ETL?

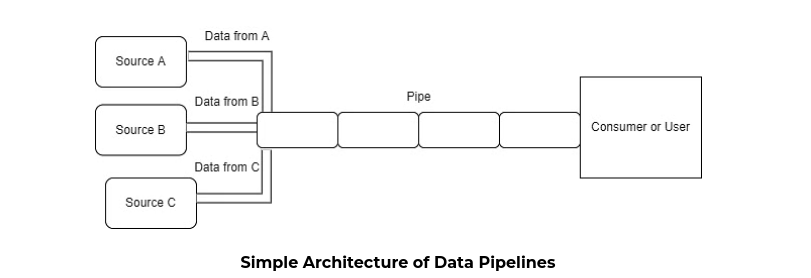
In extract-transform-load (ETL), data is obtained from multiple sources like transaction database,APIs and transformed, and stored in a single data warehouse.

**ETL Pipeline AND DATA Pipeline:**

**Data Pipeline:**

n the good old days, people walked miles to fetch water for their basic needs from water sources like wells, rivers, ponds, etc. It was a manual process and very challenging as well.

As time changed, getting water via the old traditional method was impractical. Our needs also increased; we wanted more water for various purposes. We wanted it automatically and at the same more frequently. The invention of the pipeline system automated the water supply delivery to homes via pipes from various water sources. The concept of Data pipelines is no different from the water pipeline. **We have data sources and data consumers, and to connect them, we use data pipelines.**



ETL VS Data Pipeline:

Both terms refer to the process of moving data from various sources to a single data warehouse. But they are not the same. Here are the key difference between ETL pipeline and Data Pipeline.

1. ETL pipeline are subset of Data Pipeline:

The ETL (Extract, Transform, and Load) pipeline concludes by loading data into a database or data warehouse, but a **Data Pipeline doesn't always end with loading data.** Instead, loading data can trigger new processes and flows in a Data Pipeline by triggering webhooks in other systems.

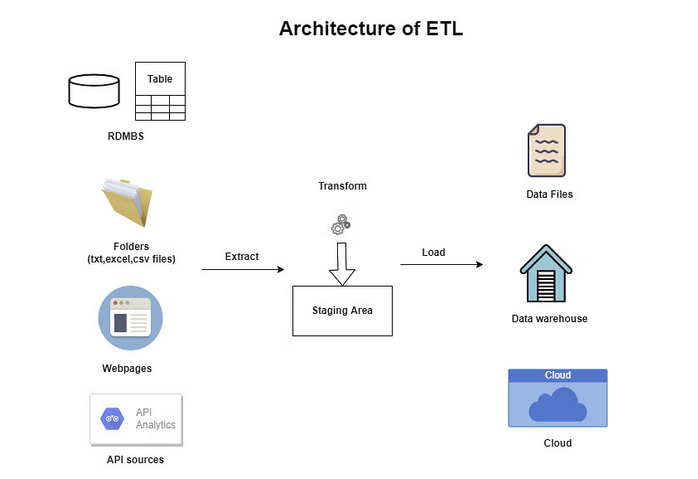
1. ETL pipeline always involves Transformation:

A Data Pipeline involves moving data between different systems but does not necessarily include transforming it. Data is extracted from a source, transformed, and then loaded into its output destination using ETL, as its acronym suggests.

1. ETL pipeline are batch processed, and data pipeline are real-time:

ETL pipelines move data in chunks at regular intervals and in batches, and the pipeline might run twice per day or at a time when system traffic is low. In contrast, a data pipeline runs as a real-time process involving streaming computations and continuously updating data.

**Architecture of ETL Pipeline:**

****

Now understanding ETL Pipeline in more details.

1. **Extract:**  This step involves the gathering data from multiple sources, eventually appearing as rows and columns in an analytics database. Data extraction meant getting information from files like Excel, CSV, Text, etc. Storing data in raw format is still prevalent, as they were the primary sources of customer information. But now, with the increase in web and mobile application usage, most businesses find valuable information for their data pipelines through web pages usingweb scraping or via API. We can gather Amazon product reviews to do sentimental analysis, provide recommendations to the buyers, scrape job postings on different company career pages, and build a job portal using this data.

There are 3 most commonly used approach for Extract:

1. Full-Extraction: The entire data is extracted from the source and pushed into the data pipeline.
2. Incremental Extraction: Each time a data extraction process runs (such as an ETL pipeline), only new data and data that has changed from the last time are collected—for example, collecting data through an API.
3. Source driven extraction: The source notifies the ETL system when data changes, triggering the ETL pipeline to extract the new data.

2. **Transform:**

All the data science professionals would be familiar with the term "Garbage in, garbage out." We know how important it is to feed good-quality data for any solution. To improve data quality, we have to apply data transformation techniques. It is a very critical step while building ETL data pipelines, as the entire value of our data lies in how well we transform it as per our needs.

Let us look at some examples of transformation steps taken to process the data.

1. *Basic Cleaning:* Converting data into a suitable format as per our requirement. For example, suppose some date input doesn't have any specific format. In that case, the user can provide it differently, like DD/MM/YY or DD/MM/YYYY or MM/DD/YYYY, etc. We can convert it into a standard format as per our needs.
2. Join Table: If our source is RDBMS or SQL tables, we might need to join or merge multiple data tables.
3. Filtering: Sometimes, we only need a subset of the data, and we can filter the relevant data and discard everything else. It will, in turn, increase the speed of our data transformation process.
4. Aggregation: applying aggregate functions such as average, minimum, maximum, median, percentile, and sum. For example, we can compute total sales revenue by region, store, or product.

Some issues that we might encounter while transforming the data are :

1. Duplicate: This is a very frequently occurring issue. Understanding whether the identical records are duplicated by mistake or legitimate is crucial. For instance, a given person probably bought the same product on the same day at the same store. If the record is duplicate, then we need to remove them.
2. Outliers: In manually entered databases, data entry errors are possible. For instance, someone might enter a person's age as 150 or 200 due to a typo.
3. Missing Data: Missing data is a common problem that can impact the conclusions drawn from the data. We can drop these values if we have data points required for our analysis even after removing the missing data points. To fix this, we can return to the extraction step and see if we can use an additional data set to fill in the missing data. It is possible to impute the missing data using means, medians, modes, regression-based imputation, etc.

**3. LOAD:**

The final step of creating ETL data pipelines is to load data, and the final destination of the processed data could vary based on the business requirement and analysis required of the data. The following are examples of target databases or data warehouses where we can store our processed data.

1. Flat files: csv,txt and excel spreadsheet are standard text file format for storing data.
2. SQL RDBMS: The SQL database is a trendy data storage where we can load our processed data.
3. Cloud

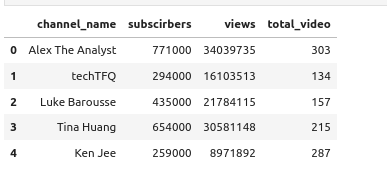
Now we have understood the basic concept . now we will extract data from youtube api . we will select 5 different channel and we will scrape their channel name, total\_videos, total\_subscriber,views etc. we need channel id to get those information. From youtube we can easily get their channel id. Also from youtube api we have generated the api\_key which we will use in our code.

**Extract:**

Our data was extracted in json format and we have pulled the required data only.

**Transformation:**

Next we write code and extract data from youtube api and our data .Initially we have got data in json format. We have extracted only those information that we need and saved in another variable. We have stored data in pandas data frame which is further again converted as it cannot be used to store in the database. Our data in data frame looks like this:



Now we need to load this data to database. I have loaded this data locally in my machine and for scheduling the task, i have pushed the data to csv file as well. And every time we run this code new csv file with different datetime stamp is created.

Now to load the data, we need to install Postgres and set up the database:

1. **Create a new PostgreSQL user**: Open your command-line interface and type the following command:

sudo -u postgres createuser --interactive

This command will prompt us for the name of the new user and ask whether it should be a superuser.

Created user biki.

1. **Create a new PostgreSQL database**: Still in the command-line interface, type the following command:

sudo -u postgres createdb etl\_assignment

1. **Open PostgreSQL interactive terminal**: Type the following command to start the PostgreSQL interactive terminal:

sudo -u postgres psql

1. **Set a password for the new user**: In the PostgreSQL interactive terminal, type the following command:

\password biki

We will be prompted to enter a password for the user. bikash123

1. **Grant privileges**: To grant all privileges on a database to a user, use the following command in the PostgreSQL interactive terminal:

GRANT ALL PRIVILEGES ON DATABASE etl\_assignment TO biki;

Now the database setup has been completed and we have create the schema as follow in db:

Create table youtube\_analysis(id PRIMARY KEY,

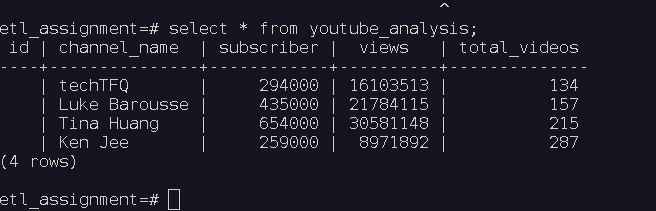
Channel\_name varchar(250),

Subscriber bigint,

Views bigint,

Total\_videos bigint);

Now the db and table has been created and inserting the data via python code our database table looks like this:



As previously mentioned, I have saved the data as CSV file too. Now we want to automate the task. We will use GitHub action for this. Our git hub action file should create a new csv file everytime the scripts run. It also should push the newly created file in github and delete the csv file older than 10 days.