Source Ananlyser

Target

Informatica Repository Manger (For Folder)

**What is a Mapping?**

Mapping is a collection of source and target objects linked together by a set of transformations. These transformations consist of a set of rules, which define the data flow and how the data is loaded into the targets.

**Source Definition**

**Transformation**

**Target Definition**

* **Links** – Links connect the source definition to different transformations and target tables. It defines how the data flows from source to target and the transformations.

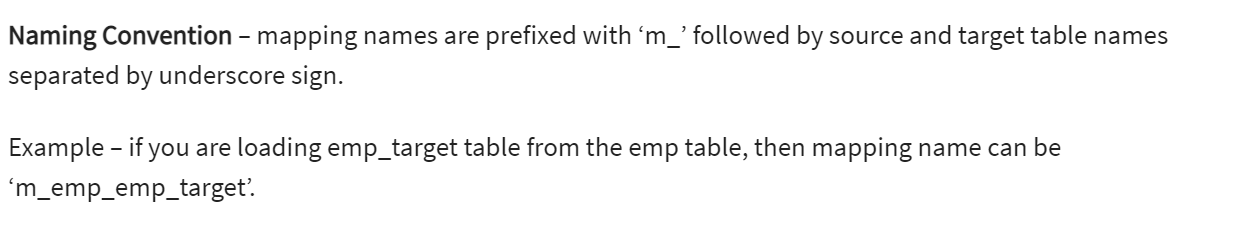
## Why do you need Mapping?

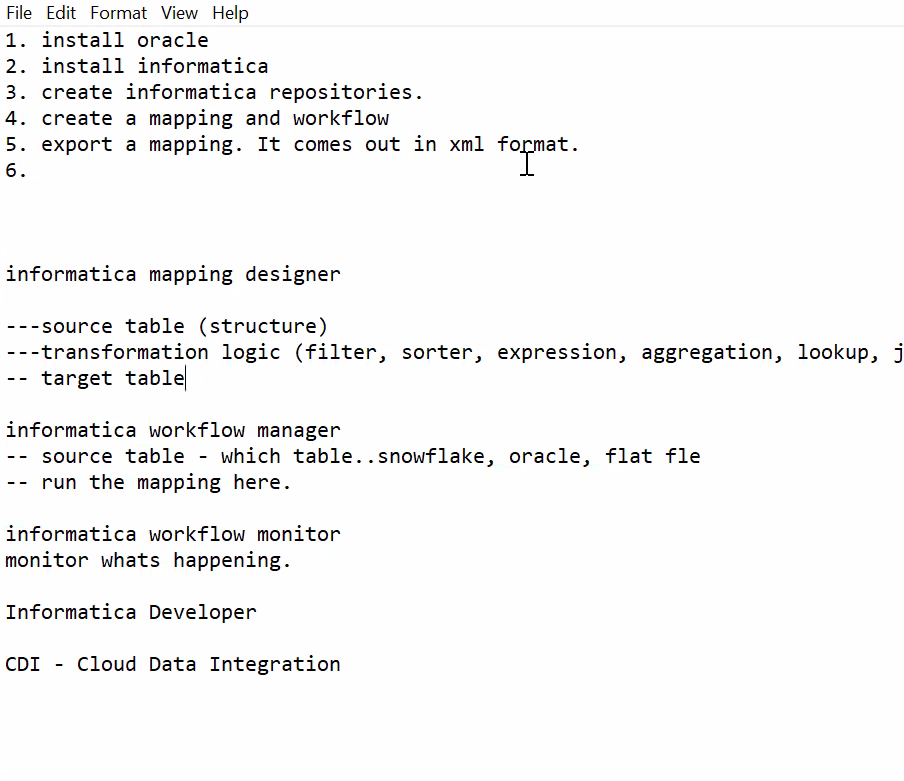
Mapping is an object in Informatica with the help of which you can define how the source data is modified before it reaches the destination or target object. Like if you have employee name as “Bill Clinton” in your source system and in the target system the requirement is to have employee name in the format as “Clinton Bill”, such operations can be designed at the mapping level. In basic terms, what you do with the source data is defined at the mapping level.

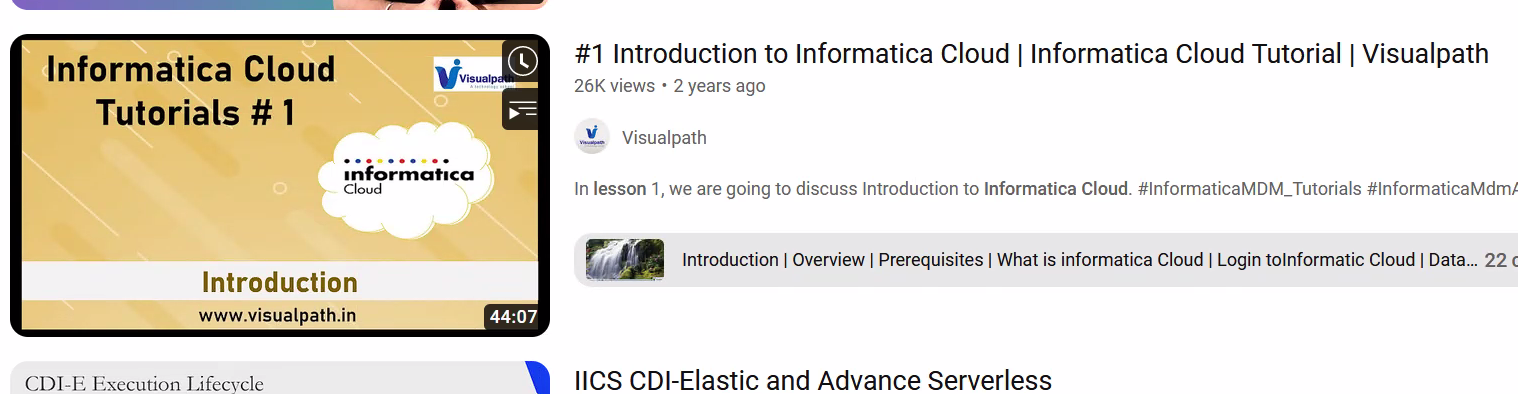
Mapping is the basic Informatica object with the help of which we can define the data transformation details and source/target object characteristics. Mappings help us to define the data transformation at the individual column levels for each row. Even in a single mapping you can handle multiple sources and targets.

**Components of Mapping**

Basic components of a mapping are

* **Source tables**
* **Mapping parameters and variables**
* **Target objects**
* **Mapping transformations**
* **Mapping Source**: Mapping sources are the objects from where you fetch the source data. It can be a database table, flat file, XML source or COBOL file source
* **Mapping target:** Mapping target is our destination objects where final processed data gets loaded. Mapping target can be a relational table of a database, a flat file or XML file. Sources and targets are mandatory in any mapping, their type can differ
* **Mapping Parameters and Variables**: Mapping parameters and variables helps you to create temporary variable objects which will help you to define and store temporary values while mapping data processing. Mapping parameters and variables are optional users defined data types, which can be created for a mapping and can be referenced and updated for a specific requirement. We will learn more about mapping parameters and variables in this section
* **Mapplets**: They are objects which consist of a set of transformation, source or targets. Mapplets are generally created to reuse the existing functionality of a set of transformations. It can be used in any no of mappings.
* 



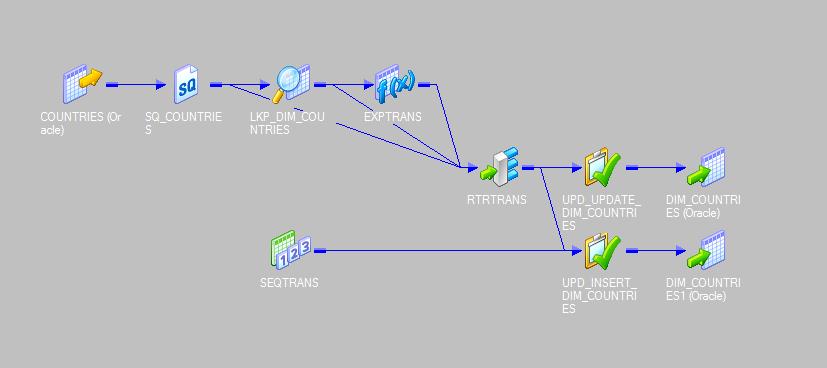


INFORMATICA TO DBT:

* 1. CREATE STAGING MODELS (FOR SOURCE TABLES) USING POSTGRESS
  2. CREATE DIMENSION MODELS (WHICH IMPLEMENTS MAPPING DIAGRAM GIVEN IN USER GUIDE)
  3. ALL TRANFORMATIONS SHOULD COME IN ONE DIMENSION FILE
  4. WITH CLAUSE (FOR EACH TRANFORMATIONS)
  5. FOLLOWED BY ALL
* Look up code example:
* SELECT state,   
  lookup.region  
  FROM {{ ref('my\_table') }}  
  JOIN {{ source('fact\_schema', 'lookup\_table') }} lookup  
  USING(state)

STEPS INVOLVED IN CREATING FACT TABLE:

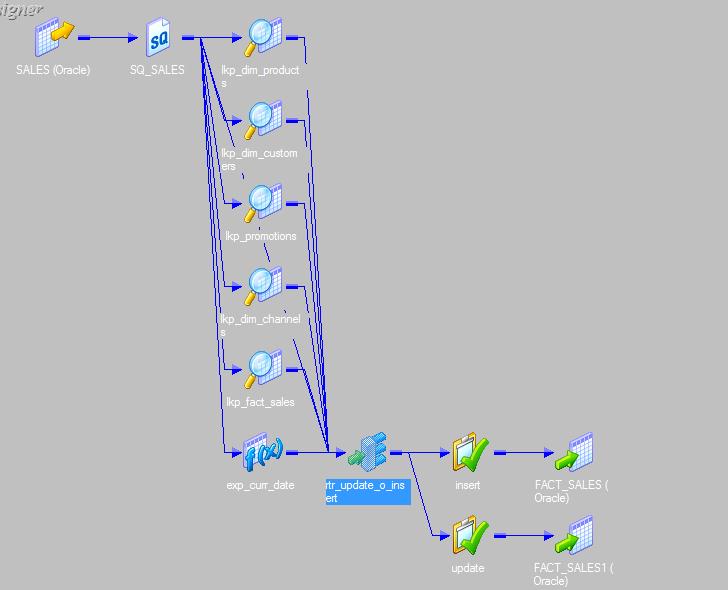
FOLLOWED THIS BELOW DIAGRM:



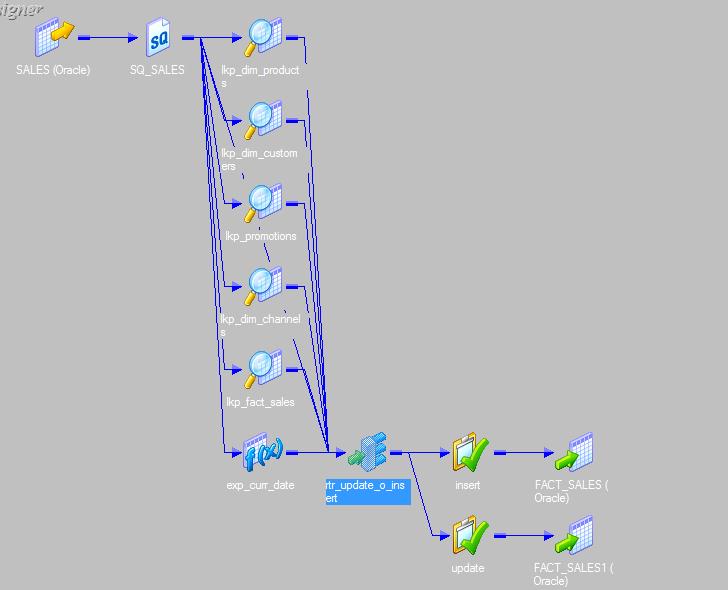
In this …

* 1. Created source clause
  2. Source qualifier from source
  3. Lookup clause with key (generation incremental)
  4. Etrans (eith sysdate)
  5. Rtrans (update separate)
  6. Rtrans (insert separate)
  7. Here we are genarting sequence key for upd\_update\_dim and as well for insert (with other cols) from 5 and 6
  8. Rtrans (upd\_dim\_udate) from rtrans\_update
  9. Rtrans (upd\_dim\_insert) from rtrans\_update
  10. Creating target tables

The above steps for dim tables



For dim (my thoughts)

1. For dim\_upd\_insert and dim\_upd\_update
2. Gave an idea of when case key is null then insert else end as cd flag
3. Gave an idea of when case key is not null then update else end as cd flag
4. Flow of with clause structure (after source, source qularifer, lkp,rtans)
5. Md5 checksum concat(two cols)
6. Rtans got splitted (by bala m and hari)
7. Sequence key code and testing incremental
8. I had pushed sequence key in lkp
9. In lkp (prod\_key) code shared same suggested by bala R
10. But it should be in dim\_upd\_insert and update (bala m and hari)
11. 

This is for fact tables

1. Creating incremental without sequence
2. Source,qualifier, lkp (prod\_key) by mistake was creating here but it can be referred from dim
3. Rtr (update and insert) alias was not given by me
4. Separate the update and insert in rtr (same like dimensions)
5. Rtr\_insert (only srs\_order\_number) combined wth all the other cols from 3
6. Rtr\_update (only lkp\_oder\_number) combined wth all the other cols from 3
7. Then final table

