Open Claims in Hadoop

OMOP Factory Overview

North America

SURVEY | US EMR | CAN EMR | Onco EMR | CLAIMS

Non North America

GER EMR | FR EMR | AU EMR | UK EMR | SURVEY

OHDSI Training

Training

OMOP Factory

- Engine development
- Operations, includes
  - Mapping, Training, Support
  - Syncing with OHDSI community
- ETL development

Data feed

E360 CB+AA+ Dataset Explorer

Customers

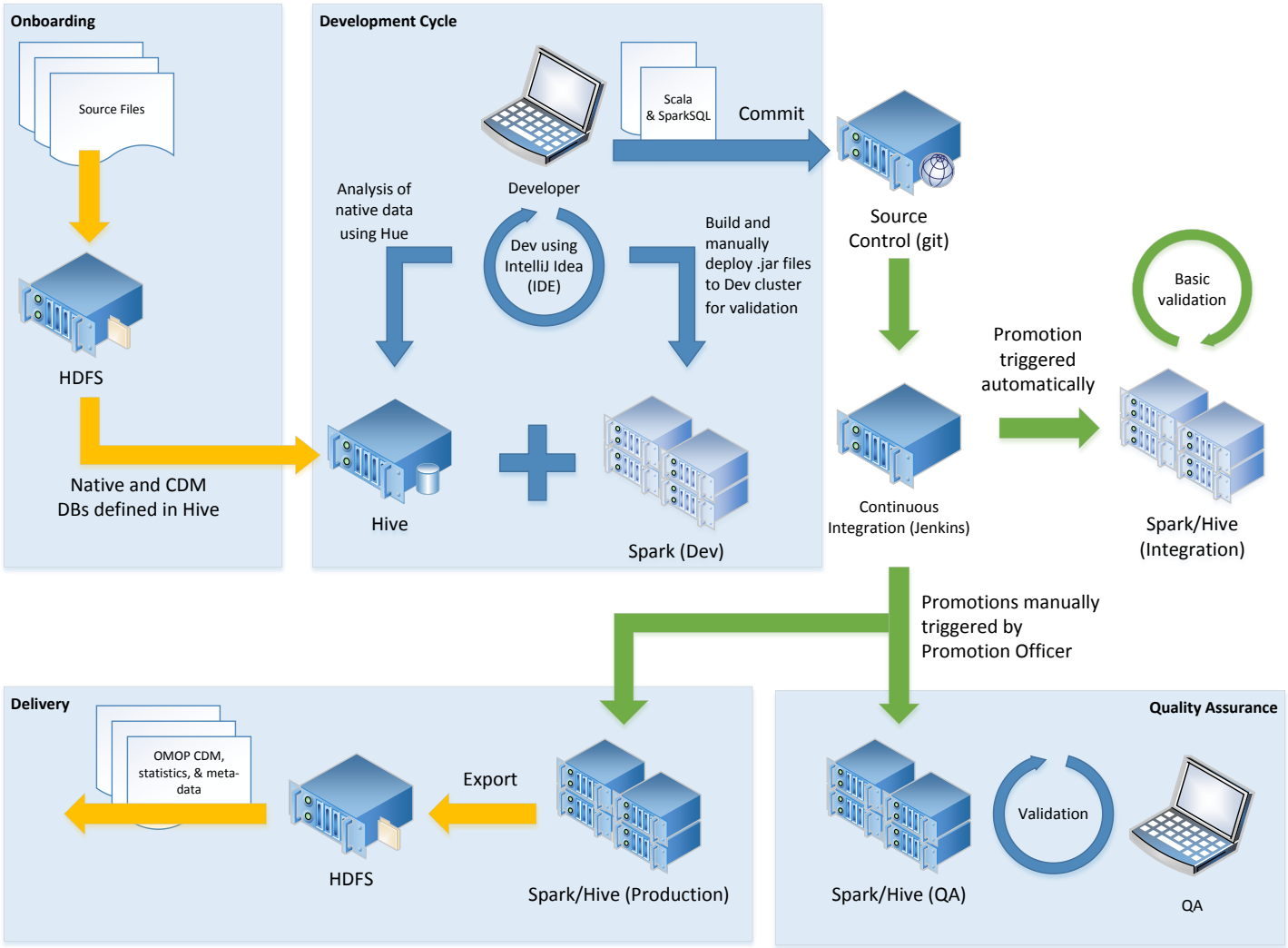
Remote Studies

QI Service

Open Claims Project

- Total size - 40TB
- Project and non-projected pre-adjudicated prescription (Rx) and medical claims (Dx)
  - Rx
    - Approximately **680 million** total prescription transactions per month / **300 million** of which are good/paid prescriptions
    - **215 million** patients observed in 2012
    - History since 2001
    - Project data since 2006
  - Dx
    - **1.1 million** of individual healthcare professionals observed within medical claim transactions in 2012
    - **162 millions** patients in 2012
    - History since 1999
    - Projected data since 2005

Hadoop Architecture



Load Native data using Spark SQL

- Copy input file from local file system to HDFS

  hdfs dfs –put PrescriberReference.txt.gz /hdfs_path/
- Define native table in hive.
- Define the layout of input file.

```
val refPrescriberSchema=StructType(Array(
  StructField("rxer_id",StringType,true),   StructField("spcl_cd",StringType,true),
  StructField("spcl_desc",StringType,true),   StructField("st_cd",StringType,true),
  StructField("zip_cd",StringType,true)))
```

- Read input file using spark read API and register as temporary table

```
//Read input file
val dfRefPrescriber=spark.read
  .format("csv")
  .schema(refPrescriberSchema)
  .option("sep","|")
  .load("/hdfs_path/PrescriberReference_201508_015288.txt.gz")

//Register as temp view
dfRefPrescriber.createOrReplaceTempView(“tmpRefPrescriber”)
```

- Insert native data to hive parquet table

```
spark.sql(“INSERT INTO $databaseName.Ref_Prescriber
            SELECT rxer_id, spcl_cd, spcl_desc, st_cd, zip_cd
              FROM tmpRefPrescriber”)
```

Native data – Prescriber table example

Native format – Prescriber input text file

```
0000131|U|UROLOGY|AL|36106
0000230|FM|FAMILY MEDICINE|AL|36701
0000255|RHU|RHEUMATOLOGY|FL|33805
0000313|U|UROLOGY|AL|36330
0000316|GS|GENERAL SURGERY|AL|36033
0000326|FM|FAMILY MEDICINE|AL|35010
0000327|GP|GENERAL PRACTICE|GA|31907
0000331|OTO|OTOLARYNGOLOGY|AL|35211
0000355|IM|INTERNAL MEDICINE|AL|35209
0000381|OBG|OBSTETRICS/GYNECOLOGY|AL|36207
```

Native format – Prescriber table in Hive

rxer_id	spcl_cd	spcl_desc	st_cd	zip_cd
0000131	U	UROLOGY	AL	36106
0000230	FM	FAMILY MEDICINE	AL	36701
0000255	RHU	RHEUMATOLOGY	FL	33805
0000313	U	UROLOGY	AL	36330
0000316	GS	GENERAL SURGERY	AL	36033
0000326	FM	FAMILY MEDICINE	AL	35010
0000327	GP	GENERAL PRACTICE	GA	31907
0000331	OTO	OTOLARYNGOLOGY	AL	35211
0000355	IM	INTERNAL MEDICINE	AL	35209
0000381	OBG	OBSTETRICS/GYNECOLOGY	AL	36207

Load CDM Table using Spark SQL - Provider

- SQL to convert native format provider data to CDM

```
etlQuery=
 s"""
   SELECT DISTINCT
        COALESCE (c.concept_id, 0) AS specialty_concept_id,
        rp.rxer_id provider_source_value,
        rp.spcl_desc specialty_source_value,
        COALESCE (c.concept_id, 0) specialty_source_concept_id
     FROM $databaseName.ref_prescriber rp
     LEFT JOIN $databaseName.source_to_concept_map stcm
      ON UPPER (rp.spcl_desc) = UPPER(STCM.source_code_description)
      AND STCM.source_vocabulary_id = 'Specialty'
     LEFT JOIN $databaseName.concept c
      ON stcm.target_concept_id = c.concept_id
      AND FROM_UNIXTIME(UNIX_TIMESTAMP()) BETWEEN c.valid_start_date AND c.valid_end_date
  """
val dfProviderStg=spark.sql(etlQuery)
```

- Execute User Defined Function to generate provider_id identity column
- Load the data from dataframe to CDM provider table.

```
Spark.sql("INSERT INTO $databaseName.provider
SELECT <column list> from tmpProvider
```

CDM – Provider table example

provider_id	provider_name	npi	dea	specialty_concept_id	care_site_id	year_of_birth	gender_concept_id	provider_source_value	specialty_source_value	specialty_source_concept_id
185	NULL	NULL	NULL	44777747	0	NULL	0	0000131	UROLOGY	44777747
116	NULL	NULL	NULL	38004453	0	NULL	0	0000230	FAMILY MEDICINE	38004453
69	NULL	NULL	NULL	44777791	0	NULL	0	0000255	RHEUMATOLOGY	44777791
47	NULL	NULL	NULL	44777747	0	NULL	0	0000313	UROLOGY	44777747
32	NULL	NULL	NULL	44777717	0	NULL	0	0000316	GENERAL SURGERY	44777717
52	NULL	NULL	NULL	38004453	0	NULL	0	0000326	FAMILY MEDICINE	38004453
120	NULL	NULL	NULL	38004446	0	NULL	0	0000327	GENERAL PRACTICE	38004446
83	NULL	NULL	NULL	38004449	0	NULL	0	0000331	OTOLARYNGOLOGY	38004449
123	NULL	NULL	NULL	38004456	0	NULL	0	0000355	INTERNAL MEDICINE	38004456
198	NULL	NULL	NULL	38004461	0	NULL	0	0000381	OBSTETRICS/GYNECOLOGY	38004461

Extract CDM data to flat file – Provider table

- Select the data from Hive and create a data frame.

```
etlQuery=
 s"""
   SELECT provider_id, provider_name, npi, dea, specialty_concept_id, care_site_id, year_of_birth,
            gender_concept_id, provider_source_value, specialty_source_value, specialty_source_concept_id,
            gender_source_value, gender_source_concept_id
     FROM $databaseName.provider
  """
val dfProvider=spark.sql(etlQuery)
```

- Extract the data from data frame to flat file using Spark Write API.

```
dfProvider.write
.format("csv")
.option("sep","|")
.csv("/hdfs_path/provider")
```

Output File

```
127955||||45756833|0||0|1867664|VASCULAR/INTERVENTION RAD|45756833||0
202787||||45756833|0||0|3188695|VASCULAR/INTERVENTION RAD|45756833||0
151455||||45756833|0||0|1957079|VASCULAR/INTERVENTION RAD|45756833||0
82383||||45756833|0||0|0796851|VASCULAR/INTERVENTION RAD|45756833||0
28430||||38004446|0||0|0223438|GENERAL PRACTICE|38004446||0
68230||||38004446|0||0|0725954|GENERAL PRACTICE|38004446||0
```

QuintilesIMS™

For questions/inquiries, contact:
Mui Van Zandt
Email: mui.vanzandt@quintilesims.com
Phone: (415) 692-9835

quintilesims.com