# Homework 1
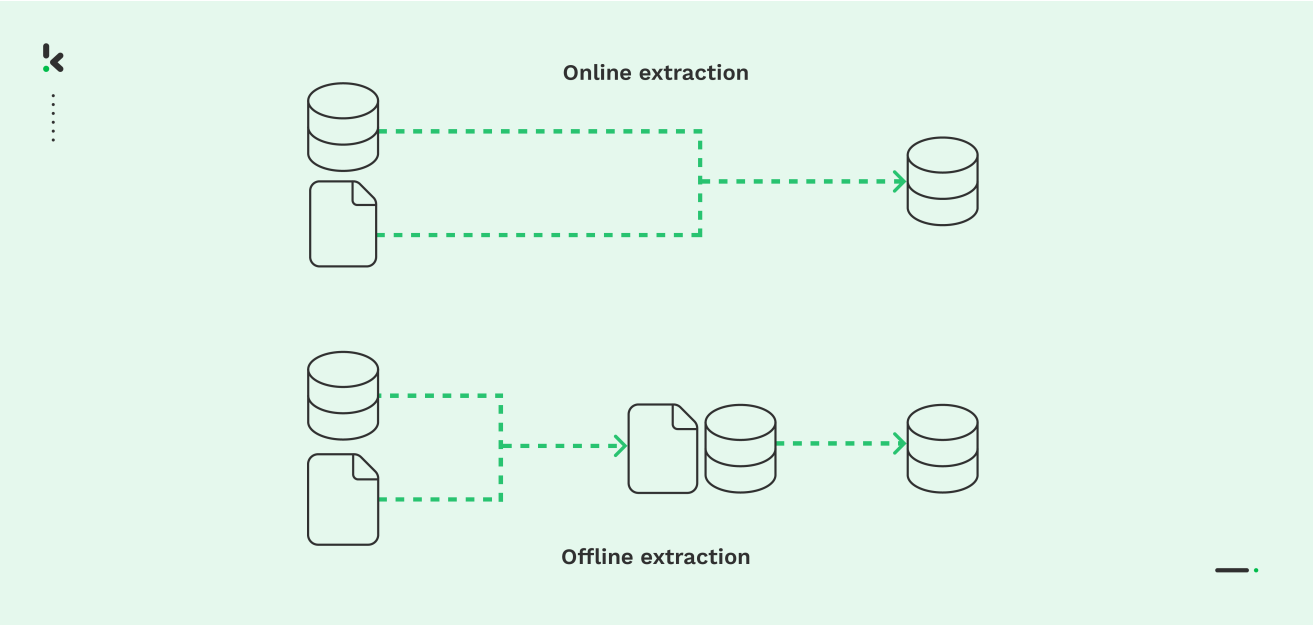
Date: 27/03/2024

StudentID: ChauTND2

## Ex1: Online vs Offline extraction. Example?
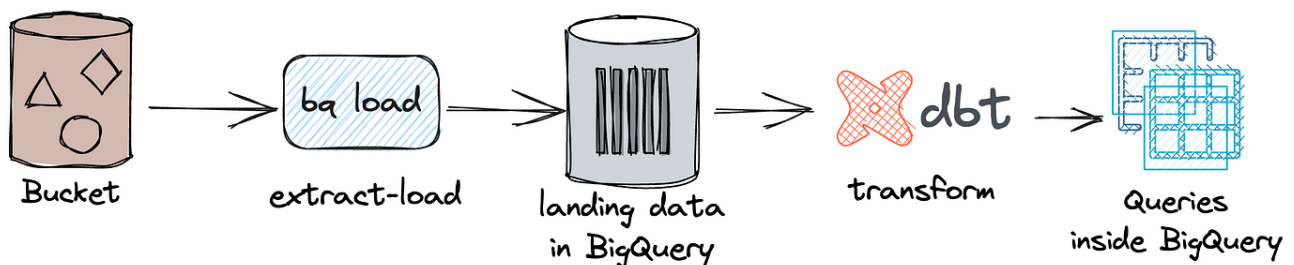


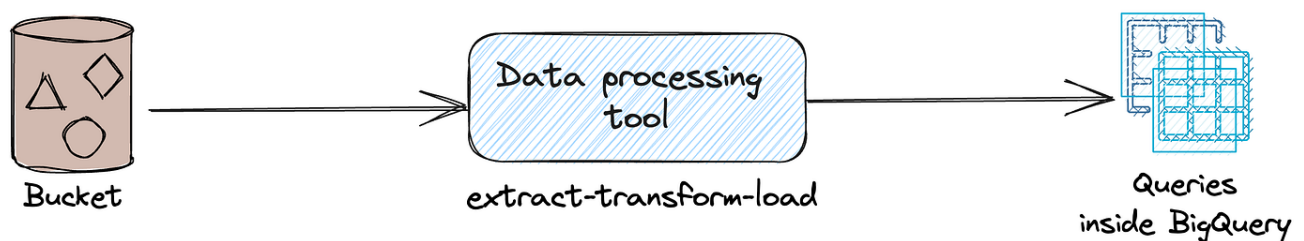|  | Online extraction | Offline extraction |
|---|---|---|
| Definitions | Online extraction is the process of extracting data by interacting directly with the data source (querying a database, API, etc.), in real-time or near real-time | Offline extraction entails extracting data from the source and storing it elsewhere, such as dump files, archive logs, backups, or data lakes. This is used when real-time interaction with the data source is not feasible or practical |
| Use cases | Online extraction is used when there is a need for real-time analysis (sales, stocks, etc.) or the data tends to change quickly and dynamically | Offline extraction is used when there is no need for real-time analysis, such as for studies, historical trend analysis, periodic reporting, etc. |
| Examples | A retail company analyses real-time sales trends by maintaining a direct connection to continuously access the latest transaction data | Daily transaction logs are extracted and stored in a data lake. These logs are later accessed for various purposes, like monthly sales report |

## Ex2: ETL vs ELT?

|  | ETL | ELT |
|---|---|---|
| Meaning | Extract Transform Load | Extract Load Transform |
| Process | Extract data from various sources, transformed according to business rules or requirements, and then loaded into a data warehous or data mart | Extract data, load it into a storage system (like a data lake or data warehouse) in its raw form, and transformation occurs later, often within the storage system itself or during query execution |
| Advantages | Assure data quality, as all data are cleansed, transformed and normalized before storing in DW | Preserve the entire raw data, offering flexibility for various use cases |
| Disadvantages | May result in missing some parts of the data due to pre-transformation | Requires more storage |
| When to use | Suited for smaller datasets or when transformation requirements are well-defined and consistent | Preferred for handling large volumes of data and scenarios where data structures and transformation needs may vary over time or across different analytical purposes (like large company's data) |

## ELT



Bucket    bq load    landing data    transform    Queries
   extract-load    in BigQuery      inside BigQuery

## ETL



Bucket      Data processing tool      Queries
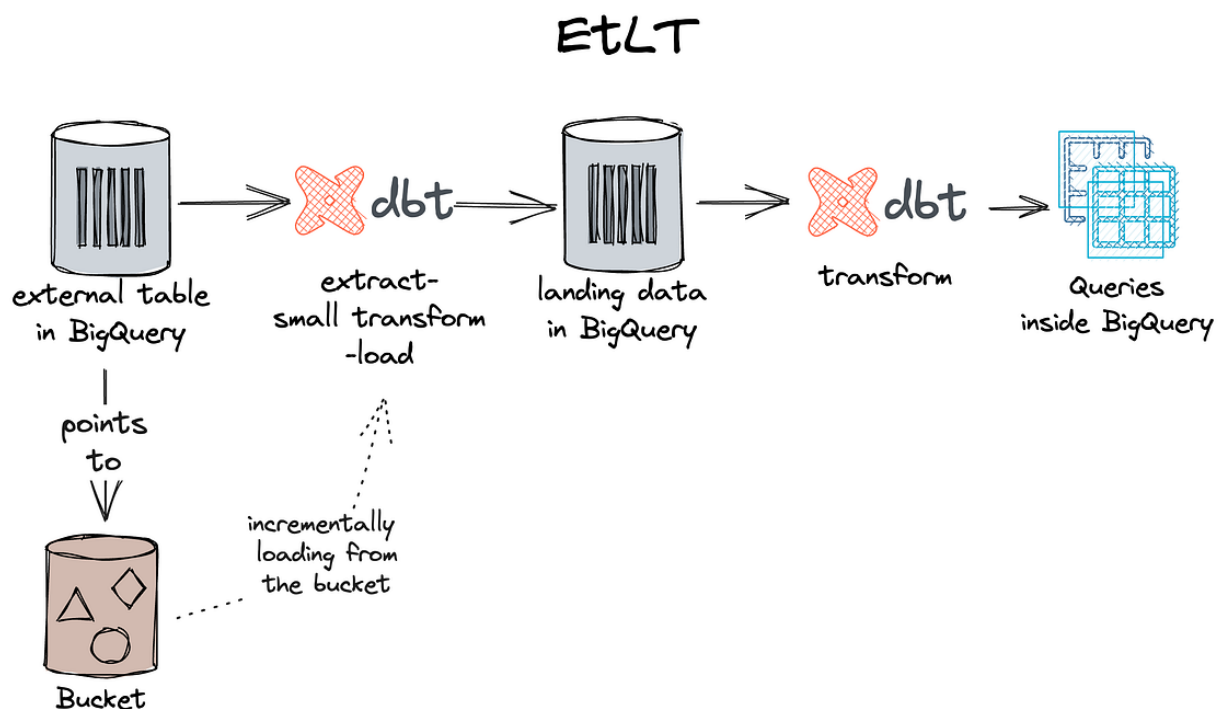     extract-transform-load      inside BigQuery

## What do companies use these days?

From my research, ETL (Extract, Transform, Load) used to be the standard procedure primarily due to the high costs of cloud storage in the past. To minimise data size while maintaining analysis quality, people needed to cleanse and filter data, retaining only essential components. However, with advancements in technology and the decreasing costs of cloud storage, the approach to data handling has evolved.

Nowadays, there's a growing trend towards preserving entire raw data sets for later transformations. This shift is driven by factors such as more affordable cloud storage, the recognition of the value of raw data for future analysis, and the increasing variety of use cases requiring access to detailed, unaltered data.

While ETL processes are still relevant, especially for specific analytical tasks requiring data transformation and aggregation, ELT enables organisations to explore new insights, support machine learning initiatives, analyse historical trends, and meet compliance requirements. However, albeit minimal, some level of transformation or standardisation may still occur as part of the data management process, so that the data is searchable, restructured in the correct format, etc. so the process could be EtTL.



EtLT

external table in BigQuery — extract- small transform -load — landing data in BigQuery — transform — Queries inside BigQuery

points to

Bucket

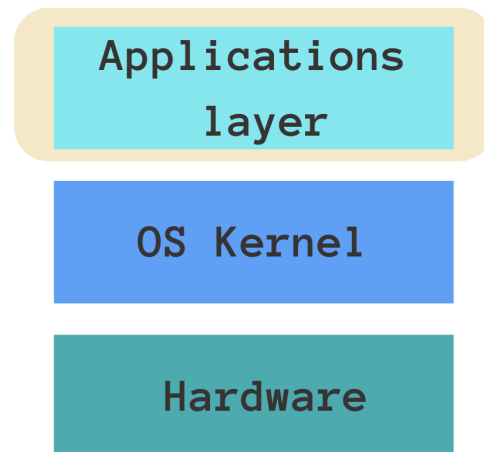incrementally loading from the bucket

# Ex3: Virtual Environment vs Virtual Machine vs Container.

VM: Virtualization of the OS Kernel +
Applications layer
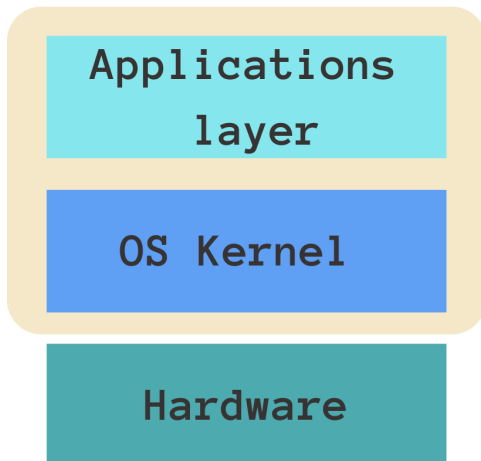
Container: Virtualization of the
Applications layer only

| Applications layer |
|---|
| OS Kernel |

| Hardware |
|---|

| Applications layer |
|---|

| OS Kernel |
|---|

| Hardware |
|---|

|  | Virtual Environment | Virtual Machine | Container |
|---|---|---|---|
| Definitions | A self-contained directory tree that contains dependency details, allowing each application install and manage their own dependencies | An emulation of a physical computer, runs their own instance of the OS Kernel and applications, mostly isolated from the host system | A software instance that encapsulates the application and its dependencies |
| How they do it | Let application install their own dependencies inside a directory separated from the global environment | Use hypervisor which abstracts the hardware physical resources, allow the VM use the resources without relying on host system | Some kernel technologies like namespaces and control groups (cgroups) |
| What they use | Utilizes the computer's hardware (CPU, memory, storage), OS kernel, and applications. Only separates the dependency packages | Use the computer's hardware (CPU, memory, storage), but use its own OS kernel and applications | Use the computer's hardware (CPU, memory, storage) and OS kernel |
| Isolation level | Isolate the dependency packages inside the environment | Isolate the OS kernel and the applications inside it, encapsulating the whole operating system (high level of isolation). | Isolate at the application level, shares the OS kernel. |
| Portability | Has limited portability due to differences in system architectures and dependencies | Portable across systems, as it encapsulates the entire operating system, but there might still be | Portable across systems, as it packages |

| | Virtual Environment | Virtual Machine | Container |
|---|---|---|---|
| | | compatibility issues due to hypervisor incompatibility | applications and their dependencies |
| Use cases | Suitable for different projects on one machine, or when all team members have the same OS | Used for projects that require a high level of isolation or OS-specific configurations | Commonly used in software development or team projects |