Informatica- Frequently Asked Questions

1. **What is meant by active and passive transformation?**

   An active transformation is the one that performs any of the following actions:

   1. Change the number of rows between transformation input and output. Example: Filter transformation

   2. Change the transaction boundary by defining commit or rollback points. Example transaction control transformation

   3. Change the row type, example Update strategy is active because it flags the rows for insert, delete, update or reject

   On the other hand a passive transformation is the one which does not change the number of rows that pass through it. Example: Expression transformation

2. **What can we do to improve the performance of Informatica Aggregator Transformation?**

   Aggregator performance improves dramatically if records are sorted before passing to the aggregator and "sorted input" option under aggregator properties is checked. The record set should be sorted on those columns that are used in Group By operation.

3. **What are the different lookup cache(s)?**

   Informatica Lookups can be cached or un-cached (No cache). And Cached lookup can be either static or dynamic. A **static cache** is one which does not modify the cache once it is built and it remains same during the session run. On the other hand, a **dynamic cache** is refreshed during the session run by inserting or updating the records in cache based on the incoming source data. By default, Informatica cache is static cache.

   A lookup cache can also be divided as **persistent** or **non-persistent** based on whether Informatica retains the cache even after the completion of session run or deletes it.

4. **How can we update a record in target table without using Update strategy?**

   A target table can be updated without using 'Update Strategy'. For this, we need to define the key in the target table in Informatica level and then we need to connect the key and the field we want to update in the mapping Target. In the session level, we should set the target property as "Update as Update" and check the "Update" check-box.

   Let's assume we have a target table "Customer" with fields as "Customer ID", "Customer Name" and "Customer Address". Suppose we want to update "Customer Address" without an Update Strategy. Then we have to define "Customer ID" as primary key in Informatica level and we will have to connect Customer ID and Customer Address fields in the mapping. If the session properties are set correctly as described above, then the mapping will only update the customer address field for all matching customer IDs.

5. **Why is Sorter an Active Transformation?**

   This is because we can select the "distinct" option in the sorter property.

   When the Sorter transformation is configured to treat output rows as distinct, it assigns all ports as part of the sort key. The Integration Service discards duplicate rows compared during the sort operation. The number of Input Rows will vary as compared with the Output rows and hence it is an Active transformation.

6. **What is a Source Qualifier? What are the tasks we can perform using a SQ and why it is an ACTIVE transformation?**

A **Source Qualifier** is an Active and Connected Informatica transformation that reads the rows from a relational database or flat file source.

- We can configure the **SQ** to **join** [Both **INNER** as well as **OUTER JOIN**] data originating from the same source database.
- We can use a source **filter** to reduce the number of rows the Integration Service queries.
- We can specify a number for **sorted ports** and the Integration Service adds an ORDER BY clause to the default SQL query.
- We can choose **Select Distinct** option for relational databases and the Integration Service adds a SELECT DISTINCT clause to the default SQL query.
- Also we can write **Custom/Used Defined SQL** query which will override the default query in the SQ by changing the default settings of the transformation properties.
- Also we have the option to write **Pre** as well as **Post SQL** statements to be executed before and after the SQ query in the source database.

Since the transformation provides us with the property **Select Distinct**, when the Integration Service adds a SELECT DISTINCT clause to the default SQL query, which in turn affects the number of rows returned by the Database to the Integration Service and hence it is an Active transformation.

7. **What is a Filter Transformation and why it is an Active one?**

   A **Filter** transformation is an **Active** and **Connected** transformation that can filter rows in a mapping.

   Only the rows that meet the **Filter Condition** pass through the Filter transformation to the next transformation in the pipeline. TRUE and FALSE are the implicit return values from any filter condition we set. If the filter condition evaluates to NULL, the row is assumed to be FALSE.

   The numeric equivalent of FALSE is zero (0) and any non-zero value is the equivalent of TRUE.

   As an **ACTIVE** transformation, the Filter transformation may change the number of rows passed through it. A filter condition returns TRUE or FALSE for each row that passes through the transformation, depending on whether a row meets the specified condition. Only rows

that return TRUE pass through this transformation. Discarded rows do not appear in the session log or reject files.

8. State the limitations where we cannot use Joiner in the mapping pipeline.

The Joiner transformation accepts input from most transformations. However, following are the limitations:

- Joiner transformation cannot be used when either of the input pipeline contains an **Update Strategy** transformation.
- Joiner transformation cannot be used if we connect a **Sequence Generator** transformation directly before the Joiner transformation.

9. What are the different types of Joins available in Joiner Transformation?

In SQL, a join is a relational operator that combines data from multiple tables into a single result set. The Joiner transformation is similar to an SQL join except that data can originate from different types of sources.

The Joiner transformation supports the following **types of joins**:

- **Normal**
- **Master Outer**
- **Detail Outer**
- **Full Outer**

10. What is a Sequence Generator Transformation?

A **Sequence Generator** transformation is a **Passive** and **Connected** transformation that generates numeric values. It is used to create unique primary key values, replace missing primary keys, or cycle through a sequential range of numbers. This transformation by **default** contains **ONLY Two OUTPUT** ports namely **CURRVAL** and **NEXTVAL**. We cannot edit or delete these ports neither we cannot add ports to this unique transformation. We can create approximately two billion unique numeric values with the widest range from 1 to 2147483647

11. What is an Aggregator Transformation?

An aggregator is an Active, Connected transformation which performs aggregate calculations like **AVG**, **COUNT**, **FIRST**, **LAST**, **MAX**, **MEDIAN**, **MIN**, **PERCENTILE**, **STDDEV**, **SUM** and **VARIANCE**.

12. What are the performance considerations when working with Aggregator Transformation?

Filter the unnecessary data before aggregating it. Place a Filter transformation in the mapping before the Aggregator transformation to reduce unnecessary aggregation. Improve performance by connecting only the necessary input/output ports to subsequent transformations, thereby reducing the size of the data cache. Use Sorted input which reduces the amount of data cached and improves session performance.

13. What differs when we choose Sorted Input for Aggregator Transformation?

Integration Service creates the index and data caches files in memory to process the Aggregator transformation. If the Integration Service requires more space as allocated for the index and data cache sizes in the transformation properties, it stores overflow values in cache files i.e. paging to disk. One way to increase session performance is to increase the index and data cache sizes in the transformation properties. But when we check Sorted Input the Integration Service uses memory to process an Aggregator transformation it does not use cache files.

14. What is a Rank Transform?

Rank is an Active Connected Informatica transformation used to select a set of top or bottom values of data.

15. What is a RANK port and RANKINDEX?

Rank port is an input/output port use to specify the column for which we want to rank the source values. By default Informatica creates an output port RANKINDEX for each Rank transformation. It stores the ranking position for each row in a group.

16. What are the restrictions of Rank Transformation?

- We can connect ports from only one transformation to the Rank transformation.
- We can select the top or bottom rank.
- We need to select the Number of records in each rank.
- We can designate only one Rank port in a Rank transformation.

17. How can you get ranks based on different groups?

Rank transformation lets us group information. We can configure one of its input/output ports as a group by port. For each unique value in the group port, the transformation creates a group of rows falling within the rank definition (top or bottom, and a particular number in each rank)

18. What is a Sorter Transformation?

Sorter Transformation is an Active, Connected Informatica transformation used to sort data in ascending or descending order according to specified sort keys. The Sorter transformation contains only input/output ports.

19. How does Sorter handle Case Sensitive sorting?

The Case Sensitive property determines whether the Integration Service considers case when sorting data. When we enable the Case Sensitive property, the Integration Service sorts uppercase characters higher than lowercase characters.

20. What is a Union Transformation?

The Union transformation is an Active, Connected non-blocking multiple input group transformation use to merge data from multiple pipelines or sources into one pipeline branch. Similar to the UNION ALL SQL statement, the Union transformation does not remove duplicate rows.

21. What is Persistent Lookup Cache?

Lookups are cached by default in Informatica. Lookup cache can be either non-persistent or persistent. The Integration Service saves or deletes lookup cache files after a successful session run based on whether the Lookup cache is checked as persistent or not.

22. What are the transformations that are not supported in Mapplet?

Normalizer, Cobol sources, XML sources, XML Source Qualifier transformations, Target definitions, Pre- and post- session Stored Procedures, Other Mapplets.

23. What is the difference between Reusable transformation and Mapplet?

Any Informatica Transformation created in the in the Transformation Developer or a non-reusable promoted to reusable transformation from the mapping designer which can be used in multiple mappings is known as Reusable Transformation. When we add a reusable transformation to a mapping, we actually add an instance of the transformation. Since the instance of a reusable transformation is a pointer to that transformation, when we change the transformation in the Transformation Developer, its instances reflect these changes.

A Mapplet is a reusable object created in the Mapplet Designer which contains a **set of transformations** and lets us reuse the transformation logic in multiple mappings. A Mapplet can contain as many transformations as we need. Like a reusable transformation when we use a mapplet in a mapping, we use an instance of the mapplet and any change made to the mapplet is inherited by all instances of the mapplet.

24. What is the difference between Static and Dynamic Lookup Cache?

We can configure a Lookup transformation to cache the corresponding lookup table. In case of static or read-only lookup cache the Integration Service caches the lookup table at the beginning of the session and does not update the lookup cache while it processes the Lookup transformation.

In case of dynamic lookup cache the Integration Service dynamically inserts or updates data in the lookup cache and passes the data to the target. The dynamic cache is synchronized with the target.

25. How to Delete duplicate row using Informatica using aggregator?

Other ways to handle duplicate records in source batch run is to use an **Aggregator Transformation** and using the **Group By** checkbox on the ports having duplicate occurring data. Here you can have the flexibility to select the *last or the first* of the duplicate column value records.