

Repository related Questions

Q. What is the difference between PowerCenter and PowerMart?

With PowerCenter, you receive all product functionality, including the ability to register multiple servers, share metadata across repositories, and partition data.

A PowerCenter license lets you create a single repository that you can configure as a global repository, the core component of a data warehouse.

PowerMart includes all features except distributed metadata, multiple registered servers, and data partitioning. Also, the various options available with PowerCenter (such as PowerCenter Integration Server for BW, PowerConnect for IBM DB2, PowerConnect for IBM MQSeries, PowerConnect for SAP R/3, PowerConnect for Siebel, and PowerConnect for PeopleSoft) are not available with PowerMart.

Q. What are the new features and enhancements in PowerCenter 5.1?

The major features and enhancements to PowerCenter 5.1 are:

a) Performance Enhancements

- **High precision decimal arithmetic.** The Informatica Server optimizes data throughput to increase performance of sessions using the Enable Decimal Arithmetic option.
- **To_Decimal and Aggregate functions.** The Informatica Server uses improved algorithms to increase performance of To_Decimal and all aggregate functions such as percentile, median, and average.
- **Cache management.** The Informatica Server uses better cache management to increase performance of Aggregator, Joiner, Lookup, and Rank transformations.
- **Partition sessions with sorted aggregation.** You can partition sessions with Aggregator transformation that use sorted input. This improves memory usage and increases performance of sessions that have sorted data.

b) Relaxed Data Code Page Validation

When enabled, the Informatica Client and Informatica Server lift code page selection and validation restrictions. You can select any supported code page for source, target, lookup, and stored procedure data.

c) Designer Features and Enhancements

- **Debug mapplets.** You can debug a mapplet within a mapping in the Mapping Designer. You can set breakpoints in transformations in the mapplet.
- **Support for slash character (/) in table and field names.** You can use the Designer to import source and target definitions with table and field names containing the slash character (/). This allows you to import SAP BW source definitions by connecting directly to the underlying database tables.

d) Server Manager Features and Enhancements

- **Continuous sessions.** You can schedule a session to run continuously. A continuous session starts automatically when the Load Manager starts. When the session stops, it

- restarts immediately without rescheduling. Use continuous sessions when reading real time sources, such as IBM MQSeries.
- **Partition sessions with sorted aggregators.** You can partition sessions with sorted aggregators in a mapping.
 - **Register multiple servers against a local repository.** You can register multiple PowerCenter Servers against a local repository.

Q. What is a repository?

The Informatica repository is a relational database that stores information, or *metadata*, used by the Informatica Server and Client tools. The repository also stores administrative information such as usernames and passwords, permissions and privileges, and product version.

We create and maintain the repository with the Repository Manager client tool. With the Repository Manager, we can also create folders to organize metadata and groups to organize users.

Q. What are different kinds of repository objects? And what it will contain?

Repository objects displayed in the Navigator can include sources, targets, transformations, mappings, mapplets, shortcuts, sessions, batches, and session logs.

Q. What is a metadata?

Designing a data mart involves writing and storing a complex set of instructions. You need to know where to get data (sources), how to change it, and where to write the information (targets). PowerMart and PowerCenter call this set of instructions metadata. Each piece of metadata (for example, the description of a source table in an operational database) can contain comments about it.

In summary, Metadata can include information such as mappings describing how to transform source data, sessions indicating when you want the Informatica Server to perform the transformations, and connect strings for sources and targets.

Q. What are folders?

Folders let you organize your work in the repository, providing a way to separate different types of metadata or different projects into easily identifiable areas.

Q. What is a Shared Folder?

A shared folder is one, whose contents are available to all other folders in the same repository. If we plan on using the same piece of metadata in several projects (for example, a description of the CUSTOMERS table that provides data for a variety of purposes), you might put that metadata in the shared folder.

Q. What are mappings?

A mapping specifies how to move and transform data from sources to targets. Mappings include source and target definitions and transformations. Transformations describe how the Informatica Server transforms data. Mappings can also include shortcuts, reusable transformations, and mapplets. Use the Mapping Designer tool in the Designer to create mappings.

Q. What are mapplets?

You can design a mapplet to contain sets of transformation logic to be reused in multiple mappings within a folder, a repository, or a domain. Rather than recreate the same set of transformations each time, you can create a mapplet containing the transformations, then add instances of the mapplet to individual mappings. Use the Mapplet Designer tool in the Designer to create mapplets.

Q. What are Transformations?

A transformation generates, modifies, or passes data through ports that you connect in a mapping or mapplet. When you build a mapping, you add transformations and configure them to handle data according to your business purpose. Use the Transformation Developer tool in the Designer to create transformations.

Q. What are Reusable transformations?

You can design a transformation to be reused in multiple mappings within a folder, a repository, or a domain. Rather than recreate the same transformation each time, you can make the transformation reusable, then add instances of the transformation to individual mappings. Use the Transformation Developer tool in the Designer to create reusable transformations.

Q. What are Sessions and Batches?

Sessions and batches store information about how and when the Informatica Server moves data through mappings. You create a session for each mapping you want to run. You can group several sessions together in a batch. Use the Server Manager to create sessions and batches.

Q. What are Shortcuts?

We can create shortcuts to objects in shared folders. Shortcuts provide the easiest way to reuse objects. We use a shortcut as if it were the actual object, and when we make a change to the original object, all shortcuts inherit the change.

Shortcuts to folders in the same repository are known as local shortcuts. Shortcuts to the global repository are called global shortcuts.

We use the Designer to create shortcuts.

Q. What are Source definitions?

Detailed descriptions of database objects (tables, views, synonyms), flat files, XML files, or Cobol files that provide source data. For example, a source definition might be the complete structure of the EMPLOYEES table, including the table name, column names and datatypes, and any constraints applied to these columns, such as NOT NULL or PRIMARY KEY. Use the Source Analyzer tool in the Designer to import and create source definitions.

Q. What are Target definitions?

Detailed descriptions for database objects, flat files, Cobol files, or XML files to receive transformed data. During a session, the Informatica Server writes the resulting data to session targets. Use the Warehouse Designer tool in the Designer to import or create target definitions.

Q. What is Dynamic Data Store?

The need to share data is just as pressing as the need to share metadata. Often, several data marts in the same organization need the same information. For example, several data marts may

need to read the same product data from operational sources, perform the same profitability calculations, and format this information to make it easy to review.

If each data mart reads, transforms, and writes this product data separately, the throughput for the entire organization is lower than it could be. A more efficient approach would be to read, transform, and write the data to one central data store shared by all data marts. Transformation is a processing-intensive task, so performing the profitability calculations once saves time.

Therefore, this kind of dynamic data store (DDS) improves throughput at the level of the entire organization, including all data marts. To improve performance further, you might want to capture incremental changes to sources. For example, rather than reading all the product data each time you update the DDS, you can improve performance by capturing only the inserts, deletes, and updates that have occurred in the PRODUCTS table since the last time you updated the DDS.

The DDS has one additional advantage beyond performance: when you move data into the DDS, you can format it in a standard fashion. For example, you can prune sensitive employee data that should not be stored in any data mart. Or you can display date and time values in a standard format. You can perform these and other data cleansing tasks when you move data into the DDS instead of performing them repeatedly in separate data marts.

Q. When should you create the dynamic data store? Do you need a DDS at all?

To decide whether you should create a dynamic data store (DDS), consider the following issues:

- **How much data do you need to store in the DDS?** The one principal advantage of data marts is the selectivity of information included in it. Instead of a copy of everything potentially relevant from the OLTP database and flat files, data marts contain only the information needed to answer specific questions for a specific audience (for example, sales performance data used by the sales division). A dynamic data store is a hybrid of the galactic warehouse and the individual data mart, since it includes all the data needed for all the data marts it supplies. If the dynamic data store contains nearly as much information as the OLTP source, you might not need the intermediate step of the dynamic data store. However, if the dynamic data store includes substantially less than all the data in the source databases and flat files, you should consider creating a DDS staging area.
-
- **What kind of standards do you need to enforce in your data marts?** Creating a DDS is an important technique in enforcing standards. If data marts depend on the DDS for information, you can provide that data in the range and format you want everyone to use. For example, if you want all data marts to include the same information on customers, you can put all the data needed for this standard customer profile in the DDS. Any data mart that reads customer data from the DDS should include all the information in this profile.
-
- **How often do you update the contents of the DDS?** If you plan to frequently update data in data marts, you need to update the contents of the DDS at least as often as you update the individual data marts that the DDS feeds. You may find it easier to read data directly from source databases and flat file systems if it becomes burdensome to update the DDS fast enough to keep up with the needs of individual data marts. Or, if particular data marts need updates significantly faster than others, you can bypass the DDS for these fast update data marts.
-
- **Is the data in the DDS simply a copy of data from source systems, or do you plan to reformat this information before storing it in the DDS?** One advantage of the dynamic data store is that, if you plan on reformatting information in the same fashion for

several data marts, you only need to format it once for the dynamic data store. Part of this question is whether you keep the data normalized when you copy it to the DDS.

-
- **How often do you need to join data from different systems?** On occasion, you may need to join records queried from different databases or read from different flat file systems. The more frequently you need to perform this type of heterogeneous join, the more advantageous it would be to perform all such joins within the DDS, then make the results available to all data marts that use the DDS as a source.

Q. What is a Global repository?

The centralized repository in a domain, a group of connected repositories. Each domain can contain one global repository. The global repository can contain common objects to be shared throughout the domain through global shortcuts. Once created, you cannot change a global repository to a local repository. You can promote an existing local repository to a global repository.

Q. What is Local Repository?

Each local repository in the domain can connect to the global repository and use objects in its shared folders. A folder in a local repository can be copied to other local repositories while keeping all local and global shortcuts intact.

Q. What are the different types of locks?

There are five kinds of locks on repository objects:

- **Read lock.** Created when you open a repository object in a folder for which you do not have write permission. Also created when you open an object with an existing write lock.
- **Write lock.** Created when you create or edit a repository object in a folder for which you have write permission.
- **Execute lock.** Created when you start a session or batch, or when the Informatica Server starts a scheduled session or batch.
- **Fetch lock.** Created when the repository reads information about repository objects from the database.
- **Save lock.** Created when you save information to the repository.

Q. After creating users and user groups, and granting different sets of privileges, I find that none of the repository users can perform certain tasks, even the Administrator.

Repository privileges are limited by the database privileges granted to the database user who created the repository. If the database user (one of the default users created in the Administrators group) does not have full database privileges in the repository database, you need to edit the database user to allow all privileges in the database.

Q. I created a new group and removed the Browse Repository privilege from the group. Why does every user in the group still have that privilege?

Privileges granted to individual users take precedence over any group restrictions. Browse Repository is a default privilege granted to all new users and groups. Therefore, to remove the privilege from users in a group, you must remove the privilege from the group, and every user in the group.

Q. I do not want a user group to create or edit sessions and batches, but I need them to access the Server Manager to stop the Informatica Server.

To permit a user to access the Server Manager to stop the Informatica Server, you must grant them both the Create Sessions and Batches, and Administer Server privileges. To restrict the user from creating or editing sessions and batches, you must restrict the user's write permissions on a folder level.

Alternatively, the user can use *pmcmd* to stop the Informatica Server with the Administer Server privilege alone.

Q. How does read permission affect the use of the command line program, *pmcmd*?

To use *pmcmd*, you do not need to view a folder before starting a session or batch within the folder. Therefore, you do not need read permission to start sessions or batches with *pmcmd*. You must, however, know the exact name of the session or batch and the folder in which it exists.

With *pmcmd*, you can start any session or batch in the repository if you have the Session Operator privilege or execute permission on the folder.

Q. My privileges indicate I should be able to edit objects in the repository, but I cannot edit any metadata.

You may be working in a folder with restrictive permissions. Check the folder permissions to see if you belong to a group whose privileges are restricted by the folder owner.

Q. I have the Administer Repository Privilege, but I cannot access a repository using the Repository Manager.

To perform administration tasks in the Repository Manager with the Administer Repository privilege, you must also have the default privilege Browse Repository. You can assign Browse Repository directly to a user login, or you can inherit Browse Repository from a group.

Questions related to Server Manager

Q. What is Event-Based Scheduling?

When you use event-based scheduling, the Informatica Server starts a session when it locates the specified indicator file. To use event-based scheduling, you need a shell command, script, or batch file to create an indicator file when all sources are available. The file must be created or sent to a directory local to the Informatica Server. The file can be of any format recognized by the Informatica Server operating system. The Informatica Server deletes the indicator file once the session starts.

Use the following syntax to ping the Informatica Server on a UNIX system:

```
pmcmd ping [{user_name | %user_env_var} {password | %password_env_var}]  
[hostname:]portno
```

Use the following syntax to start a session or batch on a UNIX system:

```
pmcmd start {user_name | %user_env_var} {password | %password_env_var} [hostname:]portno  
[folder_name:]{session_name | batch_name} [:pf=param_file] session_flag wait_flag
```

Use the following syntax to stop a session or batch on a UNIX system:

```
pmcmd stop {user_name | %user_env_var} {password | %password_env_var}  
[hostname:]portno[folder_name:]{session_name | batch_name} session_flag
```

Use the following syntax to stop the Informatica Server on a UNIX system:

```
pmcmd stopserver {user_name | %user_env_var} {password | %password_env_var}  
[hostname:]portno
```

Q. What are the different types of Commit intervals?

The different commit intervals are:

- **Target-based commit.** The Informatica Server commits data based on the number of target rows and the key constraints on the target table. The commit point also depends on the buffer block size and the commit interval.
- **Source-based commit.** The Informatica Server commits data based on the number of source rows. The commit point is the commit interval you configure in the session properties.

Designer Questions

Q. What are the tools provided by Designer?

The Designer provides the following tools:

- **Source Analyzer.** Use to import or create source definitions for flat file, XML, Cobol, ERP, and relational sources.
- **Warehouse Designer.** Use to import or create target definitions.
- **Transformation Developer.** Use to create reusable transformations.
- **Mapplet Designer.** Use to create mapplets.
- **Mapping Designer.** Use to create mappings.

Q. What is a transformation?

A transformation is a repository object that generates, modifies, or passes data. You configure logic in a transformation that the Informatica Server uses to transform data. The Designer provides a set of transformations that perform specific functions. For example, an Aggregator transformation performs calculations on groups of data.

Each transformation has rules for configuring and connecting in a mapping. For more information about working with a specific transformation, refer to the chapter in this book that discusses that particular transformation.

You can create transformations to use once in a mapping, or you can create reusable transformations to use in multiple mappings.

Q. What are the different types of Transformations? (Mascot)

a) **Aggregator transformation:** The Aggregator transformation allows you to perform aggregate calculations, such as averages and sums. The Aggregator transformation is unlike the Expression transformation, in that you can use the Aggregator transformation to perform calculations on groups. The Expression transformation permits you to perform calculations on a row-by-row basis only. (Mascot)

b) **Expression transformation:** You can use the Expression transformations to calculate values in a single row before you write to the target. For example, you might need to adjust employee salaries, concatenate first and last names, or convert strings to numbers. You can use the Expression transformation to perform any non-aggregate calculations. You can also use the Expression transformation to test conditional statements before you output the results to target tables or other transformations.

c) **Filter transformation:** The Filter transformation provides the means for filtering rows in a mapping. You pass all the rows from a source transformation through the Filter transformation, and then enter a filter condition for the transformation. All ports in a Filter transformation are input/output, and only rows that meet the condition pass through the Filter transformation.

d) **Joiner transformation:** While a Source Qualifier transformation can join data originating from a common source database, the Joiner transformation joins two related heterogeneous sources residing in different locations or file systems.

e) **Lookup transformation:** Use a Lookup transformation in your mapping to look up data in a relational table, view, or synonym. Import a lookup definition from any relational database to which both the Informatica Client and Server can connect. You can use multiple Lookup transformations in a mapping.

The Informatica Server queries the lookup table based on the lookup ports in the transformation. It compares Lookup transformation port values to lookup table column values based on the lookup condition. Use the result of the lookup to pass to other transformations and the target.

Q. What is the difference between Aggregate and Expression Transformation? (Mascot)

Q. What is Update Strategy?

When we design our data warehouse, we need to decide what type of information to store in targets. As part of our target table design, we need to determine whether to maintain all the historic data or just the most recent changes.

The model we choose constitutes our *update strategy*, how to handle changes to existing records.

Update strategy flags a record for update, insert, delete, or reject. We use this transformation when we want to exert fine control over updates to a target, based on some condition we apply. For example, we might use the Update Strategy transformation to flag all customer records for update when the mailing address has changed, or flag all employee records for reject for people no longer working for the company.

Q. Where do you define update strategy?

We can set the Update strategy at two different levels:

- **Within a session.** When you configure a session, you can instruct the Informatica Server to either treat all records in the same way (for example, treat all records as inserts), or use instructions coded into the session mapping to flag records for different database operations.
- **Within a mapping.** Within a mapping, you use the Update Strategy transformation to flag records for insert, delete, update, or reject.

Q. What are the advantages of having the Update strategy at Session Level?

Q. What is a lookup table? (KPIT Infotech, Pune)

The lookup table can be a single table, or we can join multiple tables in the same database using a lookup query override. The Informatica Server queries the lookup table or an in-memory cache of the table for all incoming rows into the Lookup transformation.

If your mapping includes heterogeneous joins, we can use any of the mapping sources or mapping targets as the lookup table.

Q. What is a Lookup transformation and what are its uses?

We use a Lookup transformation in our mapping to look up data in a relational table, view or synonym.

We can use the Lookup transformation for the following purposes:

- ❖ Get a related value. For example, if our source table includes employee ID, but we want to include the employee name in our target table to make our summary data easier to read.
- ❖ Perform a calculation. Many normalized tables include values used in a calculation, such as gross sales per invoice or sales tax, but not the calculated value (such as net sales).
- ❖ Update slowly changing dimension tables. We can use a Lookup transformation to determine whether records already exist in the target.

Q. What are connected and unconnected Lookup transformations?

We can configure a connected Lookup transformation to receive input directly from the mapping pipeline, or we can configure an unconnected Lookup transformation to receive input from the result of an expression in another transformation.

An unconnected Lookup transformation exists separate from the pipeline in the mapping. We write an expression using the :LKP reference qualifier to call the lookup within another transformation.

A common use for unconnected Lookup transformations is to update slowly changing dimension tables.

Q. What is the difference between connected lookup and unconnected lookup?

Differences between Connected and Unconnected Lookups:

Connected Lookup	Unconnected Lookup
Receives input values directly from the pipeline.	Receives input values from the result of a :LKP expression in another transformation.
We can use a dynamic or static cache	We can use a static cache
Supports user-defined default values	Does not support user-defined default values

Q. What is Sequence Generator Transformation? (Mascot)

The Sequence Generator transformation generates numeric values. We can use the Sequence Generator to create unique primary key values, replace missing primary keys, or cycle through a sequential range of numbers.

The Sequence Generation transformation is a connected transformation. It contains two output ports that we can connect to one or more transformations.

Q. What are the uses of a Sequence Generator transformation?

We can perform the following tasks with a Sequence Generator transformation:

- o Create keys
- o Replace missing values
- o Cycle through a sequential range of numbers

Q. What are the advantages of Sequence generator? Is it necessary, if so why?

We can make a Sequence Generator reusable, and use it in multiple mappings. We might reuse a Sequence Generator when we perform multiple loads to a single target.

For example, if we have a large input file that we separate into three sessions running in parallel, we can use a Sequence Generator to generate primary key values. If we use different Sequence Generators, the Informatica Server might accidentally generate duplicate key values. Instead, we can use the same reusable Sequence Generator for all three sessions to provide a unique value for each target row.

Q. How is the Sequence Generator transformation different from other transformations?

The Sequence Generator is unique among all transformations because we cannot add, edit, or delete its default ports (NEXTVAL and CURRVAL).

Unlike other transformations we cannot override the Sequence Generator transformation properties at the session level. This protects the integrity of the sequence values generated.

Q. What does Informatica do? How it is useful?

Q. What is the difference between Informatica version 1.7.2 and 1.7.3?

Q. What are the complex filters used till now in your applications?

Q. Features of Informatica

Q. Have you used Informatica? which version?

Q. How do you set up a schedule for data loading from scratch? describe step-by-step.

Q. How do you use mapplet?

Q. What are the different data source types you have used with Informatica?

Q. Is it possible to run one loading session with one particular target and multiple types of data sources?