# Demo Script SAP HANA

This file provides a copy source for many of the steps in the SAP HANA and Hadoop Trial.

~~~~~

CREATE /wiki-data

mkdir /wiki-data

chmod 777 /wiki-data

cd /wiki-data

~~~~~

COPY data from HIVE file to HANA

wget http://wikipedia-pagecounts-hive-results.s3.amazonaws.com/year=2013/month=03/000000 --output-document=2013-03-000000

wget http://wikipedia-pagecounts-hive-results.s3.amazonaws.com/year=2013/month=03/000001 --output-document=2013-03-000001

wget http://wikipedia-pagecounts-hive-results.s3.amazonaws.com/year=2013/month=03/000002 --output-document=2013-03-000002

wget http://wikipedia-pagecounts-hive-results.s3.amazonaws.com/year=2013/month=04/000000 --output-document=2013-04-000000

wget http://wikipedia-pagecounts-hive-results.s3.amazonaws.com/year=2013/month=04/000001 --output-document=2013-04-000001

wget http://wikipedia-pagecounts-hive-results.s3.amazonaws.com/year=2013/month=04/000002 --output-document=2013-04-000002

wget http://wikipedia-pagecounts-hive-results.s3.amazonaws.com/year=2013/month=05/000000 --output-document=2013-05-000000

wget http://wikipedia-pagecounts-hive-results.s3.amazonaws.com/year=2013/month=05/000001 --output-document=2013-05-000001

wget http://wikipedia-pagecounts-hive-results.s3.amazonaws.com/year=2013/month=05/000002 --output-document=2013-05-000002

~~~~~

Verify files copied and the number of lines per file

ls -l 2013\*

wc -l 2013\*

~~~~~

CREATE SCHEMA

-- Create the schema for the Wikipedia page hit data and related tables

CREATE SCHEMA "WIKIDATA";

-- Create the COLUMN table for the staging data

CREATE COLUMN TABLE "WIKIDATA"."STAGE-PAGEHITS"

("PROJECTCODE" VARCHAR(50),

"PAGENAME" VARCHAR(2000),

"YEAR" VARCHAR(4),

"MONTH" VARCHAR(2),

"DAY" VARCHAR(2),

"HOUR" VARCHAR(2),

"PAGEHITCOUNTFORHOUR" BIGINT,

"BYTESDOWNLOADEDFORHOUR" BIGINT

);

~~~~~

IMPORT files

-- Import the first delimited text file

IMPORT FROM CSV FILE '/wiki-data/2013-03-000000'

INTO "WIKIDATA"."STAGE-PAGEHITS"

WITH RECORD DELIMITED BY '\n'

FIELD DELIMITED BY ''

ERROR LOG '/wiki-data/import-2013-03-000000.err';

~~~~~

SELECT count of records imported

SELECT COUNT(\*) FROM "WIKIDATA"."STAGE-PAGEHITS";

~~~~~

Check Error files

cd /wiki-data/

more import-2013-03-000000.err

~~~~~

DROP Table and Create with Changes

-- Drop the staging table

DROP TABLE "WIKIDATA"."STAGE-PAGEHITS";

-- Create the COLUMN table for the staging data

CREATE COLUMN TABLE "WIKIDATA"."STAGE-PAGEHITS"

("PROJECTCODE" VARCHAR(50),

"PAGENAME" VARCHAR(2000),

"YEAR" VARCHAR(4),

"MONTH" VARCHAR(2),

"DAY" VARCHAR(2),

"HOUR" VARCHAR(2),

"PAGEHITCOUNTFORHOUR" BIGINT,

"BYTESDOWNLOADEDFORHOUR" VARCHAR(50)

);

~~~~~

Import the files

-- First import and verify the count for the first file

IMPORT FROM CSV FILE '/wiki-data/2013-03-000000' INTO "WIKIDATA"."STAGE-PAGEHITS" WITH RECORD DELIMITED BY '\n' FIELD DELIMITED BY '' ERROR LOG '/wiki-data/import-2013-03-000000.err';

SELECT COUNT(\*) FROM "WIKIDATA"."STAGE-PAGEHITS";

-- Import the rest of the Hive files

IMPORT FROM CSV FILE '/wiki-data/2013-03-000001' INTO "WIKIDATA"."STAGE-PAGEHITS" WITH RECORD DELIMITED BY '\n' FIELD DELIMITED BY '' ERROR LOG '/wiki-data/import-2013-03-000001.err';

IMPORT FROM CSV FILE '/wiki-data/2013-03-000002' INTO "WIKIDATA"."STAGE-PAGEHITS" WITH RECORD DELIMITED BY '\n' FIELD DELIMITED BY '' ERROR LOG '/wiki-data/import-2013-03-000002.err';

IMPORT FROM CSV FILE '/wiki-data/2013-04-000000' INTO "WIKIDATA"."STAGE-PAGEHITS" WITH RECORD DELIMITED BY '\n' FIELD DELIMITED BY '' ERROR LOG '/wiki-data/import-2013-04-000000.err';

IMPORT FROM CSV FILE '/wiki-data/2013-04-000001' INTO "WIKIDATA"."STAGE-PAGEHITS" WITH RECORD DELIMITED BY '\n' FIELD DELIMITED BY '' ERROR LOG '/wiki-data/import-2013-04-000001.err';

IMPORT FROM CSV FILE '/wiki-data/2013-04-000002' INTO "WIKIDATA"."STAGE-PAGEHITS" WITH RECORD DELIMITED BY '\n' FIELD DELIMITED BY '' ERROR LOG '/wiki-data/import-2013-04-000002.err';

IMPORT FROM CSV FILE '/wiki-data/2013-05-000000' INTO "WIKIDATA"."STAGE-PAGEHITS" WITH RECORD DELIMITED BY '\n' FIELD DELIMITED BY '' ERROR LOG '/wiki-data/import-2013-05-000000.err';

IMPORT FROM CSV FILE '/wiki-data/2013-05-000001' INTO "WIKIDATA"."STAGE-PAGEHITS" WITH RECORD DELIMITED BY '\n' FIELD DELIMITED BY '' ERROR LOG '/wiki-data/import-2013-05-000001.err';

IMPORT FROM CSV FILE '/wiki-data/2013-05-000002' INTO "WIKIDATA"."STAGE-PAGEHITS" WITH RECORD DELIMITED BY '\n' FIELD DELIMITED BY '' ERROR LOG '/wiki-data/import-2013-05-000002.err';

SELECT COUNT(\*) FROM "WIKIDATA"."STAGE-PAGEHITS";

~~~~~

Check error files

ls -l \*.err

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* PART 2 \*

\* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

~~~~~

Check for duplicates

SELECT

"PROJECTCODE","PAGENAME",SUM("PAGEHITCOUNTFORHOUR"),COUNT("PAGENAME"),"YEAR", "MONTH","DAY","HOUR"

FROM

"WIKIDATA"."STAGE-PAGEHITS"

WHERE "MONTH"='03'

GROUP BY "PROJECTCODE","PAGENAME","YEAR","MONTH","DAY","HOUR"

HAVING (COUNT("PROJECTCODE")>1) and (COUNT("PAGENAME")>1) and (COUNT("YEAR")>1)

and (COUNT("MONTH")>1) and (COUNT("DAY")>1) and (COUNT("HOUR")>1)

ORDER BY 3 DESC;

~~~~~

Create the fact table

CREATE COLUMN TABLE "WIKIDATA"."PAGEHITSFACTTABLE"

("PAGENAME" VARCHAR(2000),

"YEAR" VARCHAR(4),

"MONTH" VARCHAR(2),

"DAY" VARCHAR(2),

"HOUR" VARCHAR(2),

"PAGEHITCOUNTFORHOUR" BIGINT,

"BYTESPERHOUR" BIGINT,

"EVENTDATE" VARCHAR(8),

"LANGCODE" VARCHAR(50),

"PROJCODE" VARCHAR(50)

);

~~~~~

Verify expressions for key columns used to join to dimension tables

SELECT DISTINCT "PROJECTCODE",

CASE WHEN ("PROJECTCODE" LIKE '%.%')

THEN LOCATE("PROJECTCODE",'.')-1

ELSE 0

END AS "POSITION",

CASE WHEN ("PROJECTCODE" LIKE '%.%')

THEN SUBSTRING("PROJECTCODE", 1, LOCATE("PROJECTCODE",'.')-1)

ELSE "PROJECTCODE"

END AS LANGCODE,

CASE WHEN ("PROJECTCODE" LIKE '%.%')

THEN SUBSTRING("PROJECTCODE", LOCATE("PROJECTCODE",'.')+1)

ELSE 'wp'

END AS PROJCODE

FROM "WIKIDATA"."STAGE-PAGEHITS"

WHERE Month = '03' AND Day = '01' AND Hour = '00' LIMIT 100;

~~~~~

Load data into the fact table

SELECT "PAGENAME","YEAR","MONTH","DAY","HOUR",

"PAGEHITCOUNTFORHOUR",

CASE WHEN ("BYTESDOWNLOADEDFORHOUR" LIKE '%N%')

THEN 0

ELSE TO\_BIGINT("BYTESDOWNLOADEDFORHOUR")

END AS BYTESPERHOUR,

"YEAR"||"MONTH"||"DAY" AS EVENTDATE,

CASE WHEN ("PROJECTCODE" LIKE '%.%')

THEN SUBSTRING("PROJECTCODE", 1, LOCATE("PROJECTCODE",'.')-1)

ELSE "PROJECTCODE"

END AS LANGCODE,

CASE WHEN ("PROJECTCODE" LIKE '%.%')

THEN SUBSTRING("PROJECTCODE", LOCATE("PROJECTCODE",'.')+1)

ELSE 'wp'

END AS PROJCODE

FROM "WIKIDATA"."STAGE-PAGEHITS"

INTO "WIKIDATA"."PAGEHITSFACTTABLE";

~~~~~

Exclude bad records (SELECT)

SELECT \* FROM "WIKIDATA"."PAGEHITSFACTTABLE"

WHERE "PAGEHITCOUNTFORHOUR" > 53000000;

~~~~~

Exclude bad records (DELETE)

DELETE FROM "WIKIDATA"."PAGEHITSFACTTABLE"

WHERE "PAGEHITCOUNTFORHOUR" > 53000000;

~~~~~

URLs for .csv files:

http://wikipedia-proj-lang-codes.s3.amazonaws.com/UniqueProjectCodes.csv

http://wikipedia-proj-lang-codes.s3.amazonaws.com/UniqueLanguageCodes.csv

~~~~~

Create Dimension Tables

CREATE COLUMN TABLE "WIKIDATA"."DIMPROJECT"

("ProjectTitle" VARCHAR(50) NOT NULL,

"ProjectCode" VARCHAR(10) NOT NULL

);

CREATE COLUMN TABLE "WIKIDATA"."DIMLANGUAGE"

("LanguageCode" VARCHAR(25) NOT NULL,

"Language" VARCHAR(50) NOT NULL

);

~~~~~

Examining Structure of M\_TIME\_DIMENSION

SELECT \* FROM "\_SYS\_BI"."M\_TIME\_DIMENSION" LIMIT 50;

~~~~~

Enhancing M\_TIME\_DIMENSION

-- Adding ENGLISH\_DAY\_OF\_WEEK

ALTER TABLE "\_SYS\_BI"."M\_TIME\_DIMENSION"

ADD (ENGLISH\_DAY\_OF\_WEEK VARCHAR(10) NULL);

UPDATE "\_SYS\_BI"."M\_TIME\_DIMENSION"

SET ENGLISH\_DAY\_OF\_WEEK=

CASE DAY\_OF\_WEEK\_INT

WHEN 0 THEN 'Monday'

WHEN 1 THEN 'Tuesday'

WHEN 2 THEN 'Wednesday'

WHEN 3 THEN 'Thursday'

WHEN 4 THEN 'Friday'

WHEN 5 THEN 'Saturday'

WHEN 6 THEN 'Sunday'

END

WHERE ENGLISH\_DAY\_OF\_WEEK IS NULL;

-- Adding ENGLISH\_MONTH

ALTER TABLE "\_SYS\_BI"."M\_TIME\_DIMENSION"

ADD (ENGLISH\_MONTH VARCHAR(10) NULL);

UPDATE "\_SYS\_BI"."M\_TIME\_DIMENSION"

SET ENGLISH\_MONTH=

CASE MONTH\_INT

WHEN 1 THEN 'January'

WHEN 2 THEN 'February'

WHEN 3 THEN 'March'

WHEN 4 THEN 'April'

WHEN 5 THEN 'May'

WHEN 6 THEN 'June'

WHEN 7 THEN 'July'

WHEN 8 THEN 'August'

WHEN 9 THEN 'September'

WHEN 10 THEN 'October'

WHEN 11 THEN 'November'

WHEN 12 THEN 'December'

END

WHERE ENGLISH\_MONTH IS NULL;

-- Seeing what the updated table looks like

SELECT DATE\_SAP, YEAR, QUARTER, MONTH\_INT, ENGLISH\_MONTH, WEEK, DAY\_OF\_WEEK\_INT, ENGLISH\_DAY\_OF\_WEEK

FROM "\_SYS\_BI"."M\_TIME\_DIMENSION" LIMIT 50;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* PART 3 \*

\* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

(Uses HANA Studio graphical tools, so no text to copy)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* PART 4 \*

\* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

The Lumira User is created already on the instance, but the following commands still need to be executed:

-- Grant SELECT permissions on the SCHEMAs where the

-- data resides for your Analytic Views

GRANT SELECT ON SCHEMA "WIKIDATA" TO LUMIRA;

-- Check to see that the permissions were set correctly

SELECT \* FROM "SYS"."GRANTED\_PRIVILEGES" WHERE GRANTEE = 'LUMIRA';

Note: Lumira user password is "4262Hana"