Lesson 5

Data Pipelines

In this lesson we will talk about:

Data pipelines. Origins. Types

Data Capture, Transforming & Analysing

Raw Data. Data Aggregation

Pipeline efficiency

Basic statistical functions

Origins

2025. Present time

IBM

"A data pipeline is a method in which raw data is ingested from various data sources, transformed and then ported to a data store, such as a data lake or data warehouse, for analysis."

"Data pipelines act as the piping for data science projects or business intelligence dashboards. Data can be sourced through a wide variety of places: APIs, SQL and NoSQL databases, files."

"During sourcing, data lineage is tracked to document the relationship between enterprise data in various business and IT applications, for example, where data is currently and how its stored in an environment, such as on-premises, in a data lake or in a data warehouse."

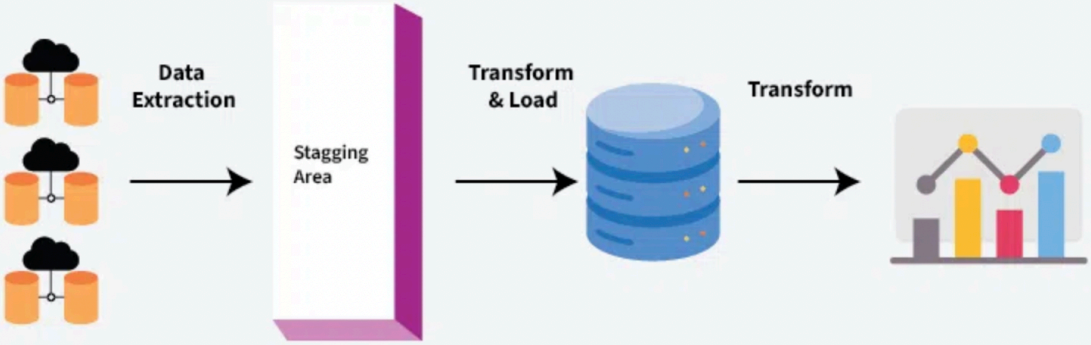
Source: ibm.com

www.geeksforgeeks.org

"Data Pipeline deals with information that is flowing from one end to another. In simple words, we can say collecting the data from various resources than processing it as per requirement and transferring it to the destination by following some sequential activities."

Source: www.geeksforgeeks.org

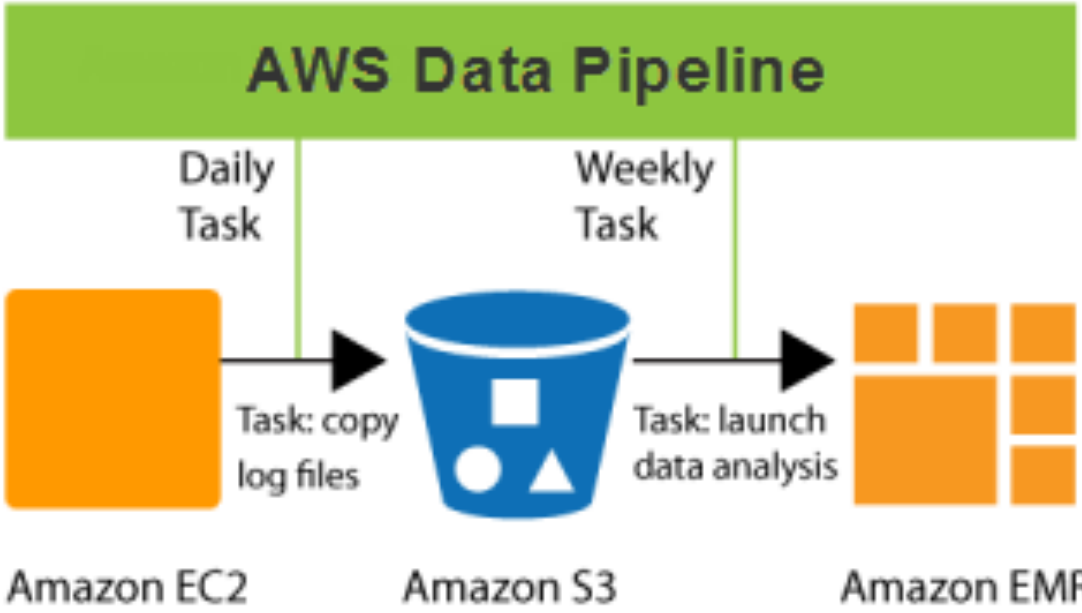
Data Pipeline Overview



AMAZON

"A data pipeline is a series of processing steps to prepare enterprise data for analysis. Organizations have a large volume of data from various sources like applications, Internet of Things (IoT) devices, and other digital channels. However, raw data is useless; it must be moved, sorted, filtered, reformatted, and analyzed for business intelligence. A data pipeline includes various technologies to verify, summarize, and find patterns in data to inform business decisions."

"Just like a water pipeline moves water from the reservoir to your taps, a data pipeline moves data from the collection point to storage. A data pipeline extracts data from a source, makes changes, then saves it in a specific destination. We explain the critical components of data pipeline architecture below."

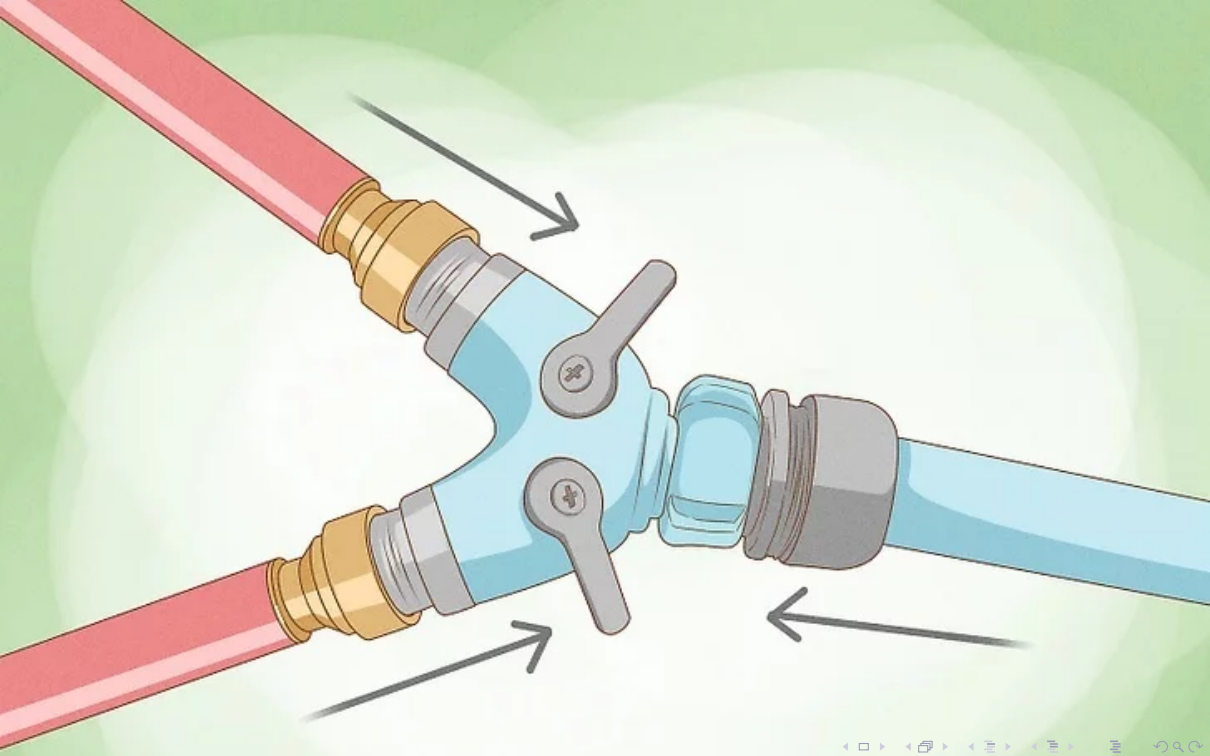

AWS Data Pipeline
Daily Task
Weekly Task
Task: copy log files
Task: launch data analysis
Amazon EC2
Amazon S3
Amazon EMR

Data pipeline = a program or what?
Lets rollback time...

1969. UNIX

The very first data pipelines

As early as 1964 Douglas McIlroy thought about a mechanism how we could connect programs same way like we connect and combine garden hoses together.



Connecting Garden Hoses

- it should connect different hoses
- as efficiently as possible
- with no leaks
- covering all the garden

Pipelines are everywhere

- transport fresh water pipelines
- collection pipelines from ground gas and oil to refineries
- wastewater pipelines
- distribution pipelines

The idea was to reuse the model from other industries to CS

"A pipeline is a mechanism for connecting the output of one program directly and conveniently into the input of another program."

— *Brian Kernighan, UNIX father*

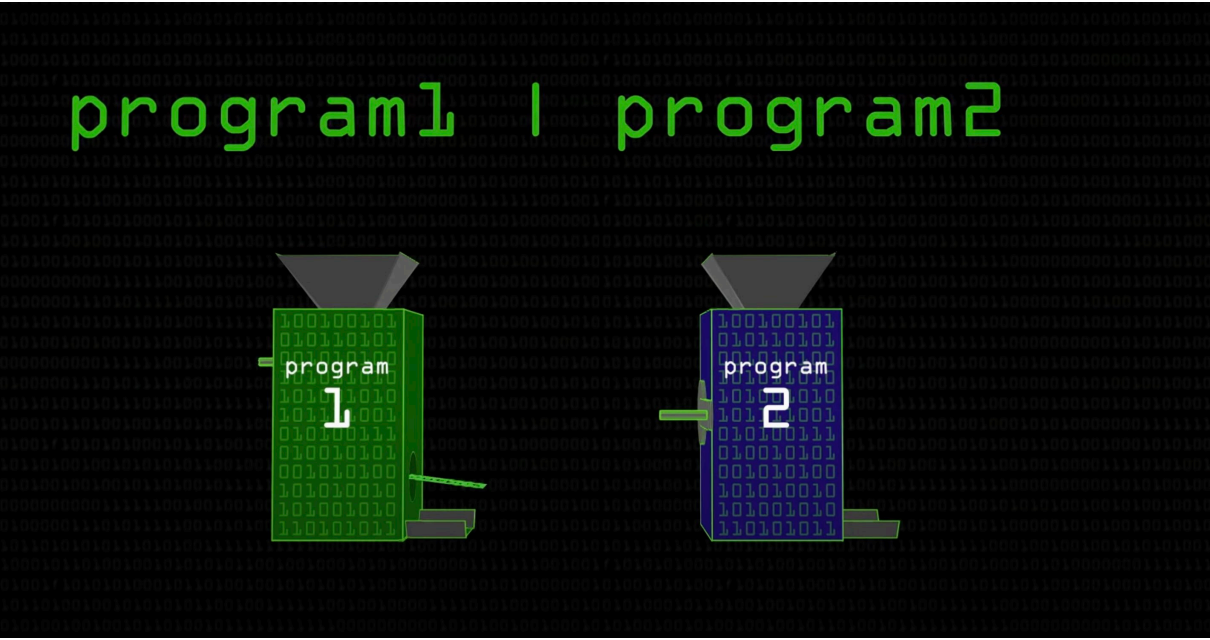
The very first models of data pipelines were defined, introduced and implemented in the operating systems.

Enter UNIX

UNIX pipelines

The very first data pipeline was implemented for UNIX based operating systems. The ideas was very simple: combine one or many programs using the **pipe** operator |

program | program | ...


program1 | program2
program
1
program
2

The pipeline: is connecting one or many programs together! It is not a single monolithic program.

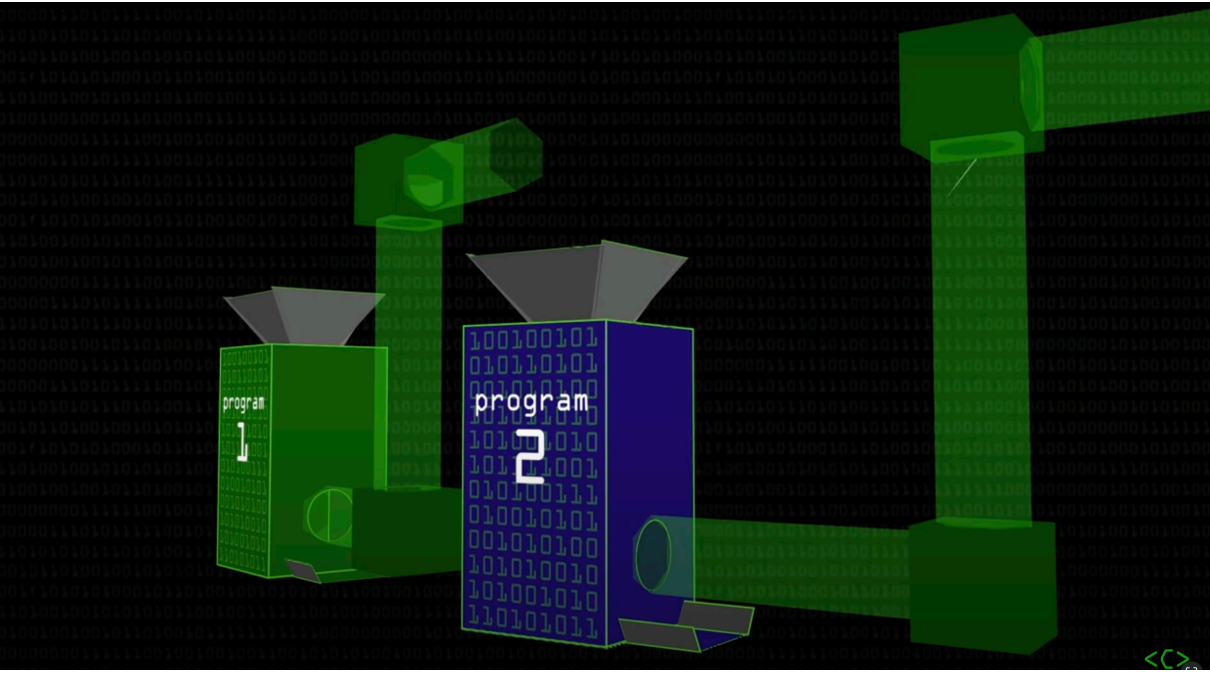
UNIX pipelines

But why connect different programs instead of having a single big program to handle all the input data?

Few reasons behind... have small, **modular** programs which can connect together to solve a problem quickly and efficiently.

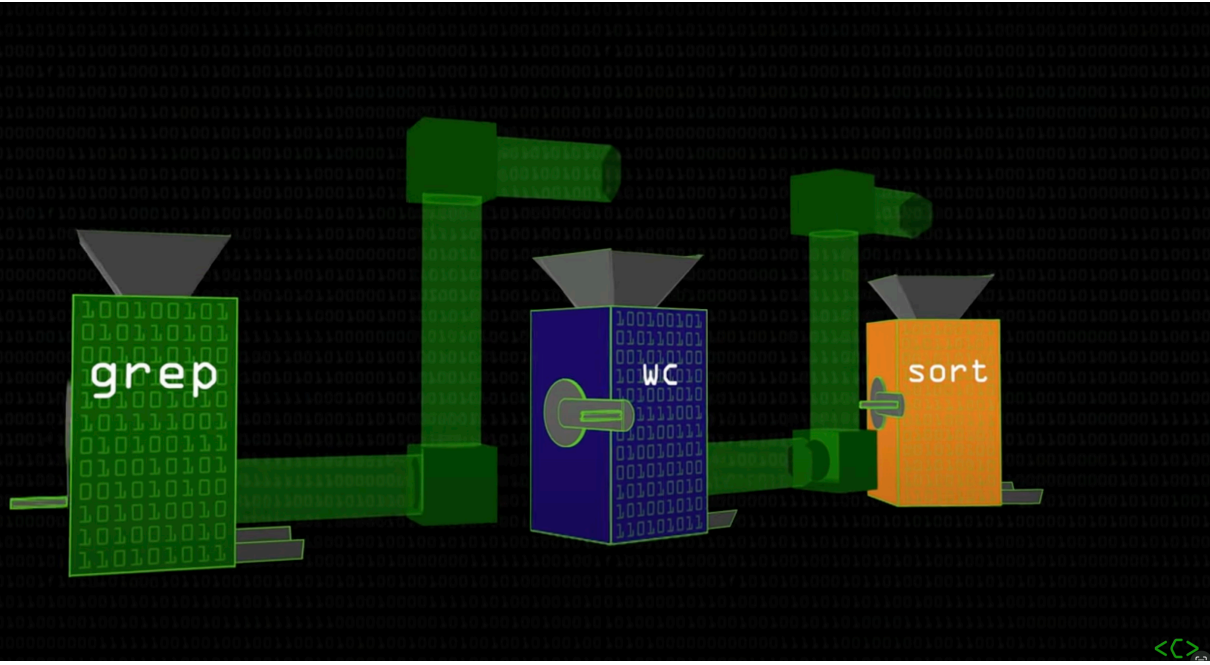
By enforcing a strict order of execution!!!

- smaller programs are easy to develop
- and are simple and cheaper to maintain
- as well the amount of data might not fit one program
- no temporary files or data in between


program
1
program
2

UNIX pipelines

Connects different UNIX programs: like ls, sort, wc, grep to solve a problem using the | pipe character


grep
wc
sort

UNIX pipelines, example

```
$ ls -lrt Images/*.png | wc -l
      140
```

Sort all images which were produced during Feb, example

```
$ ls -l Images/ | awk '{print $6,$7,$9}' | grep Feb | sort
Feb 26 treevsgraph.png
Feb 26 treevsgraph2.png
Feb 26 tsp.png
Feb 3 Alan_turing.jpg
Feb 3 Bombe.jpg
Feb 4 Queue.png
...
Feb 4 factorial2.png
Feb 4 recursion.png
```

Sort all images which were produced during Feb, example

```
$ ls -l Images/ | awk '{print $6,$7,$9}' | grep Feb | sort -V
Feb 19 hash_function4.png
Feb 19 hash_function5.png
Feb 25 binary-tree.png
Feb 26 btree.png
Feb 26 depth-height.png
Feb 26 directed-graph.png
Feb 26 graph.png
Feb 26 selfbalanced-tree.png
Feb 26 treevsgraph.png
...
```

But how about the pipeline's speed and overall performance? Should we care about?

Search for all png images measuring the elapsed time, example

```
$ time find $HOME -type f | grep png
...
/myhome/feynman_path/examples/no-entanglement.png
/myhome/feynman_path/examples/no-interference-circuit.png
/myhome/feynman_path/examples/entanglement.png
/myhome/feynman_path/examples/no-entanglement-circuit.png

real 0m9.167s
user 0m0.450s
sys 0m3.147s
```

Search and sort for all jpg and png images measuring the elapsed time, example

```
$ time find $HOME -type f | egrep 'png|jpg' | sort -V
 ...
/myhome/.vscode/extensions/osi-certified-72x60.png
/myhome/.vscode/extensions/osi-certified-72x60.png
/myhome/.vscode/extensions/osi-certified-72x60.png
/myhome/.vscode/extensions/logo.png
/myhome/.vscode/extensions/lldb.png

real 0m9.240s
user 0m0.773s
sys 0m3.034s
```

Search and sort for all jpg, png measuring the elapsed time under Windows, example

$ time find /myhome −**type** f | grep png

...

/myhome/.vscode/extensions/osi−certified−72x60.png
/myhome/.vscode/extensions/osi−certified−72x60.png
/myhome/.vscode/extensions/osi−certified−72x60.png
/myhome/.vscode/extensions/images/logo.png
/myhome/.vscode/extensions/images/lldb.png

real	2m20.786s
user	0m0.421s
sys	0m1.686s

Measuring pipeline elapsed time matters!

UNIX Pipelines

- a chain link of different parts or modules (programs)
- each part, designed to do one thing but do it well (atomicity)
- pipeline must be simple to change and expand (modularity)
- it should perform in time (performance matters)

From OS pipelines were adopted everywhere else in IT. Remember ETL?

From OS pipelines were adopted everywhere else in IT. Remember ETL?

ETL - Extract, transform, load

A three-phase computing process where data is extracted from an input source, transformed (including cleaning), and loaded somewhere else for further processing or analysis.

ETL Systems

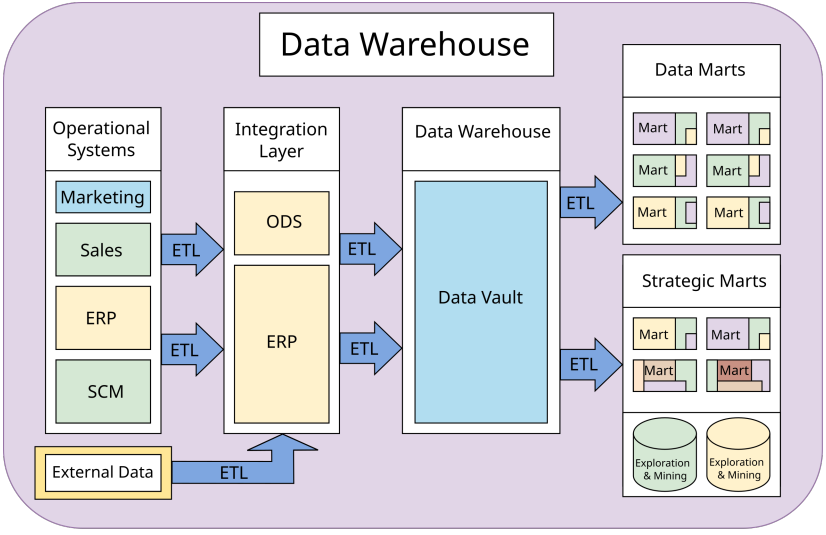
In large IT organisations, ETL systems started to be deployed as in-house or 3rd party based applications, most common close to date warehousing.

Data warehousing?

Data warehousing

"It is a system used for reporting and data analysis, a core component of business intelligence for any IT organization. Data warehouses are central repositories of data integrated from disparate sources. They store current and historical data organized in a way that is optimized for data analysis, generation of reports, and developing insights across the integrated data." (wikipedia)

Lesson 2



Source: wikipedia

Data warehousing

Data warehousing = it is a huge repository of different data sources. But how is this different from a classic transactional database? Whats the difference?

Transactional Databases

- OLTP (Online transaction processing)
- transaction-oriented applications
- Used to update data in real-time
- designed for write and read operations
- normalized data structure
- SQL and noSQL implementation
- high transaction data, multi-users accessing the system

OLTP. Normalized data

Data in OLTP systems is typically normalized to reduce redundancy and improve data integrity. It is a technique for creating database tables with suitable columns and keys by decomposing a large table into smaller logical units.

Data Warehousing

- large repository of different datasets
- historical data storage
- designed for complex read operations
- Data is denormalized to improve query performance and simplify data retrieval
- ETL mechanisms to consolidate data from multiple sources into a single central repository.

Pipelines and ETL

Pipelines started to be used for different ETL systems and when deploying data warehousing systems. Fetching or capturing data, transforming and manipulating data for a specific business reason all parts of a data pipelines.

Data pipelines vs ETL pipelines

Data pipelines and ETL pipelines or ETL systems are sometimes used. There are however some few and important differences between ETL and Data pipelines.

ETL vs Data Pipelines

- specific, clear structure and order
- Extract, Transform and Load
- not all data pipelines follow this order
- ETL is very much know for batch processing
- sometimes data pipelines do not transform data

Types of Data Pipelines

Types

Like waterpipes, there are different type of data pipelines. The model of a basic OS pipelines went beyond operating systems. But still the very same principles apply no matter of pipeline type and complexity. (elapsed time, modularity, simplicity, atomicity)

Data Pipeline Types

- batch - run at specific time, with large amounts of data at once
- streaming - processes the data in real-time
- data integration - merge data from different sources into a single central place

Data Capture (Ingestion), Transformation and Analysis

Capturing Data

"Data is collected from various sourcesincluding software-as-a-service (SaaS) platforms, internet-of-things (IoT) devices and mobile devicesand various data structures, both structured and unstructured data. Within streaming data, these raw data sources are typically known as producers, publishers, or senders." - IBM

Capturing Data

- from multiple data sources
- stored as raw data: flat files, on on a central repository
- no transformations, no changes to original data

Data Ingestion

Data capturing or data collection is also known as data ingestion. Todays every IT organization uses data ingestion as the term for capturing data from one or many data sources.

Data Transformation

This turns raw data which was previously collected into a format required by the destination data repository. A number of processes might be applied to clean or format data in a specific way.

Data Transformation, Example 1

A data stream may be ingested as a nested JSON format, and the data transformation stage will aim to flatten that JSON and extract the key fields for analysis.

Data Transformation, Example 2

Another data stream might be ingested as a XML file. We need to transform the XML input file into a SQL table to be stored in a SQL database.

Data Storage and Analysis

"The transformed data is then stored within a data repository, where it can be exposed to various stakeholders. This transformed data are typically known as consumers, subscribers, or recipients." (IBM)

Raw Data

Raw Data

Raw data, also known as primary data, are data (e.g., numbers, instrument readings, figures, etc.) collected from a source. In the context of examinations, the raw data might be described as a raw score (after test scores)." (wikipedia)

Raw Data

Raw data is fundamental to any data analysis.

The primary, original data collected from a source, which has not been transformed, aggregated or changed in any way.

The raw data is primary. The interface secondary.

Time series

All recorded observations, metrics are variable measured sequentially in time, called time series, stored as raw data. Raw data is produced by a recorder, which fetches data from a system, device or sensor, data which has not been modified or changed in any way. The data is saved as CSV flat files on disk.

Universal Format

The original recorded data, available as ascii text, CSV files or plain old sql dump files will always work with any software, user- interface or applications. The raw data never becomes obsolete, old or dated by any technology standards or new software methodologies.

Data Centric

Many software applications are putting effort in providing shiny and modern front-ends. This reflect an interface centric, rather than data centric approach. Any UI will soon become obsolete. Effort should be put in capturing and consolidating raw data. The data is primary, the user-interface secondary.

Open Data

A data centric system will always be able to process and offer original raw data in a simple manner for usage, without constraints. The data can be made available directly for download or accessible over a programmable interface which other programs and applications can use.

Not all data is good. How do you know to capture or ingest the right data?

Open Data

A data centric system will always be able to process and offer original raw data in a simple manner for usage, without constraints. The data can be made available directly for download or accessible over a programmable interface which other programs and applications can use.

Data aggregation

Aggregating Data

"Data aggregation is the process of collecting and summarizing information from various databases to create combined datasets for analysis. It helps in organizing large amounts of data into a more usable format, making it easier to derive insights and make decisions." (wikipedia)

Date warehouses. Data lakes. Data lakehouses.

Data warehouses

Are the foundations for decision support and business intelligence applications (OLAP). But these were expensive and not very good for handling unstructured data, semi-structured data.

Structured vs Unstructured Data

Data classified by its format and the existence or missing schema.

- **structured** predefined data model, includes a schema (relational database or data warehouse). A financial report is an example of structured data.
- **unstructured** no schema or data model behind. (audio or video files, web pages, sensory data)

Data lakes

Data lakes are simple storage computing systems designed to handle raw data on cheap storage for data science and machine learning. But there is no support for transactions, no data quality. In other words, a data lake stores cheaply data of any nature in any format. (Azure Data Lake Storage- ADLS)

Data lakehouses

Are combining the benefits of data lakes and data warehouses together. Provides storage and processing capabilities for different organizations built on top of the medallion data design paradigm.

Data lakehouses. Medallion architecture

Data is organized in layers, that describes the quality of data. The new architecture offers guarantee to atomicity, isolation and durability as data passes through different layers of validations and transformations: **bronze**(raw data), **silver** (validated data), **gold** (rich, business ready data)

Databricks lakehouse

Includes the following technologies:

- **compute engine**: Apache Spark, a scalable engine decoupled from storage
- **optimized storage**: Delta lake, supports file-based transaction log for ACID transactions for tables. (Delta tables)

ACID

- **Atomicity**: each transactions is seen as a single unit which either succeeds or fails completely.
- **Consistency**: ensures the data is consistent, according with the rules defined. Prevents data corruption.
- **Isolation**: ensures that concurrent execution of transactions would not be affected by other transactions.
- **Durability**: commited changes are permanent

Why data lakehouses?

- **non proprietary**: data is stored using different open formats
- **simplicity**: data is indexed using different protocols used by AI, data science, other 3rd party applications
- **performance** low latency and high availability for BI reports and advanced analytics

Pipeline efficiency

How do you measure good or bad?

- performance metrics
- data quality output
- errors

Performance metrics

- throughput
- latency (wait time)
- response time (wait time + service time)
- busy time

Quality metrics

- do we have the right output?
- is the output the right format
- is data consistent?
- can we aggregate from this data?
- ensures that data remains consistent across different parts of the system or over time.