Table of Contents

Informatica Interview Questions

Informatica Interview Questions on Transformations

Transformation Name	Type
Aggregator Transformation	Active/Connected
Expression Transformation	Passive/Connected
Filter Transformation	Active/Connected
Joiner Transformation	Active/Connected
Lookup Transformation	Passive/Connected or Unconnected
Normalizer Transformation	Active/Connected
Rank Transformation	Active/Connected
Router Transformation	Active/Connected
Sequence Generator Transformation	Passive/Connected
Source Qualifier Transformation	Active/Connected
Sorter Transformation	Active/Connected
SQL Transformation	Active or Passive/Connected
Stored Procedure Transformation	Passive/Connected or Unconnected
Transaction Control Transformation	Active/Connected
Union Transformation	Active/Connected
Update Strategy Transformation	Active/Connected

Q1. What is a transformation?

Ans A transformation is a repository object that generates, modifies, or passes data.

Q2. What is an active transformation?

Ans An active transformation is the one which changes the number of rows that pass through it.

Example: Filter transformation

Q3. What is a passive transformation?

Ans A passive transformation is the one which does not change the number of rows that pass through it.

Example: Expression transformation

Q4. What is a connected transformation?

Ans A connected transformation is connected to the data flow or connected to the other transformations in the mapping pipeline.

Example: Sorter Transformation.

Q5. What is an unconnected transformation?

Ans An unconnected transformation is not connected to other transformations in the mapping. An unconnected transformation is called within another transformation and returns a value to that transformation.

Example: Unconnected lookup transformation, unconnected stored procedure transformation

Q6. What are multi-group transformations?

Ans Transformations having multiple input and output groups are called multi-group transformations.

Examples: Custom, HTTP, Joiner, Router, Union, Unstructured Data, XML source qualifier, XML Target definition, XML parser, XML generator

Q7. List out all the transformations which use cache?

Ans Aggregator, Joiner, Lookup, Rank, Sorter.

Q8. What is blocking transformation?

Ans Transformation which blocks the input rows are called blocking transformation.

Example: Custom transformation, unsorted joiner

Q9. What is a reusable transformation?

Ans A reusable transformation is the one which can be used in multiple mappings. Reusable transformation is created in transformation developer.

Q10. How do you promote a non-reusable transformation to reusable transformation?

Ans Edit the transformation and check the Make Reusable option

Q11. How to create a non-reusable instance of reusable transformations?

Ans In the navigator, select an existing transformation and drag the transformation into the mapping workspace. Hold down the Ctrl key before you release the transformation.

Q12. Which transformation can be created only as reusable transformation but not as non-reusable transformation?

Ans External Procedure Transformation.

Informatica Interview Questions on Source Qualifier Transformation

Q1. **What is a source qualifier transformation?**

Ans A source qualifier represents the rows that the integration service reads when it runs a session. Source qualifier is an active transformation.

Q2. **Why you need a source qualifier transformation?**

Ans The source qualifier transformation converts the source data types into informatica native data types.

Q3. **What are the different tasks a source qualifier can do?**

Ans Join two or more tables originating from the same source (homogeneous sources) database.

- Filter the rows.
- Sort the data.
- Selecting distinct values from the source.
- Create custom query.
- Specify a pre-sql and post-sql.

Q4. **What is the default join in source qualifier transformation?**

Ans The source qualifier transformation joins the tables based on the primary key-foreign key relationship.

Q5. **How to create a custom join in source qualifier transformation?**

Ans When there is no primary key-foreign key relationship between the tables, you can specify a custom join using the 'user-defined join' option in the properties tab of source qualifier.

Q6. **How to join heterogeneous sources and flat files?**

Ans Use joiner transformation to join heterogeneous sources and flat file.

Q7. **How do you configure a source qualifier transformation?**

Ans

- SQL Query.
- User-Defined Join.
- Source Filter.
- Number of Sorted Ports.
- Select Distinct.
- Pre-SQL.
- Post-SQL.

Informatica Interview Questions on Filter Transformation

Q1. What is a filter transformation?

Ans A filter transformation is used to filter out the rows in mapping. The filter transformation allows the rows that meet the filter condition to pass through and drops the rows that do not meet the condition. Filter transformation is an active transformation.

Q2. Can we specify more than one filter condition in a filter transformation?

Ans We can only specify one condition in the filter transformation. To specify more than one condition, we have to use router transformation?

Q3. In which case a filter transformation acts as passive transformation?

Ans If the filter condition is set to TRUE, then it passes all the rows without filtering any data. In this case, the filter transformation acts as passive transformation.

Q4. Can we concatenate ports from more than one transformation into the filter transformation?

Ans No. The input ports for the filter must come from a single transformation.

Q5. How to filter the null values and spaces?

Ans Use the ISNULL and IS_SPACES functions

For Example: IIF (ISNULL (commission), FALSE, TRUE)

Q6. How session performance can be improved by using filter transformation?

Ans Keep the filter transformation as close as possible to the sources in the mapping. This allows the unwanted data to be discarded and the integration service processes only the required rows. If the source is relational source, use the source qualifier to filter the rows.

Informatica Interview Questions on Aggregator Transformation

Q1. What is aggregator transformation?

Ans Aggregator transformation performs aggregate calculations like sum, average, count etc. It is an active transformation, changes the number of rows in the pipeline. Unlike expression transformation (performs calculations on a row-by-row basis), an aggregator transformation performs calculations on group of rows.

Q2. What is aggregate cache?

Ans The integration service creates index and data cache in memory to process the aggregator transformation and stores the data group in index cache, row data in data cache. If the integration service requires more space, it stores the overflow values in cache files.

Q3. How can we improve performance of aggregate transformation?

Ans

- Use sorted input: Sort the data before passing into aggregator. The integration service uses memory to process the aggregator transformation and it does not use cache memory.
- Filter the unwanted data before aggregating.
- Limit the number of input/output or output ports to reduce the amount of data the aggregator transformation stores in the data cache.

Q4. What are the different types of aggregate functions?

Ans The different types of aggregate functions are listed below:

- AVG
- COUNT
- FIRST
- LAST
- MAX
- MEDIAN
- MIN
- PERCENTILE
- STDDEV
- SUM
- VARIANCE

Q5. Why cannot you use both single level and nested aggregate functions in a single aggregate transformation?

Ans The nested aggregate function returns only one output row, whereas the single level aggregate function returns more than one row. Since the number of rows returned are not same, you cannot use both single level and nested aggregate functions in the same transformation. If you include both the single level and nested functions in the same aggregator, the designer marks the mapping or mapplet as invalid. So, you need to create separate aggregator transformations.

Q6. Up to how many levels, you can nest the aggregate functions?

Ans We can nest up to two levels only.
For Example: MAX (SUM (ITEM))

Q7. What is iGuwahatiemental aggregation?

Ans The integration service performs aggregate calculations and then stores the data in historical cache. Next time when you run the session, the integration service reads only new data and uses the historical cache to perform new aggregation calculations iGuwahatiementally.

Q8. Why cannot we use sorted input option for iGuwahatiemental aggregation?

Ans In iGuwahatiemental aggregation, the aggregate calculations are stored in historical cache on the server. In this historical cache the data need not be in sorted order. If you give sorted input, the records come as pre-sorted for that particular run but in the historical cache the data may not be in the sorted order. That is why this option is not allowed.

Q9. How the NULL values are handled in Aggregator?

Ans You can configure the integration service to treat null values in aggregator functions as NULL or zero. By default, the integration service treats null values as NULL in aggregate functions.

Informatica Interview Questions on Expression Transformation

Q1. **What is an expression transformation?**

Ans An expression transformation is used to calculate values in a single row.
Example: salary+1000

Q2. **How to generate sequence numbers using expression transformation?**

Ans Create a variable port in expression transformation and iGuwahatiement it by one for every row. Assign this variable port to an output port.

Q3. **Consider the following Employee s data as source?**

Employee_id	Salary
10	1000
20	2000
30	3000
40	4000

1. Design a mapping to load the cumulative sum of salaries of employees into target table?

The target table data should look like as:

Employee_Id	Salary	cumulative_sum
10	1000	1000
20	2000	3000
30	3000	6000
40	5000	11000

Ans Connect the source Qualifier to expression transformation. In the expression transformation, create a variable port V_cum_sal and in the expression editor write V_cum_sal + salary.
Create an output port O_cum_sal and assign V_cum_sal to it.

2. Design a mapping to get the pervious row salary for the current row. If there is no pervious row exists for the current row, then the pervious row salary should be displayed as null.

The output should look like as:

Employee_Id	Salary	cumulative_sum
10	1000	Null
20	2000	1000
30	3000	2000
40	5000	3000

Ans Connect the source Qualifier to expression transformation. In the expression transformation, create a variable port V_count and iGuwahatiement it by one for each row entering the expression transformation. Also create V_salary variable port and assign the expression IIF(V_count=1, NULL, V_prev_salary) to it . Then create one more variable port V_prev_salary and assign Salary to it. Now create output port O_prev_salary and assign V_salary to it. Connect the expression transformation to the target ports.

In the expression transformation, the ports will be:

employee_id
salary
V_count=V_count+1
V_salary=IIF(V_count=1,NULL,V_prev_salary)
V_prev_salary=salary
O_prev_salary=V_salary

3. Design a mapping to get the next row salary for the current row. If there is no next row for the current row, then the next row salary should be displayed as null.

The output should look like as:

Employee_Id	Salary	cumulative_sum
10	1000	Null
20	2000	1000
30	3000	2000
40	5000	3000

Ans **Step1:** Connect the source qualifier to two expression transformation. In each expression transformation, create a variable port V_count and in the expression editor write V_count+1. Now create an output port O_count in each expression transformation. In the first expression transformation, assign V_count to O_count. In the second expression transformation assign V_count-1 to O_count.

In the first expression transformation, the ports will be:
employee_id
salary
V_count=V_count+1
O_count=V_count

In the second expression transformation, the ports will be:
employee_id
salary
V_count=V_count+1
O_count=V_count-1

Step2: Connect both the expression transformations to joiner transformation and join them on the port O_count. Consider the first expression transformation as Master and second one as detail. In the joiner specify the join type as Detail Outer Join. In the joiner transformation check the property sorted input, then only you can connect both expression transformations to joiner transformation.

Step3: Pass the output of joiner transformation to a target table. From the joiner, connect the employee_id, salary which are obtained from the first expression transformation to the employee_id, salary ports in target table. Then from the joiner, connect the salary which is obtained from the second expression transformaiton to the next_row_salary port in the target table.

4. Design a mapping to find the sum of salaries of all employees and this sum should repeat for all the rows.
The output should look like as:

Employee_Id	Salary	Salary_sum
10	1000	11000
20	2000	11000
30	3000	11000
40	5000	11000

Ans **Step1:** Connect the source qualifier to the expression transformation. In the expression transformation, create a dummy port and assign value 1 to it.

In the expression transformation, the ports will be:
employee_id
salary
O_dummy=1

Step2: Pass the output of expression transformation to aggregator. Create a new port O_sum_salary and in the expression editor write SUM (salary). Do not specify group by on any port.

In the aggregator transformation, the ports will be:
Salary
O_dummy
O_sum_salary=SUM(salary)

Step3: Pass the output of expression transformation, aggregator transformation to joiner transformation and join on the DUMMY port. In the joiner transformation check the property sorted input, then only you can connect both expression and aggregator to joiner transformation.

Step4: Pass the output of joiner to the target table.

Q4. **Consider the following employees table as source**

Department_No	Employee_Name
20	R
10	A
10	D
20	P
10	B
10	C
20	Q
20	S

1. **Design a mapping to load a target table with the following values from the above source?**

Department_No	Employee_List
10	A
10	A, B
10	A, B, C
10	A, B, C, D
20	A, B, C, D, P
20	A, B, C, D, P, Q
20	A, B, C, D, P, Q, R
20	A, B, C, D, P, Q, R, S

Ans **Step1:** Use a sorter transformation and sort the data using the sort key as department_no and then pass the output to the expression transformation. In the expression transformation, the ports will be
department_no
employee_name
V_employee_list = IIF(ISNULL(V_employee_list),employee_name,V_employee_list||','||employee_name)
O_employee_list = V_employee_list

Step2: Now connect the expression transformation to a target table.

2. **Design a mapping to load a target table with the following values from the above source?**

Department_No	Employee_List
10	A
10	A, B
10	A, B, C
10	A, B, C, D
20	P
20	P, Q
20	P, Q, R
20	P, Q, R, S

Ans **Step1:** Use a sorter transformation and sort the data using the sort key as department_no and then pass the output to the expression transformation. In the expression transformation, the ports will be
department_no

```
employee_name
V_curr_deptno=department_no
V_employee_list = IIF (V_curr_deptno! = V_prev_deptno,employee_name,V_employee_list||','||employee_name)
V_prev_deptno=department_no
O_employee_list = V_employee_list
```

Step2: Now connect the expression transformation to a target table.

1. **Design a mapping to load a target table with the following values from the above source?**

Department_No	Employee_List
10	A, B, C, D
20	P, Q, R, S

Ans The first step is same as the above problem. Pass the output of expression to an aggregator transformation and specify the group by as department_no Now connect the aggregator transformation to a target table.

Informatica Interview Questions on Joiner Transformation

Q1. What is a joiner transformation?

Ans A joiner transformation joins two heterogeneous sources. You can also join the data from the same source. The joiner transformation joins sources with at least one matching column. The joiner uses a condition that matches one or more joins of columns between the two sources.

Q2. How many joiner transformations are required to join n sources?

Ans To join n sources n-1 joiner transformations are required.

Q3. What are the limitations of joiner transformation?

Ans

- You cannot use a joiner transformation when input pipeline contains an update strategy transformation.

- You cannot use a joiner if you connect a sequence generator transformation directly before the joiner.

Q4. What are the different types of joins?

Ans

- Normal join: In a normal join, the integration service discards all the rows from the master and detail source that do not match the join condition.
- Master outer join: A master outer join keeps all the rows of data from the detail source and the matching rows from the master source. It discards the unmatched rows from the master source.
- Detail outer join: A detail outer join keeps all the rows of data from the master source and the matching rows from the detail source. It discards the unmatched rows from the detail source.
- Full outer join: A full outer join keeps all rows of data from both the master and detail rows.

Q5. What is joiner cache?

Ans When the integration service processes a joiner transformation, it reads the rows from master source and builds the index and data cached. Then the integration service reads the detail source and performs the join. In case of sorted joiner, the integration service reads both sources (master and detail) concurrently and builds the cache based on the master rows.

Q6. How to improve the performance of joiner transformation?

Ans

- Join sorted data whenever 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.

Q7. Why joiner is a blocking transformation?

Ans When the integration service processes an unsorted joiner transformation, it reads all master rows before it reads the detail rows. To ensure it reads all master rows before the detail rows, the integration service blocks all the details source while it caches rows from the master source. As it blocks the detail source, the unsorted joiner is called a blocking transformation.

Q8. What are the settings used to configure the joiner transformation?

Ans

- Master and detail source
- Type of join
- Join condition

Informatica Interview Questions on Lookup Transformation

Q1. What is a lookup transformation?

Ans A lookup transformation is used to look up data in a flat file, relational table, view, and synonym.

Q2. What are the tasks of a lookup transformation?

Ans The lookup transformation is used to perform the following tasks?

- **Get a related value**: Retrieve a value from the lookup table based on a value in the source.
- **Perform a calculation**: Retrieve a value from a lookup table and use it in a calculation.
- **Update slowly changing dimension tables**: Determine whether rows exist in a target.

Q3. How do you configure a lookup transformation?

Ans Configure the lookup transformation to perform the following types of lookups:

- Relational or flat file lookup.
- Pipeline lookup.
- Connected or unconnected lookup.
- Cached or uncached lookup.

Q4. What is a pipeline lookup transformation?

Ans A pipeline lookup transformation is used to perform lookup on application sources such as JMS, MSMQ or SAP. A pipeline lookup transformation has a source qualifier as the lookups source.

Q5. What is connected and unconnected lookup transformation?

Ans

- A connected lookup transformation is connected the transformations in the mapping pipeline. It receives source data, performs a lookup and returns data to the pipeline.

- An unconnected lookup transformation is not connected to the other transformations in the mapping pipeline. A transformation in the pipeline calls the unconnected lookup with a:LKP expression.

Q6. What are the differences between connected and unconnected lookup transformation?

Ans

- Connected lookup transformation receives input values directly from the pipeline. Unconnected lookup transformation receives input values from the result of a:LKP expression in another transformation.
- Connected lookup transformation can be configured as dynamic or static cache. Unconnected lookup transformation can be configured only as static cache.
- Connected lookup transformation can return multiple columns from the same row or insert into the dynamic lookup cache. Unconnected lookup transformation can return one column from each row.
- If there is no match for the lookup condition, connected lookup transformation returns default value for all output ports. If you configure dynamic caching, the Integration Service inserts rows into the cache or leaves it unchanged. If there is no match for the lookup condition, the unconnected lookup transformation returns null.
- In a connected lookup transformation, the cache includes the lookup source columns in the lookup condition and the lookup source columns that are output ports. In an unconnected lookup transformation, the cache includes all lookup/output ports in the lookup condition and the lookup/return port.
- Connected lookup transformation passes multiple output values to another transformation. Unconnected lookup transformation passes one output value to another transformation.
- Connected lookup transformation supports user-defined values. Unconnected lookup transformation does not support user-defined default values.

Q7. How do you handle multiple matches in lookup transformation? or what is "Lookup Policy on Multiple Match"?

Ans "Lookup Policy on Multiple Match" option is used to determine which rows that the lookup transformation returns when it finds multiple rows that match the lookup condition. You can select lookup to return first or last row or any matching row or to report an error.

Q8. What is "Output Old Value on Update"?

Ans This option is used when dynamic cache is enabled. When this option is enabled, the integration service outputs old values out of the lookup/output ports. When the Integration Service updates a row in the cache, it outputs the value that existed in the lookup cache before it updated the row based on the input data. When the Integration Service inserts a new row in the cache, it outputs null values. When you disable this property, the Integration Service outputs the same values out of the lookup/output and input/output ports.

Q9. What is "Insert Else Update" and "Update Else Insert"?

Ans These options are used when dynamic cache is enabled.

- Insert Else Update option applies to rows entering the lookup transformation with the row type of insert. When this option is enabled the integration service inserts new rows in the cache and updates existing rows when disabled, the Integration Service does not update existing rows.
- Update Else Insert option applies to rows entering the lookup transformation with the row type of update. When this option is enabled, the Integration Service updates existing rows, and inserts a new row if it is new. When disabled, the Integration Service does not insert new rows.

Q10.What are the options available to configure a lookup cache?

Ans The following options can be used to configure a lookup cache:

- Persistent cache.
- Recache from lookup source.
- Static cache.
- Dynamic cache.
- Shared Cache.
- Pre-build lookup cache.

Q11. What is a cached lookup transformation and Uncached lookup transformation?

Ans

- Cached lookup transformation: The Integration Service builds a cache in memory when it processes the first row of data in a cached

Lookup transformation. The Integration Service stores condition values in the index cache and output values in the data cache. The Integration Service queries the cache for each row that enters the transformation.

- Uncached lookup transformation: For each row that enters the lookup transformation, the Integration Service queries the lookup source and returns a value. The integration service does not build a cache.

Q12. How the integration service builds the caches for connected lookup transformation?

Ans The Integration Service builds the lookup caches for connected lookup transformation in the following ways:

- Sequential cache: The Integration Service builds lookup caches sequentially. The Integration Service builds the cache in memory when it processes the first row of the data in a cached lookup transformation.
- Concurrent caches: The Integration Service builds lookup caches concurrently. It does not need to wait for data to reach the Lookup transformation.

Q13. How the integration service builds the caches for unconnected lookup transformation?

Ans The Integration Service builds caches for unconnected Lookup transformations as sequentially.

Q14. What is a dynamic cache?

Ans The dynamic cache represents the data in the target. The Integration Service builds the cache when it processes the first lookup request. It queries the cache based on the lookup condition for each row that passes into the transformation. The Integration Service updates the lookup cache as it passes rows to the target. The integration service either inserts the row in the cache or updates the row in the cache or makes no change to the cache.

Q15. When you use a dynamic cache, do you need to associate each lookup port with the input port?

Ans Yes. You need to associate each lookup/output port with the input/output port or a sequence ID. The Integration Service uses the data in the associated port to insert or update rows in the lookup cache.

Q16. What are the different values returned by NewLookupRow port?

Ans The different values are

- 0 - Integration Service does not update or insert the row in the cache.
- 1 - Integration Service inserts the row into the cache.
- 2 - Integration Service updates the row in the cache.

Q17. What is a persistent cache?

Ans If the lookup source does not change between session runs, then you can improve the performance by creating a persistent cache for the source. When a session runs for the first time, the integration service creates the cache files and saves them to disk instead of deleting them. The next time when the session runs, the integration service builds the memory from the cache file.

Q18. What is a shared cache?

Ans You can configure multiple Lookup transformations in a mapping to share a single lookup cache. The Integration Service builds the cache when it processes the first Lookup transformation. It uses the same cache to perform lookups for subsequent Lookup transformations that share the cache.

Q19. What is unnamed cache and named cache?

Ans

- Unnamed cache: When Lookup transformations in a mapping have compatible caching structures, the Integration Service shares the cache by default. You can only share static unnamed caches.
- Named cache: Use a persistent named cache when you want to share a cache file across mappings or share a dynamic and a static cache. The caching structures must match or be compatible with a named cache. You can share static and dynamic named caches.

Q20. How do you improve the performance of lookup transformation?

Ans

- Create an index on the columns used in the lookup condition.
- Place conditions with equality operator first.
- Cache small lookup tables.
- Join tables in the database: If the source and the lookup table are in the same database, join the tables in the database rather than using a lookup transformation.

- Use persistent cache for static lookups.
- Avoid ORDER BY on all columns in the lookup source. Specify explicitly the ORDER By clause on the required columns.
- For flat file lookups, provide Sorted files as lookup source.

Informatica Interview Questions on Normalizer Transformation

Q1. What is normalizer transformation?

Ans The normalizer transformation receives a row that contains multiple-occurring columns and returns a row for each instance of the multiple-occurring data. This means it converts column data in to row data. Normalizer is an active transformation.

Q2. Which transformation is required to process the Cobol sources?

Ans Since the Cobol sources contain denormalized data, normalizer transformation is used to normalize the Cobol sources.

Q3. What is generated key and generated column id in a normalizer transformation?

Ans

- The integration service iGuwahatiements the generated key sequence number each time it process a source row. When the source row contains a multiple-occurring column or a multiple-occurring group of columns, the normalizer transformation returns a row for each occurrence. Each row contains the same generated key value.
- The normalizer transformation has a generated column ID (GCID) port for each multiple-occurring column. The GCID is an index for the instance of the multiple-occurring data. For example, if a column occurs 3 times in a source record, the normalizer returns a value of 1,2 or 3 in the generated column ID.

Q4. What is VSAM?

Ans VSAM (Virtual Storage Access Method) is a file access method for an IBM mainframe operating system. VSAM organize records in indexed or sequential flat files.

Q5. What is VSAM normalizer transformation?

Ans The VSAM normalizer transformation is the source qualifier transformation for a COBOL source definition. A COBOL source is flat file that can contain multiple-occurring data and multiple types of records in the same file.

Q6. What is pipeline normalizer transformation?

Ans Pipeline normalizer transformation processes multiple-occurring data from relational tables or flat files.

Q7. What is occurs clause and redefines clause in normalizer transformation?

Ans

- Occurs clause is specified when the source row has a multiple-occurring columns.
- A redefines clause is specified when the source has rows of multiple columns.

Informatica Interview Questions on Rank Transformation

Q1. What is rank transformation?

Ans A rank transformation is used to select top or bottom rank of data. This means, it selects the largest or smallest numeric value in a port or group. Rank transformation also selects the strings at the top or bottom of a session sort order. Rank transformation is an active transformation.

Q2. What is rank cache?

Ans The integration service compares input rows in the data cache, if the input row out-ranks a cached row, the integration service replaces the cached row with the input row. If you configure the rank transformation to rank across multiple groups, the integration service ranks iGuwahatiementally for each group it finds. The integration service stores group information in index cache and row data in data cache.

Q3. What is RANKINDEX port?

Ans The designer creates RANKINDEX port for each rank transformation. The integration service uses the rank index port to store the ranking position for each row in a group.

Q4. How do you specify the number of rows you want to rank in a rank transformation?

Ans In the rank transformation properties, there is an option 'Number of Ranks' for specifying the number of rows you want to rank.

Q5. How to select either top or bottom ranking for a column?

Ans In the rank transformation properties, there is an option 'Top/Bottom' for selecting the top or bottom ranking for a column.

Q6. Can we specify ranking on more than one port?

Ans No. We can specify to rank the data based on only one port. In the ports tab, you have to check the R option for designating the port as a rank port and this option can be checked only on one port.

Informatica Interview Questions on Router Transformation

Q1. What is a router transformation?

Ans A router is used to filter the rows in a mapping. Unlike filter transformation, you can specify one or more conditions in a router transformation. Router is an active transformation.

Q2. How to improve the performance of a session using router transformation?

Ans Use router transformation in a mapping instead of creating multiple filter transformations to perform the same task. The router transformation is more efficient in this case. When you use a router transformation in a mapping, the integration service processes the incoming data only once. When you use multiple filter transformations, the integration service processes the incoming data for each transformation.

Q3. What are the different groups in router transformation?

Ans The router transformation has the following types of groups:

- Input
- Output

Q4. How many types of output groups are there?

Ans There are two types of output groups:

- User-defined group
- Default group

Q5. Where you specify the filter conditions in the router transformation?

Ans You can create the group filter conditions in the groups tab using the expression editor.

Q6. Can you connect ports of two output groups from router transformation to a single target?

Ans No. You cannot connect more than one output group to one target or a single input group transformation.

Informatica Interview Questions on Sequence Generator Transformation

Q1. What is a sequence generator transformation?

Ans A Sequence generator transformation generates numeric values. Sequence generator transformation is a passive transformation.

Q2. What is the use of a sequence generator transformation?

Ans A sequence generator is used to create unique primary key values, replace missing primary key values or cycle through a sequential range of numbers.

Q3. What are the ports in sequence generator transformation?

Ans A sequence generator contains two output ports.

- CURRVAL.
- NEXTVAL.

Q4. What is the maximum number of sequence that a sequence generator can generate?

Ans The maximum value is 9,223,372,036,854,775,807

Q5. When you connect both the NEXTVAL and CURRVAL ports to a target, what will be the output values of these ports?

Ans **The output values are**

NEXTVAL	CURRVAL
1	2
2	3
3	4
4	5
5	6

Q6. What will be the output value, if you connect only CURRVAL to the target without connecting NEXTVAL?

Ans The integration service passes a constant value for each row.

Q7. What will be the value of CURRVAL in a sequence generator transformation?

Ans CURRVAL is the sum of "NEXTVAL" and "IGuwahatiement By" Value.

Q8. What is the number of cached values set to default for a sequence generator transformation?

Ans For non-reusable sequence generators, the number of cached values is set to zero. For reusable sequence generators, the number of cached values is set to 1000.

Q9. How do you configure a sequence generator transformation?

Ans The following properties need to be configured for a sequence generator transformation:

- Start Value.
- IGuwahatiement By.
- End Value.
- Current Value.
- Cycle.
- Number of Cached Values.

Informatica Interview Questions on Sorter Transformation

Q1. What is a sorter transformation?

Ans Sorter transformation is used to sort the data. You can sort the data either in ascending or descending order according to a specified sort key.

Q2. Why sorter is an active transformation?

Ans As sorter transformation can suppress the duplicate records in the source, it is called an active transformation.

Q3. How to improve the performance of a session using sorter transformation?

Ans Sort the data using sorter transformation before passing in to aggregator or joiner transformation. As the data is sorted, the integration service uses the memory to do aggregate and join operations and does not use cache files to process the data.

Informatica Interview Questions on SQL Transformation

Q1. What is SQL transformation?

Ans SQL transformation process SQL queries midstream in a pipeline and you can insert, update, delete and retrieve rows from a database.

Q2. How do you configure a SQL transformation?

Ans The following options are required to configure SQL transformation:

- **Mode**: Specifies the mode in which SQL transformation runs. SQL transformation supports two modes. They are script mode and query mode.
- **Database type**: The type of database that SQL transformation connects to.
- **Connection type**: Pass database connection to the SQL transformation at run time or specify a connection object.

Q3. What are the different modes in which a SQL transformation runs?

Ans SQL transformation runs in two modes. They are:

- **Script mode**: The SQL transformation runs scripts that are externally located. You can pass a script name to the transformation with each input row. The SQL transformation outputs one row for each input row.
- **Query mode**: The SQL transformation executes a query that you define in a query editor. You can pass parameters to the query to define dynamic queries. You can output multiple rows when the query has a SELECT statement.

Q4. In which cases the SQL transformation becomes a passive transformation and active transformation?

Ans If you run the SQL transformation in script mode, then it becomes passive transformation. If you run the SQL transformation in the query mode and the query has a SELECT statement, then it becomes an active transformation.

Q5. When you configure an SQL transformation to run in script mode, what are the ports that the designer adds to the SQL transformation?

Ans The designer adds the following ports to the SQL transformation in script mode:

- **ScriptName**: This is an input port. ScriptName receives the name of the script to execute the current row.
- **ScriptResult**: This is an output port. ScriptResult returns PASSED if the script execution succeeds for the row. Otherwise it returns FAILED.
- **ScriptError**: This is an output port. ScriptError returns the errors that occur when a script fails for a row.

Q6. What are the types of SQL queries you can specify in the SQL transformation when you use it in query mode?

Ans

- **Static SQL query**: The query statement does not change, but you can use query parameters to change the data. The integration service prepares the query once and runs the query for all input rows.
- **Dynamic SQL query**: The query statement can be changed. The integration service prepares a query for each input row.

Q7. What are the types of connections to connect the SQL transformation to the database available?

Ans

- **Static connection**: Configure the connection object tin the session. You must first create the connection object in workflow manager.
- **Logical connection**: Pass a connection name to the SQL transformation as input data at run time. You must first create the connection object in workflow manager.
- **Full database connection**: Pass the connect string, user name, password and other connection information to SQL transformation input ports at run time.

Q8. How do you find the number of rows inserted, updated or deleted in a table?

Ans You can enable the NumRowsAffected output port to return the number of rows affected by the INSERT, UPDATE or DELETE query statements in each input row. This NumRowsAffected option works in query mode.

Q9. What will be the output of NumRowsAffected port for a SELECT statement?

Ans The NumRowsAffected outout is zero for the SELECT statement.

Q10. When you enable the NumRowsAffected output port in script mode, what will be the output?

Ans In script mode, the NumRowsAffected port always returns NULL.

Q11. How do you limit the number of rows returned by the select statement?

Ans You can limit the number of rows by configuring the Max Output Row Count property. To configure unlimited output rows, set Max Output Row Count to zero.

Informatica Interview Questions on Stored Procedure Transformation

Q1. What is a stored procedure?

Ans A stored procedure is a precompiled collection of database procedural statements. Stored procedures are stored and run within the database.

Q2. Give some examples where a stored procedure is used?

Ans The stored procedure can be used to do the following tasks

- Check the status of a target database before loading data into it.
- Determine if enough space exists in a database.
- Perform a specialized calculation.
- Drop and recreate indexes.

Q3. What is a connected stored procedure transformation?

Ans The stored procedure transformation is connected to the other transformations in the mapping pipeline.

Q4. In which scenarios a connected stored procedure transformation is used?

Ans Run a stored procedure every time a row passes through the mapping.
Pass parameters to the stored procedure and receive multiple output parameters.

Q5. What is an unconnected stored procedure transformation?

Ans The stored procedure transformation is not connected directly to the flow of the mapping. It either runs before or after the session or is called by an expression in another transformation in the mapping.

Q6. In which scenarios an unconnected stored procedure transformation is used?

Ans

- Run a stored procedure before or after a session.
- Run a stored procedure once during a mapping, such as pre or post-session.
- Run a stored procedure based on data that passes through the mapping, such as when a specific port does not contain a null value.
- Run nested stored procedures.
- Call multiple times within a mapping.

Q7. What are the options available to specify when the stored procedure transformation needs to be run?

Ans The following options describe when the stored procedure transformation runs:

- **Normal**: The stored procedure runs where the transformation exists in the mapping on a row-by-row basis. This is useful for calling the stored procedure for each row of data that passes through the mapping, such as running a calculation against an input port. Connected stored procedures run only in normal mode.
- **Pre-load of the Source**: Before the session retrieves data from the source, the stored procedure runs. This is useful for verifying the existence of tables or performing joins of data in a temporary table.
- **Post-load of the Source**: After the session retrieves data from the source, the stored procedure runs. This is useful for removing temporary tables.
- **Pre-load of the Target**: Before the session sends data to the target, the stored procedure runs. This is useful for verifying target tables or disk space on the target system.
- **Post-load of the Target**: After the session sends data to the target, the stored procedure runs. This is useful for re-creating indexes on the database.

A connected stored procedure transformation runs only in Normal mode. An unconnected stored procedure transformation runs in all the above modes.

Q8. What is execution order in stored procedure transformation?

Ans The order in which the Integration Service calls the stored procedure used in the transformation, relative to any other stored procedures in the same mapping. Only used when the Stored Procedure Type is set to anything except Normal and more than one stored procedure exists.

Q9. What is PROC_RESULT in stored procedure transformation?

Ans PROC_RESULT is a system variable, where the output of an unconnected stored procedure transformation is assigned by default.

Q10. What are the parameter types in a stored procedure?

Ans There are three types of parameters exist in a stored procedure:

- **IN**: Input passed to the stored procedure.
- **OUT**: Output returned from the stored procedure.
- **INOUT**: Defines the parameter as both input and output. Only Oracle supports this parameter type.

Informatica Interview Questions on Transaction Control Transformation

Q1. What is a transaction control transformation?

Ans A transaction is a set of rows bound by a commit or rollback of rows. The transaction control transformation is used to commit or rollback a group of rows.

Q2. What is the commit type if you have a transaction control transformation in the mapping?

Ans The commit type is "user-defined".

Q3. What are the different transaction levels available in transaction control transformation?

Ans The following are the transaction levels or built-in variables:

- **TC_CONTINUE_TRANSACTION**: The Integration Service does not perform any transaction change for this row. This is the default value of the expression.
- **TC_COMMIT_BEFORE**: The Integration Service commits the transaction, begins a new transaction, and writes the current row to the target. The current row is in the new transaction.

- **TC_COMMIT_AFTER**: The Integration Service writes the current row to the target, commits the transaction, and begins a new transaction. The current row is in the committed transaction.
- **TC_ROLLBACK_BEFORE**: The Integration Service rolls back the current transaction, begins a new transaction, and writes the current row to the target. The current row is in the new transaction.
- **TC_ROLLBACK_AFTER**: The Integration Service writes the current row to the target, rolls back the transaction, and begins a new transaction. The current row is in the rolled back transaction.

Informatica Interview Questions on Union Transformation

Q1. What is a union transformation?

Ans A union transformation is used merge data from multiple sources similar to the UNION ALL SQL statement to combine the results from two or more SQL statements.

Q2. As union transformation gives UNION ALL output, how you will get the UNION output?

Ans Pass the output of union transformation to a sorter transformation. In the properties of sorter transformation check the option select distinct. Alternatively, you can pass the output of union transformation to aggregator transformation and in the aggregator transformation specify all ports as group by ports.

Q3. What are the guidelines to be followed while using union transformation?

Ans The following rules and guidelines need to be taken care while working with union transformation:

- You can create multiple input groups, but only one output group.
- All input groups and the output group must have matching ports. The precision, datatype, and scale must be identical across all groups.
- The Union transformation does not remove duplicate rows. To remove duplicate rows, you must add another transformation such as a Router or Filter transformation.
- You cannot use a Sequence Generator or Update Strategy transformation upstream from a Union transformation.
- The Union transformation does not generate transactions.

Q4. Why union transformation is an active transformation?

Ans Union is an active transformation because it combines two or more data streams into one. Though the total number of rows passing into the Union is the same as the total number of rows passing out of it, and the sequence of rows from any given input stream is preserved in the output, 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.

Informatica Interview Questions on Update Strategy Transformation

Q1. What is an update strategy transformation?

Ans Update strategy transformation is used to flag source rows for insert, update, delete or reject within a mapping. Based on this flagging each row will be either inserted or updated or deleted from the target. Alternatively, the row can be rejected.

Q2. Why update strategy is an active transformation?

Ans As update strategy transformation can reject rows, it is called as an active transformation.

Q3. What are the constants used in update strategy transformation for flagging the rows?

Ans

- DD_INSERT is used for inserting the rows. The numeric value is 0.
- DD_UPDATE is used for updating the rows. The numeric value is 1.
- DD_DELETE is used for deleting the rows. The numeric value is 2.
- DD_REJECT is used for rejecting the rows. The numeric value is 3.

Q4. If you place an aggregator after the update strategy transformation, how the output of aggregator will be affected?

Ans The update strategy transformation flags the rows for insert, update and delete of reject before you perform aggregate calculation. How you flag a particular row determines how the aggregator transformation treats any values in that row used in the calculation. For example, if you flag a row for delete and then later use the row to calculate the sum, the integration service subtracts the value appearing in this row. If the row had been flagged for insert, the integration service would add its value to the sum.

Q5. How to update the target table without using update strategy transformation?

Ans In the session properties, there is an option 'Treat Source Rows As'. Using this option, you can specify whether all the source rows need to be inserted, updated or deleted.

Q6. If you have an update strategy transformation in the mapping, what should be the value selected for 'Treat Source Rows As' option in session properties?

Ans The value selected for the option is 'Data Driven'. The integration service follows the instructions coded in the update strategy transformation.

Q7. If you have an update strategy transformation in the mapping and you did not selected the value 'Data Driven' for 'Treat Source Rows As' option in session, then how the session will behave?

Ans If you do not choose Data Driven when a mapping contains an Update Strategy or Custom transformation, the Workflow Manager displays a warning. When you run the session, the Integration Service does not follow instructions in the Update Strategy transformation in the mapping to determine how to flag rows.

Q8. In which files the data rejected by update strategy transformation will be written?

Ans If the update strategy transformation is configured to Forward Rejected Rows, then the integration service forwards the rejected rows to next transformation and writes them to the session reject file. If you do not select the forward reject rows option, the integration service drops rejected rows and writes them to the session log file. If you enable row error handling, the Integration Service writes the rejected rows and the dropped rows to the row error logs. It does not generate a reject file.

Informatica Scenario Based Questions with Answers

Q1. How to generate sequence numbers using expression transformation?

Ans In the expression transformation, create a variable port and iGuwahatiement it by 1. Then assign the variable port to an output port. In the expression transformation, the ports are:
V_count=V_count+1
O_count=V_count

Q2. Design a mapping to load the first 3 rows from a flat file into a target?

Ans You have to assign row numbers to each record. Generate the row numbers either using the expression transformation as mentioned above or use sequence generator transformation.
Then pass the output to filter transformation and specify the filter condition as O_count <=3

Q3. Design a mapping to load the last 3 rows from a flat file into a target?

Ans Consider the source has the following data.

Column
A
B
C
D
E

Step1: You have to assign row numbers to each record. Generate the row numbers using the expression transformation as mentioned above and call the row number generated port as O_count. Create a DUMMY output port in the same expression transformation and assign 1 to that port. So that, the DUMMY output port always returns 1 for each row.

In the expression transformation, the ports are:
V_count=V_count+1
O_count=V_count
O_dummy=1

The output of expression transformation will be:

Column	Column	Column
A	1	1
B	2	1
C	3	1
D	4	1
E	5	1

Step2: Pass the output of expression transformation to aggregator and do not specify any group by condition. Create an output port O_total_records in the aggregator and assign O_count port to it. The aggregator will return the last row by default. The output of aggregator contains the DUMMY port which has value 1 and O_total_records port which has the value of total number of records in the source.

In the aggregator transformation, the ports are:
O_dummy
O_count
O_total_records=O_count

The output of aggregator transformation will be:

O_total_records	O_dummy
5	1

Step3: Pass the output of expression transformation, aggregator transformation to joiner transformation and join on the DUMMY port. In the joiner transformation check the property sorted input, then only you can connect both expression and aggregator to joiner transformation.

In the joiner transformation, the join condition will be
O_dummy (port from aggregator transformation) = O_dummy (port from expression transformation)

The output of joiner transformation will be:

Column	O_total_records	O_dummy
A	1	5
B	2	5
C	3	5
D	4	5
E	5	5

Step4: Now pass the ouput of joiner transformation to filter transformation and specify the filter condition as O_total_records (port from aggregator) - O_count(port from expression) <=2

In the filter transformation, the filter condition will be O_total_records - O_count <=2

The output of filter transformation will be:

Column	O_total_records	O_dummy
C	3	5
D	4	5
E	5	5

Q4. **Design a mapping to load the first record from a flat file into one table A, the last record from a flat file into table B and the remaining records into table C?**

Ans This is similar to the above problem; the first 3 steps are same. In the last step instead of using the filter transformation, you have to use router transformation. In the router transformation create two output groups.

In the first group, the condition should be O_count = 1 and connect the corresponding output group to table A. In the second group, the condition should be O_count = O_total_records and connect the corresponding output group to table B. The output of default group should be connected to table C.

Q5. **Consider the following products data which contain duplicate rows.**

Column
A
B
C
C
B
D
B

1. Design a mapping to load all unique products in one table and the duplicate rows in another table.

The first table should contain the following output

Column
A
D

The second target should contain the following output

Column
B
B
B
C
C

Ans Use sorter transformation and sort the products data. Pass the output to an expression transformation and create a dummy port O_dummy and assign 1 to that port. So that, the DUMMY output port always return 1 for each row.

The output of expression transformation will be:

Product	O_dummy
A	1
B	1
B	1
B	1
C	1
C	1
D	1

Pass the output of expression transformation to an aggregator transformation. Check the group by on product port. In the aggreagtor, create an output port O_count_of_each_product and write an expression count(product).

The output of aggregator will be:

Product	O_count_of_each_product
A	1
B	3
C	2
D	1

Now pass the output of expression transformation, aggregator transformation to joiner transformation and join on the products port. In the joiner transformation checks the property sorted input, then only you can connect both expression and aggregator to joiner transformation.

The output of joiner will be:

Product	O_dummy	O_count_of_each_product
A	1	1
B	1	3
B	1	3
B	1	3
C	1	2
C	1	2
D	1	1

Now pass the output of joiner to a router transformation, create one group and specify the group condition as O_dummy = O_count_of_each_product. Then connect this group to one table. Connect the output of default group to another table.

Q6. Design a mapping to load each product once into one table and the remaining products which are duplicated into another table.
The first table should contain the following output:

Column
A
B
C
D

The second table should contain the following output

Column
B
B
C

Ans Use sorter transformation and sort the products data. Pass the output to an expression transformation and create a variable port, V_curr_product, and assign product port to it. Then create a V_count port and in the expression editor write IIF(V_curr_product = V_prev_product, V_count+1,1). Create one more variable port V_prev_port and assign product port to it. Now create an output port O_count port and assign V_count port to it.

In the expression transformation, the ports are
Product
V_curr_product=product
V_count = IIF (V_curr_product = V_prev_product, V_count+1,1)
V_prev_product = product
O_count = V_count

The output of expression transformation will be

Product	O_count
A	1
B	1
B	2
B	3
C	1
C	2
D	1

Now Pass the output of expression transformation to a router transformation, create one group and specify the condition as O_count=1. Then connect this group to one table. Connect the output of default group to another table.

Q7. Consider the following product types data as the source.

Product_Id	Product_Type
10	Video
10	Audio
20	Audio
30	Audio
40	Audio
50	Audio
10	Movie
20	Movie
30	Movie
40	Movie
50	Movie
60	Movie

Assume that there are only 3 product types are available in the source. The source contains 12 records and you don't know how many products are available in each product type.

1. **Design a mapping to select 9 products in such a way that 3 products should be selected from video, 3 products should be selected from Audio and the remaining 3 products should be selected from Movie.**

Ans

Step1: Use sorter transformation and sort the data using the key as product_type.

Step2: Connect the sorter transformation to an expression transformation. In the expression transformation, the ports will be
product_id
product_type
V_curr_prod_type=product_type
V_count = IIF(V_curr_prod_type = V_prev_prod_type,V_count+1,1)
V_prev_prod_type=product_type
O_count=V_count

Step3: Now connect the expression transformaion to a filter transformation and specify the filter condition as O_count<=3. Pass the output of filter to a target table.

2. **In the above problem Q1, if the number of products in a particular product type are less than 3, then you won't get the total 9 records in the target table. For example, see the videos type in the source data. Now design a mapping in such way that even if the number of products in a particular product type are less than 3, then you have to get those smaller / Lesser number of records from another product types. For example: If the number of products in videos are 1, then the remaining 2 records should come from audios or movies. So, the total number of records in the target table should always be 9.**

Ans **The first two steps are same as above.**

Step3: Connect the expression transformation to a sorter transformation and sort the data using the key as O_count. The ports in soter transformation will be
product_id
product_type
O_count (sort key)

Step4: Discard O_count port and connect the sorter transformation to an expression transformation. The ports in expression transformation will be
product_id
product_type
V_count=V_count+1
O_prod_count=V_count

Step5: Connect the expression to a filter transformation and specify the filter condition as O_prod_count<=9. Connect the filter transformation to a target table.

Q8. Design a mapping to convert column data into row data without using the normalizer transformation.

The source data looks like

Column1	Column1	Column1
A	B	C
D	E	F

The target table data should look like

Column
A
B
C
D
E
F

Ans Create three expression transformations with one port each. Connect col1 from Source Qualifier to port in first expression transformation. Connect col2 from Source Qualifier to port in second expression transformation. Connect col3 from source qualifier to port in third expression transformation. Create a union transformation with three input groups and each input group should have one port. Now connect the expression transformations to the input groups and connect the union transformation to the target table.

1. Design a mapping to convert row data into column data. **The source data looks like**

Id	Value
10	A
10	B
10	C
20	D
20	E
20	F

The target table data should look like

Id	Column1	Column1	Column1
10	A	B	C
20	D	E	F

Ans

Step1: Use sorter transformation and sort the data using id port as the key. Then connect the sorter transformation to the expression transformation.

Step2: In the expression transformation, create the ports and assign the expressions as mentioned below.
Id
Value
V_curr_id = id
V_count = IIF (v_curr_id=V_prev_id, V_count+1,1)
V_prev_id = id
O_col1 = IIF (V_count=1, value, NULL)
O_col2 = IIF (V_count=2, value, NULL)
O_col3 = IIF (V_count=3, value, NULL)

Step3: Connect the expression transformation to aggregator transformation. In the aggregator transformation, create the ports and assign the expressions as mentioned below.
id (specify group by on this port)
O_col1
O_col2
O_col3

Col1 = MAX (O_col1)
Col2 = MAX(O_col2)
Col3 = MAX (O_col3)

Stpe4: Now connect the ports id, col1, col2, col3 from aggregator transformation to the target table.

Q9. The source data contains only column 'id'. It will have sequence numbers from 1 to 1000.

The source data looks like as

Id
1
2
3
4
5
6
7
8

1000

Create a workflow to load only the Fibonacci numbers in the target table.

The target table data should look like as

Id
1
2
3
5
8
13

In Fibonacci series each subsequent number is the sum of previous two numbers. Here assume that the first two numbers of the Fibonacci series are 1 and 2.

Ans **STEP1:** Drag the source to the mapping designer and then in the Source Qualifier Transformation properties, set the number of sorted ports to one. This will sort the source data in ascending order. So that we will get the numbers in sequence as 1, 2, 3, ...1000

STEP2: Connect the Source Qualifier Transformation to the Expression Transformation. In the Expression Transformation, create three variable ports and one output port. Assign the expressions to the ports as shown below.

Ports in Expression Transformation:
id
v_sum = v_prev_val1 + v_prev_val2
v_prev_val1 = IIF (id=1 or id=2,1, IIF (v_sum = id, v_prev_val2, v_prev_val1))
v_prev_val2 = IIF (id=1 or id =2, 2, IIF (v_sum=id, v_sum, v_prev_val2))
o_flag = IIF (id=1 or id=2,1, IIF (v_sum=id,1,0))

STEP3: Now connect the Expression Transformation to the Filter Transformation and specify the Filter Condition as o_flag = 1

STEP4: Connect the Filter Transformation to the Target Table.

Q10. The source table contains two columns "id" and "val".
The source data looks like as below

Id	Value
1	A, B, C
2	PQ, M, N
3	ASZ, RO, LIQT

Here the "val" column contains comma delimited data and has three fields in that column. Create a workflow to split the fields in “val” column to separate rows. **The output should look like as below**.

Id	Value
1	A
1	B
1	C
2	PQ
2	M
2	N
3	ASZ
3	RO
3	LIQT

Ans **STEP1:** Connect three Source Qualifier transformations to the Source Definition

STEP2: Now connect all the three Source Qualifier transformations to the Union Transformation. Then connect the Union Transformation to the Sorter Transformation. In the sorter transformation sort the data based on Id port in ascending order.

STEP3: Pass the output of Sorter Transformation to the Expression Transformation.

The ports in Expression Transformation are:

id (input/output port)

val (input port)

v_currend_id (variable port) = id

v_count (variable port) = IIF (v_current_id != v_previous_id, 1, v_count+1)

v_previous_id (variable port) = id

o_val (output port) = DECODE (v_count, 1, SUBSTR (val, 1, INSTR (val,',',1, 1) - 1),

2, SUBSTR (val, INSTR (val, ', ', 1, 1) +1, INSTR (val, ', ', 1, 2) -INSTR (val, ', ', 1, 1) - 1),

3, SUBSTR (val, INSTR (val, ', ', 1, 2) + 1), NULL)

STEP4: Now pass the output of Expression Transformation to the Target definition. Connect id, o_val ports of Expression Transformation to the id, val ports of Target Definition.

For those who are interested to solve this problem in oracle SQL, Go to Page No **82 Q1.** The oracle sql query provides a dynamic solution where the "val" column can have varying number of fields in each row.

Q11. Take a look at the following tree structure diagram. From the tree structure, you can easily derive the parent-child relationship between the elements. For example, B is parent of D and E.

The above tree structure data is represented in a table as shown below.

Col1	Col2	Col3	Col4
A	B	D	H
A	B	D	I

A	B	E	NULL
A	C	F	NULL
A	C	G	NULL

Here in this table, column Col1 is parent of column Col2, column Col2 is parent of column Col3, column Col3 is parent of column Col4.

1. **Design a mapping to load the target table with the below data. Here you need to generate sequence numbers for each element and then you have to get the parent id. As the element "A" is at root, it does not have any parent and its parent_id is NULL.**

Id	Element	Parent_Id
1	A	NULL
2	B	1
3	C	1
4	D	2
5	E	2
6	F	3
7	G	3
8	H	4
9	I	4

I have provided the solution for this problem in Oracle SQL query. If you are interested you can Go to Page No **83 Q2.1** to see the solution.

2. **This is an extension to the problem 1. Let say column Col2 has null for all the rows, then Col1 becomes the parent of Col3 and col3 is parent of Col4. Let say both columns col2 and col3 has null for all the rows. Then col1 becomes the parent of col4. Design a mapping to accommodate these types of null conditions.**

Q12. In this problem we will see how to implement the not equal operator, greater than, greater than or equal to, less than and less than or equal to operators when joining two tables in informatica.

Consider the below sales table as an example

Table name: Sales

Product	Prod_Quantity	Price	Year
A	10	100	2010
B	15	150	2010
A	8	80	2011
B	26	260	2011

Now the problem is to identify the products whose sales is less than in the current year (In this example: 2011) when compared to the last year. Here in this example, Product A sold less in 2011 when compared with the sales in 2010.

This problem can be easily implemented with the help of SQL query as shown below

```
SELECT cy.*
FROM SALES cy, SALES py
WHERE cy.product = py.product
AND   cy.year=2011
AND   py.year=2010
AND   cy.prod_quantity < py.prod_quantity;
```

In informatica, you can specify only equal to condition in joiner. Now we will see how to implement this problem using informatica.

Ans **STEP1**: Connect two source qualifier transformations to the source definition. Call the first source qualifier transformation as sq_cy (cy means current year) and the other as sq_py (py means previous year).

STEP2: In the sq_cy source qualifier transformation, specify the source filter as price=2011. In the sq_py, specify the source filter as price = 2010

STEP3: Now connect these two source qualifier transformations to joiner transformation and make sq_cy as master, sq_py as detail. In the join condition, select the product port from master and detail.

STEP4: Now connect all the master ports and only the prod_quantity port from detail to the filter transformation. In the filter transformation specify the filter condition as prod_quantity < prod_quantity1. Here pord_quantity port is from master port and prod_quantity1 is from detail port.

STEP5: Connect all the ports except the prod_quantity1 of filter transformation to the target definition.

Q13. How to implement the not exists operator in informatica which is available in database?

Ans Implementing the Not Exists operator is very easy in informatica. For example, we want to get only the records which are available in table A and not in table B. For this use a joiner transformation with A as master and B as detail. Specify the join condition and in the join type, select detail outer join. This will get all the records from A table and only the matching records from B table.

Connect the joiner to a filter transformation and specify the filter condition as B_port is NULL. This will give the records which are in A and not in B. Then connect the filter to the target definition.

Q14. How to generate sequence numbers using expression transformation?

Ans In the expression transformation, create a variable port and iGuwahatiement it by 1. Then assign the variable port to an output port. In the expression transformation, the ports are:
V_count=V_count + 1
O_count=V_count

Q15. Design a mapping to load the first 3 rows from a flat file into a target?

Ans You have to assign row numbers to each record. Generate the row numbers either using the expression transformation as mentioned above or use sequence generator transformation. Then pass the output to filter transformation and specify the filter condition as O_count <=3

Q16. Design a mapping to load the last 3 rows from a flat file into a target?

Ans Consider the source has the following data.

Column
A
B
C
D
E

Step1: You have to assign row numbers to each record. Generate the row numbers using the expression transformation as mentioned above and call the row number generated port as O_count. Create a DUMMY output port in the same expression transformation and assign 1 to that port. So that, the DUMMY output port always return 1 for each row.

In the expression transformation, the ports are

V_count = V_count + 1

O_count = V_count

O_dummy = 1

The output of expression transformation will be

Column	O_Count	O_Dummy
A	1	1
B	2	1
C	3	1
D	4	1
E	5	1

Step2: Pass the output of expression transformation to aggregator and do not specify any group by condition. Create an output port O_total_records in the aggregator and assign O_count port to it. The aggregator will return the last row by default. The output of aggregator contains the DUMMY port which has value 1 and O_total_records port which has the value of total number of records in the source.

In the aggregator transformation, the ports are

O_dummy

O_count

O_total_records=O_count

The output of aggregator transformation will be

O_Total_Records	O_Dummy
5	1

Step3: Pass the output of expression transformation, aggregator transformation to joiner transformation and join on the DUMMY port. In the joiner transformation check the property sorted input, then only you can connect both expression and aggregator to joiner transformation.

In the joiner transformation, the join condition will be
O_dummy (port from aggregator transformation) = O_dummy (port from expression transformation)

The output of joiner transformation will be

Column	O_Count	O_Total_Records
A	1	5
B	2	5
C	3	5
D	4	5
E	5	5

Step4: Now pass the ouput of joiner transformation to filter transformation and specify the filter condition as O_total_records (port from aggregator)-O_count(port from expression) <=2

In the filter transformation, the filter condition will be
O_total_records - O_count <=2

The output of filter transformation will be

Column	O_Count	O_Total_Records
C	3	5
D	4	5
E	5	5

Q17. Design a mapping to load the first record from a flat file into one table A, the last record from a flat file into table B and the remaining records into table C?

Ans This is similar to the above problem; the first 3 steps are same. In the last step instead of using the filter transformation, you have to use router transformation. In the router transformation create two output groups.

In the first group, the condition should be O_count=1 and connect the corresponding output group to table A. In the second group, the condition should be O_count=O_total_records and connect the corresponding output group to table B. The output of default group should be connected to table C.

Q18. Consider the following products data which contain duplicate rows.

Column
A
B
C
C
B
D
B

1. Design a mapping to load all unique products in one table and the duplicate rows in another table.

The first table should contain the following output

Column
A
D

The second target should contain the following output

Column
B
B
B
C
C

Ans Use sorter transformation and sort the products data. Pass the output to an expression transformation and create a dummy port O_dummy and assign 1 to that port. So that, the DUMMY output port always return 1 for each row.

The output of expression transformation will be

Product	O_Dummy
A	1
B	1
B	1
B	1
C	1
C	1
D	1

Pass the output of expression transformation to an aggregator transformation. Check the group by on product port. In the aggregator, create an output port O_count_of_each_product and write an expression count(product).

The output of aggregator will be

Product	O_Dummy
A	1
B	3
C	2
D	1

Now pass the output of expression transformation, aggregator transformation to joiner transformation and join on the products port. In the joiner transformation check the property sorted input, then only you can connect both expression and aggregator to joiner transformation.

The output of joiner will be

Product	O_Dummy	O_Count_of_Each_Product
A	1	1
B	1	3
B	1	3
B	1	3
C	1	2
C	1	2
D	1	1

Now pass the output of joiner to a router transformation, create one group and specify the group condition as O_dummy = O_count_of_each_product. Then connect this group to one table. Connect the output of default group to another table.

1. Design a mapping to load each product once into one table and the remaining products which are duplicated into another table.
The first table should contain the following output

Column
A
B
C
D

The second table should contain the following output

Column
B
B
C

Ans Use sorter transformation and sort the products data. Pass the output to an expression transformation and create a variable port, V_curr_product, and assign product port to it. Then create a V_count port and in the expression editor write IIF (V_curr_product = V_prev_product, V_count + 1, 1). Create one more variable port V_prev_port and assign product port to it. Now create an output port O_count port and assign V_count port to it.

In the expression transformation, the ports are
Product
V_curr_product = product
V_count= IIF (V_curr_product = V_prev_product, V_count + 1, 1)
V_prev_product = product
O_count = V_count

The output of expression transformation will be

Product	O_Count
A	1
B	1
B	2
B	3
C	1
C	2
D	1

Now Pass the output of expression transformation to a router transformation, create one group and specify the condition as O_count = 1. Then connect this group to one table. Connect the output of default group to another table.

Informatica Interview Questions

Q1. While importing the relational source definition from the database, what are the metadata of source that will be imported?

Ans The metadata of the source that will be imported are:

- Source name.
- Database location.
- Column names.
- Data types.
- Key constraints.

Q2. How many ways a relational source definition can be updated and what are they?

Ans There are two ways to update the relational source definition:

- Edit the definition.
- Re-import the definition.

Q3. To import the flat file definition into the designer where should the flat file be placed?

Ans Place the flat file in local folder in the local machine

Q4. To provide support for Mainframe's source data, which files are used as a source definition?

Ans COBOL files

Q5. Which transformation is needed while using the cobol sources as source definitions?

Ans As cobol sources consists of denormalized data, normalizer transformation is required to normalize the data.

Q6. How to create or import flat file definition in to the warehouse designer?

Ans We cannot create or import flat file definition into warehouse designer directly. We can create or import the file in source analyzer and then drag it into the warehouse designer.

Q7. What is a mapplet?

Ans A mapplet is a set of transformations that you build in the mapplet designer and can be used in multiple mappings.

Q8. What is a transformation?

Ans It is a repository object that generates, modifies or passes data.

Q9. What are the designer tools for creating transformations?

Ans

- Mapping Designer.
- Transformation Developer.
- Mapplet Designer.

Q10. What are active and passive transformations?

Ans An active transformation can change the number of rows that pass through it. A passive transformation does not change the number of rows that pass through it.

Q11. What are connected or unconnected transformations?

Ans An unconnected transformation is not connected to other transformations in the mapping. Connected transformation is connected to other transformations in the mapping pipeline.

Q12. How many ways are there to create ports?

Ans There are two ways to create the ports:

- Drag the port from another transformation
- Click the add button on the ports tab.

Q13. What are the reusable transformations?

Ans Reusable transformations can be used in multiple mappings and mapplets. When you need to include this transformation into a mapping or a mapplet, an instance of it is dragged into the mapping or mapplet. Since, the instance of reusable transformation is a pointer to that transformation, any change in the reusable transformation will be inherited by all the instances.

Q14. What are the methods for creating reusable transformations?

Ans Two methods:

- Design it in the transformation developer.
- Promote a standard transformation (Non reusable) from the mapping designer. After adding a transformation to the mapping, we can promote it to the status of reusable transformation.

Q15. What are the unsupported repository objects for a mapplet?

Ans

- COBOL Source Definition.
- Joiner Transformations.
- Normalizer Transformations.
- Non reusable Sequence Generator Transformations.
- Pre or Post Session Stored Procedures.
- Target Definitions.
- Power Mart 3.5 style Look Up functions.
- XML Source Definitions.
- IBM MQ source Definitions.

Q16. What are the mapping parameters and mapping variables?

Ans

- Mapping parameter represents a constant value which is defined before running a session. A mapping parameter retains the same value throughout the entire session. A parameter can be declared either in a mapping or mapplet and can have a default value. We can specify the value of the parameter in the parameter file and the session reads the parameter value from the parameter file.
- Unlike a mapping parameter, a mapping variable represents can change throughout the session. The informatica server saves the value of mapping variable in the repository at the end of session run and uses that value next time when the session runs.

Q17. Can we use the mapping parameters or variables created in one mapping into another mapping?

Ans **NO**. We can use the mapping parameters or variables only in the transformations of the same mapping or mapplet in which we have created the mapping parameters or variables.

Q18. Can we use the mapping parameters or variables created in one mapping into any other reusable Transformation?

Ans **Yes**. As an instance of the reusable transformation created in the mapping belongs to that mapping only.

Q19. How can we improve session performance in Aggregator Transformation?

Ans Use sorted input. Sort the input on the ports which are specified as group by ports in aggregator.

Q20. What is aggregate cache in Aggregator Transformation?

Ans The aggregator stores data in the aggregate cache until it completes aggregate calculations. When we run a session that uses an aggregator transformation, the informatica server creates index and data caches in memory to process the transformation. If the informatica server requires more space, it stores overflow values in cache files.

Q21. What are the differences between joiner transformation and source qualifier transformation?

Ans A joiner transformation can join heterogeneous data sources where as a source qualifier can join only homogeneous sources. Source qualifier transformation can join data from only relational sources but cannot join flat files.

Q22. What are the limitations of joiner transformation?

Ans

- Both pipelines begin with the same original data source.
- Both input pipelines originate from the same Source Qualifier transformation.
- Both input pipelines originate from the same Normalizer transformation.
- Both input pipelines originate from the same Joiner transformation.
- Either input pipelines contain an Update Strategy transformation.
- Either input pipelines contain a connected or unconnected Sequence Generator transformation.

Q23. What are the settings that you use to configure the joiner transformation?

Ans The following settings are used to configure the joiner transformation.

- Master and detail source.
- Type of join.
- Condition of the join.

Q24. What are the join types in joiner transformation?

Ans The join types are

- Normal (Default) Join.
- Master outer Join.
- Detail outer Join.
- Full outer Join.

Q25. What are the joiner caches?

Ans When a Joiner transformation occurs in a session, the Informatica Server reads all the records from the master source and builds index and data caches based on the master rows. After building the caches, the Joiner transformation reads records from the detail source and performs joins.

Q26. What is the look up transformation?

Ans Lookup transformation is used to lookup data in a relational table, view and synonym. Informatica server queries the look up table based on the lookup ports in the transformation. It compares the lookup transformation port values to lookup table column values based on the look up condition.

Q27. Why use the lookup transformation?

Ans Lookup transformation is used to perform the following tasks.

- Get a related value.
- Perform a calculation.
- Update slowly changing dimension tables.

Q28. What are the types of lookup transformation?

Ans The types of lookup transformation are Connected and unconnected.

Q29. What is meant by lookup caches?

Ans The informatica server builds a cache in memory when it processes the first row of a data in a cached look up transformation. It allocates memory for the cache based on the amount you configure in the transformation or session properties. The informatica server stores condition values in the index cache and output values in the data cache.

Q30. What are the types of lookup caches?

Ans

- **Persistent cache**: You can save the lookup cache files and reuse them the next time the informatica server processes a lookup transformation configured to use the cache.
- **Re-cache from database**: If the persistent cache is not synchronized with the lookup table, you can configure the lookup transformation to rebuild the lookup cache.
- **Static cache**: you can configure a static or read only cache for only lookup table. By default, informatica server creates a static cache. It caches the lookup table and lookup values in the cache for each row that comes into

the transformation. When the lookup condition is true, the informatica server does not update the cache while it processes the lookup transformation.

- **Dynamic cache**: If you want to cache the target table and insert new rows into cache and the target, you can create a look up transformation to use dynamic cache. The informatica server dynamically inserts data to the target table.
- **Shared cache**: You can share the lookup cache between multiple transactions. You can share unnamed cache between transformations in the same mapping.

Q31. Which transformation should we use to normalize the COBOL and relational sources?

Ans Normalizer Transformation is used to normalize the data.

Q32. In which transformation you cannot drag ports into it?

Ans Normalizer Transformation.

Q33. How the informatica server sorts the string values in Rank transformation?

Ans When the informatica server runs in the ASCII data movement mode it sorts session data using Binary sort order. If you configure the session to use a binary sort order, the informatica server calculates the binary value of each string and returns the specified number of rows with the highest binary values for the string.

Q34. What are the rank caches?

Ans During the session, the informatica server compares an input row with rows in the data cache. If the input row out-ranks a stored row, the informatica server replaces the stored row with the input row. The informatica server stores group information in an index cache and row data in a data cache.

Q35. What is the Rank index port in Rank transformation?

Ans The Designer automatically creates a RANKINDEX port for each Rank transformation. The Informatica Server uses the Rank Index port to store the ranking position for each record in a group.

Q36. What is the Router transformation?

Ans A Router transformation is similar to a Filter transformation because both transformations allow you to use a condition to test data. However, a Filter transformation tests data for one condition and drops the rows of data that do not meet the condition. A Router transformation tests data for one or more conditions and gives you the option to route rows of data that do not meet any of the conditions to a default output group. If you need to test the same input data based on multiple conditions, use a Router Transformation in a mapping instead of creating multiple Filter transformations to perform the same task.

Q37. What are the types of groups in Router transformation?

Ans The different types of groups in router transformation are

- Input group.
- Output group.

The output group contains two types. They are

- User defined groups.
- Default group.

Q38. What are the types of data that passes between informatica server and stored procedure?

Ans Three types of data passes between the informatica server and stored procedure.

- Input/Output parameters.
- Return Values.
- Status code.

Q39. What is the status code in stored procedure transformation?

Ans Status code provides error handling for the informatica server during the session. The stored procedure issues a status code that notifies whether or not stored procedure completed successfully. This value cannot seen by the user. It only used by the informatica server to determine whether to continue running the session or stop.

Q40. What is the target load order?

Ans You can specify the target load order based on source qualifiers in a mapping. If you have the multiple source qualifiers connected to the multiple targets, you can designate the order in which informatica server loads data into the targets.

Q41. What is polling?

Ans Polling displays the updated information about the session in the monitor window. The monitor window displays the status of each session when you poll the informatica server.

Q42. In which circumstances, informatica server creates Reject files?

Ans When the informatica server encounters the DD_Reject in update strategy transformation, violates the database constraints, filed in the rows were truncated or overflowed.

Q43. What are the data movement modes in informatica?

Ans Data movement mode determines how informatica server handles the character data. You can choose the data movement mode in the informatica server configuration settings. Two types of data movement modes are available in informatica. They are ASCII mode and Unicode mode.

Q44. Define mapping and session?

Ans

- **Mapping**: It is a set of source and target definitions linked by transformation objects that define the rules for transformation.
- **Session**: It is a set of instructions that describe how and when to move data from source to targets.

Q45. Can u generate reports in Informatica?

Ans Yes. By using Metadata reporter, we can generate reports in informatica.

Q46. What is metadata reporter?

Ans It is a web-based application that enables you to run reports against repository metadata. With a metadata reporter, you can access information about the repository without having knowledge of SQL, transformation language or underlying tables in the repository.

Q47. What is the default source option for update strategy transformation?

Ans Data Driven.

Q48. What is Data driven?

Ans The informatica server follows the instructions coded in the update strategy transformations with in the mapping and determines how to flag the records for insert, update, delete or reject. If you do not choose data driven option setting, the informatica server ignores all update strategy transformations in the mapping.

Q49. What is source qualifier transformation?

Ans When you add a relational or a flat file source definition to a mapping, you need to connect it to a source qualifier transformation. The source qualifier transformation represents the records that the informatica server reads when it runs a session.

Q50. What are the tasks that source qualifier perform?

Ans

- Joins the data originating from same source data base.
- Filter records when the informatica server reads source data.
- Specify an outer join rather than the default inner join
- Specify Sorted Records.
- Select only distinct values from the source.
- Create custom query to issue a special SELECT statement for the informatica server to read the source data.

Q51. What is the default Join that Source Qualifier Provides?

Ans Equi Join

Q52. What are the basic requirements to join two sources in a source qualifier transformation using default join?

Ans

- The two sources should have primary key and foreign key relationship.
- The two sources should have matching data types.

Get Previous Row Value

Retrieve the previous row value when processing the current row.

Q1. How to get the previous row value while processing the current row in Informatica?

The source data is shown below:

Table Name: Customers

Cust_Id	Year	City
10	2001	Bangalore
10	2002	Mumbai
10	2003	Pune
10	2004	New York
20	2001	Delhi
20	2002	Guwahati
20	2003	Hyderabad

The question is for each customer when processing the record for current row, you have to get the previous row city value. If there is no previous row, then make the previous row value as null.

The output data is shown below:

Table Name: Customers_TGT

Cust_Id	Year	City	Previous_City
10	2001	Bangalore	Null
10	2002	Mumbai	Bangalore
10	2003	Pune	Mumbai
10	2004	New York	Pune
20	2001	Delhi	Null
20	2002	Guwahati	Delhi
20	2003	Hyderabad	Guwahati

Getting Previous Row Value Informatica Mapping Logic

Ans Connect the source qualifier transformation to the sorter transformation and sort the data on cust_id, year ports in ascending order.

Connect the sorter transformation to the expression transformation. In the expression transformation, create the below additional ports and assign the corresponding expressions:
cust_id (input/output port)
year (input/output port)
city (input/output port)
v_current_cust_id (variable port) = cust_id
v_act_previous_city (variable port) = IIF (v_current_cust_id = v_previous_cust_id, v_previous_city, NULL)
v_previous_city (variable port) = city
v_previous_cust_id (variable port) = cust_id
o_previous_city (output port) = v_act_previous_city

Connect the output ports of expression transformation to the target.

Load all rows except last N rows

Skip Last N rows from the source and load the remaining rows

Q1. I want to load all the records from my source, which is a file, except the last 5 records. This question can be asked interview as "How to remove the footer record which is last record"
For Example: My source file contains the following records:

Name
A
B
C
D
E
F
G

After excluding the last 5 records, i want to load A,B into the target.
How to implement a mapping logic for this in informatica?

Ans

- Connect the source qualifier transformation, NEXTVAL port of sequence generator to the sorter transformation.

- In the sorter transformation, check the key box corresponding to NEXTVAL port and change the direction to Descending.
- Create one more sequence generator transformation and a filter transformation.
- Connect the NEXTVAL port of the second sequence generator transformation to the filter and Name port of sorter transformation to filter.
- Specify the filter condition as NEXTVAL > 5.
- Save the mapping. Create a workflow and session. Save the workflow and run the workflow.

You can use the same approach to remove the footer record from the source by specifying the filter condition as NEXVAL > 1. If you have any issues in solving this problem, please do comment here.

Cumulative Sum Calculation

Find the cumulative sum of values using expression transformation.

Q1. How to find the cumulative sum of salaries of employees in informatica?

I have employees table as a source. The data in the employees table is shown below:

Table name: Employees

Dept_Id	Emp_Id	Salary
10	201	10000
10	202	20000
10	203	30000
20	301	40000
20	302	50000

I want to sort the data on the department id, employee id and then find the cumulative sum of salaries of employees in each department.

The output I shown below:

Dept_Id	Emp_Id	Salary	Cum_Salary
10	201	10000	10000
10	202	20000	30000
10	203	30000	60000
20	301	40000	40000
20	302	50000	90000

Ans Follow the below steps for implementing mapping logic in informatica. Connect the source qualifier transformation to a sorter transformation. Sort the rows on the dept_id and emp_id ports in ascending order. Connect the sorter transformation to the expression transformation. In the expression transformation, create the following additional ports and assign the corresponding expressions:

v_salary (variable port) = IIF (dept_id = v_last_dept_id, v_salary + salary, salary)
v_last_dept_id (variable port) = dept_id
o_cum_salary (output port) = v_salary

Connect the expression transformation ports to the target. Save the mapping.

Convert multiple Rows to single row (multiple Columns) in Informatica

Concatenate multiple rows into a single column.

Q1. I have the sales table as a source. The sales table contains the sales information of products for each year and month. The data in the source **table is shown below:**

Source Data: Sales Table

Year_Product	Month	Amount
1999 A	Jan	9600
1999 A	Feb	2000
1999 A	Mar	2500
2001 B	Jan	3000
2001 B	Feb	3500
2001 B	Mar	4000

The sales information of a product for each month is available in a separate row. I want to convert the rows for all the months in a specific year to a single row. **The output is shown below:**

Target Data: TGT_ Sales Table

Year_Product	Jan_Month	Feb_Month	March_Month
1999 A	9600	2000	2500
2001 B	3000	3500	4000

How to implement a mapping logic for this in informatica?

Ans Follow the below steps to implement the mapping logic for the above scenario in informatica:

- Create a new mapping.
- Drag the source into the mapping.
- Create an expression transformation.
- Drag the ports of source qualifier into the expression transformation.
- Create the below additional ports in the expression transformation and assign the corresponding expressions:
  Jan_Month (output port) = IIF (month='Jan', amount, null)
  Feb_Month (output port) = IIF (month='Feb', amount, null)
  Mar_Month (output port) = IIF (month='Mar', amount, null)
- Connect the expression transformation to an aggregator transformation. Connect only the ports year, product, Jan_Month, Feb_Month,Mar_Month ports of expression to aggregator transformation. Group by on year and product in aggregator transformation.
- Create the below additional ports in aggregator transformation and assign the corresponding expressions:
  o_Jan_Month (output port) = MAX (Jan_Month)
  o_Feb_Month (output port) = MAX (Feb_Month)
  o_Mar_Month (output port) = MAX (Mar_Month)
- Now connect the ports year, product, o_Jan_Month, o_Feb_Month, o_Mar_Month of aggregator transformation to the target.
- Save the mapping.

Load Last N Records of File into Target Table

Load the last N rows from the source into the target.

Q1. How to load only the last N rows from source file into the target table using the mapping in informatica?

First take a look at the below data in the source file:

Source Data: Products

Product
Windows
Linux
Unix
Ubuntu
Fedora
Centos
Debian

I want to load only the last record or footer into the target table. The target should contain only the product "Debain".

Follow the below steps for implementing the mapping logic in informatica

- **The mapping flow and the transformations are shown below:** SRC->SQ->EXPRESSION->SORTER->EXPRESSION->FILTER->TGT
- Create a new mapping and drag the source into the mapping. By default, it creates the source qualifier transformation.
- Now create an expression transformation and drag the ports from source qualifier into the expression transformation. In the expression transformation, create the below additional ports and assign the corresponding expressions:
  v_count (variable port) = v_count + 1
  o_count (output port) = v_count

- **The output of Expression Transformation is**

Products	O_Count
Windows	1
Linux	2
Unix	3
Ubuntu	4
Fedora	5
Centos	6
Debian	7

Now connect the expression transformation to a sorter transformation and sort the rows on the o_count port in descending order.

The output of sorter transformation is shown below:

Products
Debian
Centos
Fedora
Ubuntu
Unix
Linux
Windows

- Create another expression transformation and connect the Products port of sorter to expression transformation.
- Create the following ports in the expression transformation:
  v_count (variable port) = v_count + 1
  o_count (output port) = v_count
- Connect the expression to a filter transformation and specify the filter condition as o_count = 1.

Connect the filter to the target and save the mapping.

Load Alternative Records into Multiple Targets

Load odd number rows into one target and even numbered rows into another target.

Q1. How to load records alternatively into multiple targets in informatica? Implement mapping logic for this.

I have a source file which contains N number of records. I want to load the source records into two targets, such that first row goes into target 1, second row goes into target2, third row goes into target3 and so on.

Let see how to create a mapping logic for this in informatica with an example. **Consider the following source flat file as an example:**

Products
Informatica
Datastage
Pentaho
MSBI
Oracle
Mysql

The data in the targets should be:

Target1

Products
Informatica
Pentaho
Oracle

Target2

Products
Datastage
MSBI
Mysql

Ans The mapping flow and the transformations used are mentioned below:
SRC->SQ->EXP->RTR->TGTS

- First create a new mapping and drag the source into the mapping.
- Create an expression transformation. Drag the ports of source qualifier into the expression transformation. Create the following additional ports and assign the corresponding expressions:
  v_count (variable port) = v_count + 1
  o_count (output port) = v_count
- Create a router transformation and drag the ports (products, v_count) from expression transformation into the router transformation. Create an output group in the router transformation and specify the following filter condition:
  MOD(o_count,2) = 1
- Now connect the output group of the router transformation to the target1 and default group to target2. Save the mapping.

In the above solution, I have used expression transfromation for generating numbers. You can also use sequence generator transformation for producing sequence values.

Load Source File Name in Target

Generate the source file name and load into the target.

Q1. How to load the name of the current processing flat file along with the data into the target using informatica mapping?

We will create a simple pass-through mapping to load the data and "file name" from a flat file into the target. Assume that we have a source file "customers" and want to load this data into the target "customers_tgt". **The structures of source and target are**

Source File Name: Customers.dat

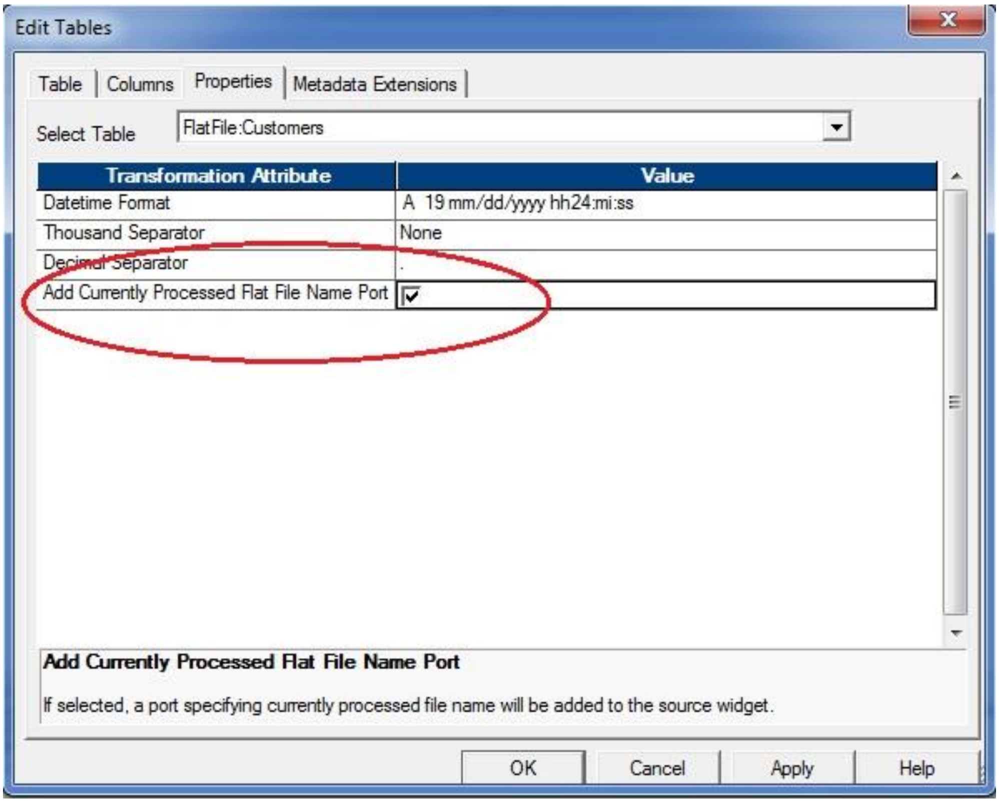
Column
Customer_Id
Location

Target: Customers_TBL

Column
Customer_Id
Location
FileName

The steps involved are:

- Login to the PowerCenter mapping designer and go to the source analyzer. You can create the flat file or import the flat file.
- Once you created a flat file, edit the source and go to the properties tab. Check the option "Add Currently Processed Flat File Name Port". This option is shown in the below image.



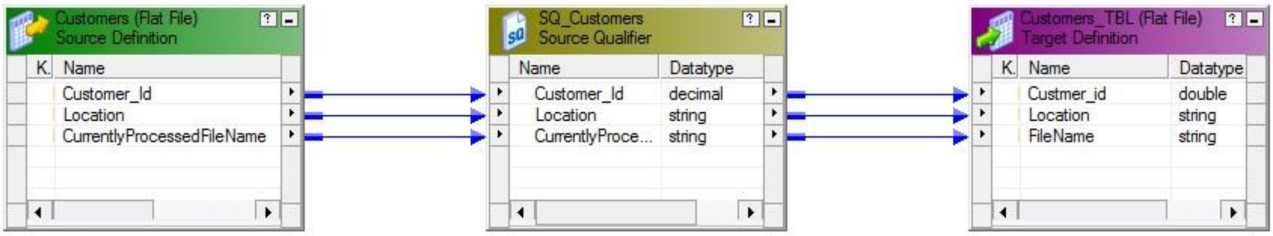
Select Table: FlatFile:Customers

Transformation Attribute	Value
Datetime Format	A 19 mm/dd/yyyy hh24:mi:ss
Thousand Separator	None
Decimal Separator	.
Add Currently Processed Flat File Name Port	☑

Add Currently Processed Flat File Name Port

If selected, a port specifying currently processed file name will be added to the source widget.

- A new port, "CurrentlyProcessedFileName" is created in the ports tab.
- Now go to the Target Designer or Warehouse Designer and create or import the target definition. Create a "Filename" port in the target.
- Go to the Mapping designer tab and create new mapping.
- Drag the source and target into the mapping. Connect the appropriate ports of source qualifier transformation to the target.
- Now create a workflow and session. Edit the session and enter the appropriate values for source and target connections.
- The mapping flow is shown in the below image



Customers (Flat File)
Source Definition

K.	Name
	Customer_Id
	Location
	CurrentlyProcessedFileName

SQ_Customers
Source Qualifier

Name	Datatype
Customer_Id	decimal
Location	string
CurrentlyProce...	string

Customers_TBL (Flat File)
Target Definition

K.	Name	Datatype
	Custmer_id	double
	Location	string
	FileName	string

The loading of the filename works for both Direct and Indirect Source filetype. After running the workflow, the data and the filename will be loaded in to the target. The important point to note is the complete path of the file will be loaded into the target. This means that the directory path and the filename will be loaded (**For Example**: /informatica/10.2/SrcFiles/Customers.dat).

If you don’t want the directory path and just want the filename to be loaded in to the target, then follow the below steps:

- Create an expression transformation and drag the ports of source qualifier transformation into it.
- Edit the expression transformation, go to the ports tab, create an output port and assign the below expression to it.

  REVERSE (SUBSTR (REVERSE (CurrentlyProcessedFileName), 1, INSTR (REVERSE (CurrentlyProcessedFileName), '/') -))

- Now connect the appropriate ports of expression transformation to the target definition.

Generate rows based on a column value – Informatica

This is to duplicate each row based on the value in the column.

Q1. How to generate or load values in to the target table based on a column value using informatica ETL tool.

I have the products table as the source and the data of the products table is shown below.

Table Name: Products

Product	Quantity
Samsung	Null
Iphone	3
LG	0
Nokia	4

Now I want to duplicate or repeat each product in the source table as many times as the value in the quantity column. **The output is**

Product	Quantity
Iphone	3
Iphone	3
Iphone	3
Nokia	4
Nokia	4
Nokia	4
Nokia	4

The Samsung and LG products should not be loaded as their quantity is NULL, 0 respectively.

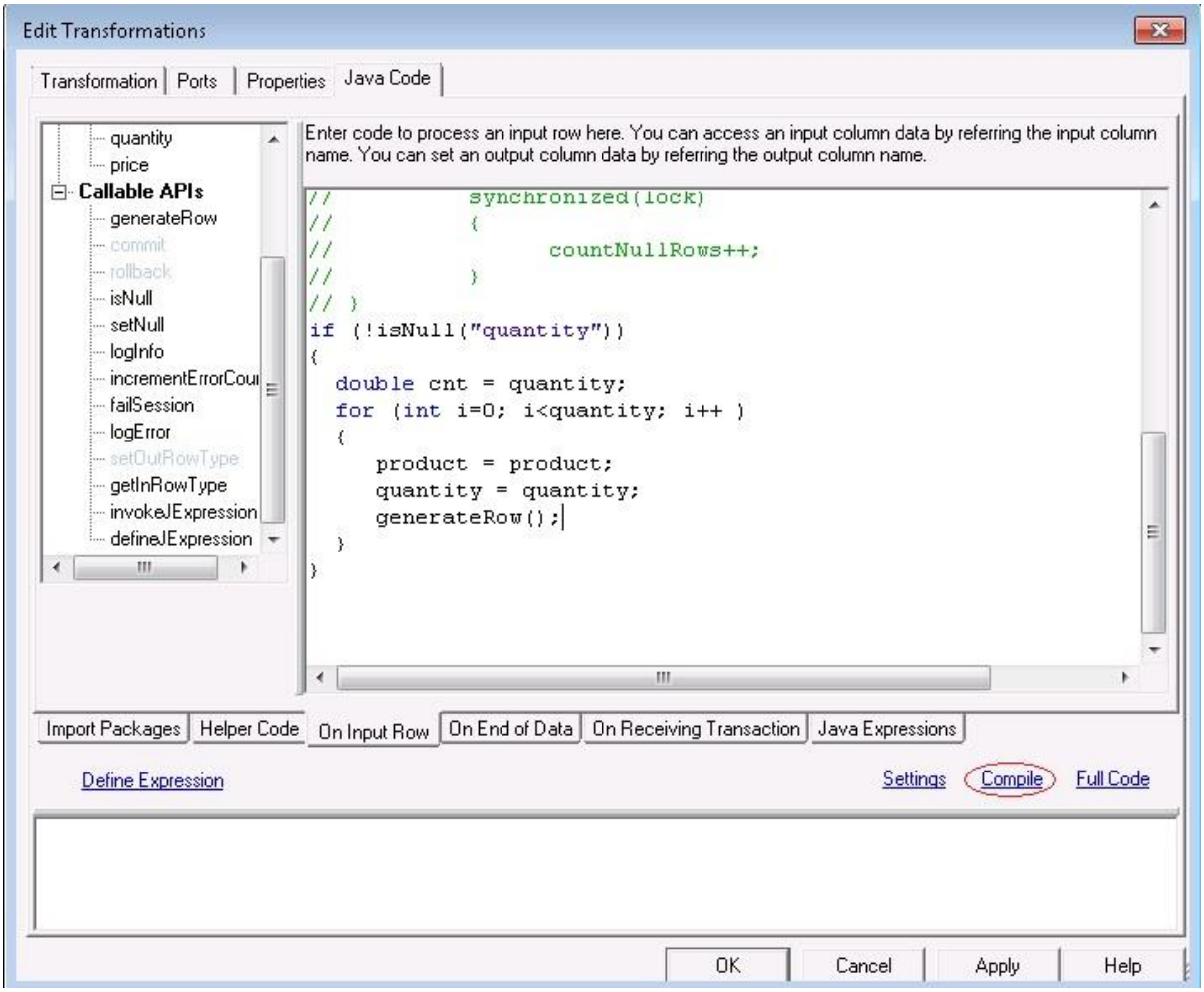
Now create informatica workflow to load the data in to the target table?

Ans Follow the below steps

- Create a new mapping in the mapping designer.
- Drag the source definition in to the mapping.
- Create the java transformation in active mode.
- Drag the ports of source qualifier transformation in to the java transformation.

- Now edit the java transformation by double clicking on the title bar of the java transformation and go to the "Java Code" tab.
- Enter the below java code in the "Java Code" tab.

```
If (!ISNULL("Quantity"))
{
double cnt = quantity;
for (int i = 1; i <= quantity; i++)
{
product = product;
quantity = quantity;
generateRow ();
} }
```



Enter code to process an input row here. You can access an input column data by referring the input column name. You can set an output column data by referring the output column name.

```
//          synchronized(lock)
//          {
//                countNullRows++;
//          }
// }
if (!isNull("quantity"))
{
  double cnt = quantity;
  for (int i=0; i<quantity; i++ )
  {
     product = product;
     quantity = quantity;
     generateRow();
  }
}
```

- Now compile the java code. The compile button is shown in red circle in the image.
- Connect the ports of the java transformation to the target.
- Save the mapping, create a workflow and run the workflow.

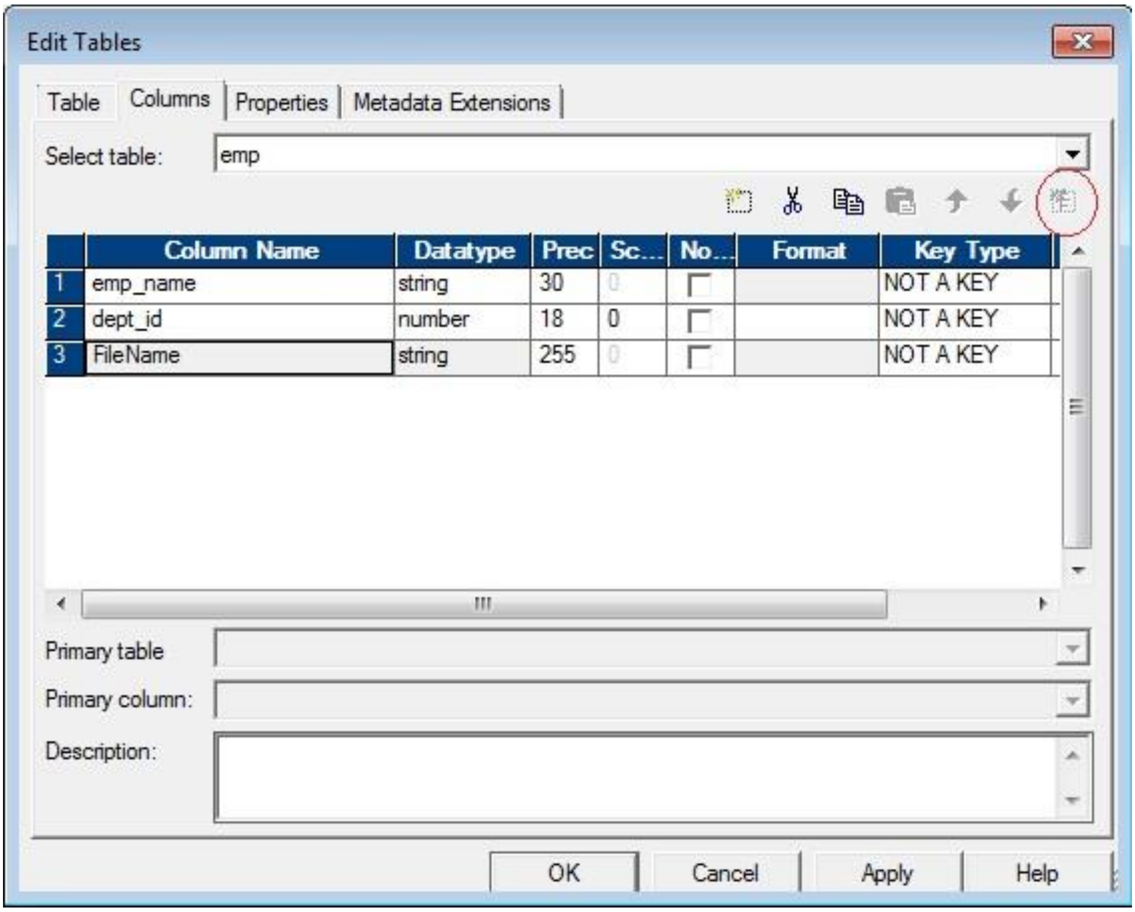
Dynamic Target Flat File Name Generation

Without specifying the target file name, generate dynamically in the mapping.

Informatica **8.x** or later versions provides a feature for generating the target files dynamically. This feature allows you to

- Create a new file for every session run.
- Create a new file for each transaction.

Informatica provides a special port,"FileName" in the Target file definition. This port you have to add explicitly. See the below diagram for adding the "FileName" port.



Select table: emp

	Column Name	Datatype	Prec	Sc...	No...	Format	Key Type
1	emp_name	string	30	0	□		NOT A KEY
2	dept_id	number	18	0	□		NOT A KEY
3	FileName	string	255	0	□		NOT A KEY

Primary table

Primary column:

Description:

Go to the Target Designer or Warehouse builder and edit the file definition. You have to click on the button indicated in red color circle to add the special port.

Now we will see some informatica mapping examples for creating the target file name dynamically and load the data.

1. Generate a new file for every session run.

Whenever the session runs you need to create a new file dynamically and load the source data into that file. To do this just follow the below steps:

STEP1: Connect the source qualifier to an expression transformation. In the expression transformation create an output port (call it as File_Name) and assign the expression as 'EMP_'|| to_char (sessstarttime, 'YYYYMMDDHH24MISS')||'.dat'
STPE2: Now connect the expression transformation to the target and connect eh File_Name port of expression transformation to the FileName port of the target file definition.

STEP3: Create a workflow and run the workflow.

Here I have used sessstarttime, as it is constant throughout the session run. If you have used sysdate, a new file will be created whenever a new transaction occurs in the session run.

The target file names created would look like **EMP_20120101125040.dat**.

2. Create a new file for every session run. The file name should contain suffix as numbers (EMP_n.dat)

In the above mapping scenario, the target flat file name contains the suffix as 'timestamp.dat'. Here we have to create the suffix as a number. So, the file names should look as EMP_1.dat, EMP_2.dat and so on. Follow the below steps:

STPE1: Go the mappings parameters and variables -> Create a new variable, $$COUNT_VAR and its data type should be Integer

STPE2: Connect the source Qualifier to the expression transformation. In the expression transformation create the following new ports and assign the expressions.
v_count (variable port) = v_count + 1
v_file_count (variable port) = IIF (v_count = 1,
SETVARIABLE ($$COUNT_VAR, $$COUNT_VAR + 1), $$COUNT_VAR)
o_file_name (output port) = 'EMP_'||v_file_count||'.dat'

STEP3: Now connect the expression transformation to the target and connect the o_file_name port of expression transformation to the FileName port of the target.

3. Create a new file once a day.

You can create a new file only once in a day and can run the session multiple times in the day to load the data. You can either overwrite the file or append the new data.
This is similar to the first problem. Just change the expression in expression transformation to 'EMP_'|| to_char (sessstarttime, 'YYYYMMDD')||'.dat'. To avoid overwriting the file, use Append If Exists option in the session properties.

4. Create a flat file based on the values in a port.

You can create a new file for each distinct values in a port. As an example consider the employees table as the source. I want to create a file for each department id and load the appropriate data into the files.

STEP1: Sort the data on department_id. You can either use the source qualifier or sorter transformation to sort the data.

STEP2: Connect to the expression transformation. In the expression transformation create the below ports and assign expressions.
v_curr_dept_id (variable port) = dept_id
v_flag (variable port) = IIF (v_curr_dept_id = v_prev_dept_id, 0, 1)
v_prev_dept_id (variable port) = dept_id
o_flag (output port) = v_flag
o_file_name (output port) = dept_id||'.dat'

STEP4: Now connect the expression transformation to the transaction control transformation and specify the transaction control condition as
IIF (o_flag = 1, TC_COMMIT_BEFORE, TC_CONTINUE_TRANSACTION)

STEP5: Now connect to the target file definition.

Reverse the Contents of Flat File

Last row should become first row and first row should become last row in the target.

Q1. I have a flat file, want to reverse the contents of the flat file which means the first record should come as last record and last record should come as first record and load into the target file.

As an example, consider the source flat file data as

Informatica Enterprise Solution.
Informatica Power center.
Informatica Power exchange.
Informatica Data quality.

The target flat file data should look as

Informatica Data quality.
Informatica Power exchange.
Informatica Power center.
Informatica Enterprise Solution.

Ans Follow the below steps for creating the mapping logic

- Create a new mapping.
- Drag the flat file source into the mapping.
- Create an expression transformation and drag the ports of source qualifier transformation into the expression transformation.
- Create the below additional ports in the expression transformation and assign the corresponding expressions
  Variable port: v_count = v_count + 1
  Output port o_count = v_count
- Now create a sorter transformation and drag the ports of expression transformation into it.
- In the sorter transformation specify the sort key as o_count and sort order as DESCENDING.
- Drag the target definition into the mapping and connect the ports of sorter transformation to the target.

Q2. Load the header record of the flat file into first target, footer record into second target and the remaining records into the third target.

The solution to this problem I have already posted by using aggregator and joiner. Now we will see how to implement this by reversing the contents of the file.

Ans

- Connect the source qualifier transformation to the expression transformation. In the expression transformation create the additional ports as mentioned above.
- Connect the expression transformation to a router. In the router transformation create an output group and specify the group condition as o_count = 1. Connect this output group to a target and the default group to sorter transformation.
- Sort the data in descending order on o_count port.
- Connect the output of sorter transformation to expression transformation (don't connect o_count port).
- Again, in the expression transformation create the same additional ports mentioned above.
- Connect this expression transformation to router and create an output group. In the output group specify the condition as o_count = 1 and connect this group to second target. Connect the default group to the third group.

Q1. Alternate Target Loading

My source is a flat file which contains N number of records. I want to load the source data into two targets such that first five records should loaded into the first target, next five records into the second target table. Again the next source five records into the first target table and so on. How to implement a Informatica mapping logic for this?

Ans

- Connect the source qualifier transformation to the expression transformation. In the expression transformation, create the below additional ports:
  v_cnt (variable port) = v_cnt + 1
  o_cnt (output port) = v_cnt

- Connect the expression transformation to the router transformation. Create two output groups in the router transformation and specify the following filter conditions:

--Filter condition for first output group
DECODE (substr (o_cnt, - 1, 1), 1, TRUE, 2, TRUE, 3, TRUE, 4, TRUE, 5, TRUE, FALSE)
--Filter condition for second output group
DECODE (substr (o_cnt, - 1, 1), 6, TRUE, 7, TRUE, 8, TRUE, 9, TRUE, 0, TRUE, FALSE)

- Connect the router transformation output groups to the appropriate targets.

Q2. Load source data in multiple session run.
I have flat file as a source which contains N number of records. My requirement is to load half of the source data into the target table in the first session run and the remaining half of the records in the second session run. Create Informatica mapping to implement this logic? Assume that the source data does not change between session runs.

Ans

- Create a mapping to find out the number of records in the source and write the count to a parameter file. Let call this parameter as $$SOURCE_COUNT.
- Create another mapping. Go to the mapping parameters and variables, create a mapping variable ($$VAR_SESSION_RUNS) with integer data type.
- Connect the source qualifier transformation to the expression transformation. In the expression transformation, create the below additional ports.
  v_Count (variable port) = v_Count + 1
  O_Run_flag (output port) = IIF ($$vAR_SESSION_RUNS = 0,
  Setvariable ($$vAR_SESSION_RUNS, 1),
  IIF (!ISNULL ($$vAR_SESSION_RUNS) and v_Count=1, 2, $$vAR_SESSION_RUNS)) O_count (output port) = V_Count

- Connect the expression transformation to the filter transformation and specify the following filter condition:
  IIF (O_Run_Flag = 1, v_count <= $$SOURCE_COUNT/2,
  IIF (O_Run_Flag = 2, v_count > $$SOURCE_COUNT/2))

- Connect the filter transformation to the target.
- Here I am assuming that you know how to use a parameter file. That is why I did not specify the complete details.

Source Qualifier Transformation Examples - Informatica

To solve these examples, create the employees and departments tables in your database. The "**create table**" statements are provided below.

```
create table DEPARTMENTS
(
DEPARTMENT_ID NUMBER (4) not null,
DEPARTMENT_NAME VARCHAR2 (15) not null,
MANAGER_ID NUMBER (6)
)
alter table DEPARTMENTS add primary key (DEPARTMENT_ID);

create table EMPLOYEES
(
EMPLOYEE_ID NUMBER (6) not null,
NAME VARCHAR2 (10),
LAST_NAME VARCHAR2 (10),
SALARY NUMBER (10,2),
MANAGER_ID NUMBER (6),
DEPARTMENT_ID NUMBER (4)
)
alter table EMPLOYEES add primary key (EMPLOYEE_ID);
alter table EMPLOYEES add foreign key (DEPARTMENT_ID) references
DEPARTMENTS (DEPARTMENT_ID);
```

Table Name: Employees

EMPLOYEE_ID	NAME	SALARY	MANAGER_ID	DEPARTMENT_ID
101	VIVEK	1000	201	10
102	HARSHAL	2000	201	10
201	SHUBHAM	5000	NULL	10
301	RAJ	7000	NULL	NULL

Table Name: Departments

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID
10	ACCOUNT	201
20	HR	501

Try Solving the below examples:

Q1. Create a mapping to join employees and departments table on "DEPARTMENT_ID " column using source qualifier transformation?

Ans

1. Source qualifier transformation can be used to join sources only from the same database.
2. Connect the source definitions of departments and employees to the same qualifier transformation.
3. As there is a primary-key, foreign-key relationship between the source tables, the source qualifier transformation by default joins the two sources on the DEPARTMENT_ID column.

Q2. Create a mapping to join employees and departments table on "MANAGER_ID" column using source qualifier transformation?

Ans

1. Connect the source definitions of departments and employees to the same qualifier transformation.
2. Go to the properties tab of source qualifier -> User Defined Join and then open the editor. Enter the join condition as DEPARTMENTS.MANAGER_ID = EMPLOYEES.MANAGER_ID. Click Ok.
3. Now connect the required ports from the source qualifier transformation to the target.

Q3 Create a mapping to get only the employees who have manager?

Ans This is very simple. Go to the properties tab of source qualifier-> Source Filter. Open the editor and enter EMPLOYEES.MANAGER_ID IS NOT NULL

Q4. Create a mapping to sort the data of employees table on DEPARTMENT_ID, SALARY?

Ans Make sure the ports order in the source qualifier transformation as shown below:

Column_Names
EMPLOYEE_ID
DEPARTMENT_ID
SALARY
EMPLOYEE_ID
NAME
LAST_NAME
MANAGER_ID

The first two ports should be DEPARTMENT_ID, SALARY and the rest of the ports can be in any order.

Now go to the properties tab of source qualifier-> Number of Sorted Ports. Make the Number of Sorted Ports value as 2.

Q5. Create a mapping to get only distinct departments in employees table?

Ans

1. The source qualifier transformation should only contain the DEPARTMENT_ID port from EMPLOYEES source definition.
2. Now go to the properties tab of source qualifier-> Select Distinct. Check the check box of Select Distinct option.

Oracle Query to split the delimited data in a column to multiple rows

Q1. Consider the following table "t" data as the source

Id	Value
1	A, B, C
2	P, Q, R, S, T
3	M, N

Here the data in value column is a delimited by comma. Now write a query to split the delimited data in the value column into multiple rows.

The output should look like as

Id	Value
1	A
1	B
1	C
2	P
2	Q
2	R
2	S
2	T
3	M
3	N

Ans

```
SELECT t.id,
CASE WHEN a.1 = 1
        THEN substr(value, 1, instr (value, ', ', 1, a.1) - 1)
        ELSE substr(value, instr (value, ', ', 1, a.1 - 1) + 1,
CASE WHEN instr (value, ', ', 1, a.1) – instr (value, ', ', 1, a.1 - 1) - 1 > 0
        THEN instr (value, ', ', 1, a.1) – instr (value, ', ', 1, a.1 - 1) – 1
        ELSE length (value)
        END)
END final_value
FROM t,
        (SELECT level 1
        FROM DUAL
        CONNECT BY LEVEL <=
        (SELECT Max (length (value) – length (replace (value, ', ', '')) + 1)
FROM t)) a
WHERE   length (value) – length (replace (value, ', ', '')) + 1 > = a.1
order by t.id, a.1;
```

Q2. The source data is represented in the form the tree structure. You can easily derive the parent-child relationship between the elements. For example, B is parent of D and E. As the element A is root element, it is at level 0. B, C are at level 1 and so on.

The above tree structure data is represented in a table as shown below.

Col1	Col2	Col3	Col4
A	B	D	H
A	B	D	I
A	B	E	NULL
A	C	F	NULL
A	C	G	NULL

Here in this table, column Col1 is parent of column Col2, column Col2 is parent of column Col3, column Col3 is parent of column Col4.

1. **Write a query to load the target table with the below data. Here you need to generate sequence numbers for each element and then you have to get the parent id. As the element "A" is at root, it does not have any parent and its parent_id is NULL.**

Id	Element	Lev	Parent_Id
1	A	0	NULL
2	B	1	1
3	C	1	1
4	D	2	2
5	E	2	2
6	F	2	3
7	G	2	3
8	H	3	4
9	I	3	4

Ans

```
WITH t1 AS
(SELECT VALUE PARENT, LEV, LEAD (value, 1) OVER (PARTITION BY r
ORDER BY lev) CHILD
FROM (SELECT c1, c2, c3, c4, ROWNUM r
FROM table_name)
UNPIVOT (value FOR lev IN (c1 as 0, c2 as 1, c3 as 2, c4 as 3))), t2 AS
(SELECT PARENT, LEV, ROWNUM SEQ
FROM (SELECT DISTINCT PARENT, LEV
FROM T1
ORDER BY LEV)), T3 AS
(SELECT DISTINCT PARENT, CHILD
FROM T1
WHERE CHILD IS NOT NULL
UNION ALL
SELECT DISTINCT NULL, PARENT
FROM T1
WHERE   LEV=0)
SELECT C.SEQ Id, T3.CHILD ELEMENT, C.LEV, P.SEQ PARENT_ID
FROM T3 INNER JOIN T2 C
ON (T3.CHILD = C.PARENT)
LEFT OUTER JOIN T2 P
ON (T3.PARENT = P.PARENT)
ORDER BY C.SEQ;
```

Note: The unpivot function works in oracle 11g.