**Loading XML into Snowflake:**

1. Using the Snowflake UI

Created a new table with an XML column using Snowflake’s VARIANT data type.

Created a new column-oriented file format for XML, to be used with the new table.

Loaded the XML data into the XML column of the new table.

This method worked well for tags.xml but it could not be done successfully for the

rest of the files as it gave "file size exceeded" error.

2. Using manual sql queries in the worksheets:

create database ubuntu\_db;

create schema unbuntu\_schema;

use database ubuntu\_db;

use schema unbuntu\_schema;

use role accountadmin;

create or replace table badges\_xml ( src\_xml variant);

select \* from information\_schema.tables where table\_name = 'BADGES\_XML';

desc table badges\_xml;

show tables like '%badges%';

create or replace file format xml\_load

type = 'XML'

compression = 'auto';

describe file format xml\_load;

show file formats like '%xml\_load%';

select \* from information\_schema.file\_formats;

SELECT src\_xml FROM post\_links\_xml;

create or replace table post\_links (creation\_date varchar, id int, link\_typeid int, post\_id int, related\_postid int);

drop table post\_links;

insert into post\_links

SELECT

value:"@CreationDate" as CreationDate,

value:"@Id" as Id,

value:"@LinkTypeId" as LinkTypeId,

value:"@PostId" as PostId,

value:"@RelatedPostId" as RelatedPostId

FROM post\_links\_xml,

LATERAL FLATTEN(to\_array(post\_links\_xml.src\_xml:"$"));

SELECT XMLGET(value, 'Count' ):"$" as "'count' ",

XMLGET( value, 'ExcerptPostId' ):"$" as "expert\_post\_id",

XMLGET( value, 'Id' ):"$" as "id",

XMLGET( value, 'TagName' ):"$" as "tag\_name",

XMLGET( value, 'WikiPostId' ):"$" as "wiki\_post\_id"

FROM post\_links\_xml,

LATERAL FLATTEN(to\_array(tags\_xml.src\_xml:"$" ));

select src\_xml from tags\_xml;

SELECT

value:"@Count" as count,

value:"@ExcerptPostId" as expert\_post\_id,

value:"@Id" as id,

value:"@TagName" as tag\_name,

value:"@WikiPostId" as wiki\_post\_id

FROM post\_links\_xml,

LATERAL FLATTEN(to\_array(tags\_xml.src\_xml:"$"));

Proceeded like this…

Although, the date did load with the help of SnowSql but when transforming the data into relational type, the values in all the field were returned as NULL. Trying a lot but still stuck on it.

Thus, could not move forward with the given queries that were supposed to be performed.

One basic dbt project was successfully created and linked to the snowflake account.

Same all is uploaded to GitHub as seen here.