– When testing update in SCD type 1, are we supposed to use insert or update query, or would any of them be just as well??

Informatica PC Case Study Assessment

- A leading healthcare player, wanted to load the transactional data of claims to dimensional and fact tables in their EDW. The transactional data is being sent in a flat text file, delimited by "|". Pipe
- The file contains the claims transactional data (claims_transaction.txt), which has to be loaded to both the dimensional table (TBL_MEMBER) and the fact table (TBL_CLAIM) in a single mapping with a single pipeline.
- The transaction file contains Claim_id ,Member_id ,Member_name and Claim_amount. ④ columns
- Member_id and Member_name have to be loaded to TBL_MEMBER. ② columns
- Claim_id, Member_id and claim_amount needs to be loaded to TBL_CLAIM. ③ colums
- The EDW data load process has mandated to track the number of records loaded for each target table using an audit table (TBL_AUDIT).

think, we Also have to specify PK for source table. check videos.

Point to note

1. There are primary / foreign key relationship exists between fact and dimensional table ← are we supposed to join any tables using joiner transformation?
2. Claim id is unique, no duplicates expected.
3. Multiple claims can be raised by the same member id.
4. There is a chance of member name being updated in the same file.
5. Only number of insert records needs to be loaded in the Audit table for the latest run.i.e update rows should not be counted.
6. All are SCD- Type 1 tables with no historical tracking

Claims Transactions – Input file

Claims_transaction.txt

DDLs

Claims_transaction_DDL_scripts.txt

Q! Me: Is it like where all data in target tables get converted into script & save in this text file?!

Final Loaded data

1. Table containing the Members data – TBL_MEMBER

INT varchar(20)

MEMBER_ID	MEMBER_NAME
M1	Member One
M2	Member 2
M3	Member 3
M4	Member 4

Me: Can use sorter transformation. Sort by Member_ID.

2. Table containing the Claims data – TBL_CLAIM

varchar(10) varchar(10) INT

CLAIM_ID	MEMBER_ID	CLAIM_AMOUNT
C1	M1	1000
C2	M2	2000
C3	M3	3000
C4	M4	4000
C5	M1	5000

Me Can use Sorter transformation. Sort by Claim_ID.

3. Table containing the Audit data – TBL_AUDIT

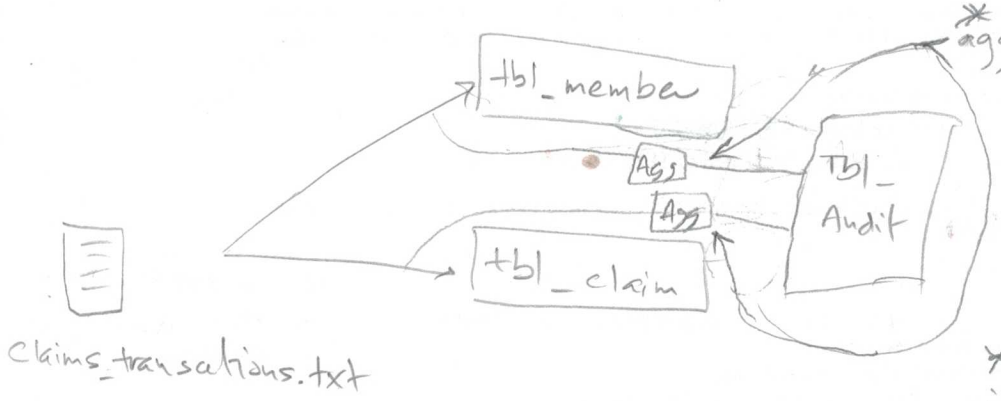
varchar25 25 25 25 25 INT

FOLDER_NAME	WORKFLOW_NAME	SESSION_NAME	MAPPING_NAME	TABLE_NAME	ROW_COUNT
Training	wf_m_specialist	s_m_specialist	m_specialist	TBL_CLAIM	5
Training	wf_m_specialist	s_m_specialist	m_specialist	TBL_MEMBER	4

Informatica built-in variable (see link shared by Michael on Discord)

- update strategy + insert ⇒ keeps inserting all source records into target every time wf is executed

- update strategy + update ⇒ updates target first time wf is executed after updating source. Subsequent wf executions will not affect Target if source is not updated



* aggregator transformations takes input from inserts only to member & claim tables (disregard update inserts)

* only first record is insert. all subsequent records are update insert

Vid (140) mentions → Versioning
→ Flagging
→ Effective Date ← I don't this is explained in the vidkos though

Informatica SCD Type 2 Case Study Assessment

- A company wants to maintain the location changes of its employees using SCD type 2 dimension method.
- The transactional data is being sent in a flat text file, with the comma delimited file.
- The file contains the employee information (Employee.csv), which has to be loaded to the dimensional table (D_Employee) and whenever there is a change in an employee's location, the history and current data need to be tracked in the following ways,

1) By flagging the records. Latest record will be set with current flag 'Y' and the older record with the current flag 'N'.

history flag ?!

2) By giving versions. Latest record will have the max version ie. Start with the version 1 and if location changes then increase the version number.

3) By End dating the record. When it comes for the first time it has the start date as SYSDATE and End date as '12/31/9999'. If location changes then the older record will be end dated with the SYSDATE -1 and the latest record will have the start date as SYSDATE and End date as '12/31/9999'.

Point to Note

1) Employee id is the unique key from the file
2) Create a sequence for EMP_Key (Primary key) each time a record is inserted into D_Employee table.

SR_NO ≡ EMP_Key ?! ← data type: number
?! Me: ?!

Employee – Input file

Employee.csv

in both cases, Target Emp_ID is PK

LKP Emp_key → flag & effective date is just for update strategy

Emp_Id

Version / Emp_key

DDLs

D_Employee.txt

??

o_new_rec
o_update_rec

Final Loaded data

1. D_Employee table with the Historical changes.

3 number | 3 number | 15 string | 30 string | 15 string | 15 string | 10 string | 1 string | 3 number | date | date

EMP_Key	EMP_Id	EMP_Name	Address	City	State	Country	Current Flag	Current_Version	Start_Dt	End_DT
1	100	John	3120 Bloomfield Square	Auburn Hills	MI	USA	Y	1	6/3/2022	12/31/9999
2	101	Jerry	105 Bloomfield Hills	Rochester Hills	NY	USA	Y	1	6/3/2022	12/31/9999
3	102	Joy	546 Barclays	Troy	MI	USA	Y	1	6/3/2022	12/31/9999
4	103	Tina	4967 Clifton Hill	Tonawanda	NY	USA	Y	1	6/3/2022	12/31/9999
5	104	Bruce	849 Young St	Buffalo	ON	Canada	Y	1	6/3/2022	12/31/9999
6	105	Tom	230 Williams St	Palo Alto	CA	USA	N	1	6/3/2022	6/5/2022
7	106	Nick	4532 Elizabeth Rd	Buffalo	ON	Canada	Y	1	6/3/2022	12/31/9999
8	107	William	5601 Main St	Williamsville	NY	USA	Y	1	6/3/2022	12/31/9999
9	105	Tom	400 Main St	Detroit	MI	USA	Y	2	6/6/2022	12/31/9999

Tom changed his address

2. Refer below scenarios attached to load D_Employee table.

SCD Type2 Scenario.xlsx

Note:

Try changing the data by creating more versions and check how the history is maintained.

employee.csv ← employee table ← source table

1. current flag (Y/N)
2. current version (1/2, etc) ← highest version number is the most current record
3. start_date / end_date

target table contains