**Types of ETL Tools**

ETL tools have been around for over 30 years. As technology has evolved in that time, different types of solutions have entered the market. There are several pure-play ETL vendors, such as Informatica, who specialize in ETL. Other tools are offered by large software vendors, such as IBM, Oracle and Microsoft. More recently, open source ETL tools and ETL cloud services have emerged. ETL tools can be categorized into the following main types:

### Batch ETL Tools

In this type of ETL tool, batch processing is used to acquire data from the source systems. The data is extracted, transformed, and loaded into the repository in batches of ETL jobs.

It’s a cost-effective method because it uses limited resources in a time-bound way.

### Real-Time ETL Tools

Data is extracted, cleansed, enriched, and loaded to the target system in real-time ETL tools. These tools offer you faster access to information and improve time to insights.

As the need to gather and analyze the data in the shortest possible time has augmented, these ETL tools are becoming more popular among businesses.

### On-Premise ETL Tools

Many companies operate legacy systems that have both the data and the repository configured on-premise. The main reason behind such an implementation is data security. That’s why companies prefer having an ETL tool deployed on-site.

### Cloud ETL Tools

As the name suggests, these tools are deployed on the cloud as various cloud-based applications form an essential part of enterprise architecture. Companies opt for cloud ETL tools to manage data transfer from these applications. Cloud-based ETL tools let businesses leverage flexibility and agility in the ETL process.

**Enterprise Software ETL**

Several software companies sell and support commercial ETL software products. These have been around the longest and tend to be the most mature in adoption and functionality. All of these products provide graphical interfaces for designing and executing ETL pipelines. They all connect to most relational databases. Some have support for non-relational data sources such as JSON and XML. A few have support for event streaming sources such as Apache Kafka.

* **Informatica PowerCenter**. This product is arguably the most mature ETL product in the market. It is used by many large companies and is well-regarded by industry analysts. It is part of a large portfolio of products, bundled as Informatica Platform. These products are perceived as IT-centric, and they are also very expensive. Informatica is less mature than some other products for semi-structured and unstructured sources.
* **IBM InfoSphere DataStage**. DataStage is a very mature ETL product that was acquired from the company Ascential. It is especially popular with IBM shops. Unlike many other ETL tools, it provides strong capabilities for working with mainframe computers. DataStage is perceived as expensive, complex to license, and overlaps with other products in the same family.
* **Oracle Data Integrator (ODI)**. Oracle’s ETL product has been available for many years. It uses a fundamentally different architecture from other ETL products. Instead of performing transformations in the ETL tool itself using a dedicated process and hardware resources, ODI moves data into the destination, then performs transformations using the features of the database or Hadoop cluster.
* **Microsoft SQL Server Integration Services (SSIS)**. SSIS is very popular among users of SQL Server. It is lower in cost than other enterprise ETL tools, and easier to use.
* **Ab Initio**. Ab Initio is a highly secretive company based in Lexington, MA. The product began as ETL and evolved to provide other offerings, such as metadata catalogs. Ab Initio is proprietary and very expensive. It claims to be higher in performance and easier to use than traditional ETL tools.
* **SAP Data Services**. SAP’s ETL tool is designed primarily for moving data between SAP applications. It is not widely used outside of these environments.
* **SAS Data Manager**. SAS has developed an ETL product with strong support for Hadoop, streaming data and machine learning, but limited support for bulk loading destination systems. Data Manager is perceived as expensive, while SAS is perceived as a vendor with high customer satisfaction.

**Open Source ETL**

Over the past 10 years, software developers have created several open source ETL products. These products are free to use. Their source code is also freely available, which allows you to extend or enhance their capabilities. These tools vary significantly in quality, integrations, ease of use, adoption and availability of support. Like the enterprise ETL tools, many of these open source ETL tools provide a graphical interface for designing and executing pipelines.

* **Talend Open Studio**. Talend’s ETL tool is the most popular open source ETL product. Open Studio generates Java code for ETL pipelines, rather than running pipeline configurations through an ETL engine. This approach gives it some performance advantages.
* **Pentaho Data Integration (PDI)**. Formerly known as Kettle, PDI is an open source ETL tool well known for its graphical interface called Spoon. PDI generates XML files to represent pipelines, and executes pipelines through its ETL engine. Pentaho was acquired by Hitachi Data Systems in 2015.
* **Hadoop**. Hadoop is a general purpose distributed computing platform. It is used to store, manipulate and analyze data of any structure. Hadoop is a complex ecosystem of open source projects, comprising over 20 different technologies. Some of these projects are used to perform ETL tasks, such as Pig, MapReduce and Spark. When used for ETL, data is typically first loaded into Hadoop’s Distributed File System (HDFS) “as is” from source systems. Sqoop is a tool used to move data from relational databases into HDFS. Once the data is stored in Hadoop, any of the projects can be used to transform and store the cleansed data in HDFS. Hive is a popular project for using SQL to define these transformations (a Hive query is compiled into MapReduce). Pig and Spark can be used as well. Other projects are used to coordinate multi-step transformations for ETL. The Hadoop ecosystem evolves very quickly, and new projects emerge frequently that are designed to perform data transformation.

**Custom ETL**

Many companies use general purpose programming languages to write their own ETL tools. This approach has the greatest flexibility, but also requires the most effort. This approach also requires users to perform their own maintenance, build their own documentation, test and perform ongoing development. Users of custom ETL often find it difficult to source help from people outside of their own team.

* **SQL**. If your data source and destination are the same, then using SQL is a very powerful option. SQL is very efficient at reading and writing data, as well as performing basic transformations. SQL is not very effective at complex transformations that involve decision trees, or calling external sources, for example. SQL is also built into the database, so no additional license fees or technologies are required. SQL is widely understood by DBAs and developers.
* **Python**. Python is a general purpose programming language. It has become a popular tool for performing ETL tasks due to its ease of use and extensive libraries for accessing databases and storage technologies. Python can be used instead of ETL tools for ETL tasks. Many data engineers use Python instead of an ETL tool because it is more flexible and more powerful for these tasks.
* **Java**. Java is another general purpose programming language that can be used to build ETL processing. Java is one of the most popular programming languages and has extensive support for different data sources and data transformation. Java and Python each present different trade-offs for ETL. When choosing between them existing skills may be the most important factor.
* **Spark & Hadoop**. Spark and Hadoop work with large datasets on clusters of computers. They make it easier to apply the power of many computers working together to perform a job on the data. This capability is especially important when the data is too large to be stored on a single computer. Today Spark and Hadoop are not as easy to use as Python, and there are far more people who know and use Python.

**ETL Cloud Services**

Amazon AWS, Google Cloud Platform and Microsoft Azure offer their own ETL capabilities as cloud services. If your data is already in one of these cloud platforms, there are a number of advantages to using their ETL services. One important difference is their integration to the proprietary data sources — traditional ETL tools cannot be used to work with data in these systems, or are less mature with these technologies. These services have the advantage of providing tight integrations to other cloud services, elasticity and use-based pricing. These solutions are also highly proprietary and only work within the framework of the cloud vendor — you cannot use them in a different cloud vendor’s platform, and you cannot move these capabilities to your own data centers.

* **AWS EMR**. Elastic MapReduce (EMR) is the Hadoop offering provided by AWS. Companies running on AWS who like to use Hadoop for ETL or ELT may use EMR. As with any Hadoop distribution, there are several tools available to perform ETL, including Hive and Spark. This solution has the advantages of being very powerful and scalable, and capable of working with structured and unstructured data. It is also elastic and users only pay for what they use. It has the downside of being very difficult to use.
* **AWS Glue**. This is a new fully managed ETL service AWS announced in late 2016. Glue is targeted at developers. It is tightly integrated into other AWS services, including data sources such as S3, RDS, and Redshift, as well as other services, such as Lambda. Glue can connect to on-premises data sources to help customers move their data to the cloud. ETL pipelines are written in Python and executed using Apache Spark and PySpark. Like most services on AWS, Glue is designed for developers to write code to take advantage of the service, and is highly proprietary — pipelines written in Glue will only work on AWS.
* **AWS Data Pipeline**. AWS Data Pipeline is cloud-based ETL. It can be used to schedule regular processing activities such as distributed data copy, SQL transforms, MapReduce applications, or even custom scripts, and is capable of running them against multiple destinations, like Amazon S3, RDS, or DynamoDB.
* **Azure Data Factory**. Data Factory is a fully managed service that connects to a wide range of cloud and on-prem data sources. It is capable of copying, transforming and enriching data, then writing the data to Azure data services as a destination. Data Factory also supports Hadoop, Spark and machine learning as transformation steps.
* **Google Cloud Dataflow**. Dataflow is a fully managed service targeted at developers for designing batch and continuous ETL jobs. Dataflow provides APIs for Java and Python for developers to connect to Google Cloud sources, apply transformations and write data into other Google Cloud destinations. Dataflow APIs are based on Apache Beam. Unlike other cloud services, Dataflow does not connect to on-prem sources.
* **Segment**. Segment is a SaaS technology that moves data between systems based on events. Companies use Segment to move data into their data warehouses, and also to move data from one application to another, such as from a marketing automation system into Salesforce. This is not strictly an ETL tool. Segment connects to many sources, including dozens of popular applications as sources and destinations, including data warehouses.
* **Stitch**. Stitch Data is a SaaS provider of ETL that is new to the market and focused on developers. This is built on an open source core called Singer.

**Staging Layer**

A staging layer is an intermediate storage area used for data processing during the ETL process. The data staging layer sits between the data sources and the data targets. Which are often data warehouse. Data Staging spaces are frequently ephemeral in nature, with their contents being wiped before performing an ETL process or shortly after it has been completed successfully. However, there are architectures for staging areas that are designed to hold data for long periods of time for preservation or debugging purposes.

Before being loaded to the target system, data from the source can be replicated, reformatted, and tested in a staging area.

Most firms today have several Data Sources to derive information. Before being loaded into the new system, the extracted data must be polished and cleansed, as well as have the right format and structure. A Staging space is useful in this situation. Data is altered, replicated as needed, linked and aggregated if necessary, and then cleansed in this intermediate layer.

A Data Staging Area is a design concept for a**Data Pipeline**. It is a location where raw/unprocessed data is stored before being modified for downstream usage. Database tables, files in a Cloud Storage System, and other staging regions are examples.

**Curation Layer**

Data is transformed into consumable data sets and it may be stored in files or tables. The purpose of the data, as well as its structure at this stage is already known. You should expect cleansing and transformations before this layer. Also, denormalization and consolidation of different objects is common. Due to all of the above, this is the most complex part of the whole Data Lake solution. In regards to organizing your data, the structure is quite simple and straightforward. For example: Purpose/Type/Files. Usually, end users are granted access only to this layer.

**Presentation Layer**

This Layer where the users get to interact with the data stored in the data warehouse. Queries and several tools will be employed to get different types of information based on the data. The information reaches the user through the graphical representation of data. Reporting Tools are used to get Business Data, and Business logic is also applied to gather several kinds of information. Meta Data Information and System operations and performance are also maintained and viewed in this layer.