150+ Concepts Heard in Data Engineering

A Comprehensive Glossary For Data Engineers

Dardan Xhymshiti Jul 19, 2020 · 25 min read ★

Photo by Gustav Gullstrand on Unsplash

Data Engineering is an attractive field. *It starts with you, a single data file, and a personal computer and ends with mountains of data and the majestic "cloud".*

In this article, I've compiled a glossary of **over 150** concepts heard in the day-to-day data engineer's life. I've divided this glossary into several topics as data, schema, *data encoding, distributed systems, architecture, as a service, cloud, data store, cache, data structures, hardware, data security,* and *measures*.

Concepts are explained in enough number of lines for you to get the idea behind. Whenever I used other sources of information, I referenced that source at the end of the section. Please, consider these sources if you want to dig deeper. In did my best to select sources with the best explanation per concept.

This glossary can be beneficial in many aspects and some of them are:

- You want to learn about data engineering and you use this glossary as a roadmap.
- You are a data engineer and you want to refresh data engineering concepts.
- You are preparing for a data engineering interview and you want to make sure that no topic/concept escapes your attention.
- You are an interviewer and you use this glossary as a pool of questions.

Before going over the glossary, I have two things to clarify:

1. This glossary will surely get updated in the future. If there is something I missed including, **please write a response and I will make sure to include it**.
2. I've skipped explaining specific tools as — let's say — Python, Spark, or Kafka. These are part of another on-going article I am writing about top technologies heard in data engineering. (I'll include the link here as soon as that article is published).

Happy learning.

Data

> A collection of facts, such as numbers, words, measurements and observations for a phenomenon.

Raw Data: The unprocessed data in the original state. (ie. as it is delivered from the source).

Processed Data: Raw Data after getting processed. Processing happens with the main goal to make data easier to use than raw data.

Big Data: Describes the data that due to *volume, velocity, variety, veracity* cannot be processed with traditional data processing tools. Read more about different definition of Big Data in my article When someone asks you what Big Data is.

Data Quality: A process of assessing whether the data does serve the intended purpose for the specific use case. When assessing the quality of the data, we usually check for data *accuracy*, *completeness*, *reliability*, *relevance* and *timeliness*. For more, read this article from Rachel Levy Sarfin on 5 Characteristics of Data Quality.

Master Data: According to Gartner, "Master data is the consistent and uniform set of identifiers and extended attributes that describes the core entities of the enterprise including customers, prospects, citizens, suppliers, sites, hierarchies and chart of accounts." [1].

SSOT: (Single Source of Truth): An organization of data in a company where data is kept centrally and is accessed from outside from other components and people. If a change is made on the data elsewhere in the system, this change should reflect in the SSOT. For example, if a company keeps client contacts in a central database, if contacts change, there is only one place where they need to be updated (ie. SSOT).

Times Series Data: Data points collected together with a timestamp that indicates when data points are generated. Example include *monitoring data, sensor data, network data, user clicks data,* etc.

Cookie: According to Wikipedia, "… is a small size of data sent from a website and stored on the user's computer by the web browser while the user is browsing." [2].

Open Data: According to Wikipedia, "Open data is the idea that some data should be freely available to everyone to use and republish as they wish, without restrictions from copyright, patents or other mechanisms of control. [3].

Personal Data: According to European Comission, "Personal data is any information that relates to an identified or identifiable living individual. Different pieces of

information, which collected together can lead to the identification of a particular person, also constitute personal data.” [4].

———————

[1] Master data management. https://www.gartner.com/en/information-technology/glossary/master-data-management-mdm

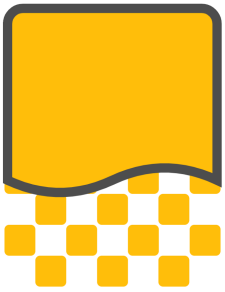
[2] HTTP cookie. https://en.wikipedia.org/wiki/HTTP_cookie

[3] Open data. https://en.wikipedia.org/wiki/Open_data

[4] Personal data: https://ec.europa.eu/info/law/law-topic/data-protection/reform/what-personal-data_en

Data Serialization

> “Data serialization (encoding, marshalling) is the process of converting structured data to a format that allows sharing or storage of the data in a form that allows recovery of its original structure.” [1] (The Hitchhiker’s Guide to Python)



Parquet: Encodes data in column-wise fashion. Data encoded in parquet is in binary format (hence not human-readable). Queries (especially aggregations) execute faster on top of data encoded in parquet. The size of the data decreases enormously because of the efficient column-wise compression. Compression is done column-by-column, hence different encoding types can be used for different columns.

Avro: Encodes data in row-wise fashion. Data encoded in Avro is in binary format (hence not human-readable). Specifies schema of the data being encoded in JSON and this schema definition is written together with the data in a file. A key feature of Avro is the robust support for schemas that change over time (schema evolution)[3].

JSON (JavaScript Object Notation): It's a widely used *key: value* encoding format of the data. JSON provides a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate.

YAML: (YAML Ain't Markup Language): It's a human-readable data-encoding format. YAML is commonly used for configuration files and excels with its minimalistic syntax.

XML (Extensible Markup Language): Is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. To perform the encoding, XML uses open and close tags. These tags are further enriched with attributes.

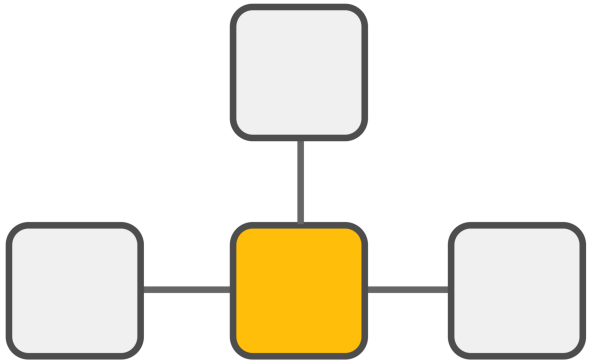
CSV (Comma Separated Values): An encoding format that stores values in rows split by a delimiter (most of the time comma, but could be a semicolon, tab, or something else). Each line of the file is a data record. A CSV file typically stores tabular data in which case each line will have the same number of fields. CSV files are human-readable.

———————

[1] Serialization. https://docs.python-guide.org/scenarios/serialization/

Schema

> “Describes the skeleton/structure of how the data is organized in a datastore (databases, files, …)”



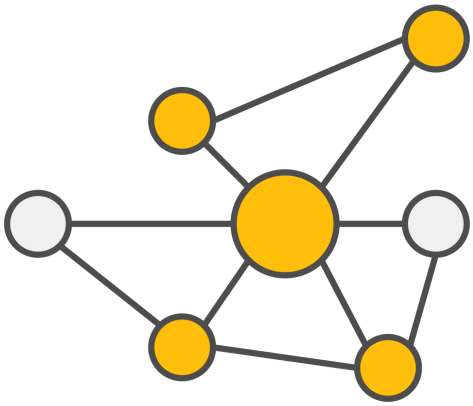
Schema-on-write: We define the schema before we deal with the data. For example, we cannot store and query data in MySQL without first defining schema (tables, column names, data types, …). A challenge arises when we need to change the schema due to — let’s say — a change in data type for a column. We need to drop the schema and load all the data again. It’s mostly seen in SQL databases.

Schema-on-read: We deal with the data before we define the schema. In this case, we store the data as soon as it arrives, without having to transform it to a strict format. The moment when you read the data is the moment when you infer the schema. It’s mostly seen in NoSQL databases. A good article to read is Schema-on-Read vs Schema-on-Write by luminousmen.

Schema evolution: Data evolves and the schema should mirror the changes made on the data. When evolving the schema, it is critical to make it compatible with the data you have, in order to not loose it. Schema evolution is necessary in case when a new column is added or a column’s data type needs to be changed.

Distributed Systems

> "... a distributed system is a collection of independent components located on different machines that share messages with each other in order to achieve common goals." [1] (Confluent)



Cluster: It's a set of connected machines (nodes) that work together to accomplish tasks. The connection between machines is done at the software level. From outside, the cluster appears as a single system.

Node: A single machine/server/virtual-machine that is part of a cluster.

Commodity Hardware: According to Suse, "Commodity hardware, sometimes known as off-the-shelf hardware, is a computer device or IT component that is relatively inexpensive, widely available and is easily interchangeable with other hardware of its type." [2].

Scalability: A quality of a distributed system that is able to cope with the load (requests, data,...) by adding or removing resources to the system.

Fault Tolerance: A quality of a distributed system that works continuously even though parts of it fail.

Robustness: A quality of a distributed system that performs well not only under expected situations but also under situations that stress designer's assumptions.

Rolling Upgrades: A strategy of upgrading software in a distributed system, when you deploy the updated software gradually to a few nodes at a time rather than deploying to all nodes simultaneously.

Vertical Scaling: Scaling by adding more resources as *CPU*, *RAM,* or *Disk* to a single machine.

Horizontal Scaling: Scaling by adding other machines to the existing pool of resources.

Replication: Replicate the same data at different machines that are connected via network. By replicating the data, we ensure fault tolerance of the system. When replicated data is distributed geographically, we reduce the latency or reads.

Partitioning: Splitting a big dataset into parts and distributing them among different nodes. Partitioning ensures scalability by distributing when queries work in different nodes.

Data Locality: Moving computation where the data resides instead of moving data where the computation resides. A practice used is Big Data systems in order to reduce the network traffic by moving computation around instead of the data.

Rebalancing partitions: The process of moving data from one cluster node to another. Usually done when data partitions increase in size unevenly or when nodes are added or removed from the cluster.

CAP Theorem: It states that a distributed system might have at most two of the following qualities simultaneously: *consistency*, *availability*, *partition tolerance*. A great article talking about this in detail is What is the CAP theorem?.

Read after write consistency: The ability to view changes (read the most recent data) right after writing those changes.

Eventual consistency: After making a write/update to the system, the system needs some time to update all the data points your write impacts throughout the nodes. If an immediate read request is made, there is a chance to receive old data.

Small Files Problem: In Hadoop, computing performance is significantly degraded when data is stored in many small files in HDFS.

_____

[1] Distributed systems. https://www.confluent.io/learn/distributed-systems/

[2] Commodity hardware. https://susedefines.suse.com/definition/commodity-hardware/

Architecture

> "... the overall design of a computing system and the logical and physical interrelationships between its components is defined as architecture. The architecture specifies the hardware, software, access methods and protocols used throughout the system." [1]. (Gartner)

Shared Memory Architecture: Many resources as CPUs, RAM chips and Disks are joined together to form a single machine. In this resource organization, any CPU has access to any part of the memory or disk. These machines are less fault tolerant and

scalability is achieved by increasing the amount of CPUs, RAM chips and disks (ie., vertical scaling).

Shared-Nothing Architecture: Commodity machines are linked together through a network and have independent access to their own machines. Scalability is achieved by adding other independent machines to the resource pool. Coordination between nodes is done at the software level.

Producer: A component that produces events (data). Consider for example a software component that generates an event every time an order is placed by a customer.

Consumer: A component that consumes events. Consider for example a software component that reacts to an order-placement event received, by sending a confirmation email to the user that did the purchase.

Event-Streaming Architecture: An architecture consisting of producers that produce stream of events and consumers that consume these events. Events are of different types like: product placed in cart, product ordered, order dispatched etc. These events are consumed and reacted upon from consumers components.

Batch Processing Architecture: An architecture consisting of one or more data stores (data lake, database, …) where the data is collected and jobs run periodically (eg. once per day) to process large chunks of the data (or all the data) to transform or perform analysis. The processed data is further stored in targeted systems in the desired format.

Lambda Architecture: Takes the advantage of both stream and batch processing advantages into a single architecture. The streaming part is responsible for live processing of the incoming data to get approximate results, and batch processing is responsible for processing the entire set of data gathered within an interval of time to get accurate results from the data.

Highly Available Architecture: An architecture that won't get negatively impacted by the load, errors, and external threats (natural disasters), such that its performance and availability is sacrificed. Such architecture contains components spread out geographically, that scale up and down depending on the incoming load, and replicas that step up whenever parts of systems fail.

Serverless Computing: According to Wikipedia "It's a cloud computing execution model in which the cloud provider runs the server, and dynamically manages the allocation of machine resources." [2].

Serverless Architecture: An architecture which components are entirely serverless components. Such architecture is less complex, takes less cost to run and less engineering time in developing and maintaining.

Incremental Architectures: An architecture that starts small, is modular, domain focused and consists of highly decoupled components.

ETL (Extract Transform Load): Describes the process of extracting data from multiple sources, transforming/processing them into a desirable format, and loading data into a final target as a database or data warehouse or data lake.

Real-time processing: When data is processed immediately after it becomes available. Example of this are critical systems as radars.

Near real-time: When data is processed with an allowed latency of seconds or minutes after it becomes available.

---

[1] Architecture. https://www.gartner.com/en/information-technology/glossary/architecture

[2] Serverless computing. https://en.wikipedia.org/wiki/Serverless_computing

As a Service

> "A business model that describes a product (software, infrastructure, ...) that is maintained and run at the owner side, and is provided to customers for an

> amount of money paid either in subscription-basis or pay-as-you-go”

IaaS (Infrastructure as a service): Provides the computing infrastructure (virtual machines, physical machines), object storages, file systems, load balancers, networking, etc. Example of IaaS are: *Amazon EC2*, *Windows Azure*, *Google Compute Engine,* etc.

PaaS (Platform as a service): According to Microsoft “… PaaS is a complete development and deployment environment in the cloud, with resources that enable you to deliver everything from simple cloud-based apps to sophisticated, cloud-enabled enterprise applications.” [1]. Examples of PaaS are *AWS Elastic Beanstalk*, *Windows Azure*, *Heroku*, *Google App Engine*, etc.

SaaS (Software as a Service): A software developed, run, and maintained from a company that licenses it on a subscription basis to customers. Example of SaaS are: *Google Apps*, *Dropbox* and *Salesforce*.

FaaS (Functions as a Service): This is also known as serverless computing. This enables developers to write only the business logic and execute it without dealing with resource provisioning and administration.

BaaS (Backend as a Service): According to Cloudflare “BaaS is a cloud service model in which developers outsource all the behind-the-scenes aspects of a web or mobile application so that they only have to write and maintain the frontend.” [2].

---

[1] What is PaaS. https://azure.microsoft.com/en-us/overview/what-is-paas/

[2] Backand as a service.
https://www.cloudflare.com/learning/serverless/glossary/backend-as-a-service-baas/

Cloud

> "... refers to servers that are accessed over the Internet, and the software and databases that run on those servers." [1]. (Cloudflare)

Cloud Computing: According to Microsoft "Cloud computing is the delivery of computing services — including servers, storage, databases, networking, software, analytics, and intelligence — over the Internet." [2].

Virtualisation: According to RedHat "… virtualization is a technology that allows you to create multiple simulated environments or dedicated resources from a single, physical hardware system." [3].

Hypervisor: According to Vmware "A hypervisor, also known as a virtual machine monitor or VMM, is software that creates and runs virtual machines (VMs). A hypervisor allows one host computer to support multiple guest VMs by virtually sharing its resources, such as memory and processing." [4].

AWS (Amazon Web Services): A cloud platform offering a lot of services, among which the following ones used mostly in data engineering: *S3* (Simple Storage Service),

EMR (Elastic Map Reduce), *Kinesis* (*Streams*, *Analytics*, *Firehose*), *Data Pipeline*, *Redshift*, *Glue*, *DynamoDB*, *Athena*, *RDS*, *Quicksight*, *Lake Formation*, *EC2*, *Lambdas*.

Microsoft Azure: A cloud platform offering a lot of services, among which the following ones used mostly in data engineering: *Blob Storage*, *Azure Databricks*, *Azure Stream Analytics*, *HDInsight*, *Azure Cosmos DB*, *Event Hubs*, *Azure Data Lake Storage*, *Data Catalog*.

GCP (Google Cloud Platform): A cloud platform offering a lot of services, among which the following ones used mostly in data engineering: *Cloud Storage*, *Compute Engine*, *Big Query*, *DataFlow*, *Cloud Functions*, *Cloud SQL*, *Cloud Bigtable*.

Pay-as-you-go: A model of payment used by cloud providers where you pay only for the individual services you need, for as long as you use them, and without requiring long-term contracts or complex licensing. You only pay for the services you consume, and once you stop using them, there are no additional costs or termination fees.

On-Demand Services: A feature of cloud computing services, which allow users to provision cloud resources whenever and wherever needed.

Lambda Functions: A function in AWS Lambda where you run code without provisioning or managing servers. AWS Lambda executes your code only when needed (in response to events, or when triggered through API calls) and scales automatically. In similar you have Azure Functions in Microsoft Azure and Cloud Functions in Google Cloud Platform.

Cloud Migration: It is the process of transferring databases, applications, and IT processes from on-premise to the cloud or from one cloud to another.

———————

[1] What is the cloud. https://www.cloudflare.com/learning/cloud/what-is-the-cloud/

[2] What is cloud computing. https://azure.microsoft.com/en-us/overview/what-is-cloud-computing/

[3] Cloud vs virtualization. https://www.redhat.com/en/topics/cloud-computing/cloud-vs-virtualization

[4] Hypervisor. https://www.vmware.com/topics/glossary/content/hypervisor

Data Store

> "A data store is a repository for persistently storing and managing collections of data which include not just repositories like databases, but also simpler store types such as files." [1]. (Wikipedia)

Database: According to Wikipedia *"A database is an organized collection of data, generally stored and accessed electronically from a computer system."* [2].

Relational Databases (SQL): Databases where data is organized in entities (tables) with attributes (columns) where relations represent the connection (relationship) among the tables. Some well-known relational databases are: *MySQL, SQL Server, PostgreSQL, Oracle.*

Table: A table in a relational database represent a unique entity which structures related data in a tabular format.

Attribute (column, field, feature, characteristic): It represents a single column of data in a table. For example, table "*Car"* contains "*brand"* as one of its attributes.

Record (tuple, data point): A row in a table.

Query: A custom request made to the database for retrieving the necessary data. A result might be composed from a single table or multiple table's data.

RDBMS (Relational Database Management System): A software system used to maintain relational databases.

Primary Key: Composition of one or more columns to generate an identifier for every record in a table. The primary key represents a unique value per row. For instance, column "*name"* is not a primary key but "*passport_id"* is.

Normalization: According to Wikipedia "*Database normalization is the process of structuring a relational database following a series of so-called normal forms 1NF, 2NF, 3NF and BCNF, to reduce data redundancy and improve data integrity."* [3]. Data integrity is achieved by ensuring that no anomalies will happen in case of insertions, updates and deletes.

Transaction: A transaction is a logical unit of work, containing one or multiple SQL statements, that performs a defined operation in a database.

#**ACID:** Describes a set of guarantees as *Atomicity*, *Consistency*, *Isolation*, and *Durability* that a database transactions makes to ensure validity even in the event of errors, power failures, etc.

Index: When a query is performed in a database, the necessary data need to get located rapidly. Indexing helps with this. When you index a column, a data structure will be created. This data structure will hold column values and the pointers where the data is stored in the disk. This data structure is mainly a B-Tree, but it can be anything. When you perform a query in an indexed column, the necessary records are quickly located and you escape the worst-case scenario of scanning the entire rows.

Hash Function: It is an algorithm that converts a given value with arbitrary length to another fixed-size value. The output of a hash function is called hash value. The most

known algorithms for hashing are MD5 (Message Digest 5) and SHA (Secure Hashing Algorithm) 1 and 2.

Non Relational Databases (NoSQL): All other databases that organize data by any other format rather than tabular fall under this category. Here you can find databases that organize data in a *document, columnar, key-value,* and *graph format*.

Key: Value Databases: Stores data in a "*key: value"* format — also known as *dictionary* or *hashtable*. Values are stored and retrieved using a *key*. Each record might have a different number and different types of fields. Other databases, for example, document ones are built on top of "*key: value"* databases. Examples are *Redis*, *Couchbase*, *Hazlecast*, *Memcached*, *ArangoDB* etc.

Document-Oriented Databases: Document is what a row is in relational database. These databases store data in "*key: value"* format, where values are encoded either in JSON, XML, or YAML or binary formats as BSON. Documents can have a flexible number of filed and data types. Data is retrieved by querying the content of documents. Examples are *MongoDB*, *Couchbase*, *CouchDB*, *RethinkDB*, *Elasticsearch*, *ArangoDB*, etc.

Graph Databases: Organize data in graph format where vertices (nodes) represent an entity as *a person* and edges that represent the relationship between two vertices. Used to store data that fits this organization model as social networking data, public transport, network topologies, etc. Examples are *AllegroGraph*, *ArangoDB*, *Apache Giraph*, *Neo4J*, *OrientDB*, etc.

Column-oriented Databases: Organize data in columns instead of rows. These databases are very efficient in analytical query processing. Examples are *Amazon DynamoDB*, *Bigtable*, *Cassandra*, Scylla, *HBase*, *Hypertable,* etc.

In-Memory Databases: A database that relies primarily on memory for data storage, in contrast to databases that store data on disk. These databases are useful in circumstances where response time is critical as in gaming and real time analysis. Examples are Redis, Hazelcast, H2, Memcached, etc.

Time Series Databases: A database that is optimized to work with time series data. These databases comes with a number of features facilitating your work with time series data. Examples are *InfluxDB, Prometheus, TimescaleDB, Druid ,* etc.

Collection: What a table is for relational databases, collections is for non-relational databases. A collection can store documents even if they do have different structure.

Document: What a row is in a relational database, a document is in a non-relational database.

Data Warehouse: It's a central repository where data from different sources is gathered. Moreover a data warehouse provides functionality for analytics and reporting for BI (business intelligence). Usually a data warehouse keeps terabytes or petabytes of structured data. Examples of data warehouse solutions are *Oracle*, *Redshift*, *MarkLogic*, etc.

Data Lake: Data lake is simply a storage used to store raw data of various formats from various data sources. Examples of data lakes are *AWS S3*, *Azure Blob, Google Cloud Storage*, etc.

OLAP (Online Analytical Processing): According to IBM, "A core component of data warehousing implementations, OLAP enables fast, flexible multidimensional data analysis for business intelligence (BI) and decision support applications." [4]. In OLAP, it's usually the historical data — let' say of last 5 years — that is queried. An example of such processing is "Company X compares sales of the first quarter versus the second quarter for 2020 in Location A ".

OLTP (Online Transaction Processing): Describes other processing done outside of data warehouses. For example queries made to insert, delete, update small quantities of data in databases fall under this category. OLTP is involved in daily operations as in e-commerce online banking, POS terminals, etc.

Dirty Reads: When transaction B reads data from database that was written from transaction A and A has not yet committed or aborted.

Dirty Writes: When transaction B overwrites data written by transaction A and A has not yet committed or aborted.

File Storage: According to IBM, "File storage is a hierarchical storage methodology used to organize and store data on a computer hard drive or on Network-Attached Storage (NAS) device. In file storage, data is stored in files, the files are organized in

folders, and the folders are organized under a hierarchy of directories and subdirectories."[5].

Block Storage: Data is chopped in blocks and then stored in separate pieces throughout the system. Each block has a unique identifier assigned. To access a file, the server's operating system uses the unique address to pull the blocks back together into the file, which takes less time than navigating through directories and file hierarchies to access a file. Examples of block storage are *SAN*, *iSCSI*, and *local disks*.

Object Storage: According to IBM, "Objects are discrete units of data that are stored in a structurally flat data environment."[6]. Every object is a simple, self-contained repository that includes the data, metadata (descriptive information associated with an object), and a unique identifying ID number. Examples of this storage are *AWS S3*, *Azure Blob Storage*, and *Google cloud storage*.

Backup: A copy of your data that you will reuse in case your production data is destroyed.

———————

[1] Data store. https://en.wikipedia.org/wiki/Data_store

[2] Database. https://en.wikipedia.org/wiki/Database

[3] Database normalization. https://en.wikipedia.org/wiki/Database_normalization

[4] OLAP. https://www.ibm.com/cloud/learn/olap

[5] Object storage. https://www.ibm.com/cloud/learn/object-storage

Cache

> "A cache is a hardware or software component that stores data so that future requests for that data can

> be served faster…”[1]. (Wikipedia)

Cache hit: Describes the event when the requested data is found in a cache.

Cache miss: Describes the event when the requested is not found in cache.

TOL (Time-to-Live): A defined interval of time for which an object can live in cache. When TOL is expired, object is either deleted or refreshed.

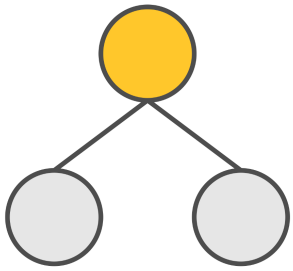
Eviction policy: A policy that decides which element to drop out when the cache is full. Examples of evicting policies are *LRU* (Least Recently Used), *FIFO* (First In First Out), *RR* (Random Replacement), etc.

---

[1] Cache. https://en.wikipedia.org/wiki/Cache_(computing)

Data Structures

> A data structure organizes data in a certain format (list, map, queue,…) and provides functionality to access and modify the data”.



Queue: A linear data structure where items are processed in FIFO (First-In-First-Out) order. Items are always appended at the end of the queue and consumed from the beginning of the queue.

Stack: A linear data structure where items are processed in LIFO (Last-In-First-Out) order. Items are always appended at the beginning and consumed from the beginning.

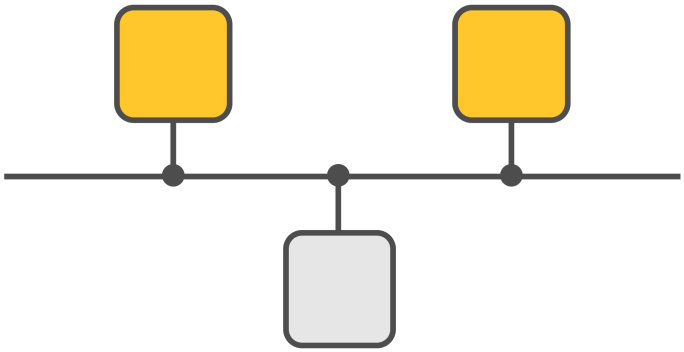
B-Tree: According to CalPonyPomona, "A B-tree is a tree data structure that keeps data sorted and allows searches, insertions, and deletions in logarithmic amortized time …optimized for systems that read and write large blocks of data. It is most commonly used in database and file systems." [1].

———————

[1] B-tree. https://www.cpp.edu/~ftang/courses/CS241/notes/b-tree.htm

Network

> Two or more computers linked together to share resources.



 # IP address (Internet Protocol Address): According to Wikipedia, "An IP address is a numerical label assigned to each device connected to a computer network that uses the Internet Protocol for communication." [1]. An example of an IP address is `192.0.2.1`.

 # TCP/IP (Transmission Control Protocol/Internet Protocol): Is a family of network protocols regulating the way how data is transferred in a network. This family contains protocols as *IP (Internet Protocol), TCP (Transmission Control Protocol), UDP (User Datagram Protocol)* and *ICMP (Internet Control Message Protocol)*.

 # UDP (User Datagram Protocol): A protocol with a lightweight mechanisms for sending data to other hosts on a network. It doesn't provide handshaking, guarantee for the delivery and ordering of messages.

 # OSI (Open Systems Interconnection): Is a conceptual framework that describes how a networking system works. Description is done by dividing the network system functionality in 7 parts: *Physical*, *Data Link*, *Network*, *Transport*, *Session*, *Presentation*, and *Application*.

 # SSH (Secure Shell): Is a method for secure remote login from one computer to another.

 # Handshaking: According to Wikipedia, "A handshake is an automated process of negotiation between two participants … through the exchange of information that establishes the protocols of a communication link at the start of the communication, before full communication begins." [2].

FTP (File TransFer Protocol): A standard protocol used by machines connected in a network, to transfer files between each other.

Certificate: According to computerhope, "A certificate or digital certificate is a unique, digitally signed document which authoritatively identifies the identity of an individual or organization. Using public key cryptography, its authenticity can be verified to ensure that the software or website you are using is legitimate." [3].

Proxy Server: According to whatismyip, "A proxy server is basically another computer which serves as a hub through which internet requests are processed." [4].

Ping: Ping is a tool used to test whether an ip address is reachable or not. Ping is accessed through terminal via the command `ping <ip-address>`.

Heartbeat: According to Wikipedia, "… a heartbeat is a periodic signal generated by hardware or software to indicate normal operation or to synchronize other parts of a computer system." [5].

NAS (Network Attached Storage): It's a single file storage server connected to a network providing data access to a group of clients. The main disadvantage of NAS is that it is not fault tolerant. If the server fails machines cannot access data anymore.

SAN (Storage Area Network): A high speed network of connected storages that store and provide access to large amount of data. It is a network of storages (multiple ones). Data is shared and replicated among different storages in the network, thus making SAN fault tolerant.

---

[1] IP Address. https://en.wikipedia.org/wiki/Internet_Protocol

[2] Handshaking. https://en.wikipedia.org/wiki/Handshaking

[3] Certificate. https://www.computerhope.com/jargon/c/certific.htm

[4] What is a proxy. https://www.whatismyip.com/what-is-a-proxy/

[5] Heartbeat. https://en.wikipedia.org/wiki/Heartbeat

Data Security

> A set of practices designed to protect digital data from unauthorized access and from physical disasters.

 # Public Key Cryptography: According to Globalsign, "Public-key cryptography, or asymmetric cryptography, is an encryption scheme that uses two mathematically related, but not identical, keys — a public key and a private key… The public key is used to encrypt and the private key is used to decrypt." [1].

 # Public Key: In a cryptosystem, the public key is used to encrypt the data. The public key can be freely shared with others. It is infeasible to compute the private key from the public key.

 # Private Key: Private key is used to decrypt data. Private key is kept secret at at the owner (ie. not shared with anyone else). Data encrypted with a specific public key *K123-public* can only be decrypted with exact associated private key *K123-private*.

 # GDPR (General Data Protection Regulation): A regulation designed by European Union and which came into force at May 25, 2018. This regulation ensures EU citizens more control over how their personal data is used. To read more about GDPR visit the official website of European Union.

Data Anonymization: According to Wikipedia, "Data anonymization is the process of removing personally identifiable information from data sets, so that the people whom the data describe remain anonymous." [2]. Data can either be removed or replaced with gibberish.

Data Breach (Data Leak): A data breach is the intentional or unintentional release of private (confidential) information to public.

Terms of Service: Terms of service are the legal agreements between a service provider and a person who wants to use that service.

Ransomware: According to Kaspersky, "Ransomware is a malicious software that infects your computer and displays messages demanding a fee to be paid in order for your system to work again… It has the ability to lock a computer screen or encrypt important, predetermined files with a password." [3].

SHA-2: A set of six cryptographic hash functions named SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224 and SHA-512/256. Numbers represent bits used per hash value.

RSA (Rivest-Shamir-Adleman): A widely used public-key cryptosystem. In a cryptosystem a public key is used to encrypt the data and a private key is used to decrypt the data.

———————

[1] What is public key cryptography. https://www.globalsign.com/en/ssl-information-center/what-is-public-key-cryptography

[2] Data anonymization. https://en.wikipedia.org/wiki/Data_anonymization

[3] Ransom ware. https://www.kaspersky.com/resource-center/definitions/what-is-ransomware

Measures

Bit: A basic unit of digital information that takes binary values of 1 or 0s.

Byte: A block of bits that usually contain 8 bits.

Terabyte: Measure the size of the data where 1 TB = 1024GB.

Petabyte: Measures the size of the data 1PB = 1024TB.

One order of magnitude: According to Study, "An order of magnitude is usually written as 10 to the nth power. The n represents the order of magnitude. If you raise a number by one order of magnitude, you are basically multiplying that number by 10. If you decrease a number by one order of magnitude, you are basically multiplying that number by 0.1." [1].

IOPS (Input-Output Operations Per Second): measures the input output performance of storage systems.

Throughput: It measures the units of information a system processes in a given amount of time. For example "*this data pipeline processes 25GB of data per minute*".

Latency: The time a request/response takes to get transmitted from sender to receiver.

Response time: The overall time, from the moment when sender submits the request till it receives the response.

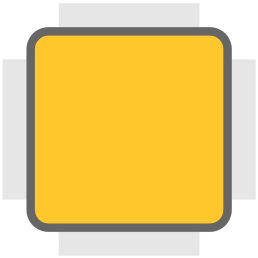
Bandwidth: Describes the maximum amount of data transmitted over an internet connection in a given amount of time.

---

[1] Order of magnitude definition examples. https://study.com/academy/lesson/order-of-magnitude-definition-examples.html

Hardware

Describes the physical parts of a computer.



 # HDD (Hard Disk Drive): According to wikipedia "A Hard Disk Drive is an electro-mechanical data storage device that uses magnetic storage to store and retrieve digital data using one or more rigid rapidly rotating platters coated with magnetic material." [1]. HDDs are cheap mechanical devices that still are widely used in storage systems.

 # SDD (Solid State Drive): According to wikipedia "A Solid State Drive is a solid-state storage device that uses integrated circuit assemblies to store data persistently, typically using flash memory …" [2]. SSDs are way more expensive than HDDs and mainly used in situations where speed of accessing the data is critical.

 # RAM (Random Access Memory): Also known as primary storage, RAM is volatile, short-term fast memory, which loads data from secondary storage (ie. HDD or SSD) for faster access from CPU. Random accessing of data in RAM has equal latency.

---

[1] Hard Disk Drive. https://en.wikipedia.org/wiki/Hard_disk_drive

[2] Solid State Drive. https://en.wikipedia.org/wiki/Solid-state_drive

Conclusion

This glossary is compiled based on several sources. Sources are mainly taken from internet web links such as Wikipedia, IBM, amazon, azure, redhat, etc.
Concepts that include a link in them, are taken from the source specified in that link.

If you want to dig deeper on the concepts, first consider the references provided. Next, I highly recommend to go through the below books if you want to make sense out of the concepts presented here.

- Designing-Data-Intensive Applications by Martin Kleppman (a must for everyone considering data engineering seriously),
- Big DataPrinciples and best practices of scalable realtime data systems by Nathan Marz and James Warren
- Distributed Systems: Principles and Paradigm by Andrew Tanenbaum

> *Finally, this article is prone to updates. I will be updating it with new concepts as they come up. Don't hesitate to write me for concepts that missed including.*

Data Engineering Data Science Data Programming Careers