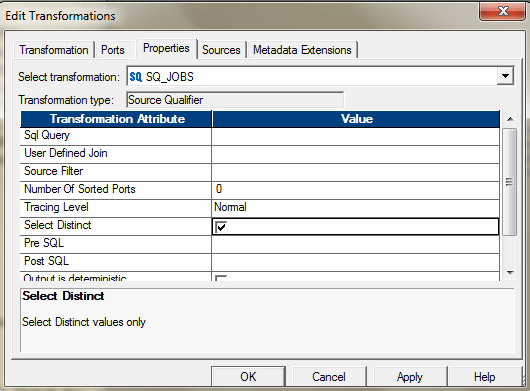
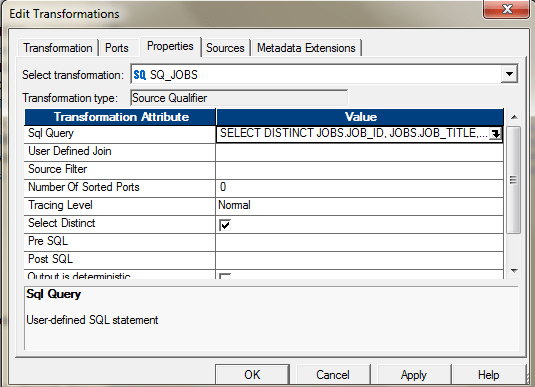
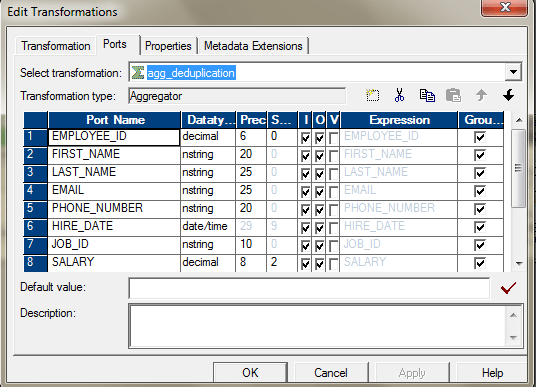
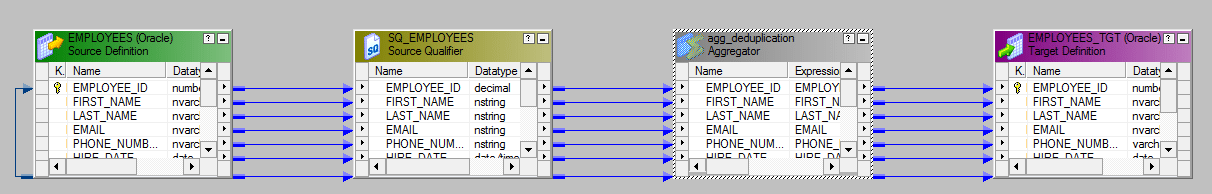
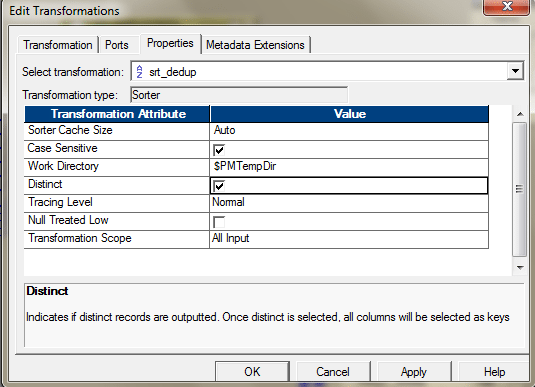
**Informatica Interview Questions (Scenario-Based):**

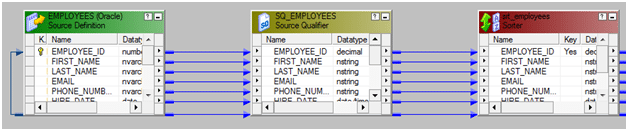
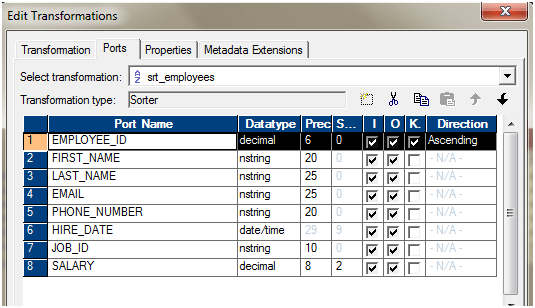
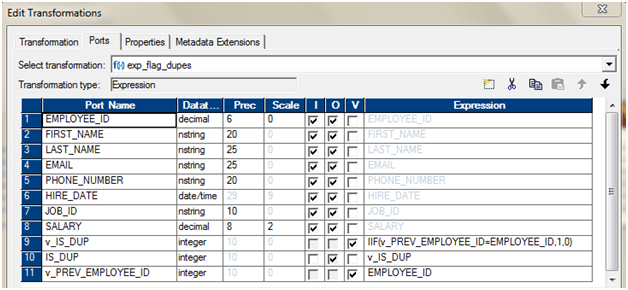
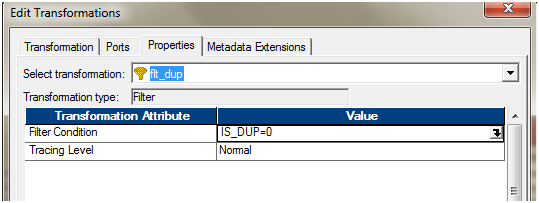
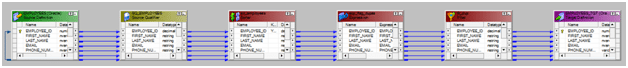
**1. Differentiate between Source Qualifier and Filter Transformation?**

|  |  |
| --- | --- |
| **Source Qualifier vs Filter Transformation** | |
| **Source Qualifier Transformation** | **Filter Transformation** |
| 1. It filters rows while reading the data from a source. | 1. It filters rows from within a mapped data. |
| 2. Can filter rows only from relational sources. | 2. Can filter rows from any type of source system. |
| 3. It limits the row sets extracted from a  source. | 3. It limits the row set sent to a target. |
| 4. It enhances performance by minimizing the number of rows used in mapping. | 4. It is added close to the source to filter out the unwanted data early and maximize performance. |
| 5. In this, filter condition uses the standard SQL to execute in the database. | 5. It defines a condition using any statement or transformation function to get either TRUE or FALSE. |

**2. How do you remove Duplicate records in Informatica? And how many ways are there to do it?**

There are several ways to remove duplicates.

1. If the source is DBMS, you can use the property in Source Qualifier to select the distinct records.Or you can also use the SQL Override to perform the same.
2. You can use, Aggregator and select all the ports as key to get the distinct values. After you pass all the required ports to the Aggregator, select all those ports , those you need to select for de-duplication. If you want to find the duplicates based on the entire columns, select all the ports as group by key.  
   The Mapping will look like this.  
   
3. You can use Sorter and use the Sort Distinct Property to get the distinct values. Configure the sorter in the following way to enable this.  
   
4. You can use, Expression and Filter transformation, to identify and remove duplicate if your data is sorted. If your data is not sorted, then, you may first use a sorter to sort the data and then apply this logic:

* Bring the source into the Mapping designer.
* Let’s assume the data is not sorted. We are using a sorter to sort the data. The Key for sorting would be Employee\_ID.  
  Configure the Sorter as mentioned below.  
  
* Use one expression transformation to flag the duplicates. We will use the variable ports to identify the duplicate entries, based on Employee\_ID.  
  
* Use a filter transformation, only to pass IS\_DUP = 0. As from the previous expression transformation, we will have IS\_DUP =0 attached to only records, which are unique. If IS\_DUP > 0, that means, those are duplicate entries.  
  
* Add the ports to the target. The entire mapping should look like this.  
  

v. When you change the property of the Lookup transformation to use the Dynamic Cache, a new port is added to the transformation. NewLookupRow.

The Dynamic Cache can update the cache, as and when it is reading the data.

If the source has duplicate records, you can also use Dynamic Lookup cache and then router to select only the distinct one.

**3. What are the differences between Source Qualifier and Joiner Transformation?**

The Source Qualifier can join data originating from the same source database. We can join two or more tables with primary key-foreign key relationships by linking the sources to one Source Qualifier transformation.

If we have a requirement to join the mid-stream or the sources are heterogeneous, then we will have to use the Joiner transformation to join the data.

**4. Differentiate between joiner and Lookup Transformation.**

Below are the differences between lookup and joiner transformation:

* In lookup we can override the query but in joiner we cannot.
* In lookup we can provide different types of operators like – “>,<,>=,<=,!=” but, in joiner only “= “ (equal to )operator is available.
* In lookup we can restrict the number of rows while reading the relational table using lookup override but, in joiner we cannot restrict the number of rows while reading.
* In joiner we can join the tables based on- Normal Join, Master Outer, Detail Outer and Full Outer Join but, in lookup this facility is not available .Lookup behaves like Left Outer Join of database.

**5. What is meant by Lookup Transformation? Explain the types of Lookup transformation.**

Lookup transformation in a mapping is used to look up data in a flat file, relational table, view, or synonym. We can also create a lookup definition from a source qualifier.

We have the following types of Lookup.

* **Relational or flat file lookup.**To perform a lookup on a flat file or a relational table.
* **Pipeline lookup.**To perform a lookup on application sources such as JMS or MSMQ.
* **Connected or unconnected lookup**.
  + A connected Lookup transformation receives source data, performs a lookup, and returns data to the pipeline.
  + An unconnected Lookup transformation is not connected to a source or target. A transformation in the pipeline calls the Lookup transformation with a: LKP expression. The unconnected Lookup transformation returns one column to the calling transformation.
* **Cached or un-cached lookup.**We can configure the lookup transformation to Cache the lookup data or directly query the lookup source every time the lookup is invoked. If the Lookup source is Flat file, the lookup is always cached.

**6. How can you increase the performance in joiner transformation?**

Below are the ways in which you can improve the performance of Joiner Transformation.

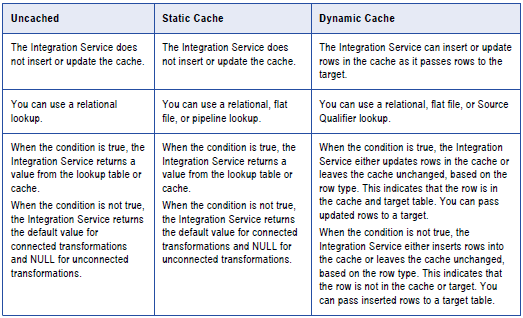
* Perform joins in a database when possible.  
  In some cases, this is not possible, such as joining tables from two different databases or flat file systems. To perform a join in a database, we can use the following options:  
  Create and Use a pre-session stored procedure to join the tables in a database.  
  Use the Source Qualifier transformation to perform the join.
* Join sorted data when possible
* For an unsorted Joiner transformation, designate the source with fewer rows as the master source.
* For a sorted Joiner transformation, designate the source with fewer duplicate key values as the master source.

**7. What are the types of Caches in lookup? Explain them.**

Based on the configurations done at lookup transformation/Session Property level, we can have following types of Lookup Caches.

* **Un- cached lookup**– Here, the lookup transformation does not create the cache. For each record, it goes to the lookup Source, performs the lookup and returns value. So for 10K rows, it will go the Lookup source 10K times to get the related values.
* **Cached Lookup**– In order to reduce the to and fro communication with the Lookup Source and Informatica Server, we can configure the lookup transformation to create the cache. In this way, the entire data from the Lookup Source is cached and all lookups are performed against the Caches.

Based on the types of the Caches configured, we can have two types of caches, Static and Dynamic.

The Integration Service performs differently based on the type of lookup cache that is configured. The following table compares Lookup transformations with an uncached lookup, a static cache, and a dynamic cache:  


**Persistent Cache**

By default, the Lookup caches are deleted post successful completion of the respective sessions but, we can configure to preserve the caches, to reuse it next time.

**Shared Cache**

We can share the lookup cache between multiple transformations. We can share an unnamed cache between transformations in the same mapping. We can share a named cache between transformations in the same or different mappings.

**8. How do you update the records with or without using Update Strategy?**

We can use the session configurations to update the records. We can have several options for handling database operations such as insert, update, delete.

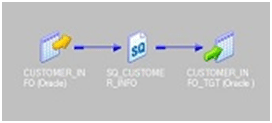
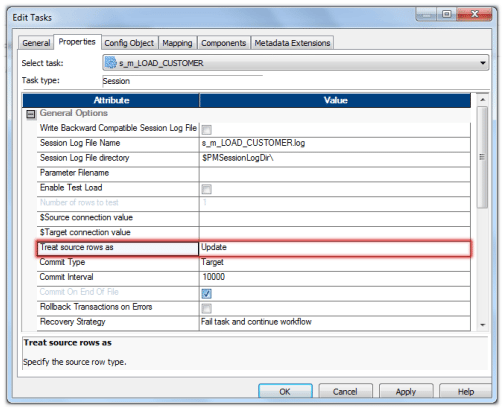
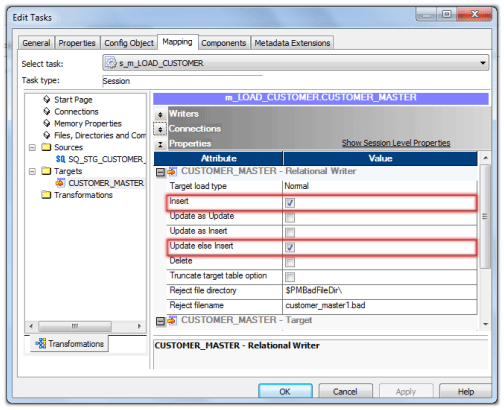
During session configuration, you can select a single database operation for all rows using the Treat Source Rows As setting from the ‘Properties’ tab of the session.

* **Insert: –** Treat all rows as inserts.
* **Delete: –** Treat all rows as deletes.
* **Update: –** Treat all rows as updates.
* **Data Driven :-** Integration Service follows instructions coded into Update Strategy flag rows for insert, delete, update, or reject.

Once determined how to treat all rows in the session, we can also set options for individual rows, which gives additional control over how each rows behaves. We need to define these options in the Transformations view on mapping tab of the session properties.

* **Insert: –** Select this option to insert a row into a target table.
* **Delete: –** Select this option to delete a row from a table.
* **Update :-** You have the following options in this situation:
  + Update as Update: – Update each row flagged for update if it exists in the target table.
  + Update as Insert: – Insert each row flagged for update.
  + Update else Insert: – Update the row if it exists. Otherwise, insert it.
* **Truncate Table: –** Select this option to truncate the target table before loading data.

Steps:

1. Design the mapping just like an ‘INSERT’ only mapping, without Lookup, Update Strategy Transformation.
2. First set Treat Source Rows As property as shown in below image.  
   
3. Next, set the properties for the target table as shown below. Choose the properties Insert and Update else Insert.  
   

These options will make the session as Update and Insert records without using Update Strategy in Target Table.

When we need to update a huge table with few records and less inserts, we can use this solution to improve the session performance.

The solutions for such situations is not to use Lookup Transformation and Update Strategy to insert and update records.

The Lookup Transformation may not perform better as the lookup table size increases and it also degrades the performance.

**9. Why update strategy and union transformations are Active? Explain with examples.**

1. The Update Strategy changes the row types. It can assign the row types based on the expression created to evaluate the rows. Like **IIF (ISNULL (CUST\_DIM\_KEY), DD\_INSERT, DD\_UPDATE).** This expression, changes the row types to Insert for which the CUST\_DIM\_KEY is NULL and to Update for which the CUST\_DIM\_KEY is not null.
2. The Update Strategy can reject the rows. Thereby with proper configuration, we can also filter out some rows. Hence, sometimes, the number of input rows, may not be equal to number of output rows.

Like **IIF (IISNULL (CUST\_DIM\_KEY), DD\_INSERT,**

**IIF (SRC\_CUST\_ID! =TGT\_CUST\_ID), DD\_UPDATE, DD\_REJECT))**

Here we are checking if CUST\_DIM\_KEY is not null then if SRC\_CUST\_ID is equal to the TGT\_CUST\_ID. If they are equal, then we do not take any action on those rows; they are getting rejected.

**Union Transformation**

In union transformation, though the total number of rows passing into the Union is the same as the total number of rows passing out of it, the positions of the rows are not preserved, i.e. row number 1 from input stream 1 might not be row number 1 in the output stream. Union does not even guarantee that the output is repeatable. Hence it is an Active Transformation.

**10. How do you load only null records into target? Explain through mapping flow.**

Let us say, this is our source

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cust\_id | Cust\_name | Cust\_amount | Cust\_Place | Cust\_zip |
| 101 | AD | 160 | KL | 700098 |
| 102 | BG | 170 | KJ | 560078 |
| NULL | NULL | 180 | KH | 780098 |

The target structure is also the same but, we have got two tables, one which will contain the NULL records and one which will contain non NULL records.

We can design the mapping as mentioned below.

**SQ –> EXP –> RTR –> TGT\_NULL/TGT\_NOT\_NULL**EXP – Expression transformation create an output port

[[](https://www.edureka.co/informatica)](https://www.edureka.co/informatica" \t "_blank)

**[Informatica Training & Certification](https://www.edureka.co/informatica" \t "_blank)**

* *[Instructor-led Sessions](https://www.edureka.co/informatica" \t "_blank)*
* *[Real-life Case Studies](https://www.edureka.co/informatica" \t "_blank)*
* *[Assignments](https://www.edureka.co/informatica" \t "_blank)*
* *[Lifetime Access](https://www.edureka.co/informatica" \t "_blank)*

[Explore Curriculum](https://www.edureka.co/informatica" \t "_blank)

O\_FLAG= IIF ( (ISNULL(cust\_id) OR ISNULL(cust\_name) OR ISNULL(cust\_amount) OR ISNULL(cust \_place) OR ISNULL(cust\_zip)), ‘NULL’,’NNULL’)  
***\*\* Assuming you need to redirect in case  any of value is null***

OR

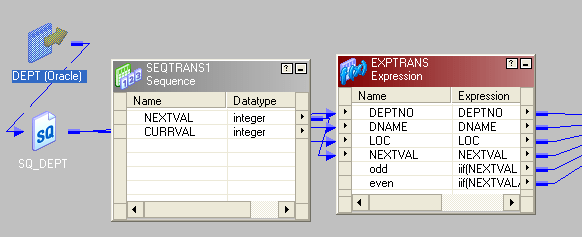
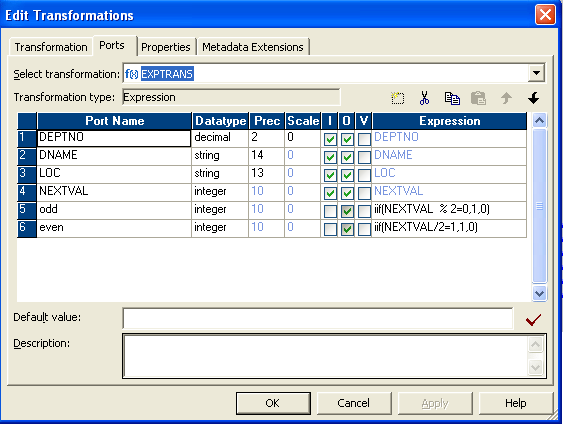
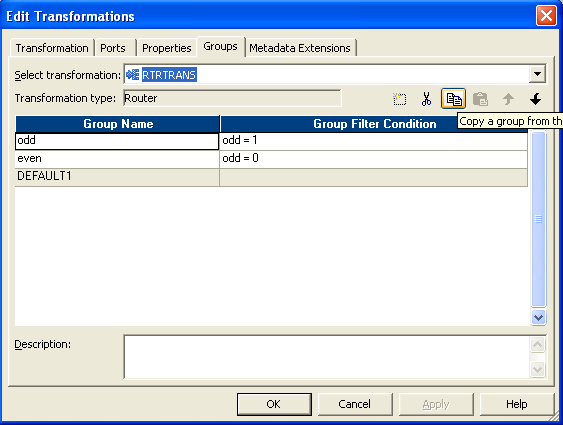
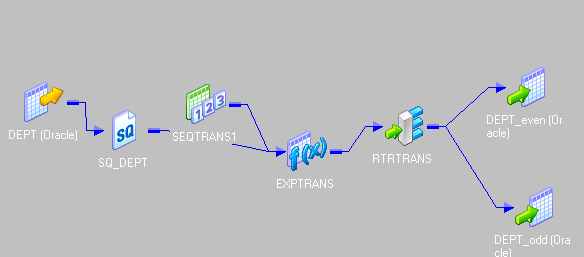
O\_FLAG= IIF ( (ISNULL(cust\_name) AND ISNULL(cust\_no) AND ISNULL(cust\_amount) AND ISNULL(cust \_place) AND ISNULL(cust\_zip)), ‘NULL’,’NNULL’)  
***\*\* Assuming you need to redirect in case all of  value is null***

RTR – Router transformation two groups

Group 1 connected to TGT\_NULL ( Expression O\_FLAG=’NULL’)  
Group 2 connected to TGT\_NOT\_NULL ( Expression O\_FLAG=’NNULL’)

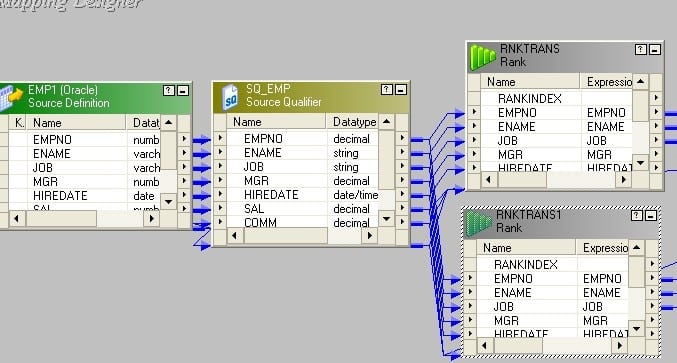
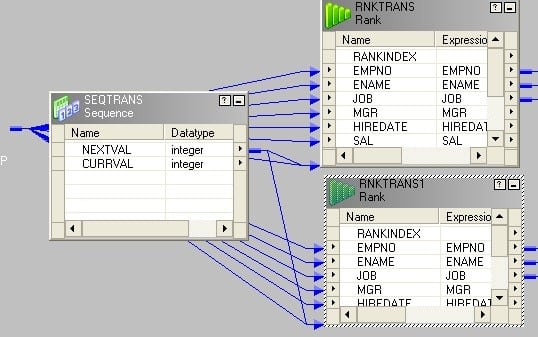
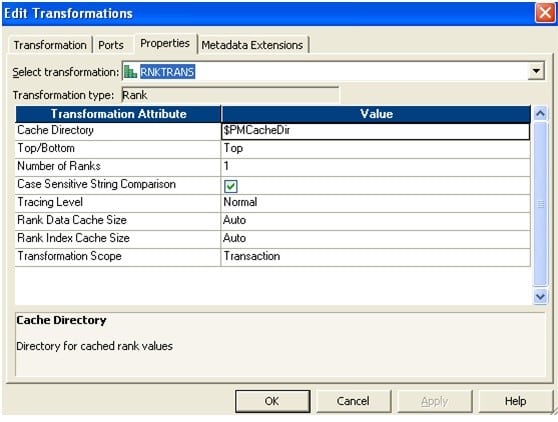
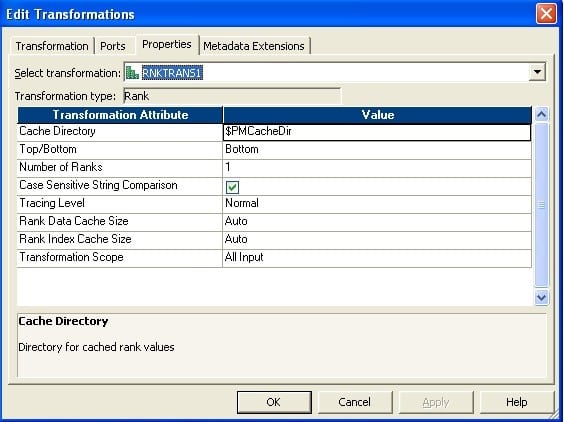
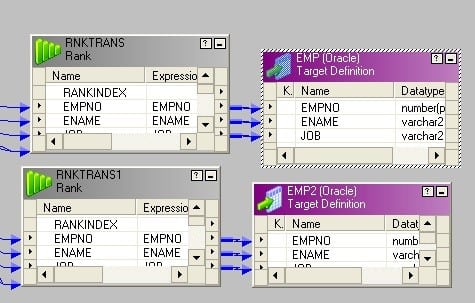
**11. How do you load alternate records into different tables through mapping flow?**

The idea is to add a sequence number to the records and then divide the record number by 2. If it is divisible, then move it to one target and if not then move it to other target.

1. Drag the source and connect to an expression transformation.
2. Add the next value of a sequence generator to expression transformation.  
   
3. In expression transformation make two port, one is “odd” and another “even”.
4. Write the expression as below  
   
5. Connect a router transformation to expression.
6. Make two group in router.
7. Give condition as below  
   
8. Then send the two group to different targets. This is the entire flow.  
   

**12. How do you load first and last records into target table? How many ways are there to do it? Explain through mapping flows.**

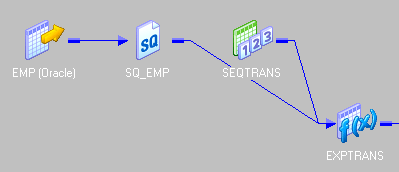
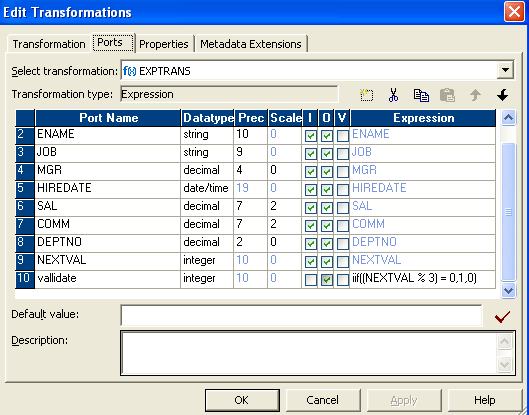
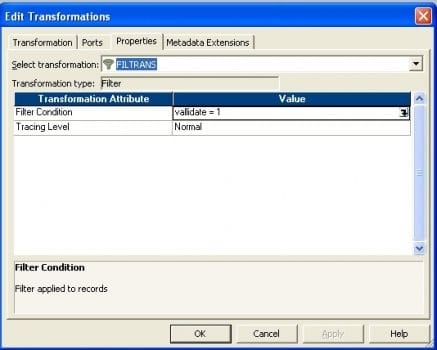
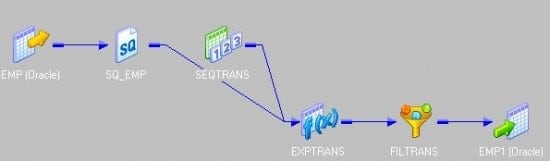
The idea behind this is to add a sequence number to the records and then take the Top 1 rank and Bottom 1 Rank from the records.

1. Drag and drop ports from source qualifier to two rank transformations.  
   
2. Create a reusable sequence generator having start value 1 and connect the next value to both rank transformations.  
   
3. Set rank properties as follows. The newly added sequence port should be chosen as Rank Port. No need to select any port as Group by Port.Rank – 1  
   
4. Rank – 2  
   
5. Make two instances of the target.  
   Connect the output port to target.  
   

**13. I have 100 records in source table, but I want to load 1, 5,10,15,20…..100 into target table. How can I do this? Explain in detailed mapping flow.**

This is applicable for any n= 2, 3,4,5,6… For our example, n = 5. We can apply the same logic for any n.

The idea behind this is to add a sequence number to the records and divide the sequence number by n (for this case, it is 5). If completely divisible, i.e. no remainder, then send them to one target else, send them to the other one.

1. Connect an expression transformation after source qualifier.
2. Add the next value port of sequence generator to expression transformation.  
   
3. In expression create a new port (validate) and write the expression as in the picture below.  
   
4. Connect a filter transformation to expression and write the condition in property as given in the picture below.  
   
5. Finally connect to target.  
   

**14. How do you load unique records into one target table and duplicate records into  a different target table?**

**Source Table:**

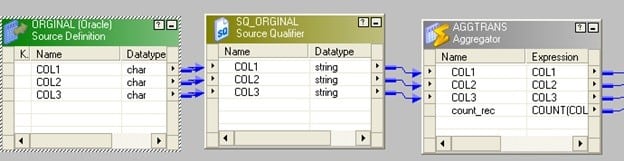
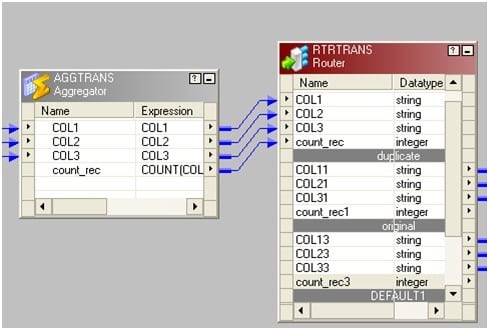
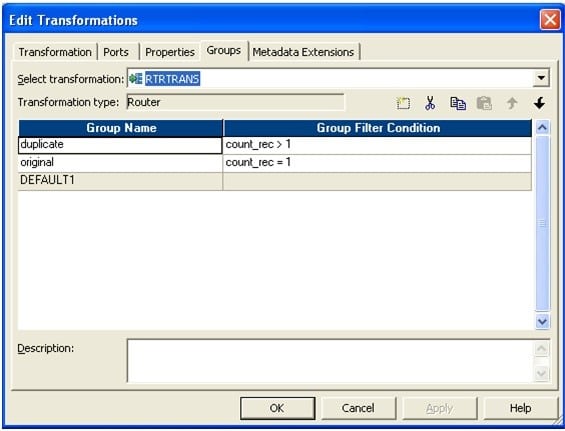
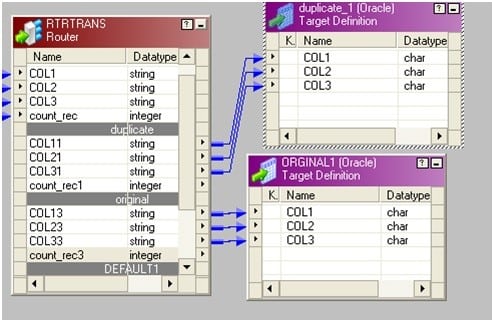
|  |  |  |
| --- | --- | --- |
| **COL1** | **COL2** | **COL3** |
| a | b | c |
| x | y | z |
| a | b | c |
| r | f | u |
| a | b | c |
| v | f | r |
| v | f | r |

**Target Table 1**: Table containing all the unique rows

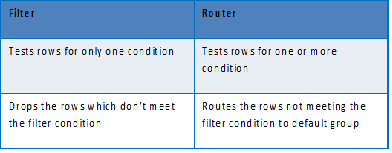
|  |  |  |
| --- | --- | --- |
| **COL1** | **COL2** | **COL3** |
| a | b | c |
| x | y | z |
| r | f | u |
| v | f | r |

**Target Table 2:** Table containing all the duplicate rows

|  |  |  |
| --- | --- | --- |
| **COL1** | **COL2** | **COL3** |
| a | b | c |
| a | b | c |
| v | f | r |

1. Drag the source to mapping and connect it to an aggregator transformation.  
   
2. In aggregator transformation, group by the key column and add a new port. Call it count\_rec to count  the key column.
3. Connect  a router to the aggregator from the previous step. In router make two groups: one named “original” and another as “duplicate”.  
   In original write count\_rec=1 and in duplicate write count\_rec>1.  
   The picture below depicts the group name and the filter conditions.  
   Connect two groups to corresponding target tables.  
   

**15. Differentiate between Router and Filter Transformation?**



**16. I have two different source structure tables, but I want to load into single target table? How do I go about it? Explain in detail through mapping flow.**

* We can use joiner, if we want to join the data sources. Use a joiner and use the matching column to join the tables.
* We can also use a Union transformation, if the tables have some common columns and we need to join the data vertically. Create one union transformation add the matching ports form the two sources, to two different input groups and send the output group to the target.  
  The basic idea here is to use, either Joiner or Union transformation, to move the data from two sources to a single target. Based on the requirement, we may decide, which one should be used.

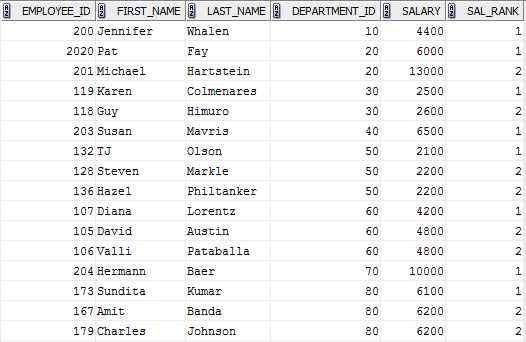
**17. How do you load more than 1 Max Sal in each Department through Informatica or write sql query in oracle?**

SQL query:

You can use this kind of query to fetch more than 1 Max salary for each**department.**

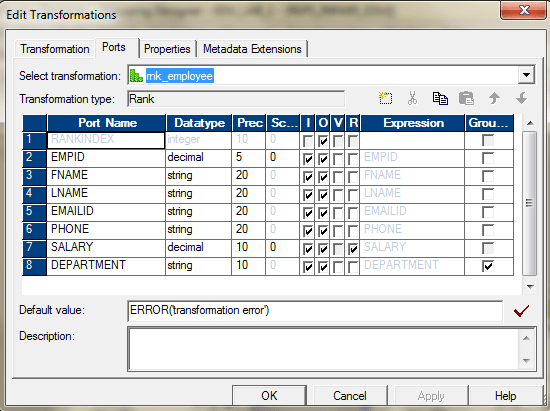
SELECT \* FROM (

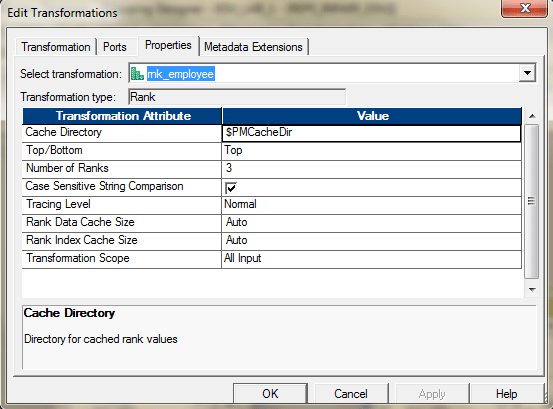
SELECT EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, DEPARTMENT\_ID, SALARY, RANK () OVER (PARTITION BY DEPARTMENT\_ID ORDER BY SALARY) SAL\_RANK FROM EMPLOYEES)

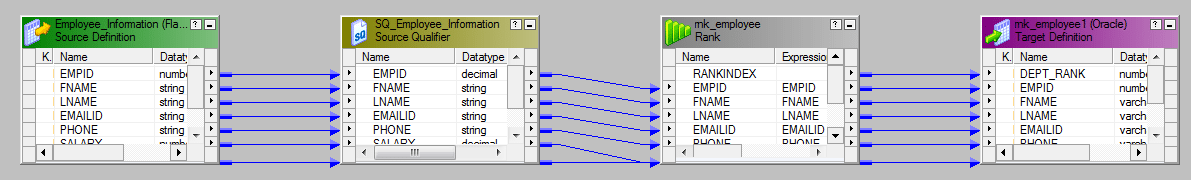
WHERE SAL\_RANK <= 2 ****

**Informatica Approach:**

We can use the Rank transformation to achieve this.

Use Department\_ID as the group key.  


In the properties tab, select Top, 3.  


The entire mapping should look like this.  


This will give us the top 3 employees earning maximum salary in their respective departments.

**18. How do you convert single row from source into three rows into target?**

We can use Normalizer transformation for this. If we do not want to use Normalizer, then there is one alternate way for this.

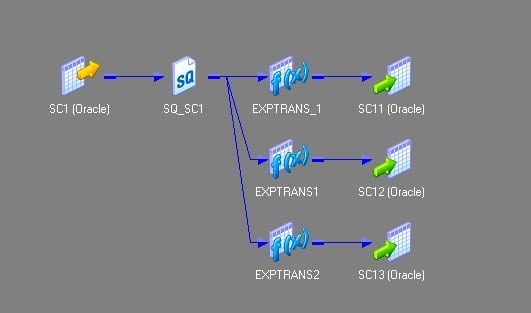
We have a source table containing 3 columns: Col1, Col2 and Col3. There is only 1 row in the table as follows:

|  |  |  |
| --- | --- | --- |
| **Col1** | **Col2** | **Col3** |
| a | b | C |

There is target table contains only 1 column Col. Design a mapping so that the target table contains 3 rows as follows:

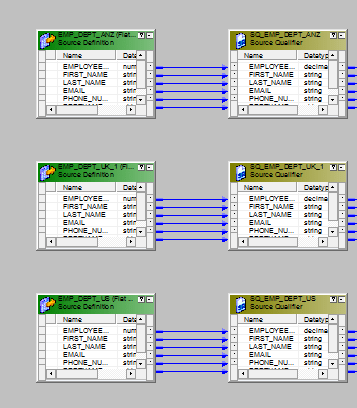
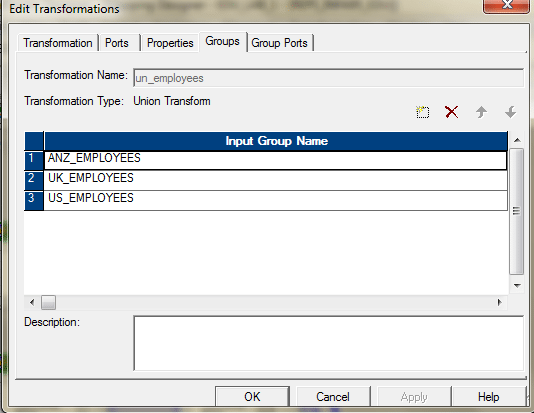
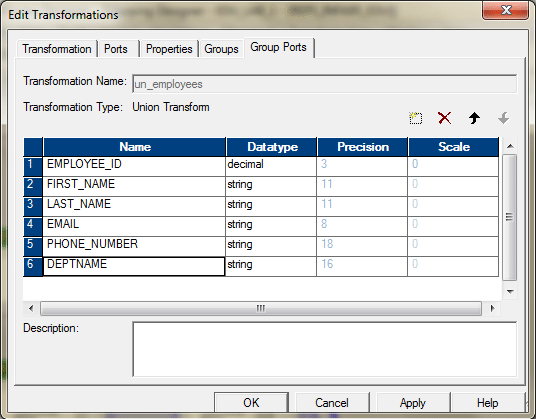
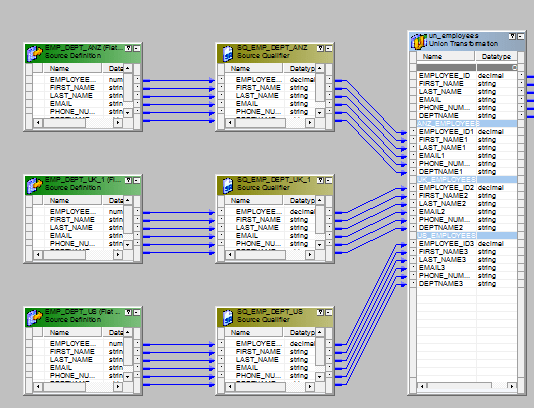
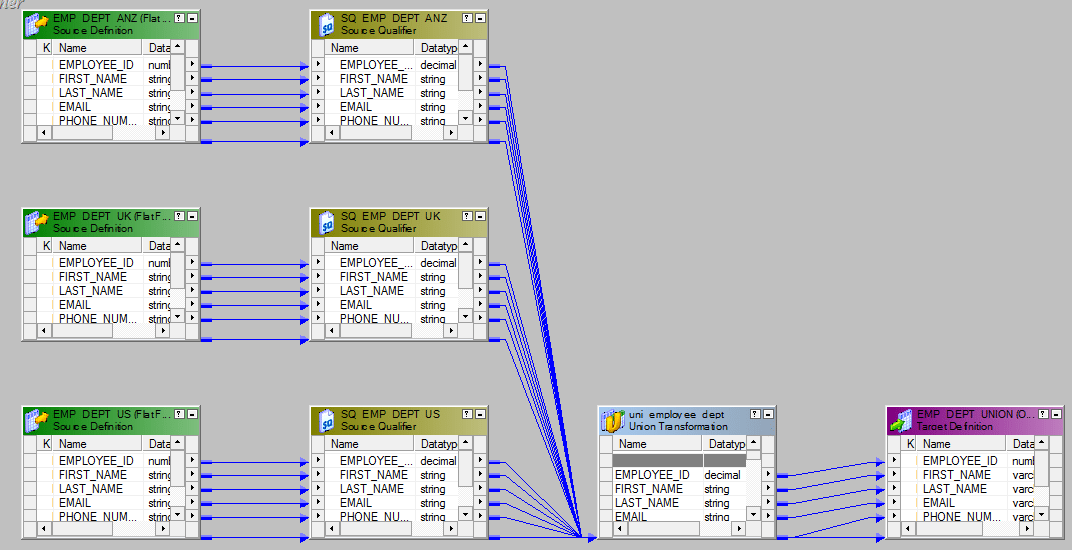
|  |
| --- |
| **Col** |
| a |
| b |
| c |

1. Create 3 expression transformations exp\_1,exp\_2 and exp\_3 with 1 port each.
2. Connect col1 from Source Qualifier to port in exp\_1.
3. Connect col2 from Source Qualifier to port in exp\_2.
4. Connect col3 from source qualifier to port in exp\_3.
5. Make 3 instances of the target. Connect port from exp\_1 to target\_1.
6. Connect port from exp\_2 to target\_2 and connect port from exp\_3 to target\_3.



**19. I have three same source structure tables. But, I want to load into single target table. How do I do this? Explain in detail through mapping flow.**

We will have to use the Union Transformation here. Union Transformation is a multiple input group transformation and it has only one output group.

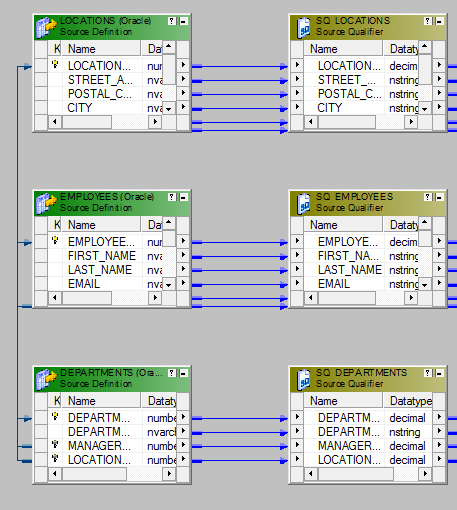
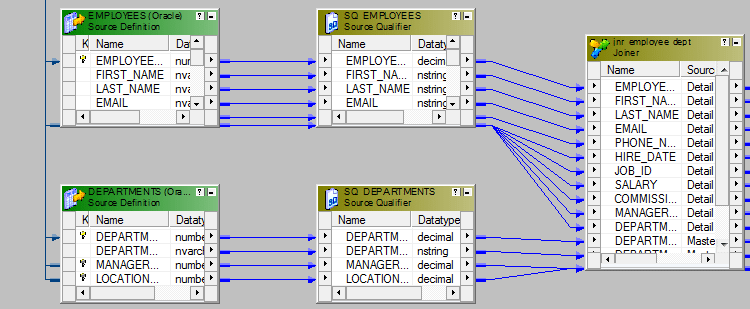
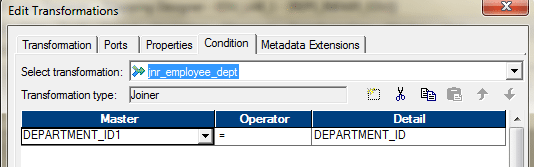
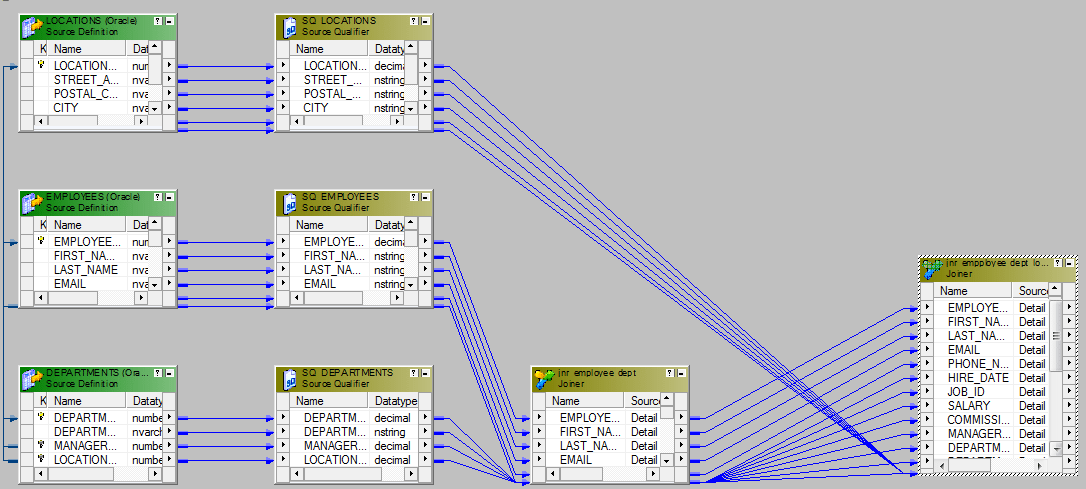
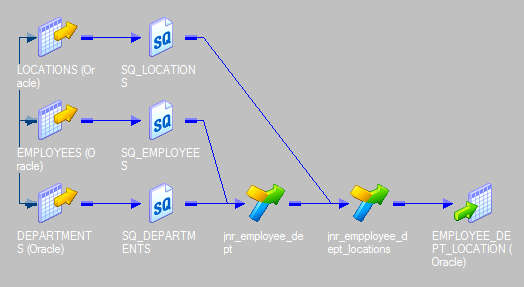
1. Drag all the sources in to the mapping designer.  
   
2. Add one union transformation and configure it as follows.Group Tab.  
   Group Ports Tab.  
   
3. Connect the sources with the three input groups of the union transformation.  
   
4. Send the output to the target or via a expression transformation to the target.The entire mapping should look like this.  
   

**20. How to join three sources using joiner? Explain though mapping flow.**

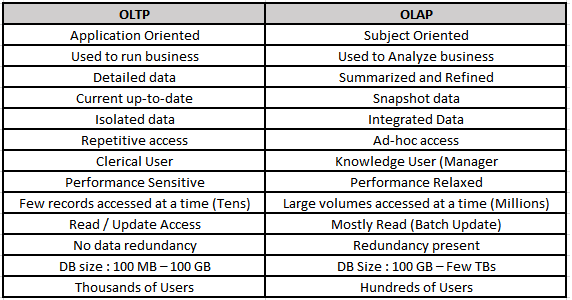
We cannot join more than two sources using a single joiner. To join three sources, we need to have two joiner transformations.

Let’s say, we want to join three tables – Employees, Departments and Locations – using Joiner. We will need two joiners. Joiner-1 will join, Employees and Departments and Joiner-2 will join, the output from the Joiner-1 and Locations table.

Here are the steps.

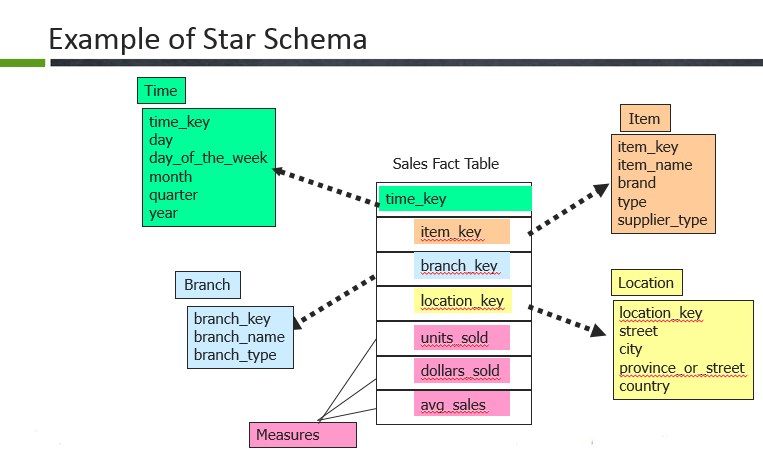
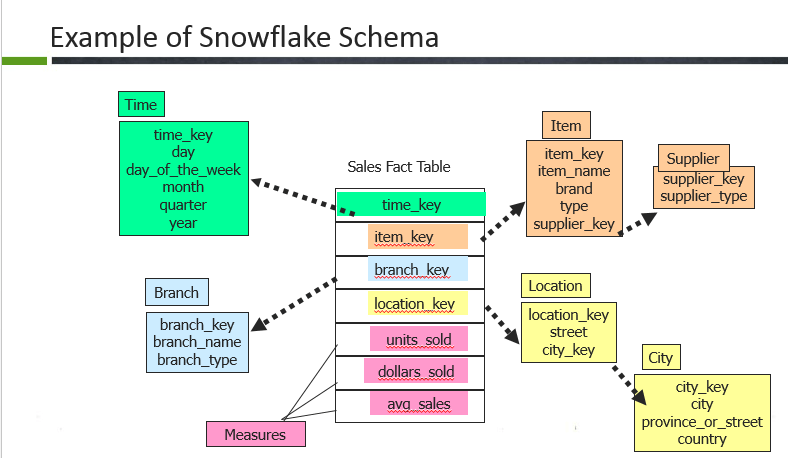
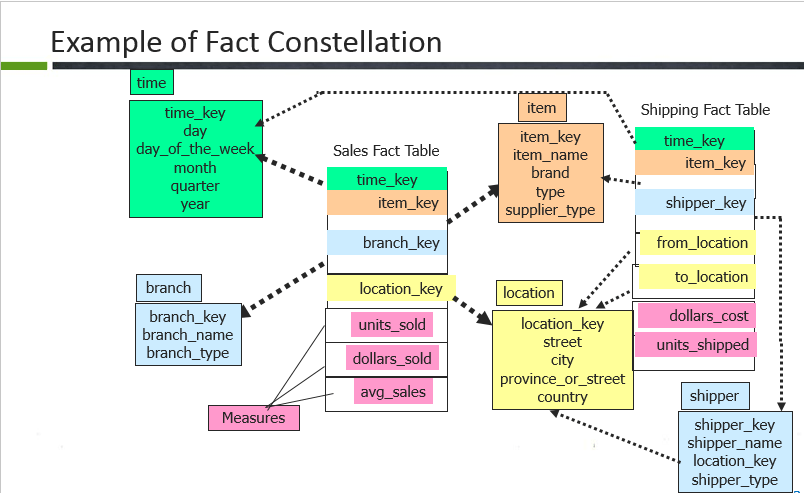
1. Bring three sources into the mapping designer.  
   
2. Create the Joiner -1 to join Employees and Departments using Department\_ID.  
     
   
3. Create the next joiner, Joiner-2. Take the Output from Joiner-1 and ports from Locations Table and bring them to Joiner-2. Join these two data sources using Location\_ID.  
   
4. The last step is to send the required ports from the Joiner-2 to the target or via an expression transformation to the target table.  
   

**21. What are the differences between OLTP and OLAP?**



**22. What are the types of Schemas we have in data warehouse and what are the difference between them?**

There are three different data models that exist.

1. Star schemaHere, the Sales fact table is a fact table and the surrogate keys of each dimension table are referred here through foreign keys. Example: time key, item key, branch key, location key. The fact table is surrounded by the dimension tables such as Branch, Location, Time and item. In the fact table there are dimension keys such as time\_key, item\_key, branch\_key and location\_keys and measures are untis\_sold, dollars sold and average sales.Usually, fact table consists of more rows compared to dimensions because it contains all the primary keys of the dimension along with its own measures.
2. Snowflake schema  
     
   In snowflake, the fact table is surrounded by dimension tables and the dimension tables are also normalized to form the hierarchy. So in this example, the dimension tables such as location, item are normalized further into smaller dimensions forming a hierarchy.
3. Fact constellations  
     
   In fact constellation, there are many fact tables sharing the same dimension tables. This examples illustrates a fact constellation in which the fact tables sales and shipping are sharing the dimension tables time, branch, item.

**23. What is Dimensional Table? Explain the different dimensions.**

Dimension table is the one that describes business entities of an enterprise, represented as hierarchical, categorical information such as time, departments, locations, products etc.

**Types of dimensions in data warehouse**

A dimension table consists of the attributes about the facts. Dimensions store the textual descriptions of the business. Without the dimensions, we cannot measure the facts. The different types of dimension tables are explained in detail below.

* **Conformed Dimension:**  
  Conformed dimensions mean the exact same thing with every possible fact table to which they are joined.  
  Eg: The date dimension table connected to the sales facts is identical to the date dimension connected to the inventory facts.
* **Junk Dimension:**A junk dimension is a collection of random transactional codes flags and/or text attributes that are unrelated to any particular dimension. The junk dimension is simply a structure that provides a convenient place to store the junk attributes.  
  Eg: Assume that we have a gender dimension and marital status dimension. In the fact table we need to maintain two keys referring to these dimensions. Instead of that create a junk dimension which has all the combinations of gender and marital status (cross join gender and marital status table and create a junk table). Now we can maintain only one key in the fact table.
* **Degenerated Dimension:**A degenerate dimension is a dimension which is derived from the fact table and doesn’t have its own dimension table.  
  Eg: A transactional code in a fact table.
* **Role-playing dimension:**Dimensions which are often used for multiple purposes within the same database are called role-playing dimensions. For example, a date dimension can be used for “date of sale”, as well as “date of delivery”, or “date of hire”.

**24. What is Fact Table? Explain the different kinds of Facts.**

The centralized table in the star schema is called the Fact table. A Fact table typically contains two types of columns. Columns which contains the measure called facts and columns, which are foreign keys to the dimension tables. The Primary key of the fact table is usually the composite key that is made up of the foreign keys of the dimension tables.

**Data Warehousing and ETL Training**

**[INFORMATICA TRAINING & CERTIFICATION](https://www.edureka.co/informatica" \o "Informatica Training & Certification" \t "_blank)**

**[Informatica Training & Certification](https://www.edureka.co/informatica" \o "Informatica Training & Certification" \t "_blank)**

*[Reviews](https://www.edureka.co/informatica" \o "Informatica Training & Certification" \t "_blank)*

**[5](https://www.edureka.co/informatica" \o "Informatica Training & Certification" \t "_blank)**[(14149)](https://www.edureka.co/informatica" \o "Informatica Training & Certification" \t "_blank)

**[TALEND FOR DATA INTEGRATION AND BIG DATA](https://www.edureka.co/talend-for-big-data" \o "Talend for Data Integration and Big data" \t "_blank)**

**[Talend for Data Integration and Big data](https://www.edureka.co/talend-for-big-data" \o "Talend for Data Integration and Big data" \t "_blank)**

*[Reviews](https://www.edureka.co/talend-for-big-data" \o "Talend for Data Integration and Big data" \t "_blank)*

**[5](https://www.edureka.co/talend-for-big-data" \o "Talend for Data Integration and Big data" \t "_blank)**[(2071)](https://www.edureka.co/talend-for-big-data" \o "Talend for Data Integration and Big data" \t "_blank)

**[DATA WAREHOUSING AND BI CERTIFICATION TRAINING](https://www.edureka.co/data-warehousing-and-bi" \o "Data Warehousing and BI Certification Training" \t "_blank)**

**[Data Warehousing and BI Certification Training](https://www.edureka.co/data-warehousing-and-bi" \o "Data Warehousing and BI Certification Training" \t "_blank)**

*[Reviews](https://www.edureka.co/data-warehousing-and-bi" \o "Data Warehousing and BI Certification Training" \t "_blank)*

**[5](https://www.edureka.co/data-warehousing-and-bi" \o "Data Warehousing and BI Certification Training" \t "_blank)**[(4024)](https://www.edureka.co/data-warehousing-and-bi" \o "Data Warehousing and BI Certification Training" \t "_blank)

**[DATA WAREHOUSING CERTIFICATION TRAINING](https://www.edureka.co/datawarehousing-self-paced" \o "Data Warehousing Certification Training" \t "_blank)**

**[Data Warehousing Certification Training](https://www.edureka.co/datawarehousing-self-paced" \o "Data Warehousing Certification Training" \t "_blank)**

*[Reviews](https://www.edureka.co/datawarehousing-self-paced" \o "Data Warehousing Certification Training" \t "_blank)*

**[4](https://www.edureka.co/datawarehousing-self-paced" \o "Data Warehousing Certification Training" \t "_blank)**[(3743)](https://www.edureka.co/datawarehousing-self-paced" \o "Data Warehousing Certification Training" \t "_blank)

Next

**Types of Facts in Data Warehouse**

A fact table is the one which consists of the measurements, metrics or facts of business process. These measurable facts are used to know the business value and to forecast the future business. The different types of facts are explained in detail below.

* **Additive:**  
  Additive facts are facts that can be summed up through all of the dimensions in the fact table. A sales fact is a good example for additive fact.
* **Semi-Additive:**  
  Semi-additive facts are facts that can be summed up for some of the dimensions in the fact table, but not the others.  
  Eg: Daily balances fact can be summed up through the customers dimension but not through the time dimension.
* **Non-Additive:**  
  Non-additive facts are facts that cannot be summed up for any of the dimensions present in the fact table.  
  Eg: Facts which have percentages, ratios calculated.

**Factless Fact Table:**

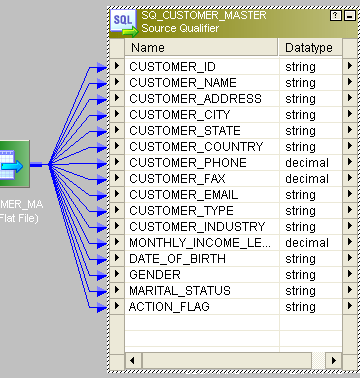
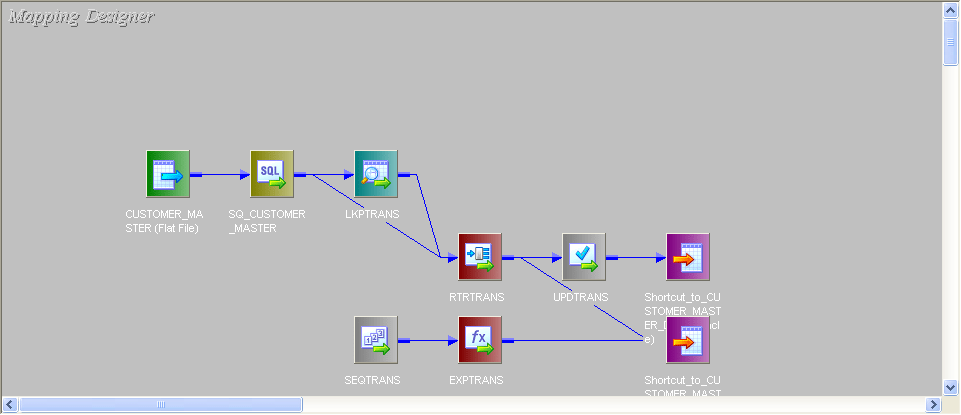
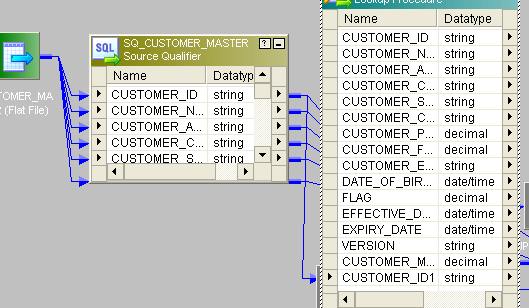
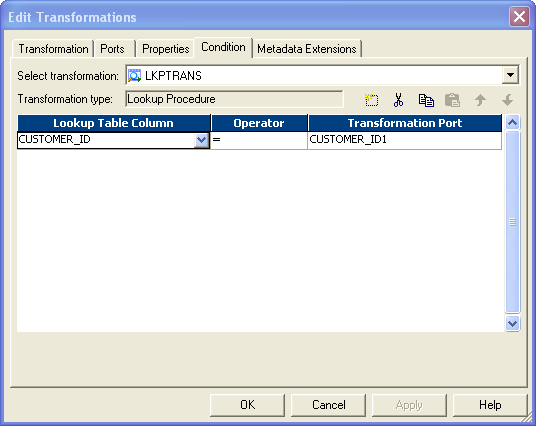
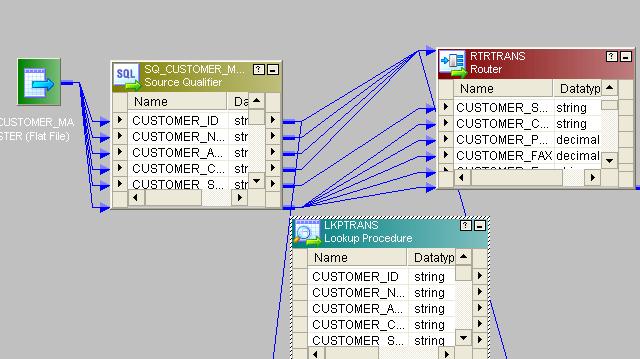
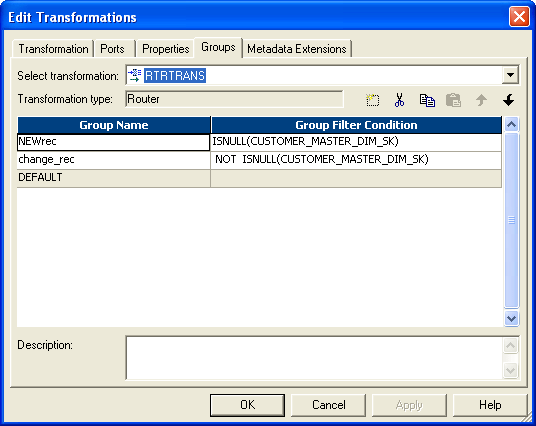
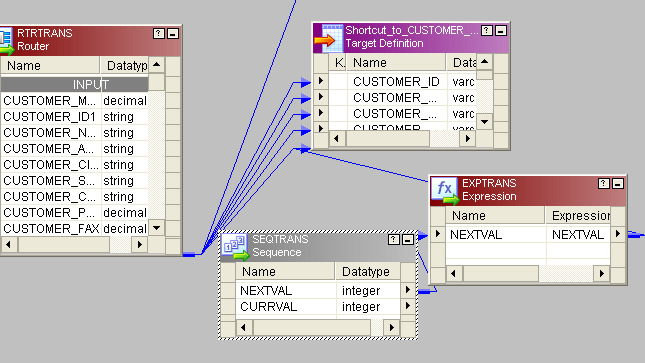
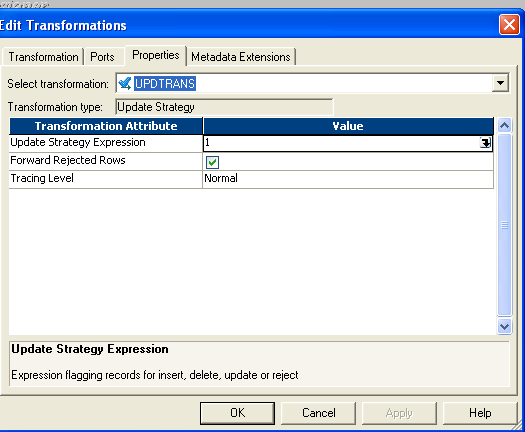
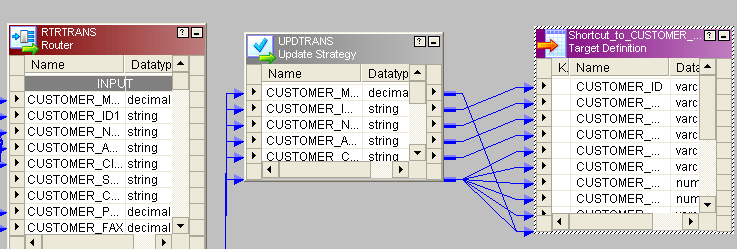
In the real world, it is possible to have a fact table that contains no measures or facts. These tables are called “Factless Fact tables”.  
E.g: A fact table which has only product key and date key is a factless fact. There are no measures in this table. But still you can get the number products sold over a period of time.

A fact table that contains aggregated facts are often called summary tables.

**25. Explain in detail about SCD TYPE 1 through mapping.**

SCD Type1 Mapping

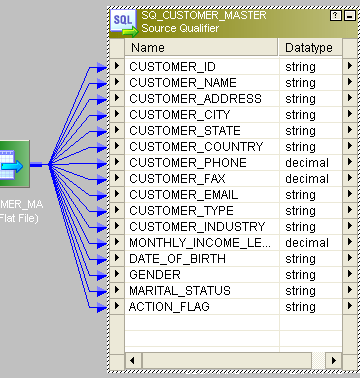
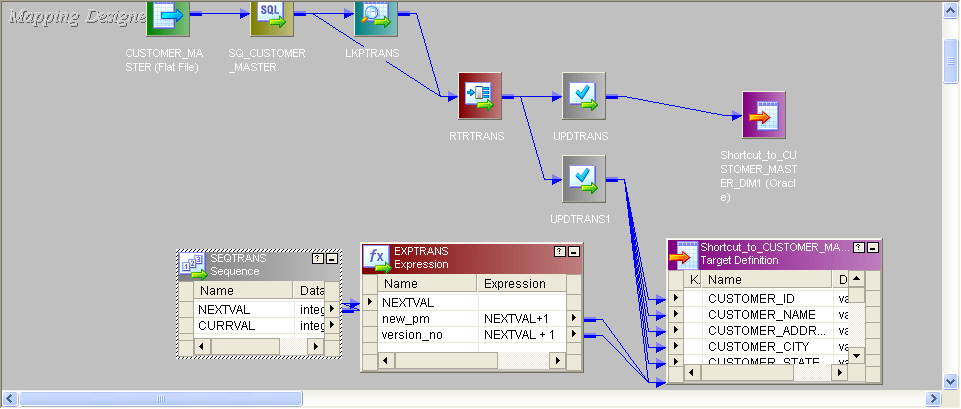
The SCD Type 1 methodology overwrites old data with new data, and therefore does not need to track historical data.

1. Here is the source.  
   
2. We will compare the historical data based on key column CUSTOMER\_ID.
3. This is the entire mapping:  
   
4. Connect lookup to source. In Lookup fetch the data from target table and send only CUSTOMER\_ID port from source to lookup.  
   
5. Give the lookup condition like this:  
   
6. Then, send rest of the columns from source to one router transformation.  
   
7. In router create two groups and give condition like this:  
   
8. For new records we have to generate new customer\_id. For that, take a sequence generator and connect the next column to expression. New\_rec group from router connect to target1 (Bring two instances of target to mapping, one for new rec and other for old rec). Then connect next\_val from expression to customer\_id column of target.  
   
9. Change\_rec group of router bring to one update strategy and give the condition like this:  
   
10. Instead of 1 you can give dd\_update in update-strategy and then connect to target.

**26. Explain in detail SCD TYPE 2 through mapping.**

**SCD Type2 Mapping**

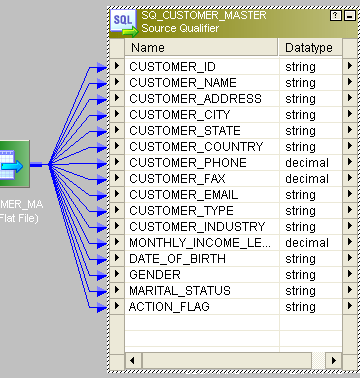
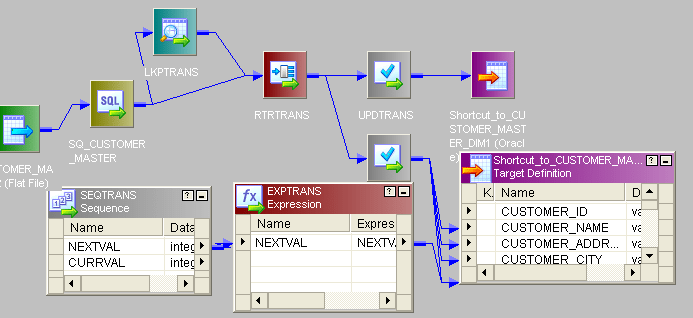
In Type 2 Slowly Changing Dimension, if one new record is added to the existing table with a new information then, both the original and the new record will be presented having new records with its own primary key.

1. To identifying new\_rec we should and one new\_pm and one vesion\_no.
2. This is the source:  
   
3. This is the entire mapping:  
   
4. All the procedures are similar to SCD TYPE1 mapping. The Only difference is, from router new\_rec will come to one update\_strategy and condition will be given dd\_insert and one new\_pm and version\_no will be added before sending to target.
5. Old\_rec also will come to update\_strategy condition will give dd\_insert then will send to target.

**27. Explain SCD TYPE 3 through mapping.**

**SCD Type3 Mapping**

In SCD Type3, there should be two columns added to identifying a single attribute. It stores one time historical data with current data.

1. This is the source:  
   
2. This is the entire mapping:  
   
3. Up to router transformation, all the procedure is same as described in SCD type1.
4. The only difference is after router, bring the new\_rec to router and give condition dd\_insert send to.
5. Create one new primary key send to target. For old\_rec send to update\_strategy and set condition dd\_insert and send to target.
6. You can create one effective\_date column in old\_rec table

**28. Differentiate between Reusable Transformation and Mapplet.**

Any Informatica Transformation created in the Transformation Developer or a non-reusable promoted to reusable transformation from the mapping designer which can be used in multiple mappings is known as **Reusable Transformation**.

When we add a reusable transformation to a mapping, we actually add an instance of the transformation. Since the instance of a reusable transformation is a pointer to that transformation, when we change the transformation in the Transformation Developer, its instances reflect these changes.

**A Mapplet** is a reusable object created in the Mapplet Designer which contains a set of transformations and lets us reuse the transformation logic in multiple mappings.

A Mapplet can contain as many transformations as we need. Like a reusable transformation when we use a mapplet in a mapping, we use an instance of the mapplet and any change made to the mapplet is inherited by all instances of the mapplet.

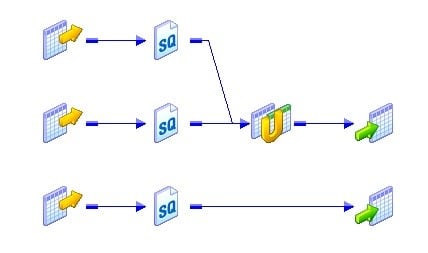
**29. What is meant by Target load plan?**

**Target Load Order**:

Target load order (or) Target load plan is used to specify the order in which the integration service loads the targets. You can specify a target load order based on the source qualifier transformations in a mapping. If you have multiple source qualifier transformations connected to multiple targets, you can specify the order in which the integration service loads the data into the targets.

**Target Load Order Group**:

A target load order group is the collection of source qualifiers, transformations and targets linked in a mapping. The integration service reads the target load order group concurrently and it processes the target load order group sequentially. The following figure shows the two target load order groups in a single mapping.



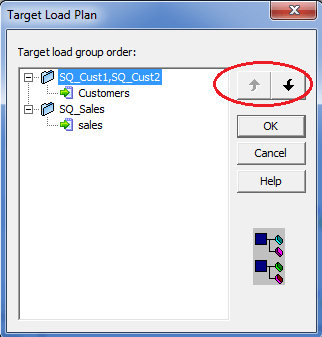
**Use of Target Load Order**:

Target load order will be useful when the data of one target depends on the data of another target. For example, the employees table data depends on the departments data because of the primary-key and foreign-key relationship. So, the departments table should be loaded first and then the employees table. Target load order is useful when you want to maintain referential integrity when inserting, deleting or updating tables that have the primary key and foreign key constraints.

**Target Load Order Setting**:

You can set the target load order or plan in the mapping designer. Follow the below steps to configure the target load order:

1. Login to the PowerCenter designer and create a mapping that contains multiple target load order groups.  
2. Click on the Mappings in the toolbar and then on Target Load Plan. The following dialog box will pop up listing all the source qualifier transformations in the mapping and the targets that receive data from each source qualifier.



1. Select a source qualifier from the list.
2. Click the Up and Down buttons to move the source qualifier within the load order.
3. Repeat steps 3 and 4 for other source qualifiers you want to reorder.
4. Click OK.

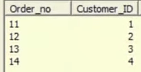
**30. Write the Unconnected lookup syntax and how to return more than one column.**

We can only return one port from the Unconnected Lookup transformation. As the Unconnected lookup is called from another transformation, we cannot return multiple columns using Unconnected Lookup transformation.

However, there is a trick. We can use the SQL override and concatenate the multiple columns, those we need to return. When we can the lookup from another transformation, we need to separate the columns again using substring.

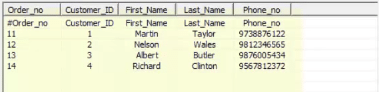
As a scenario, we are taking one source, containing the Customer\_id and Order\_id columns.

**Source:**

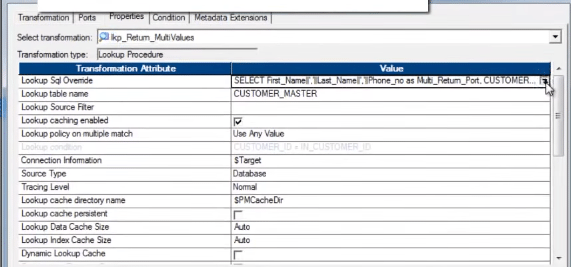


We need to look up the Customer\_master table, which holds the Customer information, like Name, Phone etc.

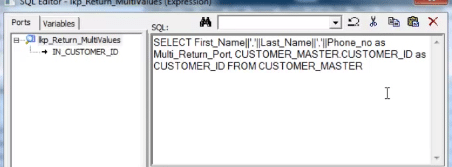
The target should look like this:



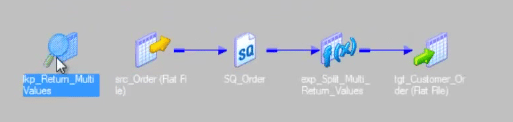
Let’s have a look at the Unconnected Lookup.



The SQL Override, with concatenated port/column:



Entire mapping will look like this.



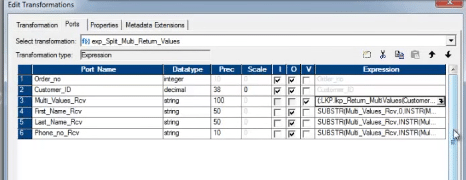
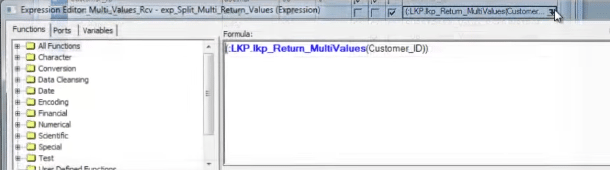
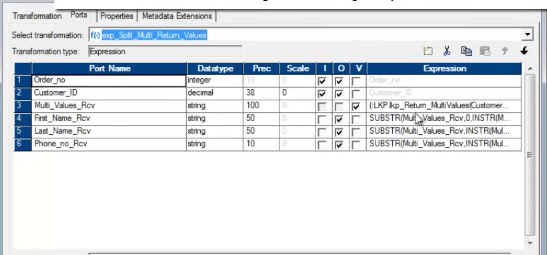
[[](https://www.edureka.co/informatica)](https://www.edureka.co/informatica" \t "_blank)

**[Informatica Training & Certification](https://www.edureka.co/informatica" \t "_blank)**

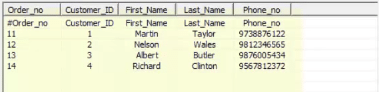
[Weekday / Weekend BatchesSee Batch Details](https://www.edureka.co/informatica" \t "_blank)

We are calling the unconnected lookup from one expression transformation.

Below is the screen shot of the expression transformation.



After execution of the above mapping, below is the target, that is populated.



I am pretty confident that after going through both these Informatica Interview Questions blog, you will be fully prepared to take Informatica Interview without any hiccups. If you wish to deep dive into Informatica with use cases, I will recommend you to go through our website and enrol at the earliest.