**[What is ETL?](http://bi-raj.blogspot.in/2010/08/what-is-etl_4081.html)**

ETL stands for Extract, Transform, and Load. ETL is a process that enables businesses to consolidate their disparate data while moving it from place to place, and it doesn't really matter that that data is in different forms or formats. The data can come from any source. ETL process is powerful enough to handle such data disparities.

**Extract**

ETL process involves extracting 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. Goal of the extraction phase is to convert the data into a single format which is appropriate for transformation processing. More often as a best practice, the Data is extracted from the source in to PREP tables, where the data is in exactly the same format as the source system, but this need not be true always. Extracting data in to the PREP tables always ensures that all the data from the source has been extracted, and is ready for applying transformations as per business requirements and load the data to the target system.

 **Transform**

Post data extraction, the next step in ETL is to apply transformations (or in simple terms the business rules), that are applicable for the extracted data from the source to derive the data for loading into the end target. Some data sources will require very little or even no manipulation of data. In other cases, one or more of the following transformation types may be required to meet the business and technical needs of the target database:

* Restricting unwanted data from propagating in to the target system (for ex, discarding null values, selecting only certain columns to load, etc).
* Automated Cleansing and decoding the data (For ex:- if the source system stores 1 for male and 2 for female, but the warehouse stores M for male and F for female)
* Deriving new calculated fields (ex:- sale\_amount = qty \* unit\_price)
* Applying filtering and sorting
* Merging data from various sources (for ex:- Lookup, Merge, etc)
* Aggregation (for example, rollup - summarizing multiple rows of data - total sales for each store, and for each region, etc.)
* Generating surrogate-key values
* Splitting a column into multiple columns (e.g., putting a comma-separated list specified as a string in one column as individual values in different columns)
* Lookup and Constraints validation to check the relevant data from tables or referential files for slowly changing dimensions.
* Data validation. If validation fails, it may result in a full, partial or no rejection of the data, and thus none, some or all the data are handed over to the next step, depending on the rule design and exception handling.

**Load**

Once the data is extracted, transformed, the data is loaded in to the target system, usually the data warehouse (DW). Depending on the requirements of the organization, this process varies widely. Some data warehouses may overwrite existing information with cumulative information, updates from the extract data is done periodically (daily, weekly, monthly etc). Other DW may add new data in a historicized form.

[**Extract / Transform Process**](http://bi-raj.blogspot.in/2010/08/extract-transform-process_02.html)

**Extract**  
Generally this is a two stage processes sometimes one stage depending on the source system. If the source system is a non standard source, such as flat file, excel files, EAV model databases, ERP systems, CRM systems, etc, then the Extract phase becomes a two stage process, i.e.,

* extract data from source to temporary storage area (sometimes called PREP area)
* load the extracted data in to the Staging area

If the source system is any operational database, then the extract the data from different operational databases for e.g. : individual stores level data of a retail chain, etc. and load into Stage area as is and proceed to the next steps to apply transformation/business rules and load from Stage area to Data warehouse/mart.  
  
**Transformation**:   
Here the following are addressed:

* The business logics to the necessary fields
* Populate the surrogate keys and check referential integrity constraints by performing look up for foreign keys.
* Aggregation of data
* Data conversion if necessary
* Implement SCD's

### [What is a staging area? Do we need it? What is the purpose of a staging area?](http://bi-raj.blogspot.in/2010/08/what-is-staging-area-do-we-need-it-what.html)

A staging area is storage area between the operational data source systems and the data presentation system. Staging area is a common place to keep extracted data from different source systems before applying business rules (transformation, cleansing). Staging area is place where you hold temporary tables on data warehouse server. Staging tables are connected to work area or fact tables. Staging area plays an important role in performing data validation, transformations, cleansing and merging before loading the data into warehouse. In the absence of the Staging area, there is a load on the source system or the data presentation system to have the data conformed to the dimension model. This is the primary reason for the existence of a staging area in any ETL system. In addition it also offers a platform for carrying out data cleansing.

### [Why do we need an ETL tool?](http://bi-raj.blogspot.in/2010/08/why-do-we-need-etl-tool.html)

ETL Tools enable extract transform and load data from operational source systems into Data Warehouse for decision making. ETL tools automate many complex T-SQL scripts. Apart from automating T-SQL scripts, ETL tools also offer parallel processing, auditing, error handling, and logging systems which can be configured easily. Also the deployment and maintenance of ETL is easier compared to maintenance of T-SQL scripts. ETL Tools are very powerful and they offer many advantages in all stages of ETL process starting from extraction data cleansing data profiling transformation debugging and loading into data warehouse when compared to the old method.

### [What are different ETL tools and Reporting Tools available in the market?](http://bi-raj.blogspot.in/2010/08/what-are-different-etl-tools-and.html)

**ETL Tools:** Informatica, Datastage, SAS, Microsoft SSIS, Oracle Warehouse Builder, Abniaio

**Repoting Tools:** Cognos, BOXI/R2, Microstrategy, Hyperion

[**Lookup Operations**](http://bi-raj.blogspot.in/2010/10/lookup-operations.html)

A Lookup operation ensures that all the references of a table are made correctly, for example, while inserting data in to a fact table, it is required to provide appropriate dimension keys so that the fact data becomes relevant, in order to load the dimension keys, a look up operation on the dimension table will be done to fetch the dimension keys for a fact data.  
  
SSIS supports, regular Lookup as well as Fuzzy Lookup operations via, Lookup Transformation and Fuzzy Lookup Transformation under the Data Flow Task.  
  
Lookup transformation performs lookups by joining data in input columns with columns in a reference dataset. You use the lookup to access additional information in a related table that is based on values in common columns. A lookup table is nothing but a 'lookup' it give values to referenced table (it is a reference) it is used at the run time it saves joins and space in terms of transformations.  
  
The Lookup transformation tries to perform an equi join between values in the transformation input and values in the reference dataset. (An equi-join means that each row in the transformation input must match at least one row from the reference dataset.) If an equi join is not possible, the Lookup transformation takes one of the following actions:

1. If there is no matching entry in the reference dataset, no join occurs. By default, the Lookup transformation treats rows without matching entries as errors. However, you can configure the Lookup transformation to redirect such rows to a no match output
2. If there are multiple matches in the reference table, the Lookup transformation returns only the first match returned by the lookup query.

The Fuzzy Lookup transformation differs from the Lookup transformation in its use of fuzzy matching. The Lookup transformation uses an equi-join to locate matching records in the reference table. It returns either an exact match or nothing from the reference table. In contrast, the Fuzzy Lookup transformation uses fuzzy matching to return one or more close matches from the reference table. A Fuzzy Lookup transformation frequently follows a Lookup transformation in a package data flow. First, the Lookup transformation tries to find an exact match. If it fails, the Fuzzy Lookup transformation provides close matches from the reference table.  
  
The transformation needs access to a reference data source that contains the values that are used to clean and extend the input data. The reference data source must be a table in a SQL Server database. The match between the value in an input column and the value in the reference table can be an exact match or a fuzzy match. However, the transformation requires at least one column match to be configured for fuzzy matching. If you want to use only exact matching, use the Lookup transformation instead.