The table defines the meaning of basic terms which are widely used within PostgreSQL's documentation and related publications. The purpose is the specification of an unambiguous terminology. Therefore it focuses on those terms, which sound similar but have a distinct meaning, and on those terms, which are different but have the same meaning. Hence it serves as a guideline for beginners as well as for authors of our documentation.

Additionally it addresses a problem within the global database community. People use basic technical terms - mostly - in a consistent and database independent way. But there are exceptions. Some of the terms listed here have a divergent meaning in other database systems. We want to make clear, which understanding the PostgreSQL community has concerning those terms.

Table xyz: Definition and meaning of widely used terms

Part 1: Fundamental Terms		
Term or **Accepted Alternative Term** (Outdated Term)	**Definition**	**Notes**
Server or **Node** (Database Server)	A server consists of real or virtual hardware where the PostgreSQL software is installed and running.	Don't confuse the term with instance. It is possible to run multiple instances on a single server. The term node is also used in the context of data structures like trees. Considering the context the meaning should become clear.

Database Cluster (Cluster)	A database cluster is a group of 3 or more databases (*template0*, *template1*, *postgres*, ...) plus their meta information. Its control- and runtime-information is stored in files which reside - in analogy to the situation of databases - in one directory and its subdirectories. This directory is often referred to as PGDATA. The database cluster is controlled and managed by one instance.	Depending on the OS the database cluster‘s default name is “data” or “main”, which is identical to the name of the directory at OS level. It is possible to create more databases in the database cluster, e.g. “MyDb”. One can create new objects, eg: “MyTable”, within the default database “postgres” as well as in new databases. The utility initdb creates a new database cluster (with 3 databases), **not** a new database within an existing database cluster. Don’t confuse the term with the term Replication Cluster. The SQL key word CLUSTER has the meaning ”arrange the sequence of rows within a table”.
Database	A database is a collection of files. With the possible exception of tablespaces (and targets of links) they reside in PGDATA/base/<oid>/. The files hold the databases data (heap, indexes), its meta data (definitions, views, ...), and system data (functions, operators, ...) of this database.	<oid> is the object id of the database within the database cluster’s meta data.
Segment	The term “segment” is a synonym for “file”. In the case of WAL files its size is 16MB (fix) and in the case of heap or index the size varies and grows up to 1 GB - the instance generates more files if heap or index grows above this limit.	
Instance (Server)	An instance is a group of processes (on a UNIX server) respectively one service (on a Windows server) plus shared buffers, which controls and manages exactly one database cluster.	Please consider the difference between the terms instance (processes, RAM) and database (files).
Schema	A schema is a logical cramp for a collection of objects like tables or functions within one database. (An analogy is a directory within a file system - but “schema” has no recursive nature.)	Every database contains the special schema “public”. It holds the names of all system tables, data types, functions, and operators of this database. It is possible to define more tables etc. within the schema “public”.

Catalog	In the case of the two schemas “information_schema” and “pg_catalog” the term “schema” is replaced by the term “catalog”.	The catalog (respectively schema) “information_schema” is required by the SQL standard and holds information about the database structure in a standardized form. The catalog (respectively schema) “pg_catalog” holds information about the database structure in a PostgreSQL specific form.
Summary: The hierarchy to any database object is: server.database_cluster.database.schema.object, eg: apollo.main.postgres.public.employee		
Heap	The heap contains the original data (values). He is realized within database files and mirrored in shared buffers.	
Index	The index contains data (as keys) and pointers to the heap. He is realized within database files and mirrored in shared buffers.	

Part 2: Fundamental Software Components

Shared Buffers	Shared buffers are RAM where parts of the database (files) are mirrored by the instance. Its different processes as well as the backend processes have concurrent access to the shared buffers. Shared buffers are constructed within the operating system’s shared memory.	
Shared Memory	RAM which may be accessed by multiple processes. The operating system uses semaphores to grant access to shared memory.	“Shared Memory” is an operating system term, especially in the UNIX family.

Semaphore	Semaphores are provided by the operating system to offer applications undividable (atomic) operations. In conjunction with shared memory they are necessary to implement inter-process communication (IPC).	
Page	A page is a contiguous block of RAM (usually 8 kB) with a PostgreSQL specific layout. This layout is identical in RAM and in files. I/O activities of the instance transfers entire pages between RAM and files.	The operating system page size, the file system block size and PostgreSQL's page size may differ from each other.
Dirty Page	A dirty page is a page within the shared buffers, which is modified since last read from disk but not yet flushed back to disk.	
Query Optimizer or **Query Planner** **(**A suggestion to unite the two terms: **optimizing query planner)**	An algorithm which creates an optimal execution plan for a given SQL statement under consideration of the results of last ANALYZE.	
Execution Plan	An execution plan is the translation of an SQL statement into a tree of nodes. Each single node describes the low level access path to data and the estimated use of resources in terms of time, I/O and RAM.	
Postmaster	Within an instance there is exactly one postmaster process. Every new connection request from a frontend process to the instance reaches this process. As a reaction to such a request, the postmaster creates a new backend process and passes the connection and all further responsibilities to him.	
Backend Process (server process)	There is one backend process per connection. He receives requests from the assigned frontend process, executes them, and sends the result back to him.	Every backend process is assigned to exactly one fronted process.

Frontend Process or **Client Process**	Frontend processes initiate connections, send requests to the assigned backend process and receive the results.	It is possible that one frontend process communicates with multiple backend processes via different connections. Frontend and backend processes, which are assigned to each other, may run on different or the same server.

Part 3: Multiversion Concurrency Control (MVCC)

Version of a Row	The data values of a single entity are stored in a contiguous memory or disc area. A later data modification does not overwrite these values but copies the complete area to a different area - by taking the modifications into account. Each one of the two (or more) areas are called a “version of a row”.	
Row	A row is that single version of the row, which is - from the perspective of the transaction - the newest one. From the perspective of the complete database and some other transactions there may be newer versions of the same row.	

Part 4: Write Ahead Logging (WAL)

WAL file or **WAL** (transaction logfile XLog file log segment file WAL segment file segment segment file)	A WAL file is a file in binary format where the log records of the complete database cluster are stored. Multiple WAL files as well as log records within a single WAL file have a certain order which represents the sequence of changes taken place in the database cluster.	The divergent term “logfile” denotes a text file where warn- and error-messages are reported.
Log record (Log entry)	A log record contains information about a single data change. He is stored in a WAL file. The log entry contains not only the data itself but also information about the physical storage location in a file, information about indexes, and VACUUM information.	Log records are the atoms of WAL files as well as of all kind of replication (directly or after a logical decoding). The term “log record” is derived from the verb “to log” - it’s not an abbreviation for “logical”.
LSN: Log Sequence Number	A LSN is a 64-bit integer representing a byte position within the sequence of WAL files.	
Logical Decoding	Logical decoding is a feature to offer consumers a stream of the ongoing data modifications. It is realized by decoding the continuously arising log records.	
Checkpoint	A checkpoint is a point in time when all dirty pages (heap plus index) of the shared buffers are flushed to disk and a special checkpoint record is written to the WAL file.	
Snapshot	A consistent (logical) view to the data in the database cluster, which is shown to a transaction as of its start-time, no matter what other transactions are doing while this transaction runs. So, every transaction has its own snapshot.	
Savepoint	A savepoint is a special mark (a timestamp) inside a transaction. Data modification actions arising after this point in time may be rolled back to the time of the savepoint.	

Part 5: Backup and Recovery

Cold Backup or **File System Level Backup**	A cold backup takes all files of the database cluster with pure operating system commands and without any support of PostgreSQL's tools.	To get a useable backup with this method the instance MUST be shut down!
Hot Backup	A hot backup is a backup which is taken with Postgr-eSQL tools during the instance is running.	
Logical Backup	A logical backup is a hot backup which creates new files in SQL syntax or in a binary format. Those files are independent from the database clusters directory. A logical backup takes a complete database or database cluster or parts of it.	A logical backup is taken with one of the tools pg_dump or pg_dumpall.
Base Backup	A base backup is a hot backup which copies all files of a database cluster (including WAL files) and creates an additional backup history file. It is not possible to take parts of the database cluster - it must be processed as a whole.	You need the pg_start/stop_backup command or the pg_base-backup command to create a base backup.
Continuous Archiving or **Log Shipping**	The term denotes the process of copying all WAL files, which are continuously created by the instance, to an additional and reliable storage medium - eg: for archiv-ing, replication, or backup purposes.	You can use archived WAL files during a recovery (but you need more than just this files - namely the files of a base backup). The term "log shipping" is used in replication scenarios.
PITR: Point in Time Recovery	A recovery process can stop at a choosable timestamp which lays between the time of taking the base backup and the time of the last COMMIT before a crash.	To use this feature it is necessary to set certain parameters for the generation of the WAL files,

Backup History File	A backup history file contains meta information and timestamps about the base backup.	A backup history file is created automatically during the generation of a base backup.

Part 6: Replication

Replicating Cluster (Cluster)	A replicating cluster is a set (transitive closure) of instances which interact in any kind of replication.	Don’t confuse the term with Database Cluster.
Master Instance (Master server Primary server)	A master instance is an instance which pushes log records to one or more standby instances (in the case of streaming replication) or sends WAL files to them (in the case of log shipping replication).	
Standby Instance (Standby server Slave server Secondary server)	A standby instance is an instance which receives and processes log records or WAL files with the aim that the standby instance contains the same data as their master instance - after a more or less small time gap.	
Warm Standby Instance (Warm Standby Server)	A standby instance is called a “warm standby instance” if it does not accept client connections.	
Hot Standby Instance (Hot Standby Server)	A standby instance is called “hot standby instance” if it accepts client connections in read-only mode.	
Log Shipping Replication **(26.2.)**	The data transfer from the master instance to the standby instance is realized by the transfer of WAL files.	In a non-replication context transfers like this are called continuous archiving or simply log shipping.

Streaming Replication (26.2.5.)	The data transfer from the master instance to the standby instance is realized by log records which are streamed over an IP connection. ??? relation to logical decoding ???	
Logical Decoding	See above.	
Replication Slot (47.2.2.)	?	
Physical replication Slot	?	
Streaming Replication Slot (26.2.5./26.2.6.)	?	
Logical Replication Slot or **Logical Slot (47.2.2.)**	?	