**Extract, transform, load (ETL)**

In computing, extract, transform and load (ETL) refers to a process in database usage and especially in data warehousing that involves:

• Extracting data from outside sources

• Transforming it to fit operational needs (which can include quality levels)

• Loading it into the end target (database, more specifically, operational data store, data mart or data warehouse)

**Extract**

Extract The first part of an ETL process involves extracting the data from the source systems. In many cases this is the most challenging aspect of ETL, as extracting data correctly will set the stage for how subsequent processes will go. ETL Architecture Pattern 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 Information Management System (IMS) or other data structures such as Virtual Storage Access Method (VSAM) or Indexed Sequential Access Method (ISAM), or even fetching from outside sources such as through web spidering or screen-scraping. The streaming of the extracted data source and load on-the-fly to the destination database is another way of performing ETL when no intermediate data storage is required. In general, the goal of the extraction phase is to convert the data into a single format which is appropriate for transformation processing.

An intrinsic part of the extraction involves the parsing of extracted data, resulting in a check if the data meets an expected pattern or structure. If not, the data may be rejected entirely or in part.

**Transform**

The transform stage applies to a series of rules or functions to 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:

• Selecting only certain columns to load (or selecting null columns not to load). For example, if the source data has three columns (also called attributes), for example roll\_no, age, and salary, then the extraction may take only roll\_no and salary. Similarly, the extraction mechanism may ignore all those records where salary is not present (salary = null).

• Translating coded values (*e.g.*, if the source system stores 1 for male and 2 for female, but the warehouse stores M for male and F for female)

• Encoding free-form values (*e.g.*, mapping "Male" to "1")

• Deriving a new calculated value (*e.g.*, sale\_amount = qty \* unit\_price)

• Sorting

• Joining data from multiple sources (*e.g.*, lookup, merge) and deduplicating the data

• Aggregation (for example, rollup — summarizing multiple rows of data — total sales for each store, and for each region, etc.)

• Generating surrogate-key values

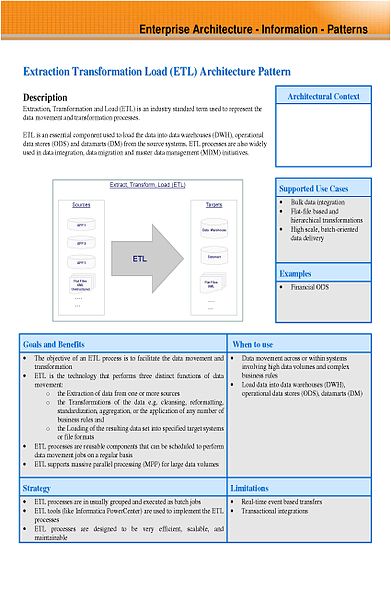
• Transposing or pivoting (turning multiple columns into multiple rows or vice versa)

• 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)

• Disaggregation of repeating columns into a separate detail table (*e.g.*, moving a series of addresses in one record into single addresses in a set of records in a linked *address* table)

• Lookup and validate the relevant data from tables or referential files for slowly changing dimensions.

• Applying any form of simple or complex 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 is handed over to the next step, depending on the rule design and exception handling. Many of the above transformations may result in exceptions, for example, when a code translation parses an unknown code in the extracted data.



**Load**

The load phase loads the data into the end target, 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, frequently updating extract data is done on daily, weekly or monthly basis. Other DW (or even other parts of the same DW) may add new data in a historicized form, for example, hourly. To understand this, consider a DW that is required to maintain sales records of the last year. Then, the DW will overwrite any data that is older than a year with newer data. However, the entry of data for any one year window will be made in a historicized manner. The timing and scope to replace or append are strategic design choices dependent on the time available and the business needs. More complex systems can maintain a history and audit trail of all changes to the data loaded in the DW.

As the load phase interacts with a database, the constraints defined in the database schema — as well as in triggers activated upon data load — apply (for example, uniqueness, referential integrity, mandatory fields), which also contribute to the overall data quality performance of the ETL process.

• For example, a financial institution might have information on a customer in several departments and each department might have that customer's information listed in a different way. The membership department might list the customer by name, whereas the accounting department might list the customer by number. ETL can bundle all this data and consolidate it into a uniform presentation, such as for storing in a database or data warehouse.

• Another way that companies use ETL is to move information to another application permanently. For instance, the new application might use another database vendor and most likely a very different Database schema. ETL can be used to transform the data into a format suitable for the new application to use.

• An example of this would be an Expense and Cost Recovery System (ECRS) such as used by accountancies, consultancies and lawyers. The data usually ends up in the time and billing system, although some businesses may also utilize the raw data for employee productivity reports to Human Resources (personnel dept.) or equipment usage reports to Facilities Management.

