ISSN: 2278-6244
Impact Factor: 6.111

EXTRACT TRANSFORM LOAD DATA WITH ETL TOOLS LIKE 'INFORMATICA'

Preeti, M. Tech., Computer Science Engineering, Ganga Institute of Technology and Management, Bahadurgarh-Jhajjar Road, Kablana, Distt. Jhajjar, Haryana

Neetu Sharma, HOD C.S.E Deptt., Ganga Institute of Technology and Management, Bahadurgarh-Jhajjar Road, Kablana, Distt. Jhajjar, Haryana

Abstract: *As we all know business intelligence (BI) is considered to have an extraordinary impact on businesses. Research activity has grown in the last years [10]. A significant part of BI systems is a well performing Implementation of the Extract, Transform, and Load (ETL) process. In typical BI projects, implementing the ETL process can be the task with the greatest effort. Here, set of generic Meta model constructs with a palette of regularly used ETL activities, is specialized, which are called templates.*

1. INTRODUCTION

We all want to load our data warehouse regularly so that it can assist its purpose of facilitating business analysis [1]. To do this, data from one or more operational systems desires to be extracted and copied into the warehouse. The process of extracting data from source systems and carrying it into the data warehouse is commonly called ETL, which stands for extraction, transformation, and loading. It is an essential phenomenon in a data warehouse. Whenever DML (data manipulation language) operations such as INSERT, UPDATE OR DELETE are issued on the source database, data extraction occurs. After data extraction and transformation have taken place, data are loaded into the data warehouse.

Extraction: The first part of an ETL process is to extract the data from the home systems. Most data warehousing projects amalgamate data from unlike source systems. Each separate system may also use a different data association format. Common data source formats are relational databases and flat files, but may contain non-relational database structures such as IMS or other data structures .Extraction converts the data into a format for transformation processing. The quantity of data is reduced by omitting any non-relevant data sets. Extraction must not negatively affect the performance of productive systems. It runs as a background task or is executed at times of low activity (e.g. during the night).

Transformation: Any transformation desirable to provide data that can be interpreted in business terms is done in the second step. Data sets are cleaned with regard to their data

quality. Eventually, they are converted to the scheme of the target database and consolidated. The transform stage applies a series of rules or functions to the extracted data to derive the data to be loaded. Some data sources will require very slight manipulation of data. Data transformations are often the most difficult and, in terms of processing time, the most costly part of the ETL process. They can range from simple data conversions to extremely complex data scrubbing techniques.

Loading: Now, the real loading of data into the data warehouse has to be done. The early Load which generally is not time-critical is great from the Incremental Load. Whereas the first phase affected productive systems, loading can have an giant effect on the data warehouse. This especially has to be taken into consideration with regard to the complex task of updating currently stored data sets. In general, incremental loading is a critical task. ETL processes can either be run in batch mode or real time. Batch jobs typically are run periodically. If intervals become as short as hours or even minutes only, these processes are called near real time. The load phase loads the data into the data warehouse. Depending on the requirements of the organization, this process ranges widely. Some data warehouses merely overwrite old information with new data.

More complex systems can maintain a history and audit trail of all changes to the data. Designing and maintaining the ETL process is often considered one of the most difficult and resource-intensive portions of a data warehouse project. Many data warehousing projects use ETL tools to manage this process. Data warehouse builders create their own ETL tools and processes, either inside or outside the database.

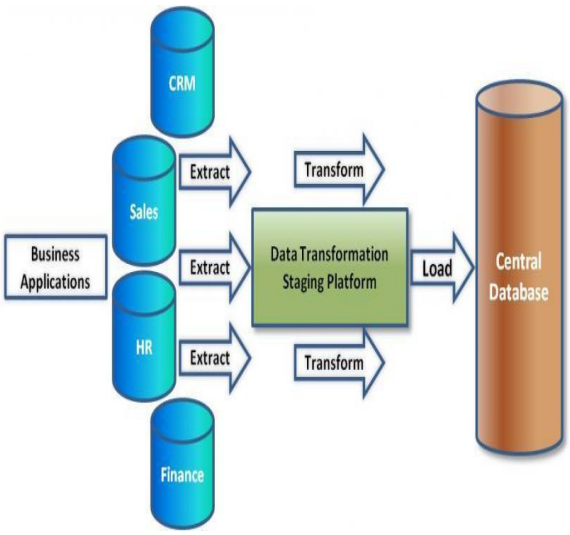


Fig. The ETL Process

ETL tools have been around for some time, have evolved, matured, and now present us with productive environments for Big Data, Data Warehouse, Business Intelligence, and analytics processing. However these are few problems with them .Such tools are very expensive and does not support small size businesses. Configuration of such tools takes lot of time. These are many ETL tools availble in the market .Some of the ETL Tools are:

- DataStage from Ascential Software
- SAS System from SAS Institute
- Informatica
- Data Integrator From BO
- Oracle Express
- Abinito
- Decision Stream From Cognos
- MS-DTS from Microsoft
- Pentaho Kettle

Informatica is the best ETL tool in the marketplace [3]. It can extract data from numerous heterogeneous sources, transforming them as per business needs and loading to target tables. It's used in Data migration and loading projects. It is a visual interface and you will be dragging and dropping with the mouse in the Designer(client Application).This graphical approach to communicate with all major databases and can move/transform data between them. It can move huge bulk of data in a very effective way. Informatica is a tool, supporting all the steps of Extraction, Transformation and Load process. Now a days Informatica is also being used as an Integration tool.

Informatica is an easy to use tool. It has got a simple visual interface like forms in visual basic. You just need to drag and drop different objects (known as transformations) and design process flow for Data extraction transformation and load. These process flow diagrams are known as mappings. Once a mapping is made, it can be scheduled to run as and when required. In the background Informatica server takes care of fetching data from source, transforming it, & loading it to the target systems/databases.

Informatica can talk with all major data sources (mainframe/RDBMS/Flat Files/XML/VSM/SAP etc), [3] can move/transform data between them. It can move huge

volumes of data in a very operational way, many a times better than even bespoke programs written for specific data movement only. It can throttle the transactions (do big updates in small chunks to avoid long locking and filling the transactional log). It can effectively join data from two distinct data sources (even a xml file can be joined with a relational table). In all, Informatica has got the ability to effectively integrate heterogeneous data sources & converting raw data into useful information.

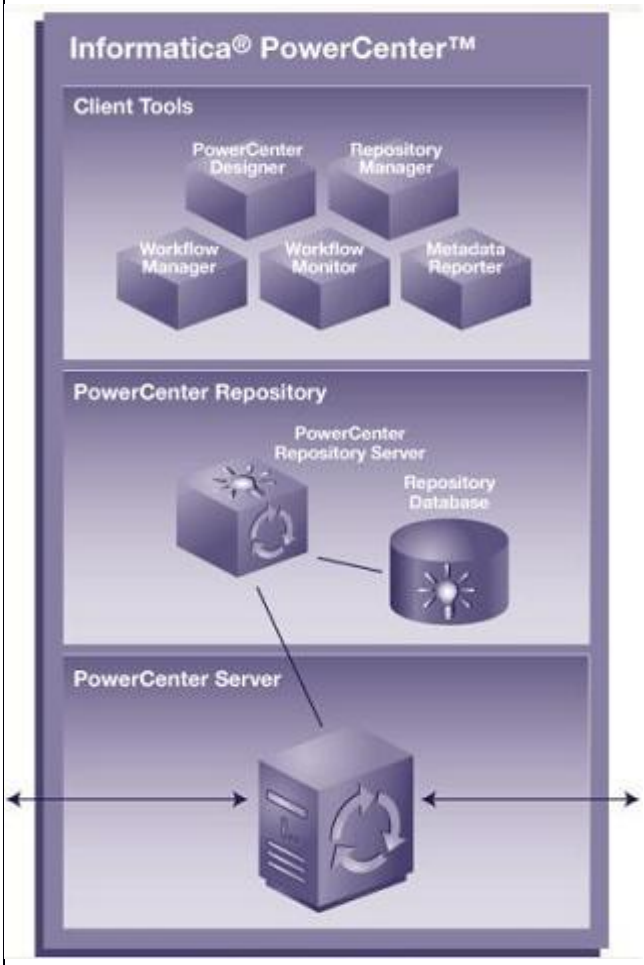
Some facts and figures about Informatica Corporation:

- Founded in 1993, based in Redwood City, California
- 1400+ Employees; 3450 + Customers; 79 of the Fortune 100 Companies
- NASDAQ Stock Symbol: INFA; Stock Price: $18.74 (09/04/2009)
- Revenues in fiscal year 2008: $455.7M
- Headquarters: Redwood City, CA
- Offices: N. & S. America, Europe, Asia Pacific
- Government organizations in 20 countries
- Partners: Over 400 major SI, ISV,OEM and On Demand

2. COMPONENTS OF INFORMATICA

Informatica provides the following integrated components:

Informatica repository. The Informatica repository is at the center of the Informatica suite. You create a set of metadata tables within the repository database that the Informatica applications and tools access. The Informatica Client and Server access the repository to save and retrieve metadata. The PowerCenter repository resides on a relational database. The repository database tables contain the instructions required to extract, transform, and load data. PowerCenter Client applications access the repository database tables through the Repository Server. You add metadata to the repository tables when you perform tasks in the PowerCenter Client application, such as creating users, analyzing sources, developing mappings or mapplets, or creating workflows. The PowerCenter Server reads metadata created in the Client application when you run a workflow. The PowerCenter Server also creates metadata, such as start and finish times of a session or session status[2].

Sources		Targets
Standard: RDBMS, Flat Files, XML, ODBC		**Standard:** RDBMS, Flat Files, XML, ODBC
Applications: SAP R/3, SAP BW, PeopleSoft, Siebel, JD Edwards, i2		**Applications:** SAP R/3, SAP BW, PeopleSoft, Siebel, JD Edwards, i2
EAI: MQ Series, Tibco, JMS, Web Services		**EAI:** MQ Series, Tibco, JMS, Web Services
Legacy: Mainframes (DB2, VSAM, IMS, IDMS, Adabas)AS400 (DB2, Flat File)		**Legacy:** Mainframes (DB2)AS400 (DB2)
Remote Sources		**Remote Targets**

You can develop global and local repositories to share metadata:

- **Global repository.** The global repository is the hub of the domain. Use the global repository to store common objects that multiple developers can use through shortcuts. These objects may include operational or Application source definitions, reusable transformations, mapplets, and mappings.
- **Local repositories.** A local repository is within a domain that is not the global repository. Use local repositories for development. From a local repository, you can create shortcuts to objects in shared folders in the global repository. These objects typically include source definitions, common dimensions and lookups, and enterprise standard transformations. You can also create copies of objects in non-shared folders.

- **Version control.** A versioned repository can store multiple copies, or versions, of an object. Each version is a separate object with unique properties. PowerCenter version control features allow you to efficiently develop, test, and deploy metadata into production.

You can connect to a repository, back up, delete, or restore repositories using *pmrep*, a command line program.

You can view much of the metadata in the Repository Manager. The Informatica Metadata Exchange (MX) provides a set of relational views that allow easy SQL access to the Informatica metadata repository.

Repository Server

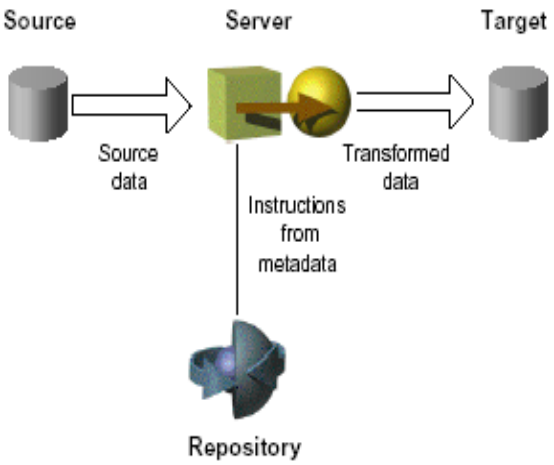
The Repository Server manages repository connection requests from client applications. For each repository database registered with the Repository Server, it configures and manages a Repository Agent process. The Repository Server also monitors the status of running Repository Agents, and sends repository object notification messages to client applications.

Informatica Client. Use the Informatica Client to manage users, define sources and targets, build mappings and mapplets with the transformation logic, and create sessions to run the mapping logic. The Informatica Client has three client applications: Repository Manager, Designer, and Workflow Manager.

- **Repository Server Administration Console.** Use the Repository Server Administration console to administer the Repository Servers and repositories.
- **Repository Manager.** Use the Repository Manager to administer the metadata repository. You can create repository users and groups, assign privileges and permissions, and manage folders and locks.
- **Designer.** Use the Designer to create mappings that contain transformation instructions for the PowerCenter Server. Before you can create mappings, you must add source and target definitions to the repository. The Designer has five tools that you use to analyze sources, design target schemas, and build source-to-target mappings:
  - **Source Analyzer.** Import or create source definitions.
  - **Warehouse Designer.** Import or create target definitions.

- **Transformation Developer.** Develop reusable transformations to use in mappings.
- **Mapplet Designer.** Create sets of transformations to use in mappings.
- **Mapping Designer.** Create mappings that the PowerCenter Server uses to extract, transform, and load data.

- **Workflow Manager.** Use the Workflow Manager to create, schedule, and run workflows. A workflow is a set of instructions that describes how and when to run tasks related to extracting, transforming, and loading data. The PowerCenter Server runs workflow tasks according to the links connecting the tasks. You can run a task by placing it in a workflow.
- **Workflow Monitor.** Use the Workflow Monitor to monitor scheduled and running workflows for each PowerCenter Server. You can choose a Gantt chart or Task view. You can also access details about those workflow runs.



InformaticaServer: The Informatica Server extracts the source data, performs the data transformation, and loads the transformed data into the targets.

The PowerCenter Server reads mapping and session information from the repository. It extracts data from the mapping sources and stores the data in memory while it applies the transformation rules that you configure in the mapping. The PowerCenter Server loads the transformed data into the mapping targets.

The PowerCenter Server can achieve high performance using symmetric multi-processing systems. The PowerCenter Server can start and run multiple workflows concurrently. It can also concurrently process partitions within a single session. When you create multiple

partitions within a session, the PowerCenter Server creates multiple database connections to a single source and extracts a separate range of data for each connection, according to the properties you configure.

3. INFORMATICA PRODUCT LINE

Informatica is a powerful ETL tool from Informatica Corporation, a leading provider of enterprise data integration software and ETL softwares.

The important products provided by Informatica Corporation is provided below:

- Power Center
- Power Mart
- Power Exchange
- Power Center Connect
- Power Channel
- Metadata Exchange
- Power Analyzer
- Super Glue

Power Center & Power Mart: Power Mart is a departmental version of Informatica for building, deploying, and managing data warehouses and data marts. Power center is used for corporate enterprise data warehouse and power mart is used for departmental data warehouses like data marts. Power Center supports global repositories and networked repositories and it can be connected to several sources. Power Mart supports single repository and it can be connected to fewer sources when compared to Power Center. Power Mart can extensibily grow to an enterprise implementation and it is easy for developer productivity through a codeless environment.

Power Exchange: Informatica Power Exchange as a stand-alone service or along with Power Center, helps organizations leverage data by avoiding manual coding of data extraction programs. Power Exchange supports batch, real time and changed data capture options in main frame(DB2, VSAM, IMS etc.,), mid-range (AS400 DB2 etc.,), and for relational databases (oracle, sql server, db2 etc) and flat files in unix, linux and windows systems.

Power Center Connect: This is adding on to Informatica Power Center. It helps to extract data and metadata from ERP systems like IBM's MQSeries, Peoplesoft, SAP, Siebel etc. and other third party applications.

Power Channel: This helps to transfer large amount of encrypted and compressed data over LAN, WAN, through Firewalls, tranfer files over FTP, etc.

Meta Data Exchange: Metadata Exchange enables organizations to take advantage of the time and effort already invested in defining data structures within their IT environment when used with Power Center. For example, an organization may be using data modeling tools, such as Erwin, Embarcadero, Oracle designer, Sybase Power Designer etc for developing data models. Functional and technical team should have spent much time and effort in creating the data model's data structures(tables, columns, data types, procedures, functions, triggers etc). By using meta deta exchange, these data structures can be imported into power center to identifiy source and target mappings which leverages time and effort. There is no need for informatica developer to create these data structures once again.

Power Analyzer: Power Analyzer provides organizations with reporting facilities. PowerAnalyzer makes accessing, analyzing, and sharing enterprise data simple and easily available to decision makers. PowerAnalyzer enables to gain insight into business processes and develop business intelligence. With PowerAnalyzer, an organization can extract, filter, format, and analyze corporate information from data stored in a data warehouse, data mart, operational data store, or other data storage models. PowerAnalyzer is best with a dimensional data warehouse in a relational database. It can also run reports on data in any table in a relational database that do not conform to the dimensional model.

Super Glue: Superglue is used for loading metadata in a centralized place from several sources. Reports can be run against this superglue to analyze meta data.

4. TYPES OF INFORMATICA PARTITIONS

Informatica provides you the option of enhancing the performance of the Informatica session by the The PowerCenter® Partitioning Option. After tuning all the performance bottlenecks we can further improve the performance by addition partitions[3].We can either go for Dynamic partitioning (number of partition passed as parameter) or Non-dynamic partition (number of partition are fixed while coding). Apart from used for optimizing the session, Informatica partition become useful in situations where we need to load huge volume of data or when we are using Informatica source which already has partitions defined, and using those partitions will allow to improve the session performance.

The partition attributes include setting the partition point, the number of partitions, and the partition types.

Partition Point: There can be one or more pipelines inside a mapping. Adding a partition point will divide this pipeline into many pipeline stages. Informatica will create one partition by default for every pipeline stage. As we increase the partition points it increases the number of threads. Informatica has mainly three types of threads –Reader, Writer and Transformation Thread. The number of partitions can be set at any partition point. We can define up to 64 partitions at any partition point in a pipeline. When you increase the number of partitions, you increase the number of processing threads, which can improve session performance. However, if you create a large number of partitions or partition points in a session that processes large amounts of data, you can overload the system.

You cannot create partition points for the following transformations:

- Source definition
- Sequence Generator
- XML Parser
- XML target
- Unconnected transformations

The partition type controls how the Integration Service distributes data among partitions at partition points. The Integration Service creates a default partition type at each partition point.

Types of partitions are:

1. Database partitioning
2. Hash auto-keys
3. Hash user keys
4. Key range
5. Pass-through
6. Round-robin

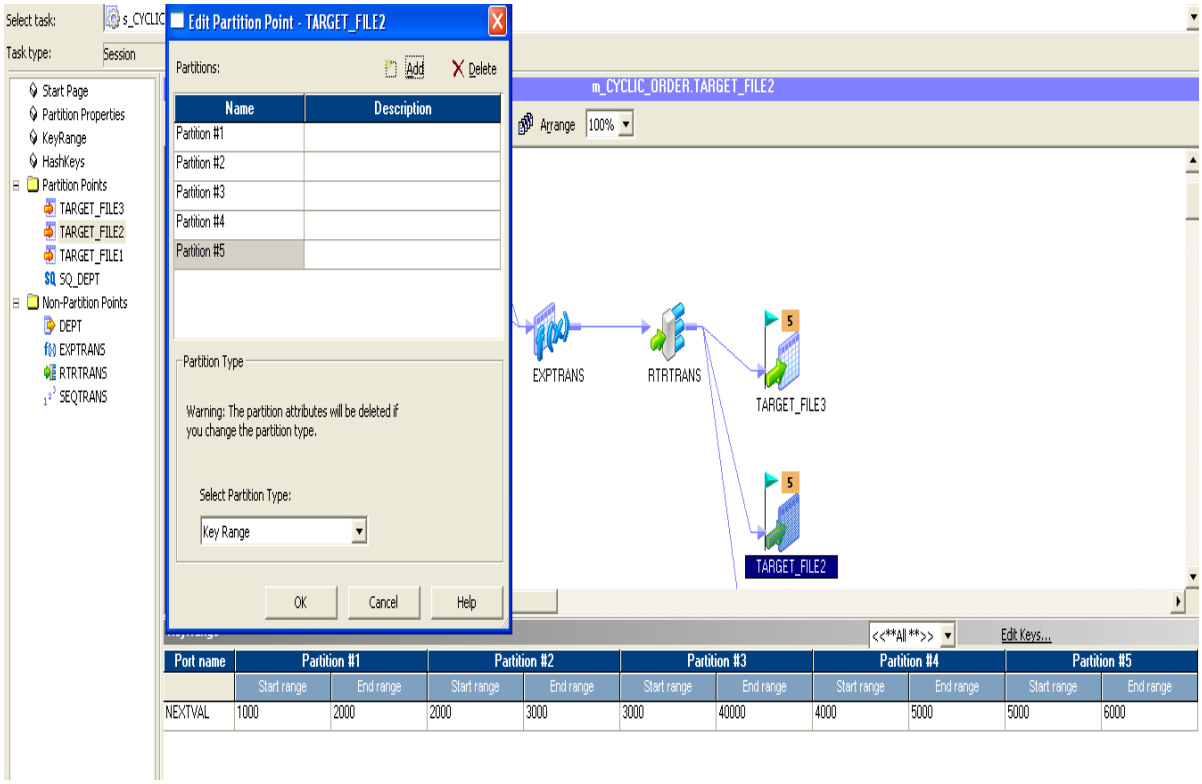
Database Partitioning: For Source Database Partitioning, Informatica will check the database system for the partition information if any and fetches data from corresponding node in the database into the session partitions. When you use Target database partitioning, the Integration Service loads data into corresponding database partition nodes.

ISSN: 2278-6244
Impact Factor: 6.111

Use database partitioning for Oracle and IBM DB2 sources and IBM DB2 targets.

Pass through: Using Pass through partition will not affect the distribution of data across partitions instead it will run in single pipeline which is by default for all your sessions. The Integration Service processes data without redistributing rows among partitions. Hence all rows in a single partition stay in the partition after crossing a pass-through partition point.

Key range: Used when we want to partition the data based on upper and lower limit. The Integration Service will distribute the rows of data based on a port or set of ports that we define as the partition key. For each port, we define a range of values. Based on the range that we define the rows are send to different partitions.



Round robin partition is used to when we want to distributes rows of data evenly to all partitions Hash auto-keys: The Integration Service uses a hash function to group rows of data among partitions. The Integration Service groups the data based on a partition key.

Hash user keys: The Integration Service uses a hash function to group rows of data among partitions. We define the number of ports to generate the partition key.

5. TRANSFORMATIONS

Informatica Transformations: A transformation is a repository object that generates, modifies, or passes data. The Designer provides a set of transformations that perform

specific functions. A transformation is a repository object that generates, modifies, or passes dataThe Designer provides a set of transformations that perform specific functions.Data passes into and out of transformations through ports that you connect in a mapping.

Transformations can be of two types:

Active Transformation: An active transformation can change the number of rows that pass through the transformation, change the transaction boundary, can change the row type. For example, Filter, Transaction Control and Update Strategy are active transformations. The key point is to note that Designer does not allow you to connect multiple active transformations or an active and a passive transformation to the same downstream transformation or transformation input group because the Integration Service may not be able to concatenate the rows passed by active transformations However, Sequence Generator transformation (SGT) is an exception to this rule[4]. A SGT does not receive data. It generates unique numeric values. As a result, the Integration Service does not encounter problems concatenating rows passed by a SGT and an active transformation.

Aggregator	performs aggregate calculations
Filter	serves as a conditional filter
Router	serves as a conditional filter (more than one filters)
Joiner	allows for heterogeneous joins
Source qualifier	represents all data queried from the source

Passive Transformation: A passive transformation does not change the number of rows that pass through it, maintains the transaction boundary, and maintains the row type. The key point is to note that Designer allows you to connect multiple transformations to the same downstream transformation or transformation input group only if all transformations in the upstream branches are passive.

Expression	performs simple calculations
Lookup	looks up values and passes to other objects
Sequence generator	generates unique ID values
Stored procedure	calls a stored procedure and captures return values
Update strategy	allows for logic to insert, update, delete, or reject data

6.1 TYPES OF TRANSFORMATION

1. Expression Transformation:

You can use the Expression transformation 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.

Calculating Values: To use the Expression transformation to calculate values for a single row, you must include the following ports**:**

- **Input or input/output ports for each value used in the calculation.** For example, when calculating the total price for an order, determined by multiplying the unit price by the quantity ordered, the input or input/output ports. One port provides the unit price and the other provides the quantity ordered.
- **Output port for the expression.** You enter the expression as a configuration option for the output port. The return value for the output port needs to match the return value of the expression. For information on entering expressions, see "Transformations" in the *Designer Guide*. Expressions use the transformation language, which includes SQL-like functions, to perform calculations

You can enter multiple expressions in a single Expression transformation. As long as you enter only one expression for each output port, you can create any number of output ports in the transformation. In this way, you can use one Expression transformation rather than creating separate transformations for each calculation that requires the same set of data.

2. Joiner Transformation:

You can use the Joiner transformation to join source data from two related heterogeneous sources residing in different locations or file systems. Or, you can join data from the same source. The Joiner transformation joins two sources with at least one matching port. The Joiner transformation uses a condition that matches one or more pairs of ports between the two sources. If you need to join more than two sources, you can add more Joiner transformations to the mapping.The Joiner transformation requires input from two separate pipelines or two branches from one pipeline.

The Joiner transformation accepts input from most transformations. However, there are some limitations on the pipelines you connect to the Joiner transformation. You cannot use a Joiner transformation in the following situations:

- Either input pipeline contains an Update Strategy transformation.
- You connect a Sequence Generator transformation directly before the Joiner transformation

The join condition contains ports from both input sources that must match for the PowerCenter Server to join two rows. Depending on the type of join selected, the Joiner transformation either adds the row to the result set or discards the row. The Joiner produces result sets based on the join type, condition, and input data sources. Before you define a join condition, verify that the master and detail sources are set for optimal performance. During a session, the PowerCenter Server compares each row of the master source against the detail source. The fewer unique rows in the master, the fewer iterations of the join comparison occur, which speeds the join process. To improve performance, designate the source with the smallest count of distinct values as the master. You can improve session performance by configuring the Joiner transformation to use sorted input. When you configure the Joiner transformation to use sorted data, the PowerCenter Server improves performance by minimizing disk input and output. You see the greatest performance improvement when you work with large data sets. When you use a Joiner transformation in a mapping, you must configure the mapping according to the number of pipelines and sources you intend to use. You can configure a mapping to join the following types of data:

- **Data from multiple sources**. When you want to join more than two pipelines, you must configure the mapping using multiple Joiner transformations.
- **Data from the same source**. When you want to join data from the same source, you must configure the mapping to use the same source

Perform joins in a database when possible.

Performing a join in a database is faster than performing a join in the session. In some cases, this is not possible, such as joining tables from two different databases or flat file systems. If you want to perform a join in a database, you can use the following options:

- Create a pre-session stored procedure to join the tables in a database.

- Use the Source Qualifier transformation to perform the join.

Join sorted data when possible.

You can improve session performance by configuring the Joiner transformation to use sorted input. When you configure the Joiner transformation to use sorted data, the PowerCenter Server improves performance by minimizing disk input and output. You see the greatest performance improvement when you work with large data sets.

For an unsorted Joiner transformation, designate as the master source the source with fewer rows. For optimal performance and disk storage, designate the master source as the source with the fewer rows. During a session, the Joiner transformation compares each row of the master source against the detail source.

3. Rank Transformation:

The Rank transformation allows you to select only the top or bottom rank of data. You can use a Rank transformation to return the largest or smallest numeric value in a port or group. You can also use a Rank transformation to return the strings at the top or the bottom of a session sort order. During the session, the PowerCenter Server caches input data until it can perform the rank calculations. You connect all ports representing the same row set to the transformation. Only the rows that fall within that rank, based on some measure you set when you configure the transformation, pass through the Rank transformation.

You can also write expressions to transform data or perform calculations. As an active transformation, the Rank transformation might change the number of rows passed through it. You might pass 100 rows to the Rank transformation, but select to rank only the top 10 rows, which pass from the Rank transformation to another transformation**.**

Rank Caches

During a session, the PowerCenter Server compares an input row with rows in the data cache. If the input row out-ranks a cached row, the PowerCenter Server replaces the cached row with the input row. If you configure the Rank transformation to rank across multiple groups, the PowerCenter Server ranks incrementally for each group it finds.

Rank Transformation Properties:

- Enter a cache directory.
- Select the top or bottom rank.

- Select the input/output port that contains values used to determine the rank. You can select only one port to define a rank.
- Select the number of rows falling within a rank.
- Define groups for ranks, such as the 10 least expensive products for each manufacturer.

The Rank transformation changes the number of rows in two different ways. By filtering all but the rows falling within a top or bottom rank, you reduce the number of rows that pass through the transformation. By defining groups, you create one set of ranked rows for each group

4. Router Transformation:

A Router transformation is similar to a Filter transformation because both transformations allow you to use a condition to test data. A Filter transformation tests data for one condition and drops the rows of data that do not meet the condition. However, a Router transformation tests data for one or more conditions and gives you the option to route rows of data that do not meet any of the conditions to a default output group**.** If you need to test the same input data based on multiple conditions, use a Router transformation in a mapping instead of creating multiple Filter transformations to perform the same task. The Router transformation is more efficient. For example, to test data based on three conditions, you only need one Router transformation instead of three filter transformations to perform this task. Likewise, when you use a Router transformation in a mapping, the PowerCenter Server processes the incoming data only once

5. Lookup Transformation:

Use a Lookup transformation in a mapping to look up data in a flat file or a relational table, view, or synonym. You can import a lookup definition from any flat file or relational database to which both the PowerCenter Client and Server can connect[6]. You can use multiple Lookup transformations in a mapping. It compares Lookup transformation port values to lookup source column values based on the lookup condition. Pass the result of the lookup to other transformations and a target.

You can use the Lookup transformation to perform many tasks, including:

- **Get a related value.** For example, your source includes employee ID, but you want to include the employee name in your target table to make your 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.** You can use a Lookup transformation to determine whether rows already exist in the target.

  You can configure the Lookup transformation to perform the following types of lookups:
- **Connected or unconnected.** Connected and unconnected transformations receive input and send output in different ways.
- **Relational or flat file lookup.** When you create a Lookup transformation, you can choose to perform a lookup on a flat file or a relational table.
- **Cached or uncached.** Sometimes you can improve session performance by caching the lookup table. If you cache the lookup, you can choose to use a dynamic or static cache. By default, the lookup cache remains static and does not change during the session. With a dynamic cache, the PowerCenter Server inserts or updates rows in the cache during the session. When you cache the target table as the lookup, you can look up values in the target and insert them if they do not exist, or update them if they do.

Note: If you use a flat file lookup, you must use a static cache.

Using Sorted Input

When you configure a flat file Lookup transformation for sorted input, the condition columns must be grouped. If the condition columns are not grouped, the PowerCenter Server cannot cache the lookup and fails the session. For best caching performance, sort the condition columns. The Lookup transformation also enables an associated ports property that you configure when you use a dynamic cache.

6. ADVANTAGES OF INFORMATICA

A comprehensive integration platform that promotes code standardization, unifies collaboration between business and IT roles, and provides capabilities that handle the high

ISSN: 2278-6244
Impact Factor: 6.111

volume and wide variety of today's business data. The Informatica Platform has eight distinct technologies designed to be a true industrial strength ETL solution. These include:

- Messaging
- Complex Event Processing
- B2B Data Exchange
- Cloud Data Integration
- Enterprise Data Integration
- Application Lifecycle Management
- Data Quality
- Master Data Management

- **Improve network speed.** Slow network connections can slow session performance. Have your system administrator determine if your network runs at an optimal speed. Decrease the number of network hops between the PowerCenter Server and databases.
- **Use multiple PowerCenter Servers.** Using multiple PowerCenter Servers on separate systems might double or triple session performance.
- **Use a server grid.** Use a collection of PowerCenter Servers to distribute and process the workload of a workflow. For information on server grids.
- **Improve CPU performance.** Run the PowerCenter Server and related machines on high performance CPUs, or configure your system to use additional CPUs.

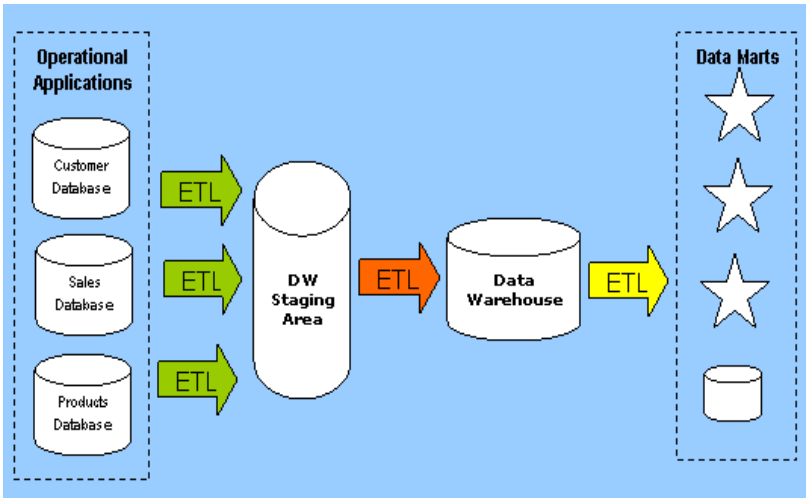
- **Configure the PowerCenter Server for ASCII data movement mode.** When all character data processed by the PowerCenter Server is 7-bit ASCII or EBCDIC, configure the PowerCenter Server for ASCII data movement mode[5].
- **Check hard disks on related machines.** Slow disk access on source and target databases, source and target file systems, as well as the PowerCenter Server and repository machines can slow session performance. Have your system administrator evaluate the hard disks on your machines.
- **Reduce paging.** When an operating system runs out of physical memory, it starts paging to disk to free physical memory. Configure the physical memory for the PowerCenter Server machine to minimize paging to disk.
- **Use processor binding.** In a multi-processor UNIX environment, the PowerCenter Server may use a large amount of system resources. Use processor binding to control processor usage by the PowerCenter Server.

7. DATA STAGING

The data staging area is the data warehouse workbench. It is the place where raw data is brought in, cleaned, combined, archived, and eventually exported to one or more data marts. The purpose of the data staging area is to get data ready for loading into a presentation server (a relational DBMS or an OLAP engine). A staging area, or landing zone, is an intermediate storage area used for data processing during the extract, transform and load (ETL) process. The data staging area sits between the data source(s) and the data target(s), which are often data warehouses, data marts, or other data repositories [1]

Data staging areas are often transient in nature, with their contents being erased prior to running an ETL process or immediately following successful completion of an ETL process. There are staging area architectures, however, which are designed to hold data for extended periods of time for archival or troubleshooting purposes. Staging areas can be implemented in the form of tables in relational databases, text-based flat files (or XML files) stored in file systems or proprietary formatted binary files stored in file systems.[2] Staging area architectures range in complexity from a set of simple relational tables in a target database to self-contained database instances or file systems.[3] Though the source systems and target systems supported by ETL processes are often relational databases, the staging areas that sit between data sources and targets need not also be relational databases.[4]

ISSN: 2278-6244
Impact Factor: 6.111



The Data Warehouse Staging Area is temporary location where data from source systems is copied. A staging area is mainly required in a Data Warehousing Architecture for timing reasons. In short, all required data must be available before data can be integrated into the Data Warehouse.[1]

The staging area in Business Intelligence is a key concept. The role of this area is to have a secure place to store the source systems data for further transformations and cleanings. Why do we do that?

Because:

- It minimizes the impact on the source systems (you don't want to re-extract everything from the source systems if your ETL failed).
- It can be used for auditing purposes (we store the data that we process).
- It eases the development process (you don't need to be bound to the operational servers).

The data staging area of the data warehouse is both a storage area and a set of processes commonly referred to as extract-transformation-load (ETL).[2] The data staging area is everything between the operational source systems and the data presentation area. It is somewhat analogous to the kitchen of a restaurant, where raw food products are transformed into a fine meal. In the data warehouse, raw operational data is transformed into a warehouse deliverable fit for user query and consumption. Similar to the restaurant's kitchen, the backroom data staging area is accessible only to skilled professionals. The data warehouse kitchen staff is busy preparing meals and simultaneously cannot be responding to customer inquiries. Customers aren't invited to eat in the kitchen. [3]

It certainly isn't safe for customers to wander into the kitchen. We wouldn't want our data warehouse customers to be injured by the dangerous equipment, hot surfaces, and sharp knifes they may encounter in the kitchen, so we prohibit them from accessing the staging area. Besides, things happen in the kitchen that customers just shouldn't be privy to. The key architectural requirement for the data staging area is that it is off-limits to business users and does not provide query and presentation services.

Once the data is extracted to the staging area, there are numerous potential transformations, such as cleansing the data (correcting misspellings, resolving domain conflicts, dealing with missing elements, or parsing into standard formats), combining data from multiple sources, deduplicating data, and assigning warehouse keys. These transformations are all precursors to loading the data into the data warehouse presentation area.[1] The data staging area is dominated by the simple activities of sorting and sequential processing. In many cases, the data staging area is not based on relational technology but instead may consist of a system of flat files. After you validate your data for conformance with the defined one-to-one and many-to-one business rules, it may be pointless to take the final step of building a fullblown third-normal-form physical database. However, there are cases where the data arrives at the doorstep of the data staging area in a third-normal-form relational format. In these situations, the managers of the data staging area simply may be more comfortable performingthe cleansing and transformation tasks using a set of normalized structures.

8. DATA STAGING PROCESS

A staging area, or landing zone, is an intermediate storage area used for data processing during the extract, transform and load (ETL) process. The data staging area sits between the data source(s) and the data target(s), which are often data warehouses, data marts, or other data repositories.[1]The data staging process imports data either as streams or files, transforms it, produces integrated, cleaned data and stages it for loading into data warehouses, data marts, or Operational Data Stores.[2]

First, Kimball distinguishes two data staging scenarios.

In (1) a data staging tool is available, and the data is already in a database. The data flow is set up so that it comes out of the source system, moves through the transformation engine, and into a staging database. The flow is illustrated in Figure One.

ISSN: 2278-6244
Impact Factor: 6.111

Figure One -- First Data Staging Scenario

In the second scenario, begin with a mainframe legacy system. Then extract the sought after data into a flat file, move the file to a staging server, transform its contents, and load transformed data into the staging database.

Figure Two illustrates this scenario.

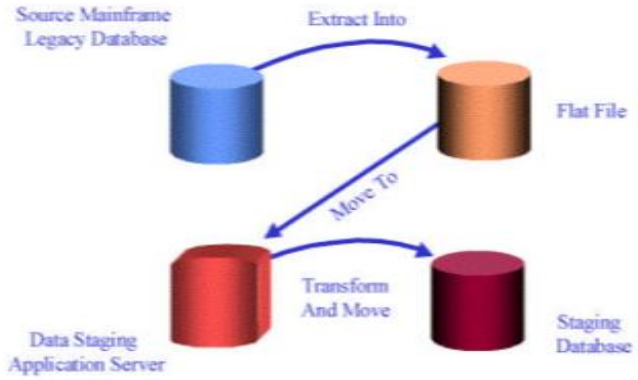


Figure Two -- Second Data Staging Scenario

We assume that the data staging area is not a query service. In other words, any database that is used for querying is assumed to be physically downstream from the data staging area. If the legacy data is already available in a relational database, then it may make sense to perform all the processing steps within the relational framework, especially if the source relational database and the eventual target presentation database are from the same vendor.[3] This makes even more sense when the source database and the target database are on the same physical machine, or when there is a convenient high-speed link between them. However, there are many variations on this theme, and in many cases it may not make sense to load the source data into a relational database. In the detailed descriptions of the processing steps, we will see that almost all the processing consists of sorting, followed by a single, sequential pass through either one or two tables.[1] This simple processing paradigm does not need the power of a relational DBMS. In fact, in some cases, it may be a

serious mistake to divert resources into loading the data into a relational database when what is needed is sequential flat-file processing. Similarly, we will see that if the raw data is not in a normalized entity-relationship (ER) format, in many cases it does not pay to load it into an ER physical model simply to check data relationships. The most important data integrity steps involving the enforcement of one-to-one and one-to-many relationships can be performed, once again, with simple sorting and sequential processing. It is acceptable to create a normalized database to support the staging processes; however, this is not the end goal. The normalized structures must be off-limits to user queries because they defeat understandability and performance. As soon as a database supports query and presentation services, it must be considered part of the data warehouse presentation area. By default, normalized databases are excluded from the presentation area, which should be strictly dimensionally structured. Regardless of whether we're working with a series of flat files or a normalized data structure in the staging area, the final step of the ETL process is the loading of data. Loading in the data warehouse environment usually takes the form of presenting the quality-assured dimensional tables to the bulk loading facilities of each data mart.[2] The target data mart must then index the newly arrived data for query performance. When each data mart has been freshly loaded, indexed, supplied with appropriate aggregates, and further quality assured, the user community is notified that the new data has been published.

The data staging area of the data warehouse is both a storage area and a set of processes commonly referred to as extract-transformation-load (ETL). The data staging area is everything between the operational source systems and the data presentation area. The process of extracting data from source systems and bringing it into the data warehouse is commonly called ETL, which stands for extraction, transformation, and loading.[3] It is a fundamental phenomenon in a data warehouse . Whenever DML (data manipulation language) operations such as INSERT, UPDATE OR DELETE are issued on the source database, data extraction occurs. After data extraction and transformation have taken place, data are loaded into the data warehouse.

Extraction: The first part of an ETL process is to extract the data from the source systems. Most data warehousing projects consolidate data from different source systems. Each separate system may also use a different data organization format. Common data source

formats are relational databases and flat files, but may include non-relational database structures such as IMS or other data structures[4] .Extraction converts the data into a format for transformation processing. The amount of data is reduced by omitting any non-relevant data sets. Extraction must not negatively affect the performance of productive systems. Extracting means reading and understanding the source data and copying the data needed for the data warehouse into the staging area for further manipulation

Transformation: Any transformation needed to provide data that can be interpreted in business terms is done in the second step. Data sets are cleaned with regard to their data quality. Eventually, they are converted to the scheme of the target database and consolidated. The transform stage applies a series of rules or functions to the extracted data to derive the data to be loaded. Some data sources will require very little manipulation of data. Data transformations are often the most complex and, in terms of processing time, the most costly part of the ETL process. They can range from simple data conversions to extremely complex data scrubbing techniques.[2]

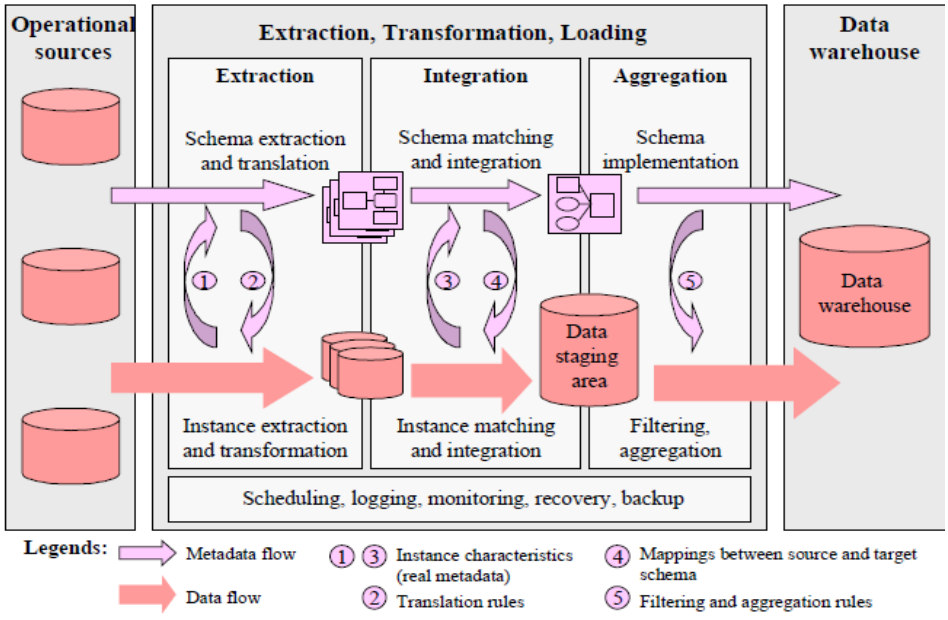


Figure 1. Steps of building a data warehouse: the ETL process

Loading: Finally, the actual loading of data into the data warehouse has to be done. The Initial Load which generally is not time-critical is distinguished from the Incremental Load. Whereas the first phase affected productive systems, loading can have an immense effect on the data warehouse. This especially has to be taken into consideration with regard to the complex task of updating currently stored data sets. In general, incremental loading is a critical task. ETL processes can either be run in batch mode or real time. Batch jobs typically

are run periodically. If intervals become as short as hours or even minutes only, these processes are called near real time. The load phase loads the data into the data warehouse. Depending on the requirements of the organization, this process ranges widely. Some data warehouses merely overwrite old information with new data.[3]

Unfortunately, there is still considerable industry consternation about whether the data that supports or results from this process should be instantiated in physical normalized structures prior to loading into the presentation area for querying and reporting. These normalized structures sometimes are referred to in the industry as the enterprise data warehouse; however, we believe that this terminology is a misnomer because the warehouse is actually much more encompassing than this set of normalized tables. The enterprise's data warehouse more accurately refers to the conglomeration of an organization's data warehouse staging and presentation areas. [4]

A normalized database for data staging storage is acceptable. However, we continue to have some reservations about this approach. The creation of both normalized structures for staging and dimensional structures for presentation means that the data is extracted, transformed, and loaded twice—once into the normalized database and then again when we load the dimensional model. Obviously, this two-step process requires more time and resources for the development effort, more time for the periodic loading or updating of data, and more capacity to store the multiple copies of the data.[1] At the bottom line, this typically translates into the need for larger development, ongoing support, and hardware platform budgets. Unfortunately, some data warehouse project teams have failed miserably because they focused all their energy and resources on constructing the normalized structures rather than allocating time to development of a presentation area that supports improved business decision making. While we believe that enterprise-wide data consistency is a fundamental goal of the data warehouse environment, there are equally effective and less costly approaches than physically creating a normalized set of tables in your staging area, if these structures don't already exist.

9. CONCLUSION

The Informatica solution for enterprise data warehousing is proven to help IT departments implement data marts and departmental data warehouses and readily scale them up to enterprise data warehousing environments. This solution serves as the foundation for all

data warehousing and enterprise data warehousing projects. It accelerates their deployment, minimizing costs and risks, by ensuring that enterprise data warehouses are populated and maintained with trustworthy, actionable, and authoritative data.

Data is and has been from the beginning created, stored, and retrieved by disparate, incompatible systems. Between 30% and 35% of all the data in the industry is still on mainframes, in languages and data structures that are archaic and generally unavailable.

The wave of specialty applications—HR, sales, accounting, ERP, manufacturing—have all contributed their share to the chaos. Informatica PowerCenter is the ETL tool that empowers an IT organization to implement highly scalable, high-performance data processing maps using a graphical interface that generates proprietary intermediate code[9]. This mapping code, when coupled with a defined data workflow plan can then be scheduled for a variety of execution strategies. Widely accepted as an industry front runner, Informatica boasts high productivity and low cost. In truth, this may be just the opposite. Perhaps high productivity at a cost is more accurate; in terms of experienced developers, administrators, and license fees. Companies who choose to use Informatica usually have very large IT teams and budgets.

REFERENCES

[1] http://dwhlaureate.blogspot.in/2012/07/informaticaetl-extract-transform-and.html
[2] http://www.dbbest.com/blog/extract-transform-load-etl-technologies-part-2/
[3] http://www.informatica.com/us/#fbid=rHNS5hx2rKv
[4] http://www.ventanaresearch.com/blog/commentblog.aspx?id=3524
[5] http://www.slideshare.net
[6] http://www.rpi.edu/datawarehouse/docs/ETL-Tool-Selection.pdf
[7] http://en.wikipedia.org/wiki/Informatica
[8] http://www.etltool.com/etl-vendors/powercenter-informatica/
[9] http://www.techopedia.com/definition/25983/informatica-powercenter
[10] www.ijsrp.org/research-paper-0214/ijsrp-p2688.pdf