

1. What is the difference between Informatica Powercenter and Informatica Cloud?

Informatica Intelligent Cloud Services is a cloud based Integration platform(iPaaS). IICS helps you integrate, synchronize all data and applications residing on your on-premise and cloud environments. It provides similar functionality as Powercenter in a better way and can be accessed via the internet. Hence in IICS, there is no need to install any client applications on the Personal computer or server. All the supported applications can be accessed from the browser and the tasks can be developed through browser UI. In Powercenter, the client applications need to be installed on your server.

2. What is a Runtime environment?

A Runtime environment is the execution platform that runs a data integration or application integration tasks. You must have at least one runtime environment setup to run tasks in your organization. Basically it is the sever upon which your data gets staged while processing. You can choose either to process via the Informatica servers or your local servers which stays behind your firewall. Informatica supports following runtime environments-

Informatica Cloud Hosted Agent,
Serverless runtime environment and
Informatica Cloud Secure Agent.

3. What is a Synchronization task?

Synchronization task helps you synchronize data between a source and target. A Synchronization task can be built easily from the IICS UI by selecting the source and target without use of any transformations like in mappings. You can also use expressions to transform the data according to your business logic or use data filters to filter data before writing it to targets and use lookup data from other objects and fetch a value. Anyone without Powercenter mapping and transformation knowledge can easily build synchronization tasks as UI guides you step by step.

4. What is a Replication task?

A Replication task allows you to replicate data from a database table or an on-premise application to a desired target. You can choose to replicate all the source rows or the rows that changes since the last runtime of the task using built in Incremental processing mechanism of Replication Task.

You can choose from three different type of operations when you replicate data to a target.

- Incremental load after initial full load
- Incremental load after initial partial load
- Full load each run

5. What is the difference between a Synchronization task and Replication task?

One of the major differences between a Synchronization task and a Replication task is that, in a synchronization task, you can transform the data before loading it to the target. However, in a Replication task, you can replicate the data from source to target without transforming the data.

A Replication task can replicate an entire schema and all the tables in it at a time which is not possible in Synchronization task.

A Replication task comes with a built-in incremental processing mechanism. In Synchronization task user needs to handle the incremental data processing.

6. Where does the metadata gets stored in Informatica Cloud (IICS)?

All the metadata gets stored in the Informatica Cloud repository. Unlike Powercenter, all the information in Informatica Cloud is stored on the server maintained by the Informatica and the user does not have access to the repository database. Hence, it is not possible to use any SQL query on metadata tables to retrieve the information like in Informatica Powercenter.

7. What metadata information gets stored in the Informatica Cloud (IICS) repository?

Informatica Cloud Services includes the IICS repository that stores various information about tasks. As you create, schedule, and run tasks, all the metadata information is written to IICS repository.

The various information that gets stored to IICS repository include:

Source and Target Metadata: Metadata information of each source and target including the field names, datatype, precision, scale and other properties.

Connection Information: The connection information to connect specific source and target systems in an encrypted format.

Mappings: All the Data integration tasks built, their dependences and rules are stored.

Schedules: The schedules created you run the task built in IICS are stored.

Logging and Monitoring information: The results of all the jobs are stored.

8. What is a Mapping Configuration task?

A Mapping Configuration Task or Mapping Task is analogous to a session in Informatica Powercenter. When you create a Mapping Task, you must select a mapping to use in the task. Mapping task allows you to process data based on the data flow logic defined in a mapping.

Optionally, you can define the following in the Mapping Task

- You can define parameters that associate with the mapping.
- Define pre and post-processing commands.
- Add advance session properties to boost the performance.
- Configure the task to run on schedule.

9. What is a taskflow in Informatica Cloud?

A Taskflow is analogous to a workflow in Informatica Powercenter. A taskflow controls the execution sequence of a mapping configuration task or a synchronization task based on the output of the previous task. To create a taskflow, you must first create the tasks and then add them to a taskflow.

The taskflow allows you to

- Run the tasks sequentially
- Run the tasks in parallel
- Make decisions based on outcome from one task before triggering the next task.

10. What is the difference between a Taskflow and Linear Taskflow?

A Linear taskflow is a simplified version of the Data Integration taskflow. A linear taskflow groups multiple Data Integration tasks and runs them serially in the specified order. If a task defined in Linear taskflow gets failed, you need to restart the entire taskflow. However, a taskflow allows you to run the tasks in parallel, provides advanced decision making capabilities and allows you to either restart from failed task or skip it when a task fails.

11. Can we run Powercenter jobs in Informatica cloud?

Yes. There is a Powercenter task available in Informatica Cloud where in user have to upload the XML file exported from Powercenter in Data Intergration and run the job as a Powercenter task. You can update an existing PowerCenter task to use a different PowerCenter XML file but cannot make changes to an imported XML. When you upload a new PowerCenter XML file to an existing PowerCenter task, the PowerCenter task deletes the old XML file and updates the PowerCenter task definition based on new XML file content.

12. How does a update strategy transformation work in Informatica Cloud?

There is no Update strategy transformation available in Information Cloud. In the target transformation in a mapping, Informatica Cloud Data Integration provides the option for the action to be performed on the target – Insert, Update, Upsert, Delete and Data Driven.

13. What is the difference between a Union transformation in Informatica Cloud vs Informatica Powercenter?

In earlier versions of Informatica Cloud, the Union transformation allows only two groups to be defined in it. Hence if three different source groups needs to be mapped to target, the user must use two Union transformations. The output of first two groups to Union1. The output of Union1 and group3 to Union2.

In the latest version, Informatica Cloud is **supporting multiple groups**. So all the input groups can be handled in a single Union transformation.

14. What is Dynamic Linking?

Informatica Cloud Data Integration allows you to create a new target files/tables at runtime. To use this feature in mappings, choose **Create New at Runtime** option in target and specify the name for the new target.

The user can choose a static filename where the target file will be replaced by a new file every time the mapping runs. The user can also choose to create a Dynamic filename so that the every time the mapping runs, target file is created with a new name.

15. In what format can you export a task present in Informatica Cloud?

Informatica Cloud Data Integration supports exporting the tasks as a zip file where the metadata gets stored in the JSON format inside the zip file. However you can also download a XML version of the tasks also which can be imported as workflows in Powercenter. But it will not support bulk export of tasks in XML format at a time. Where as you can export multiple tasks in form of JSON in a single export zip file.

16. How do you read JSON Source file in IICS?

JSON files are read using the Hierarchy Parser transformation present in IICS. The user needs to define a Hierarchical Schema that defines the expected hierarchy of the JSON file. The Hierarchical Schema can then be imported into Hierarchy Parser transformation while reading the data from input JSON files which converts the input based on the schema that is associated with the transformation. The Hierarchy Parser Transformation can also be used to read XML files in Informatica Cloud Data Integration

17. What is a Hierarchical Schema in IICS?

A Hierarchical Schema is a component where user can upload an XML or JSON sample file that define the hierarchy of output data. The Hierarchy Parser transformation converts input based on the Hierarchical schema that is associated with the transformation.

18. What is Indirect File loading and how to perform Indirect loading in IICS?

The ability to process multiple source files of same structure and properties through a single source transformation in a mapping is called **Indirect File Loading**. In order to perform Indirect loading in IICS, prepare a flat file which holds the information of all source filenames which share same file structure and properties. Pass this file as source file and select the **File List** under **Source Type** property of a source transformation in a mapping. The data from all the files listed in the source file will be processed in a single run.

19. What are the parameter types available in the Informatica Cloud?

You can add parameters to mappings to create flexible mapping templates that developers can use to create multiple mapping configuration tasks. IICS supports two types of parameters.

Input Parameter: Similar to a parameter in Powercenter. You can define an input parameter in a mapping and set the value of the parameter when you configure a mapping task. The parameter value remains constant as the value defined in mapping task or a Parameter file through out the session run.

In-Out Parameter: Similar to a variable in Powercenter. Unlike input parameters, an In-Out parameter can change each time a task runs. When you define an In-Out parameter, you can set a default value in the mapping. However, you would typically change the value of In-Out Parameter at run time using an Expression transformation using SETVARIABLE functions. The mapping saves the latest value of the parameter after the successful completion of the task. So, when the task runs again, the mapping task compares the In-Out parameter to the saved value instead of default value.

20. How many Status states are available in IICS monitor?

The various status states available in IICS are

Starting: Indicates that the task is starting.

Queued: There is a predefined number set which controls how many tasks can run together in your IICS org. If the value is set to two and if two jobs are already running, the third task you trigger enters into Queued state.

Running: The job enters the Running status from Queued status once the task is triggered completely.

Success: The task completed successfully without any issues.

Warning: The task completed with some rejects.

Failed: The task failed due to some issue.

Stopped: The parent job has stopped running, so the subtask cannot start. Applies to subtasks of replication task instances.

Aborted: The job was aborted. Applies to file ingestion task instances.

Suspended: The job is paused. Applies to taskflow instances.

21. When Source is parameterized in a Cloud mapping, the source transformation fields would be empty. Then how does the fields get propagated from source to the downstream transformations in source parameterized mappings?

In order to propagate the fields to downstream transformations when source is parameterized, initially create the mapping with actual source table. In the downstream transformation after source, select the *Field Selection Criteria* as *Named Fields* and include all the source fields in the *Incoming Fields* section of the transformation. Then change the source object to a parameter. This way the source fields are still retained in the downstream transformation even when the fields are not available in source transformation after the source is parameterized.

22. To include all incoming fields from an upstream transformation except those with dates, what should you do?

Configure two field rules in a transformation. First, use the *All Fields* rule to include all the fields coming from

upstream transformation. Then, create a *Fields by Datatypes* rule to exclude fields by data type and select Date/Time as the data type to exclude from incoming fields.

23. What are Preprocessing and postprocessing commands in IICS?

The Preprocessing and postprocessing commands are available in the Schedule tab of tasks to perform additional jobs using SQL commands or Operating system commands. The task runs preprocessing commands before it reads the source. It runs postprocessing commands after it writes to the target. The task fails if any command in the preprocessing or postprocessing scripts fail.

24. What are Field Name conflicts in IICS and how can they be resolved?

When there are fields with same name coming from different transformations into a downstream transformation like a Joiner transformation, the cloud mapping designer generates a *Field Name Conflict* error. You can either resolve the conflict by renaming the fields in the upstream transformation only or you can create a field rule in downstream transformation to *Bulk Rename* fields by adding a prefix or a suffix to all incoming fields.

25. What system variables are available in IICS to perform Incremental Loading?

IICS provides access to following system variables which can be used as a data filter variables to filter newly inserted or updated records.

\$LastRunTime returns the last time when the task ran successfully.

\$LastRunDate returns only the last date on which the task ran successfully. The values of \$LastRunDate and \$LastRunTime get stored in Informatica Cloud repository/server and it is not possible to override the values of these parameters. These parameters store the datetime value in UTC time zone.

26. What is the difference between the connected and unconnected sequence generator transformation in Informatica Cloud Data Integration?

Sequence generator can be used in two different ways in Informatica cloud. One with Incoming fields disabled and the other with incoming fields not disabled.

The difference between the sequence generator with incoming fields enabled and disabled is, when NEXTVAL field is mapped to multiple transformations,

→ Sequence generator with incoming fields **not disabled** will generate **same** sequence of numbers for each downstream transformation.

→ Sequence generator with incoming fields **disabled** will generate **Unique** sequence of numbers for each downstream transformation.

27. Explain Partitioning in Informatica Cloud Data Integration.

Partitioning is nothing but enabling the parallel processing of the data through separate pipelines. With the Partitioning enabled, you can select the number of partitions for the mapping. The DTM process then creates a reader thread, transformation thread and writer thread for each partition allowing the data to be processed concurrently, thereby reducing the execution time of the task. Partitions can be enabled by configuring the Source transformation in mapping designer.

There are two major partitioning methods supported in Informatica Cloud Data Integration.

1. Key Range Partitioning distributes the data into multiple partitions based on the partitioning key selected and range of the values defined for it. You must select a field as a partitioning key and defined the start and end ranges of the value.

2. Fixed Partitioning can be enabled for sources which are not relational or support key range partitioning. You must select the number of partitions by passing a value.

28. How to pass data from one mapping to other in Informatica Cloud Data Integration?

The data can be passed from one Mapping task to another in Informatica Cloud Data Integration through a Task flow using parameters. The Mapping Task which passes the data should have an In-Out Parameter defined using SetVariable functions. The Mapping Task which receives the data should either have an Input parameter or an In-Out Parameter defined in the mapping to read the data passed from upstream task.

Informatica Cloud (IICS) Scenario based Interview Questions

October 29, 2020

Spread the love

more

Instead of providing some scenario based Interview questions and solutions to them I would like to take a different approach here.

We shall take a "concept" and discuss what kind of scenarios based Interview questions that could be built around it.

Chapter-1: [Adding Sequence numbers to the Source records](#)

Chapter-2: [Introducing a dummy field](#)

Chapter-3: [Comparing Current record with Previous record](#)

Chapter-4: [LEAD and LAG implementation in Informatica Cloud](#)

Chapter-5: [Denormalizing data in Informatica Cloud](#)

Let us discuss in detail how to add sequence numbers to your source data and in what kind of scenarios do we need to do it.

Chapter-1: Adding Sequence numbers to the Source records

Contents

- [I. Introduction: How to add sequence numbers to your source data?](#)
- [II. Generating sequence numbers using Expression transformation](#)
- [III. Generating sequence numbers using Sequence Generator transformation](#)
- [Q1. Design a mapping to load alternate records to different tables](#)
- [Q2. Design a mapping to load alternate group of three records into three different targets](#)
- [Q3. Design a mapping to load every 150th record to the target table](#)
- [IV. Conclusion](#)

I. Introduction: How to add sequence numbers to your source data?

This can be implemented in two different ways

1. Using Expression transformation
2. Using Sequence Generator transformation

II. Generating sequence numbers using Expression transformation

In the expression transformation create a variable port and assign a value as incrementing the same variable by 1.
Next create an output port and assign the variable to the output port.
In expression transformation, the ports will be as below

$V_count = V_count + 1$

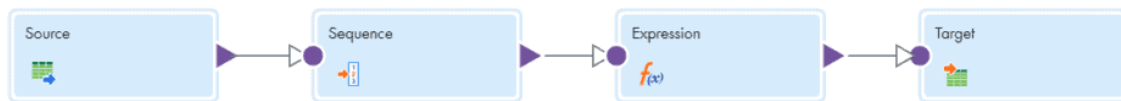
$O_count = V_count$

III. Generating sequence numbers using Sequence Generator transformation

Sequence Generator transformation is a passive and connected transformation in Informatica Cloud. It is used to generate sequence of numeric values.

Use Sequence generator transformation to

- Generate sequence of unique numbers
- Generate a cyclic sequence of numbers



Sequence Generator transformation in Informatica Cloud Mapping

To generate the sequence numbers, we always use the **NEXTVAL** column from sequence generator transformation.

Q1. Design a mapping to load alternate records to different tables

Description: The question says alternate records i.e the 1st, 3rd, 5th ... records should be loaded in Target1 and 2nd, 4th, 6th ... records should be loaded in Target2.

It means the odd and even records should be loaded in different targets.

Solution:

1. Map the fields from source to an expression transformation.
2. Create a new variable field of type integer V_count and assign value as V_count+1
3. Create a new output field of type integer Flag and assign value as MOD(V_count,2).

4. The MOD functions returns the remainder of a division calculation. So for an odd record the output will be 1 and for an even record the output will be zero when divided by 2.
5. In expression the ports will be as below

V_count = V_count+1

Flag = MOD(V_count,2)

6. Map the fields from expression to a router transformation. In router create two new output groups.

Group Name	Condition
Even	Flag=0
Odd	Flag=1

7. Map the records from Odd output group to Target1 and Even output group to Target2.

NOTE: If you are using a sequence generator transformation, there is an additional step involved. You need to map fields from source to sequence generator and then to expression. In expression create only one output port 'Flag' and assign value as MOD(NEXTVAL,2). The rest of the procedure is same.

Q2. Design a mapping to load alternate group of three records into three different targets

Description: The requirement here is to load the 1st record into 1st target, 2nd record into 2nd target, 3rd record into 3rd target. When the 4th record comes it should begin loading again from 1st target and the sequence continues.

At the end, the 1st target should contain data of row 1, 4, 7, 10

At the end, the 2nd target should contain data of row 2, 5, 8, 11

At the end, the 3rd target should contain data of row 3, 6, 9, 12

Solution:

In the earlier case as there are only two targets we used the odd/even strategy.

In this case in order to achieve this we have to increment the count of records from 1 to 3 and for the next record when the count becomes >3, we need to reset the count back to 1.

Follow the below steps to implement the solution

1. Map the fields from source to an expression transformation.

2. Create a new variable field of type integer V_count and assign value as IIF(V_count=3, 1, V_count+1)
3. Create a new output field of type integer Row_Number and assign value as V_count.
4. In expression the ports will be as below

V_count = IIF(V_count=3, 1, V_count+1)

Row_Number = V_count

5. Map the fields from expression to a router transformation. In router create three new output groups.

Group Name	Condition
Group1	Row_Number=1
Group2	Row_Number=2
Group3	Row_Number=3

6. Map the records from each group to the respective target transformations.

NOTE: This can also be implemented using a sequence generator transformation. The only change required is to replace expression with sequence generator. In the sequence generator properties **enable the Cycle option**. Set the **Cycle Start Value as 1** and **End Value as 3**. The rest of the procedure is same.

The ODD/EVEN example we discussed earlier can also be implemented using this approach by resetting the value to 1 whenever count becomes >2.

Q3. Design a mapping to load every 150th record to the target table

Description: For example, if the source have 300 records, only the 150th and 300th record should be loaded.

Solution:

As discussed in above examples, we can implement the solution here in multiple methods.

Method1:

1. Map the fields from source to an expression transformation.
2. Create a new variable field of type integer V_count and assign value as V_count+1
3. Create a new output field of type integer Flag and assign value as MOD(V_count,150).
4. In expression the ports will be as below

V_count = V_count+1

Flag = MOD(V_count,150)

5. Map the fields from expression to a Filter transformation and provide filter condition as Flag=0
6. Map the records from filter to target transformation.

Method2:

This method is similar to the above discussed method except a change in expression transformation. The ports in expression transformation should be as below

V_count = IIF(V_count=150,1,V_count+1)

Flag = V_count

The rest of the procedure is same.

IV. Conclusion

The above two methods can also be implemented using Sequence generator transformation. That makes it four different approaches to implement the solution.

Top Informatica Cloud(IICS) Salesforce Interview questions

December 25, 2020

Spread the love

[more](#)

1. How to connect Salesforce from Informatica Cloud?

In order to access data from Salesforce, a connection needs to be created initially from Administrator's tab.

The connection information created can then be used in Data Integration tasks either to retrieve data or process data into the Salesforce.

The requirements to connect to a Salesforce environment are

- User Name – User name for the Salesforce account.

- Password – Password for the Salesforce account.
- Security Token – Security token generated from the Salesforce application.
- Service URL – URL of the Salesforce service. The service URL is based on the Salesforce version.

2. When does a Security Token not required to create a Salesforce connection?

- In order to avoid using Security Token while setting up a Salesforce connection, Informatica Cloud IP ranges should be entered in the Trusted IP Ranges in the Salesforce Application.
- The Informatica Cloud Trusted IP ranges can be found by hovering over the question mark symbol available next to Security Token option in a Salesforce connection.

3. What is PK Chunking?

- PK Chunking is an advanced option that can be configured for Salesforce Bulk API tasks in a Source transformation.
- If PK Chunking is enabled, then Salesforce internally generates separate batches based on the PK Chunking size given. Each batch is a small chunk of a bulk query created based on the Primary Key(ID) of the queried records.
- It is recommended to enable PK chunking for objects with more than 10 million records. This improves performance.

- For example, let's say you enable PK chunking Account table with 10,000,000 records. Assuming a chunk size of 250,000 the query is split into 40 small queries. Each query is submitted as a separate batch.

4. How to add a filter condition in a source transformation when the source is a Salesforce object?

- When the source is a salesforce object, defining the filter condition under the Filter option available under Query Options won't work. The Filter condition needs to be defined under SOQL Filter Condition available under Advanced tab of source transformation.

- The *Filter* option under *Query Options* tab can be used only if the filter condition needs to be parameterized.
- Alternatively, the *Source Type* can be selected as *Query* and the filter condition can be passed in the source query defined.

5. How does a Boolean field from Salesforce object is processed in Informatica Cloud?

- The Boolean fields from Salesforce are read as Integer fields when read from Informatica cloud tasks. The Salesforce field values TRUE and FALSE are read as '1' and '0' by Informatica respectively.
- In order to insert/update the Boolean fields in salesforce object, define an integer field of value '1' or '0' and the value in salesforce will be updated with either TRUE or FALSE accordingly.

6. What is difference between Salesforce Bulk API and Standard API?

- Salesforce Standard API processes up to 200 records in one API call. However, Bulk API processes about 10,000 records in one API call.
- If the Salesforce object contains large data sets, then it is advisable to use Bulk API. The performance of the task would be much faster when loading large data sets using Bulk API when compared to the same job run on Standard API. Similarly, it is advisable to use Standard API when dealing with smaller data sets and the performance would be much faster compared to Bulk API.
- Standard API makes Synchronous calls in which the thread will wait until it completes its tasks before proceeding to next. In a Synchronous call, the code runs in single thread. Bulk API makes Asynchronous calls in which the thread will not wait until it completes its tasks before proceeding to next. In an Asynchronous call, the code runs in multiple threads which helps to do many tasks as background jobs.

7. Can Salesforce objects be used in IICS Lookup transformation?

- Salesforce objects can be used in a connected Lookup transformation to look up data. But the Informatica Cloud do not support using Salesforce objects in Unconnected Lookup transformation.

8. What is the difference between Soft delete and Hard delete of Salesforce records? How to enable them from Informatica Cloud?

- When you delete records from Salesforce object, they are moved to Recycle Bin and these records are said to be soft deleted. The soft deleted records can be restored.
- When you delete records permanently from Salesforce object, they are said to be hard deleted. The hard deleted records cannot be restored.
- To enable hard delete of salesforce records from IICS, check the *Hard Delete* option when you select the Delete operation in Target transformation. If the option is not selected, the delete operation is considered as soft delete.

9. How to Include archived and deleted rows in the source data queried from Informatica Cloud?

- When Salesforce objects are used as a source, by default the archived and deleted records in the salesforce object are omitted from search query. To include the archived and deleted records in search query enable the ***Include archived and deleted rows in the Source*** option available under Query Options of a source transformation.

10. What is the default timezone in which date fields are stored in Salesforce objects?

- In Salesforce by default the **DateTime** fields store the time information in UTC timezone and display the appropriate date and time to the user based on the user's personal timezone settings.

11. What are the advanced properties available for a Salesforce target?

Max Batch Size

- Maximum number of records the agent writes to a Salesforce target in one batch. Default is 200 records.
- This property is not used in Bulk API target sessions.

Set Fields to Null

- Replaces values in the target with null values from the source.
- By default, the agent does not replace values in a record with null values during an update or upsert operation.

Use SFDC Error File

- Generates the error log files for a Bulk API target session. By default, the agent does not generate the error log files.
- To generate an error log file for a Bulk API target session, select the Monitor Bulk option.

Use SFDC Success File

- Generates the success log files. By default, the agent does not generate the success log files.
- To generate a success log file for a Bulk API target session, select the Monitor Bulk option.

Salesforce API

- Defines which API method to be used to process records while loading data into Salesforce. The two options available are Standard API and Bulk API.
- By default, the agent uses the Salesforce Standard API.

12. What are the advanced properties available for a Salesforce target with Bulk load?

Monitor Bulk:

- You can enable a Bulk API task for monitoring. With monitoring enabled, the Data Integration service requests the status of each batch from the Salesforce and logs them in the session log.
- By default, the Data Integration service does not monitor Bulk API jobs. Without monitoring, the activity log and session log contains information about batch creation, but does not contain details about batch processing or accurate job statistics.

Enable Serial Mode

- The Salesforce service can perform a parallel or serial load for a Bulk API task. By default, it performs a parallel load.
- In a parallel load, the Salesforce service writes batches to target at the same time. In a serial load, the Salesforce service writes batches to targets in the order it receives them.
- Use a parallel load to increase performance when you are not concerned about the target load order. Use a serial load when you want to preserve the target load order.

Test your Understanding

Informatica Cloud Data Integration (IICS) Transformations Guide
March 9, 2021

Spread the love

more

Contents

- [1. Introduction to Informatica Cloud Transformations](#)
 - [Connected Transformation](#)
 - [Unconnected Transformation](#)
 - [Active Transformation](#)
 - [Passive Transformation](#)
- [1. Source Transformation in Informatica Cloud](#)
 - [1.1 Database objects as Source Object](#)
 - [1.2 Flat Files as Source Object](#)
 - [1.3 Fields section in Source Transformation](#)
 - [1.4 Partitions section in Source Transformation](#)
- [2. Filter Transformation in Informatica Cloud](#)

- [2.1 Types of Filter Conditions in Informatica Cloud](#)
- [2.2 Simple Filter Condition](#)
- [2.3 Advanced Filter Condition](#)
- [2.4 Parametrization of Filter condition](#)
- [2.5 Filter Performance Tuning](#)
- [3. Router Transformation in Informatica Cloud](#)
 - [3.1 Output Groups in Router Transformation](#)
 - [3.2 Default Group in Router Transformation](#)
 - [3.3 Filter Condition types in Router Transformation](#)
- [4. Expression Transformation in Informatica Cloud](#)
 - [4.1 Expression Transformation Use Cases](#)
 - [4.2 Expression Transformation Field Types](#)
 - [4.3 Expression Macros](#)
- [5. Sorter Transformation in Informatica Cloud](#)
 - [5.1 Sorter Transformation Properties](#)
 - [5.2 Configuring Sort Fields and Sort Order](#)
 - [5.3 Parameterizing the Sort Condition](#)
 - [5.4 Sorter Transformation Advanced Properties](#)
- [6. Aggregator Transformation in Informatica Cloud](#)
 - [6.1 Aggregator Transformation Properties](#)
 - [6.2 Types of Aggregator Cache](#)
 - [6.3 How Sorted Input increases Performance in Aggregator?](#)
 - [6.4 What happens when no fields are selected as Group By Fields in Aggregator?](#)
- [7. Sequence Transformation in Informatica Cloud](#)
 - [7.1 Sequence Generator Fields](#)
 - [7.2 Sequence Generator Types in Informatica Cloud](#)
 - [7.3 Sequence Generator Properties](#)
- [8. More Transformations.....](#)
 - [Informatica Cloud \(IICS\) REST V2 Connector & WebServices Transformation](#)
 - [Shared Sequences in Informatica Cloud \(IICS\)](#)
 - [Joiner Transformation in Informatica Cloud \(IICS\)](#)
 - [Normalizer Transformation in Informatica Cloud \(IICS\)](#)
 - [Union Transformation in Informatica Cloud \(IICS\)](#)
 - [Lookup Transformation in Informatica Cloud \(IICS\)](#)

I. Introduction to Informatica Cloud Transformations

Transformations are mapping objects which represents the operation that needs to be performed on the data in the Informatica Cloud Data Integration mappings. There are several transformations that are offered by Informatica Cloud with each having its own properties and operation that it performs.

The transformations in Informatica Cloud are classified based on their connectivity and how they handle the rows passing through them.

Based on connectivity, the transformations are classified as

1. Connected Transformation
2. Unconnected Transformation

Connected Transformation

A Connected transformation is an inline transformation which stays in the flow of the mapping and connected to other transformations in the mapping.

Unconnected Transformation

An Unconnected transformation is not connected to any other transformation in the mapping. They are usually called within another transformation and returns a value to that transformation.

Based on how the transformation handles the rows passing through it, the transformations are classified as

1. Active Transformation
2. Passive Transformation

Active Transformation

An Active transformation can change the number of rows passing through it. Also a transformation is considered active when it can change the transaction boundary or position of the rows passing through it.

Any transformation that splits or combines the data streams or reduce, expand or sort the data is an active transformation because it cannot be guaranteed that when the data passes through it, the number of rows and their position in the data stream are always unchanged.

Passive Transformation

A Passive transformation does not change the number of rows passing through it, maintains the transaction boundary and position of the rows passing through it.

Let us discuss about the various transformation available in Informatica Cloud.

1. Source Transformation in Informatica Cloud

Source transformation is an active and connected transformation in Informatica Cloud. It is used to read and extract the data from the objects connected as source. The Source transformation can read data from a single source object or multiple source objects based on the connection type.

When you configure a Source transformation, you define Source properties on the following tabs of the Properties panel:

- **Source tab:** Select the Connection, Source Type and Source Object. You can also configure the advanced properties depending upon the source connection.
- **Fields tab:** Configure the fields of the object selected as source.
- **Partitions tab:** Configure the partitioning type based on the source connection.

1.1 Database objects as Source Object

When the connection type is any database connection, you could select the source type as

- **Single Object:** A single database object can be selected as a source.
- **Multiple Objects:** Multiple database objects with existing relationship at database level could be selected as sources.
- **Query:** A SQL query on the database objects can be used to fetch data from the database.
- **Parameter:** The entire source object could be parameterized in the mapping for which the value could be passed from a mapping task. In the mapping task you could again select a single or multiple source objects as source or even select Query option to pass SQL query as source.

Under *Query Options* tab of Source transformation

- You can configure conditions to *Filter* data read from database object.
- You can configure fields to *Sort* the data read from database object.
- You can select only distinct rows to be read from database object by checking *Select Distinct* option.

Query

Options in Source Transformation

Filter and Sort options are not supported when a custom query is used as source.

Under *Advanced* tab of Source transformation

- You can pass the *Pre SQL* and *Post SQL* commands.
- Adjust Tracing Level

1.2 Flat Files as Source Object

When the connection type is a Flat File connection, you could select the source type as

- **Single Object:** A single flat file object can be selected as a source.

- **File List:** You can pass a filename with list of flat file names with same file structure to process all the files through a single source transformation. This is called [Indirect File loading](#).
- **Command:** You can pass a script file as input which passes the source flat file name(s).
- **Parameter:** The source flat file name can be parameterized in mapping for which the value could be passed from a mapping task.

Filter and Sort options are not supported for Flat File connection type.

The properties in the source transformation varies as per the connection type selected.

1.3 Fields section in Source Transformation



Fields Section in Source Transformation

1. New fields can be added from the *Fields* section of source transformation. If the field is not present in the database object during the mapping run, the task fails.
2. Existing fields can be deleted from the *Fields* section. During the mapping run the Integration service will not try to read the field from the database that is deleted in the source transformation.
3. The fields can be sorted in the ascending order, descending order or existing Native order based on the field name.
4. Source fields metadata can be modified (i.e. modifying the field's datatype, precision and scale) from the *Fields* section by clicking on *Options* and select *Edit Metadata*.
5. The changes in the source object fields can be synchronized by clicking the Synchronize button. You can choose between synchronizing *All Fields* or *New Fields* only.
6. For Flat File sources, an additional field can be added which gives the source flat file name as value by selecting the *Add Currently Processed Filename field* in the *Fields* section.

1.4 Partitions section in Source Transformation

Partitioning enables the parallel processing of the data through separate pipelines.

There are two major partitioning methods supported in Informatica Cloud Data Integration.

1. Key Range Partitioning
2. Fixed Partitioning

The Key Range partitioning method is supported for relational sources. Fixed partitioning is supported for non-relational sources such as flat file sources.

For a detailed information about partitions, check out the article on [Informatica Cloud Partitioning](#).

2. Filter Transformation in Informatica Cloud

Filter transformation is an active and connected transformation in Informatica Cloud. It is used to filter data out of the data flow based on a specified filter condition.

The filter condition defined in the filter transformation is an expression that returns either TRUE or FALSE. The default return value from filter transformation is TRUE. That means you can add a filter transformation in the mapping without any condition defined and it allows all the records pass through it.

Similarly, you can define filter condition as FALSE which acts as a logical stop of the flow of the mapping as no records will be passed further. This Helps while checking the logic of the mapping in case of some problems.

2.1 Types of Filter Conditions in Informatica Cloud

In Filter transformation, filter conditions can be defined in three different types

- Simple
- Advanced
- Completely Parameterized

Filter conditions are case sensitive. You can use the following operators in filter transformation

- = (equals)
- < (less than)
- > (greater than)
- < = (less than or equal to)
- > = (greater than or equal to)
- != (not equals)

2.2 Simple Filter Condition

You can create one or more simple filter conditions. A simple filter condition includes a field name, operator, and value.

When you define more than one simple filter condition, the mapping task evaluates the conditions in the order that you specify using the AND logical operator.

Simple Filter Condition

In the above example, the filter condition will be treated as

DEP_ID = '10' and SALARY = '10000'

2.3 Advanced Filter Condition

You can use an Advanced filter condition to define a complex filter condition. When you configure an advanced filter condition, you can incorporate multiple conditions using the AND or OR logical operators.

Advanced Filter Condition

When you change the filter condition type from simple to advanced, the Mapping Designer automatically converts the simple filter conditions to the advanced filter condition.

2.4 Parametrization of Filter condition

The filter condition could be completely parameterized in filter transformation by creating an Input parameter of type expression for which value could be passed during the mapping runtime or from mapping task.

Completely Parametrizing the Filter condition

Additionally, the field name and the value of the field could be parameterized and used in the simple and advanced filter conditions.

\$DEP_ID\$ = \$DEP_Value\$

Using

Parameters in Filter condition

In the above example, we have created two input parameters. One of type Field and other of type String.

When the parameters are used in filter condition, the simple filter condition to advanced filter condition conversion is not supported. You need to manually enter the condition.

2.5 Filter Performance Tuning

- Use the Filter transformation as close to the source as possible in the mapping. This will reduce the number of rows to be processed in the downstream transformations.
- In case of relational sources, if possible use the filter condition in the source transformation. This will reduce the number of rows to be read from the source.

3. Router Transformation in Informatica Cloud

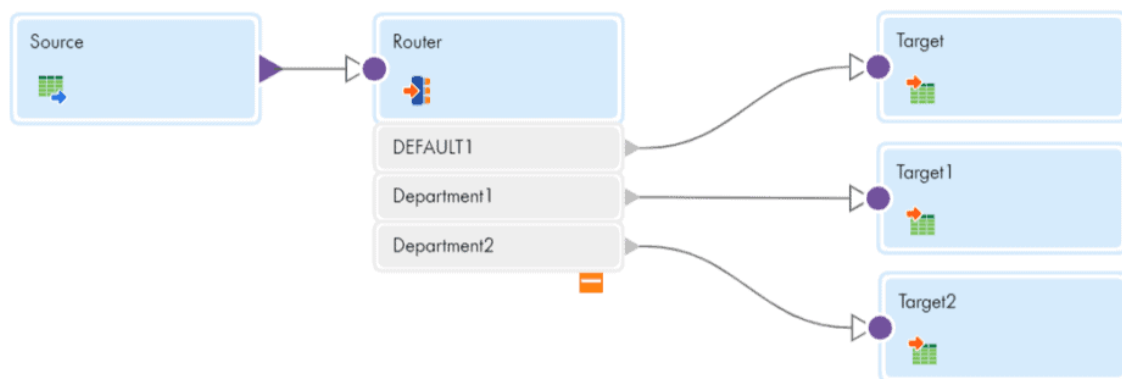
Router transformation is an active and connected transformation in Informatica Cloud. It is similar to filter transformation except that multiple filter conditions can be defined in the router transformation. Where as in filter transformation you can specify only one condition and drops the rows that do not satisfy the condition.

3.1 Output Groups in Router Transformation

You can create multiple groups in router transformation with each having its own filter condition. In the below example you can see multiple groups defined for each department.

Output Groups in Router Transformation

For each output group defined, a new data flow will be created which can be passed to downstream transformations as shown below.



Router transformation with multiple output groups routing data to multiple targets

3.2 Default Group in Router Transformation

By default, there will be a **DEFAULT** group that comes with in router transformation. The rows which do not satisfy any filter condition are passed through the default group. In our example, departments with ID other then 10 and 20 will be passed through DEFAULT group.

Router filter conditions are not if and else. If a rows satisfies the filter condition in multiple groups, the router transformation passes data from all the output groups that satisfy the condition.

3.3 Filter Condition types in Router Transformation

The filter condition types supported in Router transformation is similar to that what we have already discussed in Filter transformation. You can define a simple or advanced filter condition or completely or partially parameterize the filter condition based on the requirement.

4. Expression Transformation in Informatica Cloud

Expression transformation is a passive and connected transformation in Informatica Cloud. Expression transformation is used for row-wise manipulation. The Expression transformation accepts the row-wise data, manipulates it, and passes it to the downstream transformation.

4.1 Expression Transformation Use Cases

Expression transformation can be used to

- concatenate or split incoming field values
- create new fields
- Insert hard coded values into a field
- perform non-aggregate calculations

4.2 Expression Transformation Field Types

In Expression transformation new fields of below type can be created.

- Output Field
- Variable Field
- Input Macro Field
- Output Macro Field

Create an **Output Field** to perform certain operation for each record in the dataflow and pass it to the downstream transformation.

Create a **Variable Field** for calculations that you want to use within the transformation. The variable fields are not passed to the downstream transformations. They are usually used to hold a temporary value and can be used directly in the output fields created in the expression transformation.

4.3 Expression Macros

Informatica Cloud supports Expression Macros which allow you to create repetitive and complex expressions in mappings. Use Input and Output Macro Field types to implement expression macros.

An **Input Macro Field** represent the data of multiple source fields. An **Output Macro Field** represents the calculations that you want to perform on each input source field.

For example, you need to trim spaces in source fields data before loading into target. To implement this, you need to apply the TRIM logic for each field separately. Using expression macros, this can be implemented using one Input Macro field and Output Macro field.

To learn more, check out the detailed article on [Expression Macros](#) in Informatica Cloud.

5. Sorter Transformation in Informatica Cloud

Sorter transformation is an active and connected transformation in Informatica Cloud. Sorter transformation is used to sort the data based on incoming fields either in ascending or descending order.

5.1 Sorter Transformation Properties

When you configure a Sorter transformation, you define Sorter properties on the following tabs of the Properties panel:

- **Sort tab:** Define the Fields and Sort order on which data coming from upstream transformation should be sorted.
- **Advanced tab:** Define advanced properties like Distinct of Sorter transformation.

5.2 Configuring Sort Fields and Sort Order

When you specify multiple sort conditions, the mapping task sorts each condition sequentially. The mapping task treats each successive sort condition as a secondary sort of the previous sort condition. You can modify the order of sort conditions.

Sort Fields in Sorter Transformation

In the above example, the records are sorted department wise first and later the records in each department are sorted based on salary.

5.3 Parameterizing the Sort Condition

Informatica Cloud supports parameterizing the sort condition in the mapping where you can define the sort fields and the sort order when you run the mapping or when you configure the mapping task.

5.4 Sorter Transformation Advanced Properties

Sorter transformation can also remove duplicate records from the incoming data. This can be enabled by selecting the **Distinct** option in the **Advanced** tab.

General

Incoming Fields

Sort

Advanced

Tracing Level: ?

Normal

Sorter Cache Size: ?

☒ Auto ☐ Value

☒ Case Sensitive ?

Work Directory: ?

\$PMTempDir

☒ Distinct ?

☐ Null Treated Low ?

Transformation Scope: ?

All Input

☒ Optional ?

Advanced properties of Sorter Transformation

The other properties that can be configured from the Advanced tab of sorter transformation are

- **Case Sensitive:** When enabled, the transformation sorts uppercase characters higher than lower case characters.
- **Null Treated Low:** Treats a null value as lower than any other value. For example, if you configure a descending sort condition, rows with a null value in the sort field appear after all other rows.

6. Aggregator Transformation in Informatica Cloud

Aggregator transformation is an active and connected transformation in Informatica Cloud. It is Used to perform aggregate calculations such as sums, averages, counts on groups of data.

6.1 Aggregator Transformation Properties

When you configure a Aggregator transformation, you define Aggregator properties on the following tabs of the Properties panel:

- **Group By tab** – Configure Group by Fields to define how to group data for aggregate expressions.
- **Aggregate tab** – Configure an Aggregate field to define aggregate calculations. You can use aggregate functions, conditional clauses and non-aggregate functions in aggregate fields.
- **Advanced tab** – To improve job performance, you can configure an Aggregator transformation to use sorted data. To configure it, on the Advanced tab, select Sorted Input

For example, when you wanted to calculate average salary of employees department wise, select the department_id as *Group By Field* under **Group By** tab.

Create new aggregate field Avg_Salary and assign value as AVG(Salary) under **Aggregate** tab of Aggregator transformation.

General

Incoming Fields

Group By

Aggregate

Advanced

Select fields to group by so that the resulting records can be set of rows with the same values for some fields.

Group by: Not Parameterized

Group by Fields

Field Name

DEP_ID

Group By fields in Aggregator Transformation

General

Incoming Fields

Group By

Aggregate

Advanced

Create simple aggregate expressions. You can also use expression macros to create complex aggregate

☐ Allow additional fields and expressions during task creation

Aggregate

Field Name

Expression

Avg_Salary

AVG(Salary)

Aggregate expression in Aggregator Transformation

Aggregator transformation is used to remove duplicates. This can be achieved when all the incoming fields are selected as Group By fields

General

Incoming Fields

Group By

Aggregate

Advanced

Tracing Level: ?

Normal

☒ Sorted Input ?

Cache Directory: ?

\$PMCacheDir

Aggregator Data Cache Size: ?

☒ Auto ☐ Value

Aggregator Index Cache Size: ?

☒ Auto ☐ Value

Transformation Scope: ?

All Input

☒ Optional ?

Advanced Properties of Aggregator Transformation

6.2 Types of Aggregator Cache

Aggregator uses Index Cache and Data Cache

- Index Cache stores group values as configured in Group By Fields.
- Data Cache stores calculations based on Group By Fields.

6.3 How Sorted Input increases Performance in Aggregator?

When sorted input option is not enabled, the aggregator does not know when a group by field would come. So it holds the entire data and process record by record.

When Sorted Input is enabled, the group by fields are expected to be sent in a sorted order. It creates an Index cache on the first group by fields defined and starts adding their values in the data cache.

When the task reads data for different group, it performs aggregate calculations for the cached group, and then continues with the next group.

There by a set of data is already forwarded from aggregator to the downstream transformation while it is making aggregate calculations on the next group.

6.4 What happens when no fields are selected as Group By Fields in Aggregator?

When no fields are selected as *Group By* field, aggregator creates a default index and it will keep on overriding each record in the data cache. So finally, the last record of the data will be sent as output.

7. Sequence Transformation in Informatica Cloud

Sequence Generator transformation is a passive and connected transformation in Informatica Cloud. It is used to generate sequence of numeric values.

Use Sequence generator transformation to

- Generate sequence of unique numbers
- Generate a cyclic sequence of numbers

7.1 Sequence Generator Fields

The Sequence Generator transformation has two output fields, NEXTVAL and CURRVAL of datatype big int. No other ports can be added and default ports can't be removed.

Output Fields in Sequence Transformation

Use the NEXTVAL field to generate a sequence of numbers based on the Initial Value and Increment By properties.

CURRVAL port value is always NEXTVAL+1. If you connect only the CURRVAL port without connecting the NEXTVAL port, then the mapping task generates a constant value for each row.

7.2 Sequence Generator Types in Informatica Cloud

Sequence generator can be used in two different ways in Informatica cloud. One with Incoming fields disabled and the other with incoming fields not disabled.

Sequence Generator with Incoming fields not disabled

Sequence

Generator with Incoming fields disabled

In order to disable the incoming fields, navigate to *Advanced* tab of sequence generator and check the *Disable incoming fields* option.

The screenshot shows the configuration interface for a Sequence generator. At the top, there are three tabs: 'Properties', 'Preview', and 'Sequence'. The 'Sequence' tab is active, indicated by a blue underline and a small icon. Below the tabs, on the left, is a vertical sidebar with five options: 'General', 'Incoming Fields', 'Sequence', 'Advanced', and 'Generated Fields'. The 'Advanced' option is selected and highlighted with a blue underline. The main area of the interface displays the configuration for the 'Advanced' tab. It includes a 'Tracing Level' dropdown menu set to 'Normal'. Below this, there are two checked checkboxes: 'Optional' and 'Disable incoming fields'. Each checkbox has a small question mark icon next to it.

Disabling Incoming fields in Sequence generator

The difference between the sequence generator with incoming fields enabled and disabled is when NEXTVAL field is mapped to multiple transformations

- Sequence generator with incoming fields not disabled will generate **same** sequence of numbers for each downstream transformation.
- Sequence generator with incoming fields disabled will generate **Unique** sequence of numbers for each downstream transformation.

To generate the same sequence of numbers when incoming fields are disabled, you can place an Expression transformation between the Sequence Generator and the downstream transformations to stage the sequence of numbers.

7.3 Sequence Generator Properties

- **Initial Value:** It is the first value that is generated by the sequence generator transformation. The default value is 1.
- **Increment By:** – This is the number by which you want to increment the sequence values. The default value is 1.
- **End value:** It is the maximum value that the sequence generator transformation generates.
- **Cycle:** If enabled, after reaching the end value, the transformation restarts from the initial value. By default, the option is disabled. If disabled and the sequence reaches the end value specified with rows still to process, the task would fail with overflow error.
- **Cycle Start Value:** Start value you want the sequence generator to use when cycle option is enabled. Default is 0.
- **Number of Cached Values:** The default value is 0. When the value is >0, the mapping task caches the number of sequential values specified and updates the value in repository. Once the cached values are used up it will again go to the repository. If there are unused sequence numbers in the cached values, the task will discard them.
Use this option when multiple partitions use the same Sequence Generator at the same time to ensure each partition receives unique values.
The disadvantage of using Number of Cached Values greater than zero are:
 - Accessing the repository multiple times during the session.
 - Discarding of unused cached values, causing discontinuous sequence numbers
- **Reset:** If enabled, the mapping task generates values based on the original Initial Value for each run. Default is disabled.