**Workflow document**

Nabu

Curation Using Almaren

Oct 1, 2020

Version 0.1



Contents

[1. Almaren code using Zeppelin 2](#_Toc52467210)

[2. Curation Using Almaren 5](#_Toc52467211)

[**2.1. Prerequisites:** 5](#_Toc52467212)

[**2.2. Steps to be followed:** 5](#_Toc52467213)

# Almaren code using Zeppelin

Users can develop and test the Almaren code in Apache Zeppelin notebook.

**Apache Zeppelin URL:**

<https://arch-data-engineering-master3.arch-env.u55l-xcud.cloudera.site/arch-data-engineering/cdp-proxy/zeppelin/#/>

Below is an example of PCSDW triple creation that we can run it in zeppelin notebook.

-----------------------------------------------------------------------------------------------------------------------

import com.github.music.of.the.ainur.almaren.builder.Core.Implicit

import com.github.music.of.the.ainur.almaren.Almaren

import org.apache.spark.sql.DataFrame

val almaren = Almaren("pcsdw")

val res = almaren.builder.sourceSql("""

with RESULT\_LX\_RPT\_VW AS (

SELECT STUDY\_ID,STUDY\_NUMBER, STUDY\_GROUP\_ID, SRC\_GROUP\_NUMBER,PHASE\_ID, DAY\_OF\_PHASE, MEAS\_AREA, SPECIES\_NAME, MEAS\_NAME, NUMERIC\_VALUE AS RESULT, 'Cmpd (Root)' as ENTITY1\_TYPE,

MEAS\_NAME as ENTITY2,

MEAS\_AREA as ENTITY2\_TYPE,

MEAS\_UNIT\_NAME as RESULT\_TYPE

FROM preclinical.t\_ddmlcnp1\_pcsdw\_result\_lx\_rpt\_vw

WHERE MEAS\_AREA IN ('Chemistry Measurement','Hematology Measurement','Urine Chemistry Measurement','Urinalysis Measurement')

AND SPECIES\_NAME IN ('Rat')

AND MEAS\_NAME IN ('Leukocytes','Neutrophil','Basophil','Eosinophil','Monocytes','Platelet','High RNA Reticulocytes')

AND NUMERIC\_VALUE IS NOT NULL ),

STUDYGROUP\_DOP\_SUMMARY\_VW AS (

SELECT STUDY\_GROUP\_ID, PHASE\_ID, DAY\_OF\_PHASE, AVG\_VALUE AS VEHICLE\_GROUP\_MEAN,1 AS STRENGTH

FROM preclinical.t\_ddmlcnp1\_pcsdw\_studygroup\_dop\_summary\_vw ),

STUDY\_GROUP\_COMPOUND\_VW AS (

SELECT STUDY\_NUMBER, STUDY\_ID,STUDY\_GROUP\_ID ,SRC\_GROUP\_NUMBER,TEST\_ARTICLE as ENTITY1

FROM preclinical.t\_ddmlcnp1\_pcsdw\_study\_group\_compound\_vw

),

JOINED\_DATA AS (

select \*,CASE WHEN NORMALIZED\_RESULT >0 THEN "Increasing"

WHEN NORMALIZED\_RESULT <0 THEN "Decreasing"

WHEN NORMALIZED\_RESULT = 0 THEN "No Effect" else NORMALIZED\_RESULT end as REL\_TYPE

from (

SELECT A.\*, B.VEHICLE\_GROUP\_MEAN, ENTITY1, (A.RESULT - B.VEHICLE\_GROUP\_MEAN) / B.VEHICLE\_GROUP\_MEAN AS NORMALIZED\_RESULT,B.STRENGTH

FROM RESULT\_LX\_RPT\_VW A INNER JOIN STUDYGROUP\_DOP\_SUMMARY\_VW B ON

A.STUDY\_GROUP\_ID = B.STUDY\_GROUP\_ID

AND A.PHASE\_ID = B.PHASE\_ID

AND A.DAY\_OF\_PHASE = B.DAY\_OF\_PHASE INNER JOIN STUDY\_GROUP\_COMPOUND\_VW C

ON A.STUDY\_ID = C.STUDY\_ID and A.STUDY\_GROUP\_ID = C.STUDY\_GROUP\_ID and A.STUDY\_NUMBER = C.STUDY\_NUMBER AND A.SRC\_GROUP\_NUMBER= C.SRC\_GROUP\_NUMBER

)X )

SELECT RESULT, ENTITY1, ENTITY1\_TYPE, ENTITY2, ENTITY2\_TYPE, RESULT\_TYPE, REL\_TYPE,STRENGTH, METADATA

FROM (

SELECT struct(MEAS\_AREA,SPECIES\_NAME,MEAS\_NAME,STUDY\_GROUP\_ID,PHASE\_ID,DAY\_OF\_PHASE,VEHICLE\_GROUP\_MEAN,NORMALIZED\_RESULT) as METADATA,

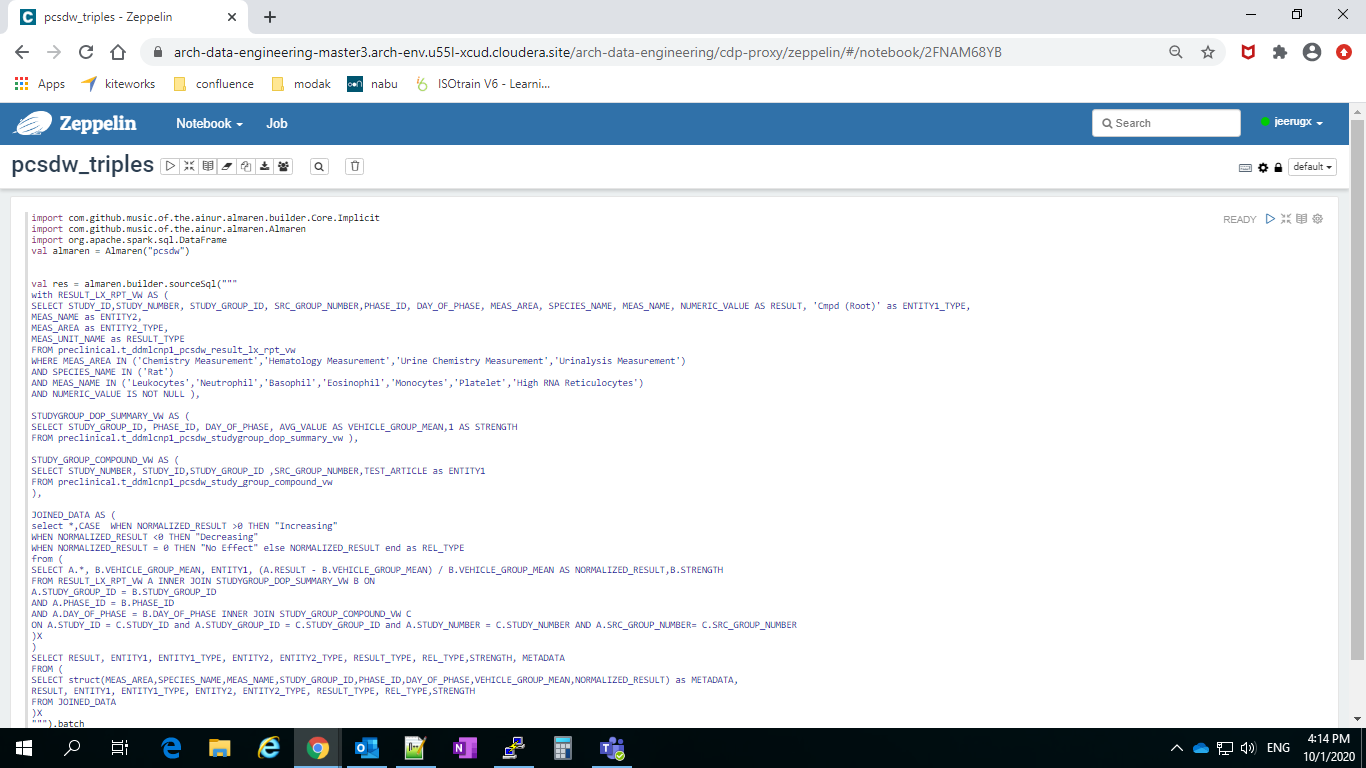
RESULT, ENTITY1, ENTITY1\_TYPE, ENTITY2, ENTITY2\_TYPE, RESULT\_TYPE, REL\_TYPE,STRENGTH

FROM JOINED\_DATA

)X

""").batch

res.write.mode("OVERWRITE").option("format", "parquet").option("path","s3a://arch-data-lake-dev/data/warehouse/integrated/preclinical.db/pcsdw\_triples\_v2").saveAsTable("preclinical.pcsdw\_triples\_v2")



# Curation Using Almaren

## **2.1. Prerequisites:**

1. Almaren code should be uploaded in AbbVie GitHub.
2. Access to **kosh** database where nabu data model is residing.
   1. Postgres server details

|  |  |
| --- | --- |
| Host name | arch-nabu-dev-instance-1.caurfctzkbwj.us-east-1.rds.amazonaws.com |
| Port | 5432 |
| Database name | kosh |

* 1. We can use any account which has all privileges like SELECT, INSERT, UPDATE. For example, **SVC-ARCH-PG3-RW**

1. Access to ARCH-CDP environment (ARCH-DATA-ENGINEERING cluster)

## **2.2. Steps to be followed:**

1. Create a data movement by following below steps:
   1. Run below query on Nabu PostgreSQL instance by changing necessary parameters:

*INSERT INTO nabu.data\_movement\_physical*

*(data\_movement\_name, contact\_info, data\_movement\_type, refresh\_freq, dataplace\_sub\_component\_id, data\_movement\_engine\_id, workflow\_engine\_id, workflow\_id, valid\_to\_ts, valid\_from\_ts, crt\_by, crt\_ts, mod\_by, mod\_ts, data\_movement\_additional\_info)*

*VALUES('<data\_movement\_name>', '{}', 'curation', '<Provide cron expression of schedule time>', 0, 0, 1, 34, '9999-12-31', now(), 'Modak', now(), 'Modak', now(),*

*'{*

*"curation\_details": {*

*"directory\_path": "<git project directory name>",*

*"file\_path": "<path of the almaren code we need to execute as present in git>",*

*"git\_url": "<git clone url using ssh>"*

*}*

*-- “note”:”advanced options is optional”*

*--, “advanced\_options”: {*

*-- "driver\_memory": "<driver memory needed to run the specific job>",*

*--"executor\_memory: "<executor memory needed to run the specific job>",*

*--"num\_executors": "<num executors needed",*

*--"no\_of\_cores": "<number of cores>",*

*--"git\_branch": "<git tag name or branch name>",*

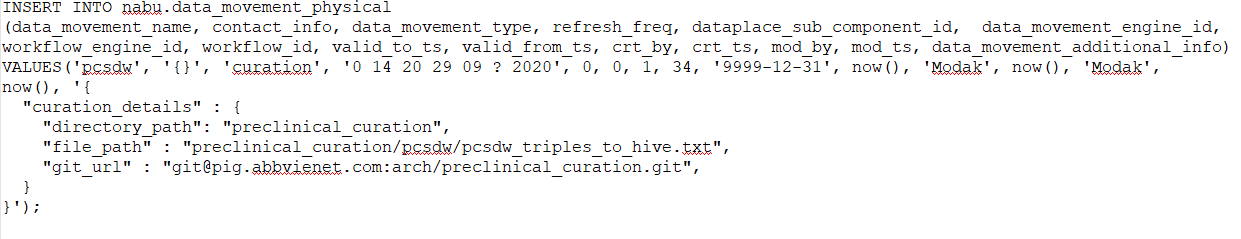
*--"packages": "<external packages needed for the respective Almaren code>"*

*--}*

*}');*

Note: Please change the above parameters as shown below.

|  |  |  |
| --- | --- | --- |
| **Placeholder** | **Example** | **Reference** |
| <git project directory name> | *preclinical\_curation* |  |
| <path of the almaren code we need to execute as present in git> | *preclinical\_curation/pcsdw/pcsdw\_triples\_to\_hive.txt* |  |
| <git clone url using ssh> | *git@pig.abbvienet.com:arch/preclinical\_curation.git* |  |

Example query:

1. You can confirm your data\_movement details after insertion by running below query

*select \**

*from nabu.data\_movement\_physical*

*where 1=1*

*and data\_movement\_name=’<datamovement\_name as mentioned above>’*

*and data\_movement\_type='curation'*

*and valid\_to\_ts=’9999-12-31’*

Example:

*select \**

*from nabu.data\_movement\_physical*

*where 1=1*

*and data\_movement\_name=’* *preclinical\_curation’*

*and data\_movement\_type='curation'*

*and valid\_to\_ts=’9999-12-31’*

This query shows all the details regarding your curation pipeline. Save the data movement id for further modifications of data movement.

1. To reschedule the job, you can update the refresh frequency by running below query

*update nabu.data\_movement\_physical*

*set refresh\_freq=’<Provide new cron expression>’*

*where data\_movement\_id=<data\_movement\_id>*

*and data\_movement\_type='curation'*

*and valid\_to\_ts=’9999-12-31’*

*Example:*

*update nabu.data\_movement\_physical*

*set refresh\_freq='0 45 19 29 09 ? 2020'*

*where data\_movement\_id=215*

*and data\_movement\_type='curation'*

*and valid\_to\_ts=’9999-12-31’*

1. *To check the status of the job run below query with your data\_movement\_id.*

*with get\_batch\_id as (*

*select batch\_id*

*from(*

*select row\_number()over(order by flow\_ts desc) as rownum,batch\_id*

*from nabu.flow\_status*

*where flow\_name ~\*'curation\_<data\_movement\_id>'*

*and status\_code\_id = 10*

*)a*

*where rownum = 1*

*)*

*select batch\_id, status*

*from (*

*select \**

*,case when status\_code\_id = 7 then 'error'*

*when process\_context = 'jdbc\_insert\_script\_details' and status\_code\_id = 6 then 'success'*

*else 'running'*

*end as status*

*,row\_number() over(partition by batch\_id order by task\_ts desc) as rownum*

*from nabu.job\_load\_result*

*where batch\_id in (select batch\_id from get\_batch\_id)*

*)x*

*where rownum = 1*

1. Once the data movement is completed. We can verify the data by checking in Hive for the table which you have created in your Almaren code.