Collibra

Scala / Python Technical Challenge for Lineage

Your task is to create a program that will extract technical lineage from input.XML.

What is technical lineage?
In this context, lineage is a graph where nodes are columns and relations represent transfer of the data. There are many ways to generate lineage, the most common would be to use SQL - lets say 'create view TECHVIEW as select A from B;' This would create a lineage between table B and TECHVIEW: (B > A) --> (TECHVIEW > A).

However sometimes, we want to transfer data across many databases. This is where ETL tools come in. They are programs that allow you to load data from one DB, work with them and then save it somewhere else.

What is in file input.XML
It is a code representing lineage inside Informatica PowerCenter ETL tool. Elements SOURCE and TARGET are db objects, elements TRANSFORMATION are objects (tables) inside PowerCenter, CONNECTOR objects define the lineage between all of these objects.

The main idea you need to know is that there is a hierarchy of objects in informatica that looks a bit like this: WORKFLOW > SESSION > MAPPING > TRANSFORMATION.
Workflow defines what sessions to launch, sessions are just instances of mapping and inside that are used transformations.

Your goal is therefore to find the lineage that will be created by running the workflow in input.XML.

Output format

I would suggest that you create three output files, all ideally in JSON format:
- all db objects and their ids (ideally in a tree structure db > schema > table > column).
- you can count on objects on the same level in hierarchy having unique names.
ie. {
  db1: {
    sch1: {
      tab1: {
        col1: {
         id: 1
        },
        col2: { id: 2}
      }
    }
  },
  db2 : {...}
}
- all informatica objects and their ids (also some kind of tree structure ending with ... > transformation > column (aka transformfield))
- again - you can count on objects on the same level in hierarchy having unique names.
ie. {
  repo: {workflow: {session: {mapping: {transformation: { col1: {id: 3}, col2: {id: 4}}}}}}
}
- column lineage - just list of tuples id1 -> id2, where id1 and id2 are ids of columns defined in the first two files.
ie. [
 [1,3], [2,4]
]