004q2/ascii/demo';
ALTER TABLE aers_demo_v3 ADD PARTITION (YEAR=2004, QUARTER=3) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2004q3/ascii/demo';
ALTER TABLE aers_demo_v3 ADD PARTITION (YEAR=2004, QUARTER=4) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2004q4/ascii/demo';
ALTER TABLE aers_demo_v3 ADD PARTITION (YEAR=2005, QUARTER=1) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2005Q1/ascii/demo';
ALTER TABLE aers_demo_v3 ADD PARTITION (YEAR=2005, QUARTER=2) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2005Q2/ascii/demo';

ALTER TABLE aers_demo_v4 ADD PARTITION (YEAR=2005, QUARTER=3) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2005Q3/ascii/demo';
ALTER TABLE aers_demo_v4 ADD PARTITION (YEAR=2005, QUARTER=4) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2005Q4/ascii/demo';
ALTER TABLE aers_demo_v4 ADD PARTITION (YEAR=2006, QUARTER=1) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2006Q1/ascii/demo';
ALTER TABLE aers_demo_v4 ADD PARTITION (YEAR=2006, QUARTER=2) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2006Q2/ascii/demo';
ALTER TABLE aers_demo_v4 ADD PARTITION (YEAR=2006, QUARTER=3) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2006Q3/ascii/demo';
ALTER TABLE aers_demo_v4 ADD PARTITION (YEAR=2006, QUARTER=4) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2006Q4/ascii/demo';
ALTER TABLE aers_demo_v4 ADD PARTITION (YEAR=2007, QUARTER=1) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2007Q1/ascii/demo';
ALTER TABLE aers_demo_v4 ADD PARTITION (YEAR=2007, QUARTER=2) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2007Q2/ascii/demo';
ALTER TABLE aers_demo_v4 ADD PARTITION (YEAR=2007, QUARTER=3) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2007Q3/ascii/demo';
ALTER TABLE aers_demo_v4 ADD PARTITION (YEAR=2007, QUARTER=4) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2007Q4/ascii/demo';

```

```
ALTER TABLE aers_demo_v4 ADD PARTITION (YEAR=2008, QUARTER=1) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2008Q1/ascii/demo';
ALTER TABLE aers_demo_v4 ADD PARTITION (YEAR=2008, QUARTER=2) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2008Q2/ascii/demo';
ALTER TABLE aers_demo_v4 ADD PARTITION (YEAR=2008, QUARTER=3) LOCATION
'/data/aers/quarterly_files/aers_ascii_2008q3/ascii/demo';
ALTER TABLE aers_demo_v4 ADD PARTITION (YEAR=2008, QUARTER=4) LOCATION
'/data/aers/quarterly_files/aers_ascii_2008q4/ascii/demo';
ALTER TABLE aers_demo_v4 ADD PARTITION (YEAR=2009, QUARTER=1) LOCATION
'/data/aers/quarterly_files/aers_ascii_2009q1/ascii/demo';
ALTER TABLE aers_demo_v4 ADD PARTITION (YEAR=2009, QUARTER=2) LOCATION
'/data/aers/quarterly_files/aers_ascii_2009q2/ascii/demo';
ALTER TABLE aers_demo_v4 ADD PARTITION (YEAR=2009, QUARTER=3) LOCATION
'/data/aers/quarterly_files/aers_ascii_2009q3/ascii/demo';
ALTER TABLE aers_demo_v4 ADD PARTITION (YEAR=2009, QUARTER=4) LOCATION
'/data/aers/quarterly_files/aers_ascii_2009q4/ascii/demo';
ALTER TABLE aers_demo_v4 ADD PARTITION (YEAR=2010, QUARTER=1) LOCATION
'/data/aers/quarterly_files/aers_ascii_2010q1/ascii/demo';
ALTER TABLE aers_demo_v4 ADD PARTITION (YEAR=2010, QUARTER=2) LOCATION
'/data/aers/quarterly_files/aers_ascii_2010q2/ascii/demo';
ALTER TABLE aers_demo_v4 ADD PARTITION (YEAR=2010, QUARTER=3) LOCATION
'/data/aers/quarterly_files/aers_ascii_2010q3/ascii/demo';
ALTER TABLE aers_demo_v4 ADD PARTITION (YEAR=2010, QUARTER=4) LOCATION
'/data/aers/quarterly_files/aers_ascii_2010q4/ascii/demo';
ALTER TABLE aers_demo_v4 ADD PARTITION (YEAR=2011, QUARTER=1) LOCATION
'/data/aers/quarterly_files/aers_ascii_2011q1/ascii/demo';
ALTER TABLE aers_demo_v4 ADD PARTITION (YEAR=2011, QUARTER=2) LOCATION
'/data/aers/quarterly_files/aers_ascii_2011q2/ascii/demo';
ALTER TABLE aers_demo_v4 ADD PARTITION (YEAR=2011, QUARTER=3) LOCATION
'/data/aers/quarterly_files/aers_ascii_2011q3/ascii/demo';
ALTER TABLE aers_demo_v4 ADD PARTITION (YEAR=2011, QUARTER=4) LOCATION
'/data/aers/quarterly_files/aers_ascii_2011q4/ascii/demo';
ALTER TABLE aers_demo_v4 ADD PARTITION (YEAR=2012, QUARTER=1) LOCATION
```

```

'/data/aers/quarterly_files/aers_ascii_2012q1/ascii/demo';
ALTER TABLE aers_demo_v4 ADD PARTITION (YEAR=2012, QUARTER=2) LOCATION
'/data/aers/quarterly_files/aers_ascii_2012q2/ascii/demo';
ALTER TABLE aers_demo_v4 ADD PARTITION (YEAR=2012, QUARTER=3) LOCATION
'/data/aers/quarterly_files/aers_ascii_2012q3/ascii/demo';

ALTER TABLE aers_demo_v5 ADD PARTITION (YEAR=2012, QUARTER=4) LOCATION
'/data/aers/quarterly_files/faers_ascii_2012q4/ascii/demo';
ALTER TABLE aers_demo_v5 ADD PARTITION (YEAR=2013, QUARTER=1) LOCATION
'/data/aers/quarterly_files/faers_ascii_2013q1/ascii/demo';
ALTER TABLE aers_demo_v5 ADD PARTITION (YEAR=2013, QUARTER=2) LOCATION
'/data/aers/quarterly_files/faers_ascii_2013q2/ascii/demo';
ALTER TABLE aers_demo_v5 ADD PARTITION (YEAR=2013, QUARTER=3) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2013q3/ascii/demo';
ALTER TABLE aers_demo_v5 ADD PARTITION (YEAR=2013, QUARTER=4) LOCATION
'/data/aers/quarterly_files/FAERS_ASCII_2013Q4/ascii/demo';
ALTER TABLE aers_demo_v5 ADD PARTITION (YEAR=2014, QUARTER=1) LOCATION
'/data/aers/quarterly_files/faers_ascii_2014q1/ascii/demo';
ALTER TABLE aers_demo_v5 ADD PARTITION (YEAR=2014, QUARTER=2) LOCATION
'/data/aers/quarterly_files/faers_ascii_2014q2/ascii/demo';

ALTER TABLE aers_demo_v6 ADD PARTITION (YEAR=2014, QUARTER=3) LOCATION
'/data/aers/quarterly_files/FAERS_ASCII_2014Q3/ascii/demo';
ALTER TABLE aers_demo_v6 ADD PARTITION (YEAR=2014, QUARTER=4) LOCATION
'/data/aers/quarterly_files/FAERS_ASCII_2014Q4/ascii/demo';
ALTER TABLE aers_demo_v6 ADD PARTITION (YEAR=2015, QUARTER=1) LOCATION
'/data/aers/quarterly_files/FAERS_ASCII_2015Q1/ascii/demo';
ALTER TABLE aers_demo_v6 ADD PARTITION (YEAR=2015, QUARTER=2) LOCATION
'/data/aers/quarterly_files/faers_ascii_2015q2/ascii/demo';
ALTER TABLE aers_demo_v6 ADD PARTITION (YEAR=2015, QUARTER=3) LOCATION
'/data/aers/quarterly_files/faers_ascii_2015q3/ascii/demo';
ALTER TABLE aers_demo_v6 ADD PARTITION (YEAR=2015, QUARTER=4) LOCATION
'/data/aers/quarterly_files/faers_ascii_2015q4/ascii/demo';

```

```
ALTER TABLE aers_demo_v6 ADD PARTITION (YEAR=2016, QUARTER=1) LOCATION  
'/data/aers/quarterly_files/faers_ascii_2016q1/ascii/demo';  
ALTER TABLE aers_demo_v6 ADD PARTITION (YEAR=2016, QUARTER=2) LOCATION  
'/data/aers/quarterly_files/faers_ascii_2016q2/ascii/demo';  
ALTER TABLE aers_demo_v6 ADD PARTITION (YEAR=2016, QUARTER=3) LOCATION  
'/data/aers/quarterly_files/faers_ascii_2016q3/ascii/demo';  
ALTER TABLE aers_demo_v6 ADD PARTITION (YEAR=2016, QUARTER=4) LOCATION  
'/data/aers/quarterly_files/faers_ascii_2016q4/ascii/demo';  
ALTER TABLE aers_demo_v6 ADD PARTITION (YEAR=2017, QUARTER=1) LOCATION  
'/data/aers/quarterly_files/faers_ascii_2017q1/ascii/demo';
```

Creating a Single Demographic View using subset of columns

```
CREATE DATABASE IF NOT EXISTS  
  aers  
;  
  
USE aers;  
  
DROP VIEW IF EXISTS aers_demo_view  
;  
  
CREATE VIEW  
  aers_demo_view(  
    ISR COMMENT 'case id',  
    EVENT_DT COMMENT 'Event date',  
    AGE COMMENT 'age of patient',  
    AGE_COD COMMENT 'days,months years',  
    SEX COMMENT 'M or F',  
    YEAR,
```

```
    QUARTER
)
COMMENT 'Demographic data required by MedSideFx app'
AS
SELECT
    ISR,
    EVENT_DT,
    AGE,
    AGE_COD,
    GNDR_COD,
    YEAR,
    QUARTER
FROM
    aers_demo_v1
UNION ALL
SELECT
    ISR,
    EVENT_DT,
    AGE,
    AGE_COD,
    GNDR_COD,
    YEAR,
    QUARTER
FROM
    aers_demo_v2
UNION ALL
SELECT
    ISR,
    EVENT_DT,
    AGE,
    AGE_COD,
    GNDR_COD,
    YEAR,
```

```
    QUARTER
FROM
  aers_demo_v3
UNION ALL
SELECT
  ISR,
  EVENT_DT,
  AGE,
  AGE_COD,
  GNDR_COD,
  YEAR,
  QUARTER
FROM
  aers_demo_v4
UNION ALL
SELECT
  primaryid AS ISR,
  EVENT_DT,
  AGE,
  AGE_COD,
  GNDR_COD,
  YEAR,
  QUARTER
FROM
  aers_demo_v5
UNION ALL
SELECT
  primaryid AS ISR,
  EVENT_DT,
  AGE,
  AGE_COD,
  SEX AS GNDR_COD,
  YEAR,
```

```
QUARTER  
FROM  
  aers_demo_v6  
;
```

Creating Drug Tables

```
CREATE DATABASE IF NOT EXISTS  
  aers  
;  
  
USE aers;  
  
DROP TABLE IF EXISTS aers_drug_v1  
;  
CREATE EXTERNAL TABLE aers_drug_v1 (  
  ISR BIGINT,  
  DRUG_SEQ BIGINT,  
  ROLE_COD STRING,  
  DRUGNAME STRING,  
  VAL_VBM INT,  
  ROUTE STRING,  
  DOSE_VBM STRING,  
  DECHAL STRING,  
  RECHAL STRING ) PARTITIONED BY (YEAR INT, QUARTER INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY '$'  
;  
  
DROP TABLE IF EXISTS aers_drug_v2  
;  
  
CREATE EXTERNAL TABLE aers_drug_v2 (  

```

```

ISR BIGINT,
DRUG_SEQ BIGINT,
ROLE_COD STRING,
DRUGNAME STRING,
VAL_VBM INT,
ROUTE STRING,
DOSE_VBM STRING,
DECHAL STRING,
RECHAL STRING,
LOT_NUM STRING,
EXP_DT BIGINT,
NDA_NUM STRING) PARTITIONED BY (YEAR INT, QUARTER INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY '$'
;

DROP TABLE IF EXISTS aers_drug_v3
;
CREATE EXTERNAL TABLE aers_drug_v3 (
primaryid BIGINT,
caseid STRING,
drug_seq BIGINT,
role_cod STRING,
drugname STRING,
val_vbm STRING,
route STRING,
dose_vbm STRING,
cum_dose_chr STRING,
cum_dose_unit STRING,
dechal STRING,
rechal STRING,
lot_nbr STRING,
exp_dt STRING,
nda_num STRING,
dose_amt STRING,

```



```

dose_unit STRING,
dose_form STRING,
dose_freq STRING) PARTITIONED BY (YEAR INT, QUARTER INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY '$'
;

DROP TABLE IF EXISTS aers_drug_v4
;
CREATE EXTERNAL TABLE aers_drug_v4 (
primaryid BIGINT,
caseid STRING,
drug_seq BIGINT,
role_cod STRING,
drugname STRING,
val_vbm STRING,
route STRING,
dose_vbm STRING,
cum_dose_chr STRING,
cum_dose_unit STRING,
dechal STRING,
rechal STRING,
lot_num STRING,
exp_dt STRING,
nda_num STRING,
dose_amt STRING,
dose_unit STRING,
dose_form STRING,
dose_freq STRING) PARTITIONED BY (YEAR INT, QUARTER INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY '$'
;

DROP TABLE IF EXISTS aers_drug_v5
;
CREATE EXTERNAL TABLE aers_drug_v5 (
primaryid  BIGINT,

```

```
caseid  STRING,
drug_seq BIGINT,
role_cod STRING,
drugname STRING,
prod_ai  STRING,
val_vbm  STRING,
route    STRING,
dose_vbm STRING,
cum_dose_chr  STRING,
cum_dose_unit STRING,
dechal    STRING,
rechal    STRING,
lot_num   STRING,
exp_dt    STRING,
nda_num   STRING,
dose_amt  STRING,
dose_unit STRING,
dose_form STRING,
dose_freq STRING) PARTITIONED BY (YEAR INT, QUARTER INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY '$'
;
```

Partitioning Drug Tables

```
USE aers;

ALTER TABLE aers_drug_v1 ADD PARTITION (YEAR=1999, QUARTER=1) LOCATION
'/data/aers/quarterly_files/aers_ascii_1999q1/ascii/drug';
ALTER TABLE aers_drug_v1 ADD PARTITION (YEAR=1999, QUARTER=2) LOCATION
'/data/aers/quarterly_files/aers_ascii_1999q2/ascii/drug';
ALTER TABLE aers_drug_v1 ADD PARTITION (YEAR=1999, QUARTER=3) LOCATION
'/data/aers/quarterly_files/aers_ascii_1999q3/ascii/drug';
ALTER TABLE aers_drug_v1 ADD PARTITION (YEAR=1999, QUARTER=4) LOCATION
'/data/aers/quarterly_files/aers_ascii_1999q4/ascii/drug';
ALTER TABLE aers_drug_v1 ADD PARTITION (YEAR=2000, QUARTER=1) LOCATION
'/data/aers/quarterly_files/aers_ascii_2000q1/ascii/drug';
ALTER TABLE aers_drug_v1 ADD PARTITION (YEAR=2000, QUARTER=2) LOCATION
'/data/aers/quarterly_files/aers_ascii_2000q2/ascii/drug';
ALTER TABLE aers_drug_v1 ADD PARTITION (YEAR=2000, QUARTER=3) LOCATION
'/data/aers/quarterly_files/aers_ascii_2000q3/ascii/drug';
ALTER TABLE aers_drug_v1 ADD PARTITION (YEAR=2000, QUARTER=4) LOCATION
'/data/aers/quarterly_files/aers_ascii_2000q4/ascii/drug';
ALTER TABLE aers_drug_v1 ADD PARTITION (YEAR=2001, QUARTER=1) LOCATION
'/data/aers/quarterly_files/aers_ascii_2001q1/ascii/drug';
ALTER TABLE aers_drug_v1 ADD PARTITION (YEAR=2001, QUARTER=2) LOCATION
'/data/aers/quarterly_files/aers_ascii_2001q2/ascii/drug';
ALTER TABLE aers_drug_v1 ADD PARTITION (YEAR=2001, QUARTER=3) LOCATION
'/data/aers/quarterly_files/aers_ascii_2001q3/ascii/drug';
ALTER TABLE aers_drug_v1 ADD PARTITION (YEAR=2001, QUARTER=4) LOCATION
'/data/aers/quarterly_files/aers_ascii_2001q4/ascii/drug';
ALTER TABLE aers_drug_v1 ADD PARTITION (YEAR=2002, QUARTER=1) LOCATION
'/data/aers/quarterly_files/aers_ascii_2002q1/ascii/drug';

ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2002, QUARTER=2) LOCATION
```

```

'/data/aers/quarterly_files/aers_ascii_2002q2/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2002, QUARTER=3) LOCATION
'/data/aers/quarterly_files/aers_ascii_2002q3/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2002, QUARTER=4) LOCATION
'/data/aers/quarterly_files/aers_ascii_2002q4/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2003, QUARTER=1) LOCATION
'/data/aers/quarterly_files/aers_ascii_2003q1/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2003, QUARTER=2) LOCATION
'/data/aers/quarterly_files/aers_ascii_2003q2/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2003, QUARTER=3) LOCATION
'/data/aers/quarterly_files/aers_ascii_2003q3/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2003, QUARTER=4) LOCATION
'/data/aers/quarterly_files/aers_ascii_2003q4/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2004, QUARTER=1) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2004q1/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2004, QUARTER=2) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2004q2/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2004, QUARTER=3) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2004q3/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2004, QUARTER=4) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2004q4/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2005, QUARTER=1) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2005Q1/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2005, QUARTER=2) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2005Q2/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2005, QUARTER=3) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2005Q3/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2005, QUARTER=4) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2005Q4/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2006, QUARTER=1) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2006Q1/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2006, QUARTER=2) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2006Q2/ascii/drug';

```

```

ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2006, QUARTER=3) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2006Q3/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2006, QUARTER=4) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2006Q4/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2007, QUARTER=1) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2007Q1/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2007, QUARTER=2) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2007Q2/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2007, QUARTER=3) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2007Q3/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2007, QUARTER=4) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2007Q4/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2008, QUARTER=1) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2008Q1/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2008, QUARTER=2) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2008Q2/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2008, QUARTER=3) LOCATION
'/data/aers/quarterly_files/aers_ascii_2008q3/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2008, QUARTER=4) LOCATION
'/data/aers/quarterly_files/aers_ascii_2008q4/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2009, QUARTER=1) LOCATION
'/data/aers/quarterly_files/aers_ascii_2009q1/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2009, QUARTER=2) LOCATION
'/data/aers/quarterly_files/aers_ascii_2009q2/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2009, QUARTER=3) LOCATION
'/data/aers/quarterly_files/aers_ascii_2009q3/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2009, QUARTER=4) LOCATION
'/data/aers/quarterly_files/aers_ascii_2009q4/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2010, QUARTER=1) LOCATION
'/data/aers/quarterly_files/aers_ascii_2010q1/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2010, QUARTER=2) LOCATION
'/data/aers/quarterly_files/aers_ascii_2010q2/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2010, QUARTER=3) LOCATION

```

```

'/data/aers/quarterly_files/aers_ascii_2010q3/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2010, QUARTER=4) LOCATION
'/data/aers/quarterly_files/aers_ascii_2010q4/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2011, QUARTER=1) LOCATION
'/data/aers/quarterly_files/aers_ascii_2011q1/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2011, QUARTER=2) LOCATION
'/data/aers/quarterly_files/aers_ascii_2011q2/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2011, QUARTER=3) LOCATION
'/data/aers/quarterly_files/aers_ascii_2011q3/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2011, QUARTER=4) LOCATION
'/data/aers/quarterly_files/aers_ascii_2011q4/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2012, QUARTER=1) LOCATION
'/data/aers/quarterly_files/aers_ascii_2012q1/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2012, QUARTER=2) LOCATION
'/data/aers/quarterly_files/aers_ascii_2012q2/ascii/drug';
ALTER TABLE aers_drug_v2 ADD PARTITION (YEAR=2012, QUARTER=3) LOCATION
'/data/aers/quarterly_files/aers_ascii_2012q3/ascii/drug';

ALTER TABLE aers_drug_v3 ADD PARTITION (YEAR=2012, QUARTER=4) LOCATION
'/data/aers/quarterly_files/faers_ascii_2012q4/ascii/drug';
ALTER TABLE aers_drug_v3 ADD PARTITION (YEAR=2013, QUARTER=1) LOCATION
'/data/aers/quarterly_files/faers_ascii_2013q1/ascii/drug';
ALTER TABLE aers_drug_v3 ADD PARTITION (YEAR=2013, QUARTER=2) LOCATION
'/data/aers/quarterly_files/faers_ascii_2013q2/ascii/drug';

ALTER TABLE aers_drug_v4 ADD PARTITION (YEAR=2013, QUARTER=3) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2013q3/ascii/drug';
ALTER TABLE aers_drug_v4 ADD PARTITION (YEAR=2013, QUARTER=4) LOCATION
'/data/aers/quarterly_files/FAERS_ASCII_2013Q4/ascii/drug';
ALTER TABLE aers_drug_v4 ADD PARTITION (YEAR=2014, QUARTER=1) LOCATION
'/data/aers/quarterly_files/faers_ascii_2014q1/ascii/drug';
ALTER TABLE aers_drug_v4 ADD PARTITION (YEAR=2014, QUARTER=2) LOCATION
'/data/aers/quarterly_files/faers_ascii_2014q2/ascii/drug';

```

```
ALTER TABLE aers_drug_v5 ADD PARTITION (YEAR=2014, QUARTER=3) LOCATION  
'/data/aers/quarterly_files/FAERS_ASCII_2014Q3/ascii/drug';  
ALTER TABLE aers_drug_v5 ADD PARTITION (YEAR=2014, QUARTER=4) LOCATION  
'/data/aers/quarterly_files/FAERS_ASCII_2014Q4/ascii/drug';  
ALTER TABLE aers_drug_v5 ADD PARTITION (YEAR=2015, QUARTER=1) LOCATION  
'/data/aers/quarterly_files/FAERS_ASCII_2015Q1/ascii/drug';  
ALTER TABLE aers_drug_v5 ADD PARTITION (YEAR=2015, QUARTER=2) LOCATION  
'/data/aers/quarterly_files/faers_ascii_2015q2/ascii/drug';  
ALTER TABLE aers_drug_v5 ADD PARTITION (YEAR=2015, QUARTER=3) LOCATION  
'/data/aers/quarterly_files/faers_ascii_2015q3/ascii/drug';  
ALTER TABLE aers_drug_v5 ADD PARTITION (YEAR=2015, QUARTER=4) LOCATION  
'/data/aers/quarterly_files/faers_ascii_2015q4/ascii/drug';  
ALTER TABLE aers_drug_v5 ADD PARTITION (YEAR=2016, QUARTER=1) LOCATION  
'/data/aers/quarterly_files/faers_ascii_2016q1/ascii/drug';  
ALTER TABLE aers_drug_v5 ADD PARTITION (YEAR=2016, QUARTER=2) LOCATION  
'/data/aers/quarterly_files/faers_ascii_2016q2/ascii/drug';  
ALTER TABLE aers_drug_v5 ADD PARTITION (YEAR=2016, QUARTER=3) LOCATION  
'/data/aers/quarterly_files/faers_ascii_2016q3/ascii/drug';  
ALTER TABLE aers_drug_v5 ADD PARTITION (YEAR=2016, QUARTER=4) LOCATION  
'/data/aers/quarterly_files/faers_ascii_2016q4/ascii/drug';  
ALTER TABLE aers_drug_v5 ADD PARTITION (YEAR=2017, QUARTER=1) LOCATION  
'/data/aers/quarterly_files/faers_ascii_2017q1/ascii/drug';
```

Creating a Single Drug View using subset of columns

```
CREATE DATABASE IF NOT EXISTS  
aers
```

```
;

USE aers;

DROP VIEW IF EXISTS aers_drug_view
;

CREATE VIEW
  aers_drug_view(
    ISR COMMENT 'case id',
    DRUG_SEQ COMMENT 'Drug Seq',
    DRUGNAME COMMENT 'Name of Drug',
    YEAR,
    QUARTER
  )
  COMMENT 'Drug data required by MedSideFx app'
  AS
SELECT
  ISR,
  DRUG_SEQ,
  DRUGNAME,
  YEAR,
  QUARTER
FROM
  aers_drug_v1
UNION ALL
SELECT
  ISR,
  DRUG_SEQ,
  DRUGNAME,
  YEAR,
  QUARTER
FROM
```



```
aers_drug_v2
UNION ALL
SELECT
  primaryid as ISR,
  drug_seq,
  drugname,
  YEAR,
  QUARTER
FROM
  aers_drug_v3
UNION ALL
SELECT
  primaryid as ISR,
  drug_seq,
  drugname,
  YEAR,
  QUARTER
FROM
  aers_drug_v4
UNION ALL
SELECT
  primaryid as ISR,
  drug_seq,
  drugname,
  YEAR,
  QUARTER
FROM
  aers_drug_v5
;
```

Creating Reaction Tables

```
CREATE DATABASE IF NOT EXISTS
  aers
;

USE aers;

DROP TABLE IF EXISTS aers_reac_v1
;
CREATE EXTERNAL TABLE aers_reac_v1 (
  ISR BIGINT,
  PT STRING)
PARTITIONED BY (YEAR INT, QUARTER INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY '$'
;

DROP TABLE IF EXISTS aers_reac_v2
;
CREATE EXTERNAL TABLE aers_reac_v2 (
  primaryid BIGINT,
  caseid  BIGINT,
  pt STRING)
PARTITIONED BY (YEAR INT, QUARTER INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY '$'
;

DROP TABLE IF EXISTS aers_reac_v3
;
CREATE EXTERNAL TABLE aers_reac_v3 (
  primaryid BIGINT,
  caseid  BIGINT,
  pt STRING,
  drug_rec_act STRING)
```

```
PARTITIONED BY (YEAR INT, QUARTER INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY '$'  
;
```

Partitioning Reaction Tables

```
USE aers;  
  
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=1999, QUARTER=1) LOCATION  
'/data/aers/quarterly_files/aers_ascii_1999q1/ascii/reac';  
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=1999, QUARTER=2) LOCATION  
'/data/aers/quarterly_files/aers_ascii_1999q2/ascii/reac';  
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=1999, QUARTER=3) LOCATION  
'/data/aers/quarterly_files/aers_ascii_1999q3/ascii/reac';  
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=1999, QUARTER=4) LOCATION  
'/data/aers/quarterly_files/aers_ascii_1999q4/ascii/reac';  
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2000, QUARTER=1) LOCATION  
'/data/aers/quarterly_files/aers_ascii_2000q1/ascii/reac';  
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2000, QUARTER=2) LOCATION  
'/data/aers/quarterly_files/aers_ascii_2000q2/ascii/reac';  
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2000, QUARTER=3) LOCATION  
'/data/aers/quarterly_files/aers_ascii_2000q3/ascii/reac';  
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2000, QUARTER=4) LOCATION  
'/data/aers/quarterly_files/aers_ascii_2000q4/ascii/reac';  
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2001, QUARTER=1) LOCATION  
'/data/aers/quarterly_files/aers_ascii_2001q1/ascii/reac';  
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2001, QUARTER=2) LOCATION  
'/data/aers/quarterly_files/aers_ascii_2001q2/ascii/reac';  
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2001, QUARTER=3) LOCATION  
'/data/aers/quarterly_files/aers_ascii_2001q3/ascii/reac';
```

```

ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2001, QUARTER=4) LOCATION
'/data/aers/quarterly_files/aers_ascii_2001q4/ascii/reac';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2002, QUARTER=1) LOCATION
'/data/aers/quarterly_files/aers_ascii_2002q1/ascii/reac';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2002, QUARTER=2) LOCATION
'/data/aers/quarterly_files/aers_ascii_2002q2/ascii/reac';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2002, QUARTER=3) LOCATION
'/data/aers/quarterly_files/aers_ascii_2002q3/ascii/reac';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2002, QUARTER=4) LOCATION
'/data/aers/quarterly_files/aers_ascii_2002q4/ascii/reac';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2003, QUARTER=1) LOCATION
'/data/aers/quarterly_files/aers_ascii_2003q1/ascii/reac';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2003, QUARTER=2) LOCATION
'/data/aers/quarterly_files/aers_ascii_2003q2/ascii/reac';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2003, QUARTER=3) LOCATION
'/data/aers/quarterly_files/aers_ascii_2003q3/ascii/reac';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2003, QUARTER=4) LOCATION
'/data/aers/quarterly_files/aers_ascii_2003q4/ascii/reac';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2004, QUARTER=1) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2004q1/ascii/reac';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2004, QUARTER=2) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2004q2/ascii/reac';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2004, QUARTER=3) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2004q3/ascii/reac';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2004, QUARTER=4) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2004q4/ascii/reac';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2005, QUARTER=1) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2005Q1/ascii/reac';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2005, QUARTER=2) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2005Q2/ascii/reac';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2005, QUARTER=3) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2005Q3/ascii/reac';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2005, QUARTER=4) LOCATION

```

```

'/data/aers/quarterly_files/AERS_ASCII_2005Q4/ascii/react';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2006, QUARTER=1) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2006Q1/ascii/react';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2006, QUARTER=2) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2006Q2/ascii/react';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2006, QUARTER=3) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2006Q3/ascii/react';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2006, QUARTER=4) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2006Q4/ascii/react';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2007, QUARTER=1) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2007Q1/ascii/react';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2007, QUARTER=2) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2007Q2/ascii/react';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2007, QUARTER=3) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2007Q3/ascii/react';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2007, QUARTER=4) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2007Q4/ascii/react';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2008, QUARTER=1) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2008Q1/ascii/react';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2008, QUARTER=2) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2008Q2/ascii/react';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2008, QUARTER=3) LOCATION
'/data/aers/quarterly_files/aers_ascii_2008q3/ascii/react';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2008, QUARTER=4) LOCATION
'/data/aers/quarterly_files/aers_ascii_2008q4/ascii/react';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2009, QUARTER=1) LOCATION
'/data/aers/quarterly_files/aers_ascii_2009q1/ascii/react';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2009, QUARTER=2) LOCATION
'/data/aers/quarterly_files/aers_ascii_2009q2/ascii/react';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2009, QUARTER=3) LOCATION
'/data/aers/quarterly_files/aers_ascii_2009q3/ascii/react';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2009, QUARTER=4) LOCATION
'/data/aers/quarterly_files/aers_ascii_2009q4/ascii/react';

```

```

ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2010, QUARTER=1) LOCATION
'/data/aers/quarterly_files/aers_ascii_2010q1/ascii/reac';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2010, QUARTER=2) LOCATION
'/data/aers/quarterly_files/aers_ascii_2010q2/ascii/reac';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2010, QUARTER=3) LOCATION
'/data/aers/quarterly_files/aers_ascii_2010q3/ascii/reac';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2010, QUARTER=4) LOCATION
'/data/aers/quarterly_files/aers_ascii_2010q4/ascii/reac';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2011, QUARTER=1) LOCATION
'/data/aers/quarterly_files/aers_ascii_2011q1/ascii/reac';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2011, QUARTER=2) LOCATION
'/data/aers/quarterly_files/aers_ascii_2011q2/ascii/reac';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2011, QUARTER=3) LOCATION
'/data/aers/quarterly_files/aers_ascii_2011q3/ascii/reac';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2011, QUARTER=4) LOCATION
'/data/aers/quarterly_files/aers_ascii_2011q4/ascii/reac';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2012, QUARTER=1) LOCATION
'/data/aers/quarterly_files/aers_ascii_2012q1/ascii/reac';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2012, QUARTER=2) LOCATION
'/data/aers/quarterly_files/aers_ascii_2012q2/ascii/reac';
ALTER TABLE aers_reac_v1 ADD PARTITION (YEAR=2012, QUARTER=3) LOCATION
'/data/aers/quarterly_files/aers_ascii_2012q3/ascii/reac';

ALTER TABLE aers_reac_v2 ADD PARTITION (YEAR=2012, QUARTER=4) LOCATION
'/data/aers/quarterly_files/faers_ascii_2012q4/ascii/reac';
ALTER TABLE aers_reac_v2 ADD PARTITION (YEAR=2013, QUARTER=1) LOCATION
'/data/aers/quarterly_files/faers_ascii_2013q1/ascii/reac';
ALTER TABLE aers_reac_v2 ADD PARTITION (YEAR=2013, QUARTER=2) LOCATION
'/data/aers/quarterly_files/faers_ascii_2013q2/ascii/reac';
ALTER TABLE aers_reac_v2 ADD PARTITION (YEAR=2013, QUARTER=3) LOCATION
'/data/aers/quarterly_files/AERS_ASCII_2013q3/ascii/reac';
ALTER TABLE aers_reac_v2 ADD PARTITION (YEAR=2013, QUARTER=4) LOCATION
'/data/aers/quarterly_files/FAERS_ASCII_2013Q4/ascii/reac';

```

```

ALTER TABLE aers_reac_v2 ADD PARTITION (YEAR=2014, QUARTER=1) LOCATION
'/data/aers/quarterly_files/faers_ascii_2014q1/ascii/reac';
ALTER TABLE aers_reac_v2 ADD PARTITION (YEAR=2014, QUARTER=2) LOCATION
'/data/aers/quarterly_files/faers_ascii_2014q2/ascii/reac';

ALTER TABLE aers_reac_v3 ADD PARTITION (YEAR=2014, QUARTER=3) LOCATION
'/data/aers/quarterly_files/FAERS_ASCII_2014Q3/ascii/reac';
ALTER TABLE aers_reac_v3 ADD PARTITION (YEAR=2014, QUARTER=4) LOCATION
'/data/aers/quarterly_files/FAERS_ASCII_2014Q4/ascii/reac';
ALTER TABLE aers_reac_v3 ADD PARTITION (YEAR=2015, QUARTER=1) LOCATION
'/data/aers/quarterly_files/FAERS_ASCII_2015Q1/ascii/reac';
ALTER TABLE aers_reac_v3 ADD PARTITION (YEAR=2015, QUARTER=2) LOCATION
'/data/aers/quarterly_files/faers_ascii_2015q2/ascii/reac';
ALTER TABLE aers_reac_v3 ADD PARTITION (YEAR=2015, QUARTER=3) LOCATION
'/data/aers/quarterly_files/faers_ascii_2015q3/ascii/reac';
ALTER TABLE aers_reac_v3 ADD PARTITION (YEAR=2015, QUARTER=4) LOCATION
'/data/aers/quarterly_files/faers_ascii_2015q4/ascii/reac';
ALTER TABLE aers_reac_v3 ADD PARTITION (YEAR=2016, QUARTER=1) LOCATION
'/data/aers/quarterly_files/faers_ascii_2016q1/ascii/reac';
ALTER TABLE aers_reac_v3 ADD PARTITION (YEAR=2016, QUARTER=2) LOCATION
'/data/aers/quarterly_files/faers_ascii_2016q2/ascii/reac';
ALTER TABLE aers_reac_v3 ADD PARTITION (YEAR=2016, QUARTER=3) LOCATION
'/data/aers/quarterly_files/faers_ascii_2016q3/ascii/reac';
ALTER TABLE aers_reac_v3 ADD PARTITION (YEAR=2016, QUARTER=4) LOCATION
'/data/aers/quarterly_files/faers_ascii_2016q4/ascii/reac';
ALTER TABLE aers_reac_v3 ADD PARTITION (YEAR=2017, QUARTER=1) LOCATION
'/data/aers/quarterly_files/faers_ascii_2017q1/ascii/reac';

```

Creating a Single Reaction View using subset of columns

```
CREATE DATABASE IF NOT EXISTS
  aers
;

USE aers;

DROP VIEW IF EXISTS aers_reac_view
;

CREATE VIEW
  aers_reac_view(
    ISR COMMENT 'case id',
    PT COMMENT 'Reaction Text',
    YEAR,
    QUARTER
  )
  COMMENT 'Reaction data required by MedSideFx app'
  AS
SELECT
  ISR,
  PT,
  YEAR,
  QUARTER
FROM
  aers_reac_v1
UNION ALL
SELECT
  primaryid as ISR,
  pt,
  YEAR,
```



```
    QUARTER  
FROM  
  aers_reac_v2  
UNION ALL  
SELECT  
  primaryid as ISR,  
  pt,  
  YEAR,  
  QUARTER  
FROM  
  aers_reac_v3  
;
```

Joining Demographic, Drug and Reaction tables

We create a joined table using the following script (this uses the column subset views

aers_demo_view

aers_drug_view

aers_reac_view

```
set hive.execution.engine=mr;
USE
  aers
;
DROP TABLE IF EXISTS
  demo_drug_reac_combo
;
CREATE TABLE IF NOT EXISTS
  demo_drug_reac_combo
AS
WITH
  ddrug
AS
(SELECT DISTINCT
  *
FROM
  aers.aers_drug_view
)
SELECT DISTINCT
  drug.isr,
  lower(drug.drugname) as drugname,
  lower(reac.pt) as pt,
  drug.year,
  demo.age,
  demo.age_cod,
  demo.age_norm,
```

```

demo.age_group
FROM
  (SELECT DISTINCT
    d.isr,
      d.year,
    lower(d.drugname) as drugname
  from
    (SELECT
      isr,
      count(distinct drug_seq) as ginti
    FROM
      ddrug ff
    GROUP BY
      isr
    HAVING
      ginti = 1
    ) g
  JOIN
    ddrug d
  ON
    g.isr = d.isr
  ) drug
JOIN
  (SELECT
    isr,
      lower(pt) as pt
  FROM
    aers.aers_reac_view
  ) reac
ON
  reac.isr = drug.isr
LEFT OUTER JOIN
(select

```

```

isr,
age,
age_cod,
age_norm,
CASE
  WHEN age_norm >= 0 and age_norm < 5 THEN '0-4'
  WHEN age_norm >= 5 and age_norm < 10 THEN '5-9'
  WHEN age_norm >= 10 and age_norm < 15 THEN '10-14'
  WHEN age_norm >= 15 and age_norm < 20 THEN '15-19'
  WHEN age_norm >= 20 and age_norm < 25 THEN '20-24'
  WHEN age_norm >= 25 and age_norm < 30 THEN '25-29'
  WHEN age_norm >= 30 and age_norm < 35 THEN '30-34'
  WHEN age_norm >= 35 and age_norm < 40 THEN '35-39'
  WHEN age_norm >= 40 and age_norm < 45 THEN '40-44'
  WHEN age_norm >= 45 and age_norm < 50 THEN '45-49'
  WHEN age_norm >= 50 and age_norm < 55 THEN '50-54'
  WHEN age_norm >= 55 and age_norm < 60 THEN '55-59'
  WHEN age_norm >= 60 and age_norm < 65 THEN '60-64'
  WHEN age_norm >= 65 and age_norm < 70 THEN '65-69'
  WHEN age_norm >= 70 and age_norm < 75 THEN '70-74'
  WHEN age_norm >= 75 and age_norm < 80 THEN '75-79'
  WHEN age_norm >= 80 and age_norm < 85 THEN '80-84'
  WHEN age_norm >= 85 and age_norm < 90 THEN '85-89'
  WHEN age_norm >= 90 and age_norm < 95 THEN '90-94'
  WHEN age_norm >= 95 and age_norm < 100 THEN '95-99'
  WHEN age_norm >= 100 and age_norm < 105 THEN '100-104'
  WHEN age_norm >= 105 and age_norm < 110 THEN '105-109'
  WHEN age_norm >= 110 and age_norm < 115 THEN '110-114'
  WHEN age_norm >= 115 and age_norm < 120 THEN '115-119'
  WHEN age_norm >= 120 and age_norm < 125 THEN '120-124'
END
  as age_group
from

```

```

(select
  isr,
  age,
  age_cod,
  CASE
    WHEN age_cod = 'YR' THEN age * 1
    WHEN age_cod = 'MON' THEN age / 12
    WHEN age_cod = 'HR' THEN age / 8760
    WHEN age_cod = 'DY' THEN age / 365
    WHEN age_cod = 'SEC' THEN age / 31536000
    WHEN age_cod = 'WK' THEN age / 52
    WHEN age_cod = 'DEC' THEN age * 10
    WHEN age_cod = 'MIN' THEN age * 525600
  END
  as age_norm
from
  aers.aers_demo_view
where
  age IS NOT NULL
AND
  age_cod IS NOT NULL
AND
  age_cod <> "
)
aa
) demo
ON
  reac.isr = drug.isr
AND
  demo.isr = drug.isr
AND
  demo.isr = reac.isr

```

Creating detail views of Demographic, Drug, Reaction(all column variations included)

Demographic Merged View (all column variations included)

```
USE aers;

DROP VIEW IF EXISTS aers_demo_merged
;
CREATE VIEW aers_demo_merged (
  isr,
  caseid,
  caseversion,
  i_f_code,
  FOLL_SEQ,
  IMAGE,
  event_dt,
  mfr_dt,
  init_fda_dt,
  fda_dt,
  rept_cod,
  auth_num,
  mfr_num,
  mfr_sndr,
  lit_ref,
  age,
  age_cod,
  age_grp,
  sex,
  BEST_ISR,
  e_sub,
```

```
wt,  
wt_cod,  
rept_dt,  
to_mfr,  
occp_cod,  
DEATH_DT,  
CONFID,  
reporter_country,  
occr_country,  
year,  
quarter  
)  
AS  
-- demoHeaderVersion1  
SELECT  
ISR,  
CASE_ as caseid,  
-1 as caseversion,  
I_F_COD,  
FOLL_SEQ,  
IMAGE,  
EVENT_DT,  
MFR_DT,  
-1 as init_fda_dt,  
FDA_DT,  
REPT_COD,  
" as auth_num,  
MFR_NUM,  
MFR_SNDR,  
" as lit_ref,  
AGE,  
AGE_COD,  
" as age_grp,
```

```

GNDR_COD,
" as BEST_ISR,
" as E_SUB,
-1.0 as WT,
" as WT_COD,
-1 as REPT_DT,
" as to_mfr,
" as occp_cod,
-1 as DEATH_DT,
" as CONFID,
" as REPORTER_COUNTRY,
" as occr_country,
year,
quarter
FROM
aers.aers_demo_v1

UNION ALL

-- demoHeaderVersion2
SELECT
ISR,
CASE_ as caseid,
-1 as caseversion,
I_F_COD,
FOLL_SEQ,
IMAGE,
EVENT_DT,
MFR_DT,
-1 as init_fda_dt,
FDA_DT,
REPT_COD,
" as auth_num,

```



```

MFR_NUM,
MFR_SNDR,
" as lit_ref,
AGE,
AGE_COD,
" as age_grp,
GNDR_COD,
BEST_ISR,
E_SUB,
WT,
WT_COD,
REPT_DT,
to_mfr,
OCCP_COD,
DEATH_DT,
CONFID,
" as REPORTER_COUNTRY,
" as occr_country,
year,
quarter
from
    aers.aers_demo_v2

UNION ALL

-- demoHeaderVersion3
SELECT
    ISR,
    CASE_ as caseid,
    -1 as caseversion,
    I_F_COD,
    FOLL_SEQ,
    IMAGE,

```

```
EVENT_DT,  
MFR_DT,  
-1 as init_fda_dt,  
FDA_DT,  
REPT_COD,  
" as auth_num,  
MFR_NUM,  
MFR_SNDR,  
" as lit_ref,  
AGE,  
AGE_COD,  
" as age_grp,  
GNDR_COD,  
" as BEST_ISR,  
E_SUB,  
WT,  
WT_COD,  
REPT_DT,  
to_mfr,  
OCCP_COD,  
DEATH_DT,  
CONFID,  
" as REPORTER_COUNTRY,  
" as occr_country,  
year,  
quarter  
from  
aers.aers_demo_v3  
  
UNION ALL  
  
-- demoHeaderVersion4  
SELECT
```

```
ISR,  
CASE_ as caseid,  
-1 as caseversion,  
I_F_COD,  
FOLL_SEQ,  
IMAGE,  
EVENT_DT,  
MFR_DT,  
-1 as init_fda_dt,  
FDA_DT,  
REPT_COD,  
" as auth_num,  
MFR_NUM,  
MFR_SNDR,  
" as lit_ref,  
AGE,  
AGE_COD,  
" as age_grp,  
GNDR_COD,  
-1 as BEST_ISR,  
E_SUB,  
WT,  
WT_COD,  
REPT_DT,  
to_mfr,  
OCCP_COD,  
DEATH_DT,  
CONFID,  
REPORTER_COUNTRY,  
" as occr_country,  
year,  
quarter  
FROM
```

```
aers.aers_demo_v4
```

```
UNION ALL
```

```
-- demoHeaderVersion5
```

```
SELECT
```

```
primaryid as isr,  
caseid,  
caseversion,  
i_f_code as I_F_COD,  
-1 as FOLL_SEQ,  
" as IMAGE,  
event_dt,  
mfr_dt,  
init_fda_dt,  
fda_dt,  
rept_cod,  
" as auth_num,  
mfr_num,  
mfr_sndr,  
" as lit_ref,  
age,  
age_cod,  
" as age_grp,  
gndr_cod,  
-1 as BEST_ISR,  
e_sub,  
wt,  
wt_cod,  
rept_dt,  
to_mfr,  
occp_cod,  
-1 as DEATH_DT,
```

```
" as CONFID,  
reporter_country,  
occr_country,  
year,  
quarter  
FROM  
aers.aers_demo_v5  
  
UNION ALL  
  
-- demoHeaderVersion6  
SELECT  
primaryid as isr,  
caseid,  
caseversion,  
i_f_code as I_F_COD,  
-1 as FOLL_SEQ,  
" as IMAGE,  
event_dt,  
mfr_dt,  
init_fda_dt,  
fda_dt,  
rept_cod,  
auth_num,  
mfr_num,  
mfr_sndr,  
lit_ref,  
age,  
age_cod,  
age_grp,  
sex as gndr_cod,  
-1 as BEST_ISR,  
e_sub,
```

```
wt,  
wt_cod,  
rept_dt,  
to_mfr,  
occp_cod,  
-1 as DEATH_DT,  
" as CONFID,  
reporter_country,  
occr_country,  
year,  
quarter  
FROM  
aers.aers_demo_v6
```

Drug Merged View (all column variations included)

```
USE aers;  
  
DROP VIEW IF EXISTS aers_drug_merged  
;  
CREATE VIEW aers_drug_merged (  
  isr,  
  caseid,  
  drug_seq,  
  role_cod,  
  drugname,  
  prod_ai,  
  val_vbm,  
  route,  
  dose_vbm,
```

```
cum_dose_chr,  
cum_dose_unit,  
dechal,  
rechal,  
lot_num,  
exp_dt,  
nda_num,  
dose_amt,  
dose_unit,  
dose_form,  
dose_freq  
)  
AS  
-- drugHeaderVersion1  
SELECT  
isr,  
" as caseid,  
drug_seq,  
role_cod,  
drugname,  
" as prod_ai,  
val_vbm,  
route,  
dose_vbm,  
" as cum_dose_chr,  
" as cum_dose_unit,  
dechal,  
rechal,  
" as lot_num,  
" as exp_dt,  
" as nda_num,  
" as dose_amt,  
" as dose_unit,
```

```
" as dose_form,
" as dose_freq
FROM
  aers.aers_drug_v1

UNION ALL

-- drugHeaderVersion2
SELECT
  isr,
  " as caseid,
  drug_seq,
  role_cod,
  drugname,
  " as prod_ai,
  val_vbm,
  route,
  dose_vbm,
  " as cum_dose_chr,
  " as cum_dose_unit,
  dechal,
  rechal,
  lot_num,
  exp_dt,
  nda_num,
  " as dose_amt,
  " as dose_unit,
  " as dose_form,
  " as dose_freq
FROM
  aers.aers_drug_v2

UNION ALL
```



```
-- drugHeaderVersion3
```

```
SELECT
```

```
  primaryid as isr,  
  caseid,  
  drug_seq,  
  role_cod,  
  drugname,  
  " as prod_ai,  
  val_vbm,  
  route,  
  dose_vbm,  
  cum_dose_chr,  
  cum_dose_unit,  
  dechal,  
  rechal,  
  lot_nbr as lot_num,  
  exp_dt,  
  nda_num,  
  dose_amt,  
  dose_unit,  
  dose_form,  
  dose_freq
```

```
FROM
```

```
  aers.aers_drug_v3
```

```
UNION ALL
```

```
-- drugHeaderVersion4
```

```
SELECT
```

```
  primaryid as isr,  
  caseid,  
  drug_seq,
```

```
role_cod,  
drugname,  
" as prod_ai,  
val_vbm,  
route,  
dose_vbm,  
cum_dose_chr,  
cum_dose_unit,  
dechal,  
rechal,  
lot_num,  
exp_dt,  
nda_num,  
dose_amt,  
dose_unit,  
dose_form,  
dose_freq  
FROM  
aers.aers_drug_v4  
  
UNION ALL  
  
-- drugHeaderVersion5  
SELECT  
primaryid as isr,  
caseid,  
drug_seq,  
role_cod,  
drugname,  
prod_ai,  
val_vbm,  
route,  
dose_vbm,
```

```
cum_dose_chr,  
cum_dose_unit,  
dechal,  
rechal,  
lot_num,  
exp_dt,  
nda_num,  
dose_amt,  
dose_unit,  
dose_form,  
dose_freq  
FROM  
aers.aers_drug_v5
```

Reac Merged View (all column variations included)

```
USE aers;  
  
DROP VIEW IF EXISTS aers_reac_merged  
;  
CREATE VIEW aers_reac_merged (  
  isr,  
  caseid,  
  pt,  
  drug_rec_act  
)  
  
AS  
  
-- reacHeaderVersion1
```

```
SELECT
  isr,
  -1 as caseid,
  pt,
  " as drug_rec_act
FROM
  aers.aers_reac_v1

UNION ALL

-- reacHeaderVersion2
SELECT
  primaryid as isr,
  caseid,
  pt,
  " as drug_rec_act
FROM
  aers.aers_reac_v2

UNION ALL

-- reacHeaderVersion3
SELECT
  primaryid as isr,
  caseid,
  pt,
  drug_rec_act
FROM
  aers.aers_reac_v3
```

Now we are ready for some analysis

AgeGroup Query

This query finds the top 25 Age Groups affected by drugname specified in the query

```
select
  age_group as lyrica_age_groups,
  count (distinct isr) as age_group_counts
from
  aers.demo_drug_reac_combo
where
  lower(drugname) like '%lyrica%'
and
  age_group is not null
group by
  age_group
order by
  age_group_counts desc limit 25
```

Running this query in Hive displays the following results (remember results may vary depending on how many quarters of data you have loaded into HDFS and configured as partitions in Hive). The first column from the left is age group and the second column is age group counts

lyrica_age_groups	age_group_counts
55-59	1277
60-64	1239
50-54	1204
65-69	978
70-74	963
45-49	914
75-79	768
80-84	688
40-44	677
35-39	445
85-89	380
30-34	298
25-29	234
90-94	167
20-24	105
15-19	86
95-99	40
10-14	12
0-4	11
5-9	5
100-104	4

Reactions Query

This query finds the top 25 Reactions for a drugname specified in the query

```

select
  pt as lyrica_symptoms,
  count (distinct isr) as count_reactions
from
  aers.demo_drug_reac_combo
where
  lower(drugname) like '%lyrica%'
group by
  pt
order by
  count_reactions desc limit 25

```

Running this query in Hive displays the following results (remember results may vary depending on how many quarters of data you have loaded into HDFS and configured as partitions in Hive). The first column from the left is Reaction description and the second column is Reaction description counts

lyrica_symptoms	count_reactions
pain	3930
drug ineffective	2969
weight increased	2157
dizziness	1848
malaise	1515
somnolence	1371
feeling abnormal	1355
pain in extremity	1138
vision blurred	1089
oedema peripheral	968

insomnia	897	
headache	765	
fall	750	
nausea	740	
gait disturbance	692	
death	690	
fatigue	641	
withdrawal syndrome	595	
drug withdrawal syndrome	594	
depression	554	
dyspnoea	526	
tremor	516	
memory impairment	508	
suicidal ideation	504	
confusional state	503	
+-----+	+-----+	+-----+

Side effects reported by Year Query

This query finds the top 25 Years where the side effects were reported for a drugname specified in the query

select
year ,
count (distinct isr) as count_years
from
aers.demo_drug_reac_combo
where
lower(drugname) like '%lyrica%'
group by


```
year
order by
year desc limit 25
```

Running this query in Hive displays the following results (remember results may vary depending on how many quarters of data you have loaded into HDFS and configured as partitions in Hive). The first column from the left is Year and the second column is Side effects counts reported in that year

year	count_years
2017	1101
2016	4394
2015	4455
2014	5055
2013	4243
2012	4744
2011	1928
2010	6068
2009	779
2008	844
2007	737
2006	2349
2005	134

Find out which Drugs have a specific side effect

Say we want to find out the top 25 drugs where the reaction had the term “hepatic”

```
select
  drugname as drug_with_hepatic_side_effect,
  count(*) as counts_drugs
from
  aers.demo_drug_reac_combo
where
  lower(pt) like '%hepatic%'
group by
  drugname
order by
  counts_drugs desc
limit 25
```

Running this query in Hive displays the following results (remember results may vary depending on how many quarters of data you have loaded into HDFS and configured as partitions in Hive). The first column from the left is Drugname and the second column is Reaction counts for the term “Hepatic” reported

drug_with_hepatic_side_effect	counts_drugs
enbrel	1309
baycol	1257
humira	984
nexavar	967
rezulin	734
cymbalta	660
avonex	619

	tysabri		600	
	lipitor		547	
	vioxx		454	
	rebif		441	
	acetaminophen		407	
	paracetamol		398	
	gilenya		390	
	avandia		378	
	acutane		326	
	remicade		292	
	baraclude		283	
	exjade		274	
	crestor		256	
	gleevec		243	
	tracleer		238	
	lamisil		225	
	januvia		225	
	sutent		192	
	+-----+-----+			

Compare the side effects of two medicines

Compare the symptom counts for two medicines

Say you want to compare the count of symptoms reported for lipitor versus crestor

```
select
  ld.pt as symptoms,
  ld.ginti as lipitor_counts,
  rd.ginti as crestor_counts
```

```
from
(select
  pt,
  count (distinct isr) as ginti
from
  aers.demo_drug_reac_combo
where
  lower(drugname) like '%lipitor%'
group by
  pt
order by
  ginti desc
limit 25) ld
join
(select
  pt,
  count (distinct isr) as ginti
from
  aers.demo_drug_reac_combo
where
  lower(drugname) like '%crestor%'
group by
  pt
order by
  ginti desc
limit 25) rd
on
ld.pt = rd.pt
```

The comparison results are as follows

symptoms	lipitor_counts	crestor_counts
type 2 diabetes mellitus	2957	503
myalgia	2482	1165
death	1720	268
pain in extremity	951	740
muscle spasms	818	543
arthralgia	806	386
muscular weakness	693	376
myocardial infarction	682	756
fatigue	674	409
pain	619	537
asthenia	604	397
blood creatine phosphokinase increased	531	266
cerebrovascular accident	478	328
rhabdomyolysis	454	512
blood cholesterol increased	428	511
gait disturbance	403	276
malaise	381	631
diabetes mellitus	319	257

Compare the age groups counts reported for two medicines

Say you want to compare the counts of age groups that have reported side effects for lipitor versus crestor

```
select
  ld.age_group,
  ld.ginti as lipitor_counts,
  rd.ginti as crestor_counts
from
  (select
    age_group,
    count (distinct isr) as ginti
  from
    aers.demo_drug_reac_combo
  where
    lower(drugname) like '%lipitor%'
  and
    age_group is not null
  group by
    age_group
  order by
    ginti desc
  limit 25) ld
join
  (select
    age_group,
    count (distinct isr) as ginti
  from
    aers.demo_drug_reac_combo
  where
    lower(drugname) like '%crestor%')
```

```

and
  age_group is not null
group by
  age_group
order by
  ginti desc
limit
  25) rd
on
  ld.age_group = rd.age_group

```

The comparison results are as follows

age_group	lipitor_counts	crestor_counts
60-64	1301	1324
55-59	1281	1149
65-69	1256	1109
70-74	1128	846
50-54	1021	903
75-79	854	690
80-84	646	506
45-49	633	562
40-44	417	356
85-89	336	233
35-39	176	144
90-94	149	67
30-34	100	82
95-99	37	7
25-29	31	33

20-24	28	7	
0-4	16	9	
15-19	13	10	
5-9	8	4	
100-104	5	1	
10-14	3	4	
+-----+	+-----+	+-----+	+

Compare the side effect counts reported by year for two medicines

Say you want to compare the side effects counts by year that have been reported for lipitor versus crestor

```

select
  ld.year as symptoms,
  ld.ginti as lipitor_counts,
  rd.ginti as crestor_counts
from
  (select
    year,
    count (distinct isr) as ginti
  from
    aers.demo_drug_reac_combo
  where
    lower(drugname) like '%lipitor%'
  group by
    year
  order by
    year desc
  limit

```



```

25) ld
join
(select
  year,
  count (distinct isr) as ginti
from
  aers.demo_drug_reac_combo
where
  lower(drugname) like '%crestor%'
group by
  year
order by
  year desc
limit
  25) rd
on
ld.year = rd.year

```

The comparison results are as follows

symptoms	lipitor_counts	crestor_counts
2017	148	105
2016	966	490
2015	1888	392
2014	2807	854
2013	1672	702
2012	3318	2639

2011	2424	2754	
2010	3726	1817	
2009	662	1742	
2008	678	1051	
2007	895	1052	
2006	760	150	
2005	619	565	
2004	644	1012	
2003	477	22	
+-----+	+-----+	+-----+	+