ETL vs ELT

INTRODUCTION – STAGES IN DATA PROCESSING

Using ELT (Extract Load Transform) instead of traditional ETL (Extract Transform Load) will save your team valuable data resources and development cycles. More importantly, it will provide more accurate and flexible data capabilities for your analytics team. The most advanced data companies in the world (Google and Facebook) are using this new data-management technique.

In order for retailers to be able to decipher and manage their data, there are three important stages through which the data must travel before it arrives in the hands of the end user. These steps are:

Extract- The process by which your data is extracted from a data source

Transform- Changing the source data into a digestible format

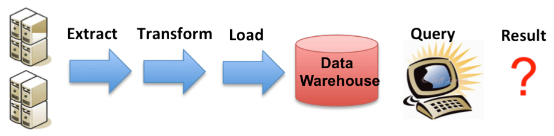
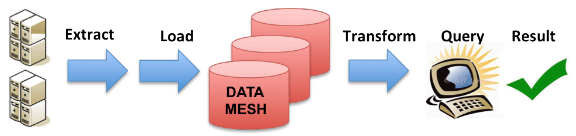
Load- Delivering new data into the platform where it can be leveraged by analytics

While all three steps are necessary in the formation of digestible data, the order in which these stages are performed can have a fundamental effect on the meaning and the overall outcome of the solution. Historically, enterprise data platforms have had little choice but to embrace an ETL standard, whereby the transformation of the data happens before it is loaded into the platform. However, a few emergent technologies are providing approaches that challenge the ETL tradition, by staging the loading of the data into the platform *before* it gets translated. This new form of data processing is referred to simply as "ELT," or Extract Load Transform.

Benefits of ELT:

- Re-Transform your data if you change your approach later
- Reconcile against raw data after the Transform
- Create new attributes, derived fields, or fix errors at any point
- Scalable to support larger volumes of data

ETL OR ELT – WHAT’S THE DIFFERENCE?

ETL: Extract Transform Load	ELT: Extract Load Transform
Extract Transform and Load (ETL), is the most common way in which existing data systems migrate data from collection point to its end state. In this method, data is “extracted” from the data sources using an extraction tool, and then is transformed. Once the data is in the format desired, it is loaded into the data platform where it is stored. However, ***this means that the transform cannot easily be modified after the fact, as the extracted data may no longer exist in the source or staging database. If any bugs are discovered in the transform, or if we want to change it in any way, we cannot easily do so.***	ELT is a new approach that eliminates some of the drawbacks of ETL. The main difference between the two approaches is WHEN the transform happens. With ELT, the transform is done at query-processing time, rather than in an up-front step outside the Data Mesh. ***This means, that each individual query can either rely on the default transform, or implement its own. It decouples extract and load from the transform, allowing us to study the impact of different transforms and to easily correct any errors in the default transform by replacing it.***
 Extract Transform Load Data Warehouse Query Result ?	 Extract Load DATA MESH Transform Query Result

Let’s take a look at this from a retailer’s point of view. RETAILER-X is a multi-channel retailer that would like to reward their loyal customers. Data on these customers exists in two databases, one that was originally deployed to support eCommerce and the other that was deployed to support an in-store loyalty card program.

MEET RETAILER-X’S SHOPPERS:

Jane Smith Bob Miller Sam Biggs Susan Biggs

Here is how the data RETAILER-X has collected on their customers may look in their data tables:

ONLINE CUSTOMER TABLE

Customer ID	First	Last	Email
17	Jane	Smith	jane@gmail.com
29	Bob	Miller	bob@hotmail.com
37	Susan	Biggs	biggsfamily77@yahoo.com

LOYALTY CARD TABLE

Candid	Email	First	Last
999	jane@gmail.com	J.	Smith
966	bmiller85@billco.com	Bob	Miller
939	biggsfamily77@yahoo.com	Susan	Biggs
988	biggsfamily77@yahoo.com	Sam	Biggs

RETAILER-X wants to bring these two tables together into a single table so that they have a better understanding of their customers' shopping behavior. If RETAILER-X chooses to adopt the old-fashioned ETL approach, they have to decide upfront what transform to use. Let's assume they decide to join the tables based on email address. In this stage, the transform step searches for all cases where there is a match between the email address in the Online table and the email address in the Loyalty Card table. The result of this transform is shown in the following EDW table:

EDW MERGED CUSTOMER TABLE

EDW Cust. ID	Online Cust. ID	Loyalty Card ID	Email	First	Last
A	17	999	jane@gmail.com	J.	Smith
B	37	939	biggsfamily77@yahoo.com	Susan	Biggs
C	37	988	biggsfamily77@yahoo.com	Sam	Biggs
WHERE IS BOB???					

ERROR! We were unable to match Bob! **ERROR!** We confused Sam & Susan!

Errors like these are common in transform steps as in most cases there is no overall data model that contemplates independent tables being joined. Even if RETAILER-X was able to find and identify these errors and come up with a better transform that causes fewer of them, they will not be able to easily re-process data from the original sources.

BENEFITS OF ELT

Aside from correcting errors, there are other reasons why we might wish to change a transform. For example, assume that RETAILER-X starts gathering loyalty card numbers online by offering shoppers who provide them an extra 10% off. This data goes into the Online database:

ONLINE CUSTOMER TABLE + LOYALTY CARD				
Loyalty Card ID	Customer ID	First	Last	Email
966	29	Bob	Miller	bob@hotmail.com

Now RETAILER-X can create a new transform that matches on this data. With ELT, the execution of the transform is delayed until query time, which means that the query applies to all data that has ever entered Data Mesh, not just newly transformed data. Furthermore, before making it the default transform, we can carefully study the effects it will have by running a battery of standard queries and reports with both the existing default transform and the proposed new query.

Additionally, many of the analytics we power need the ability to look at the data in different views, and combine additional data sources at transform time. This is why having access to the raw data at transform time becomes critical to the long and short-term results in your data processing efforts.

WHY CAN {RR} DELIVER ELT WHEN OTHER DATA PLATFORMS CANNOT?

The entire {rr} platform is built upon an infrastructure that is accustomed to handling high volumes of ELT data at an extremely fast pace. While other platforms are built primarily to store data, the {rr} platform was built to support personalization, providing a very flexible, highly intelligent platform that allows us to answer high volumes of queries an unparalleled pace.