What is ETL?

ETL is a process that extracts the data from different source systems, then transforms the data (like applying calculations, concatenations, etc.) and finally loads the data into the Data Warehouse system. Full form of ETL is Extract, Transform and Load.
It's tempting to think a creating a Data warehouse is simply extracting data from multiple sources and loading into database of a Data warehouse. This is far from the truth and requires a complex ETL process. The ETL process requires active inputs from various stakeholders including developers, analysts, testers, top executives and is technically challenging.

In order to maintain its value as a tool for decision-makers, Data warehouse system needs to change with business changes. ETL is a recurring activity (daily, weekly, monthly) of a Data warehouse system and needs to be agile, automated, and well documented.

Why do you need ETL?

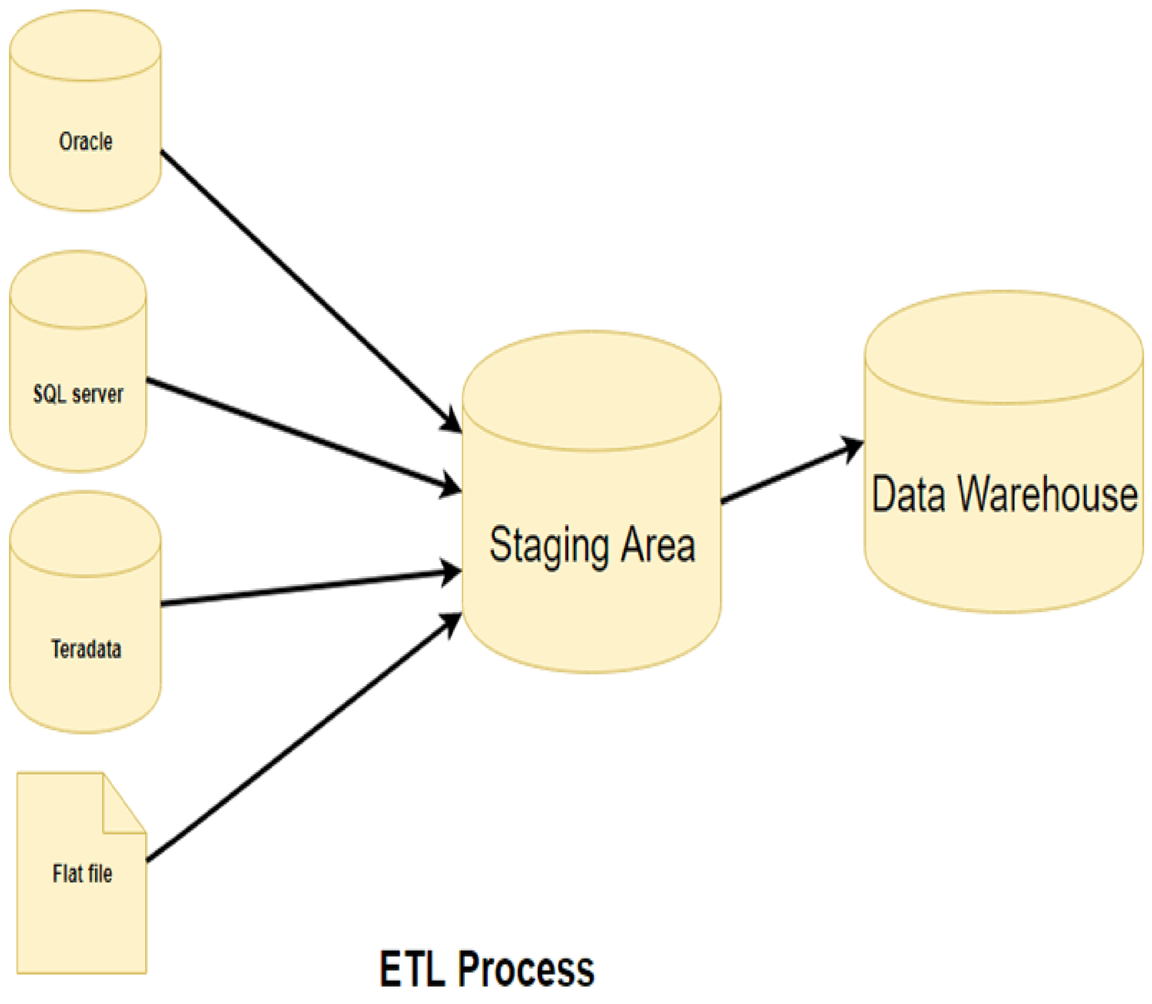
There are many reasons for adopting ETL in the organization:

- It helps companies to analyze their business data for taking critical business decisions.
- Transactional databases cannot answer complex business questions that can be answered by ETL example.
- A Data Warehouse provides a common data repository
- ETL provides a method of moving the data from various sources into a data warehouse.
- As data sources change, the Data Warehouse will automatically update.
- Well-designed and documented ETL system is almost essential to the success of a Data Warehouse project.
- Allow verification of data transformation, aggregation and calculations rules.
- ETL process allows sample data comparison between the source and the target system.
- ETL process can perform complex transformations and requires the extra area to store the data.

•ETL helps to Migrate data into a Data Warehouse. Convert to the various formats and types to adhere to one consistent system.
•ETL is a predefined process for accessing and manipulating source data into the target database.
•ETL in data warehouse offers deep historical context for the business.
•It helps to improve productivity because it codifies and reuses without a need for technical skills.

ETL Process in Data Warehouses

ETL is a 3-step process



Step 1) Extraction

In this step of ETL architecture, data is extracted from the source system into the staging area. Transformations if any are done in staging area so that performance of source system in not degraded. Also, if corrupted data is copied directly from the source into Data warehouse database, rollback will be a challenge. Staging area gives an opportunity to validate extracted data before it moves into the Data warehouse.

Data warehouse needs to integrate systems that have different

DBMS, Hardware, Operating Systems and Communication Protocols. Sources could include legacy applications like Mainframes, customized applications, Point of contact devices like ATM, Call switches, text files, spreadsheets, ERP, data from vendors, partners amongst others.

Hence one needs a logical data map before data is extracted and loaded physically. This data map describes the relationship between sources and target data.

Three Data Extraction methods:

1.Full Extraction
2.Partial Extraction- without update notification.
3.Partial Extraction- with update notification

Irrespective of the method used, extraction should not affect performance and response time of the source systems. These source systems are live production databases. Any slow down or locking could effect company's bottom line.

Some validations are done during Extraction:

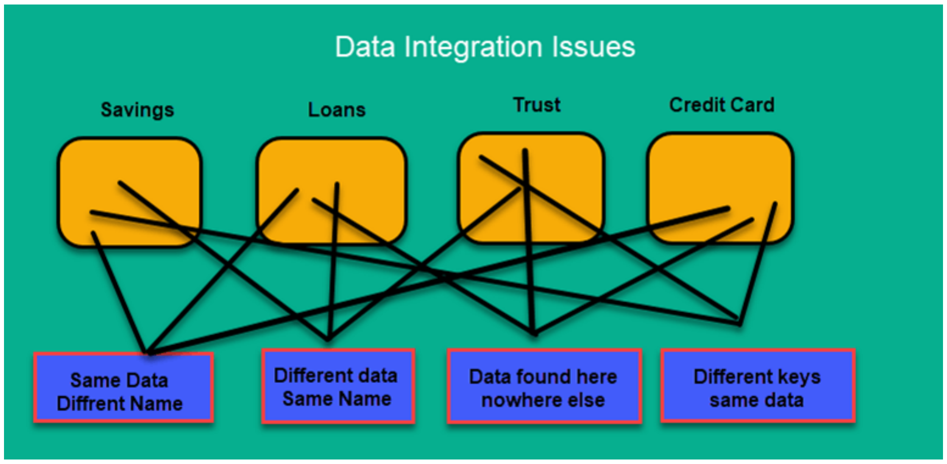
•Reconcile records with the source data
•Make sure that no spam/unwanted data loaded
•Data type check
•Remove all types of duplicate/fragmented data
•Check whether all the keys are in place or not

Step 2) Transformation

Data extracted from source server is raw and not usable in its original form. Therefore it needs to be cleansed, mapped and transformed. In fact, this is the key step where ETL process adds value and changes data such that insightful BI reports can be generated.

It is one of the important ETL concepts where you apply a set of functions on extracted data. Data that does not require any transformation is called as **direct move** or **pass through data**.

In transformation step, you can perform customized operations on data. For instance, if the user wants sum-of-sales revenue which is not in the database. Or if the first name and the last name in a table is in different columns. It is possible to concatenate them before loading.



Following are Data Integrity Problems:

1. Different spelling of the same person like Jon, John, etc.
2. There are multiple ways to denote company name like Google, Google Inc.
3. Use of different names like Cleaveland, Cleveland.
4. There may be a case that different account numbers are generated by various applications for the same customer.
5. In some data required files remains blank
6. Invalid product collected at POS as manual entry can lead to mistakes.

Validations are done during this stage

- Filtering – Select only certain columns to load
- Using rules and lookup tables for Data standardization

•Character Set Conversion and encoding handling
•Conversion of Units of Measurements like Date Time Conversion, currency conversions, numerical conversions, etc.
•Data threshold validation check. For example, age cannot be more than two digits.
•Data flow validation from the staging area to the intermediate tables.
•Required fields should not be left blank.
•Cleaning (for example, mapping NULL to 0 or Gender Male to “M” and Female to “F” etc.)
•Split a column into multiples and merging multiple columns into a single column.
•Transposing rows and columns,
•Use lookups to merge data
•Using any complex data validation (e.g., if the first two columns in a row are empty then it automatically reject the row from processing)

Step 3) Loading

Loading data into the target datawarehouse database is the last step of the ETL process. In a typical Data warehouse, huge volume of data needs to be loaded in a relatively short period (nights). Hence, load process should be optimized for performance.

In case of load failure, recover mechanisms should be configured to restart from the point of failure without data integrity loss. Data Warehouse admins need to monitor, resume, cancel loads as per prevailing server performance.

Types of Loading:

•**Initial Load** — populating all the Data Warehouse tables
•**Incremental Load** — applying ongoing changes as when needed periodically.
•**Full Refresh** —erasing the contents of one or more tables and reloading with fresh data.

Load verification

•Ensure that the key field data is neither missing nor null.
•Test modeling views based on the target tables.
•Check that combined values and calculated measures.
•Data checks in dimension table as well as history table.

•Check the BI reports on the loaded fact and dimension table.

Best practices ETL process

Following are the best practices for ETL Process steps:

Never try to cleanse all the data:

Every organization would like to have all the data clean, but most of them are not ready to pay to wait or not ready to wait. To clean it all would simply take too long, so it is better not to try to cleanse all the data.

Never cleanse Anything:

Always plan to clean something because the biggest reason for building the Data Warehouse is to offer cleaner and more reliable data.

Determine the cost of cleansing the data:

Before cleansing all the dirty data, it is important for you to determine the cleansing cost for every dirty data element.

To speed up query processing, have auxiliary views and indexes:

To reduce storage costs, store summarized data into disk tapes. Also, the trade-off between the volume of data to be stored and its detailed usage is required. Trade-off at the level of granularity of data to decrease the storage costs.

Overview of Extraction in Data Warehouses

Extraction is the operation of extracting data from a source system for further use in a data warehouse environment. This is the first step of the ETL process. After the extraction, this data can be transformed and loaded into the data warehouse. The source systems for a data warehouse are typically transaction processing applications. For example, one of the source systems for a sales analysis data warehouse might be an order entry system that records all of the current order activities.
Designing and creating the extraction process is often one of the most time-consuming tasks in the ETL process and, indeed, in the entire data warehousing process. The source systems might be very complex and poorly documented, and thus determining which data needs to be extracted can be difficult. The data has to be extracted normally not only once, but several times in a periodic manner to supply all changed data to the warehouse and keep it up-to-date. Moreover, the

source system typically cannot be modified, nor can its performance or availability be adjusted, to accommodate the needs of the data warehouse extraction process.

These are important considerations for extraction and ETL in general. This chapter, however, focuses on the technical considerations of having different kinds of sources and extraction methods. It assumes that the data warehouse team has already identified the data that will be extracted, and discusses common techniques used for extracting data from source databases.

Designing this process means making decisions about the following two main aspects:

•Which extraction method do I choose?This influences the source system, the transportation process, and the time needed for refreshing the warehouse.

•How do I provide the extracted data for further processing?This influences the transportation method, and the need for cleaning and transforming the data.

Introduction to Extraction Methods in Data Warehouses

The extraction method you should choose is highly dependent on the source system and also from the business needs in the target data warehouse environment. Very often, there’s no possibility to add additional logic to the source systems to enhance an incremental extraction of data due to the performance or the increased workload of these systems. Sometimes even the customer is not allowed to add anything to an out-of-the-box application system.

The estimated amount of the data to be extracted and the stage in the ETL process (initial load or maintenance of data) may also impact the decision of how to extract, from a logical and a physical perspective. Basically, you have to decide how to extract data logically and physically.

Logical Extraction Methods

There are two kinds of logical extraction:

•Full Extraction

•Incremental Extraction

Full Extraction

The data is extracted completely from the source system. Since this extraction reflects all the data currently available on the source system, there’s no need to keep track of changes to the data source since the last successful extraction. The source data will be provided as-is and no additional logical information (for example, timestamps) is necessary on the source site. An example for a full extraction may be an export file of a distinct table or a remote SQL statement scanning the complete source table.

Incremental Extraction

At a specific point in time, only the data that has changed since a well-defined event back in history will be extracted. This event may be the last time of extraction or a more complex business event like the last booking day of a fiscal period. To identify this delta change there must be a possibility to identify all the changed information since this specific time event. This information can be either provided by the source data itself like an application column, reflecting the last-changed timestamp or a change table where an appropriate additional mechanism keeps track of the changes besides the originating transactions. In most cases, using the latter method means adding extraction logic to the source system.
Many data warehouses do not use any change-capture techniques as part of the extraction process. Instead, entire tables from the source systems are extracted to the data warehouse or staging area, and these tables are compared with a previous extract from the source system to identify the changed data. This approach may not have significant impact on the source systems, but it clearly can place a considerable burden on the data warehouse processes, particularly if the data volumes are large.

Physical Extraction Methods

Depending on the chosen logical extraction method and the capabilities and restrictions on the source side, the extracted data can be physically extracted by two mechanisms. The data can either be extracted online from the source system or from an offline structure. Such an offline structure might already exist or it might be generated by an extraction routine.
There are the following methods of physical extraction:

- Online Extraction
- Offline Extraction

Online Extraction

The data is extracted directly from the source system itself. The extraction process can connect directly to the source system to access the source tables themselves or to an intermediate system that stores the data in a preconfigured manner (for example, snapshot logs or change tables). Note that the intermediate system is not necessarily physically different from the source system.
With online extractions, you need to consider whether the distributed transactions are using original source objects or prepared source objects.

Offline Extraction

The data is not extracted directly from the source system but is staged explicitly outside the original source system. The data already has an existing structure (for example, redo logs, archive logs or transportable tablespaces) or was created by an extraction routine.

You should consider the following structures:
•Flat filesData in a defined, generic format. Additional information about the source object is necessary for further processing.
•Dump filesOracle-specific format. Information about the containing objects is included.
•Redo and archive logsInformation is in a special, additional dump file.
•Transportable tablespaces

Change Data Capture