CONTENTS

CONFIGURATION FILES

1. POSTGRESQL.CONF

Purpose: The main configuration file for PostgreSQL, controlling server behavior and database settings.

Location: By default, located in the PostgreSQL data directory, typically specified during installation.

Contents:

General parameters for database operations, including memory management, logging, query optimization, and replication settings.

Configurable categories include File Locations, Resource Usage, Query Tuning, Write-Ahead Logging (WAL), Replication, Autovacuum, Client Connection Defaults, and more.

Description from file:

- This file is read on server startup and when the server receives a SIGHUP signal.
- If you edit the file on a running system, you have to SIGHUP the server for the changes to take effect, run "pg_ctl reload –D path/to/data directory", or execute "SELECT pg_reload_conf()".
- Some parameters, which are marked below, require a server shutdown and restart to take effect.
- Any parameter can also be given as a command-line option to the server, e.g., "postgres -c log_connections=on".
- Some parameters can be changed at run time with the "SET" SQL command.

Key Usage: The settings here influence the overall performance and functionality of the database. Adjustments can be made to optimize resource usage, configure connections, manage authentication, and tune queries.

IMPORTANT CONFIGURATION:

1. File Locations:

Parameters that define the storage locations for PostgreSQL's data, logs, and other key files.

Parameter	*Default Value*	*Description*
data_directory	Environment set	Path to the PostgreSQL data directory.
hba_file	pg_hba.conf	Path to the host-based authentication (HBA) file.
ident_file	pg_ident.conf	Path to the file mapping OS usernames to database usernames.
external_pid_file	None	Path to store the server's process ID file for easier management.

2. Connections and Authentication

Configures client connections, authentication methods, and connection limits.

Parameter	*Default Value*	*Description*
listen_addresses	Localhost or "*"	IP addresses that PostgreSQL listens to for connections.
port	5432	Port number PostgreSQL listens on.
max_connections	100	Maximum concurrent connections.
authentication_timeout	1min	Timeout for client authentication.
ssl	off	Enables SSL connections for secure communication.
password_encryption	scram-sha-256	Specifies method for storing encrypted passwords.

3. Resource Usage (except WAL)

Defines how memory, CPU, and disk resources are allocated for optimal performance.

Parameter	*Default Value*	*Description*
shared_buffers	128MB	Amount of memory used for shared data across connections.
work_mem	4MB	Memory allocated per query operation, like sorting.
maintenance_work_mem	64MB	Memory used for maintenance operations (e.g., VACUUM, indexing).
effective_cache_size	4GB	Estimate of available OS cache memory for the planner.
temp_buffers	8MB	Memory allocated for temporary tables for each session.
max_files_per_process	1000	Maximum open files per process; helps manage resource limits on the OS level.

4. Write-Ahead Log (WAL)

Parameters that manage PostgreSQL's WAL for data durability and crash recovery.

Parameter	*Default Value*	*Description*
wal_level	replica	Sets level of information in WAL for backup and replication.
wal_buffers	-1 (auto-tuned)	Memory for WAL records; tuning improves write-heavy workload performance.
checkpoint_timeout	5min	Time between automatic checkpoints.
max_wal_size	1GB	Maximum total size of WAL files before a checkpoint is forced.
min_wal_size	80MB	Minimum size of WAL files retained.

5. Replication

Manages replication settings to create high availability and standby environments.

Parameter	*Default Value*	*Description*
max_wal_senders	10	Maximum number of replication sender processes.
synchronous_commit	on	Forces transactions to wait for WAL write confirmation.
hot_standby	off	Enables read-only queries on standby servers.
max_standby_streaming_delay	30s	Delay tolerance for read queries on standby.
synchronous_standby_names	''	Defines standby servers that must acknowledge receipt of WAL.

6. Query Tuning

Parameters that guide the query planner and executor to optimize performance.

Parameter	*Default Value*	*Description*
effective_cache_size	4GB	Guides query planner on available memory; larger values can improve performance.
random_page_cost	4.0	Cost factor for non-sequential page access in disk storage.
seq_page_cost	1.0	Cost factor for sequential disk scans.
cpu_tuple_cost	0.01	Cost of processing a row in a query.
join_collapse_limit	8	Limits number of tables planner joins together automatically.

7. Reporting and Logging

Configures logging to monitor and troubleshoot PostgreSQL operations.

Parameter	*Default Value*	*Description*
log_destination	stderr	Destination for log output (stderr, csvlog, syslog).
logging_collector	off	Enables log file generation.
log_directory	pg_log	Directory for log storage.
log_filename	postgresql-%Y-%m-%d_%H%M%S.log	Filename format for log files.
log_min_duration_statement	-1 (disabled)	Logs SQL statements exceeding the set time in ms.
log_statement	none	Logs each SQL statement type (none, ddl, mod, all).

8. Statistics

Tracks query and usage statistics, crucial for performance tuning.

Parameter	*Default Value*	*Description*
track_activities	on	Enables tracking of active sessions.
track_counts	on	Collects database statistics for performance analysis.
track_io_timing	off	Enables IO timing statistics, useful for tuning queries.
stats_temp_directory	pg_stat_tmp	Directory for temporary statistics files.

9. Autovacuum

Automates database maintenance tasks like table vacuuming and analyzing.

Parameter	*Default Value*	*Description*
autovacuum	on	Enables autovacuum feature.
autovacuum_max_workers	3	Maximum number of autovacuum processes.
autovacuum_naptime	1min	Delay between autovacuum cycles.
autovacuum_vacuum_scale_factor	0.2	Fraction of table size to trigger autovacuum.
autovacuum_analyze_scale_factor	0.1	Fraction of table size to trigger auto-analyze.

10. Client Connection Defaults

Sets default configurations for client connections, including timeouts and locales.

Parameter	*Default Value*	*Description*
application_name	''	Name of the client application, visible in pg_stat_activity.
search_path	"$user", public	Schema search path.
statement_timeout	0 (disabled)	Time limit for executing a query.
idle_in_transaction_session_timeout	0	Terminates idle transactions after specified time.
client_encoding	SQL_ASCII	Sets default client encoding.

11. Lock Management

Manages table and row-level locking behavior to control concurrency.

Parameter	*Default Value*	*Description*
deadlock_timeout	1s	Timeout to detect deadlocks.
max_locks_per_transaction	64	Maximum locks per transaction.
lock_timeout	0 (disabled)	Sets max wait time for acquiring a lock.

12. Version and Platform Compatibility

Ensures compatibility with previous PostgreSQL versions and different platforms.

Parameter	*Default Value*	*Description*
default_transaction_isolation	read committed	Sets default transaction isolation level.
timezone	UTC	Sets the server's time zone.
integer_datetimes	on	Controls use of integer or floating-point storage for timestamps.

13. Error Handling

Defines handling and logging of errors within the database.

Parameter	*Default Value*	*Description*
log_error_verbosity	default	Sets detail level for error messages (terse, default, verbose).
client_min_messages	notice	Minimum message level sent to the client.
log_min_messages	warning	Minimum message level logged.

14. Config File Includes and Customized Options

Extends configuration using additional files or custom-defined parameters.

Parameter	*Default Value*	*Description*
include	''	Path to an additional configuration file for modular configs.
include_dir	''	Directory containing additional config files.
custom_variable_classes	''	Allows defining custom parameter namespaces.

2. PG_HBA.CONF (HOST-BASED AUTHENTICATION FILE)

Purpose: Controls client authentication by defining who can connect to PostgreSQL, from which hosts, and what authentication methods are required.

Location: Also found in the PostgreSQL data directory.

Contents:

Contains rules for host-based authentication, using parameters like type (local, host, hostssl), database, user, address, and method.

Configures access control for both local and remote connections, specifying whether users must authenticate using methods like password, SCRAM-SHA-256, or trust.

IMPORTANT CONFIGURATION:

1. Authentication Records

This file controls: which hosts are allowed to connect, how clients are authenticated, which PostgreSQL user names they can use, which databases they can access.

Type	*Database*	*User*	*Address*	*Method*	*Description*
local	all	all	-	trust	Allow all local connections without a password.
host	mydb	specific_user	192.168.1.0/24	md5	Require password for specific_user connecting to mydb over TCP/IP.
host	all	admin_user	192.168.2.0/24	scram-sha-256	Require SCRAM-SHA-256 authentication for admin_user from this subnet.
hostssl	all	all	203.0.113.0/24	md5	Require md5 authentication for all users over SSL from specified IP.

2. Include Records

This file allows the inclusion of external files or directories holding more records

Directive	*Value*	*Description*
include	'pg_hba_custom.conf'	Includes additional configuration from the specified file.
include_dir	'pg_hba.d'	Includes all configuration files from the specified directory.

Key Usage: This file is crucial for database security, allowing administrators to control access by restricting IP addresses and enforcing authentication methods.

3. PG_IDENT.CONF (USER IDENTITY MAPPING)

Purpose: Maps operating system user names to PostgreSQL database roles, allowing user authentication based on identity mappings.

Location: Typically located alongside the pg_hba.conf and postgresql.conf files.

Contents:

Contains mappings with columns specifying map name, system user, and PostgreSQL user.

Enables PostgreSQL users to log in as different roles without needing a matching operating system username.

IMPORTANT CONFIGURATION:

Key Usage: Useful in environments where OS user names differ from PostgreSQL roles, or where different systems need access to the database without direct role naming requirements.

COMMANDS LIST

GENERAL COMMAND IN PSQL:

Command	Description	Example
\bind [PARAM]...	Set query parameters.	`\bind date='2024-10-25'`
\copyright	Show PostgreSQL usage and distribution terms.	`\copyright`
\crosstabview [COLUMNS]	Execute query and display result in crosstab format.	`SELECT department, gender, COUNT(*) FROM employees GROUP BY 1, 2; \crosstabview department gender`
\errverbose	Show most recent error message with maximum verbosity.	`\errverbose`
\g [(OPTIONS)] [FILE]	Execute query, optionally sending result to file or pipe; `\g` alone is equivalent to a semicolon.	`SELECT * FROM customers; \g output.txt`
\gdesc	Describe result of query without executing it.	`SELECT * FROM employees; \gdesc`
\gexec	Execute query, then execute each value in its result.	`SELECT 'SELECT * FROM '
\gset [PREFIX]	Execute query and store result in psql variables.	`SELECT COUNT(*) AS total FROM employees; \gset my_` (retrieves result as `:my_total`)
\gx [(OPTIONS)] [FILE]	As `\g`, but forces expanded output mode.	`SELECT * FROM orders; \gx`
\q	Quit `psql`.	`\q`
\watch [[i=]SEC] [c=N]	Execute query every SEC seconds, up to N times.	`SELECT COUNT(*) FROM orders; \watch 5`

HELP COMMAND IN PSQL

Command	Description	Example
`\? [commands]`	Show help on backslash commands.	`\?` (lists all backslash commands)
`\? options`	Show help on `psql` command-line options.	`\? options`
`\? variables`	Show help on special variables available in `psql`.	`\? variables`
`\h [NAME]`	Show help on syntax of SQL commands, or `*` for all commands.	`\h SELECT` (shows syntax for `SELECT`) or `\h *` (lists all SQL commands)

QUERY BUFFER COMMANDS

Command	Description	Example
`\e [FILE] [LINE]`	Edit the query buffer or a specified file in the external editor.	`\e myquery.sql` (opens `myquery.sql` in editor)
`\ef [FUNCNAME [LINE]]`	Edit the definition of a function in the external editor.	`\ef my_function` (opens `my_function` in editor)
`\ev [VIEWNAME [LINE]]`	Edit the definition of a view in the external editor.	`\ev my_view` (opens `my_view` in editor)
`\p`	Display the contents of the current query buffer.	`\p` (shows query buffer content)
`\r`	Clear the current query buffer.	`\r` (empties query buffer)
`\w FILE`	Write the contents of the query buffer to a specified file.	`\w output.sql` (saves buffer content to `output.sql`)

INPUT/OUTPUT COMMANDS

Command	Description	Example
`\copy ...`	Perform SQL `COPY` operation with a data stream to or from the client host.	`\copy my_table TO 'output.csv' CSV HEADER;`
`\echo [-n] [STRING]`	Write a string to standard output; use `-n` to suppress the newline.	`\echo -n "Hello, PostgreSQL!"`
`\i FILE`	Execute commands from a specified file.	`\i script.sql`
`\ir FILE`	Similar to `\i`, but relative to the current script's location.	`\ir ../another_script.sql`
`\o [FILE]`	Redirect all query results to a specified file or pipe.	`\o results.txt`
`\qecho [-n] [STRING]`	Write a string to the output stream defined by `\o`; `-n` suppresses the newline.	`\qecho -n "Saving output to file..."`
`\warn [-n] [STRING]`	Write a string to standard error; `-n` suppresses the newline.	`\warn "This is a warning!"`

CONDITIONAL COMMANDS:

Command	Description	Example
\if EXPR	Starts a conditional block, executing commands if the expression is true.	`\if :my_var > 10` *Execute block if `my_var` is greater than 10.*
\elif EXPR	Provides an alternative condition within the conditional block if the initial `\if` was false.	`\elif :my_var == 5` *Execute block if `my_var` equals 5.*
\else	Specifies the final alternative in the conditional block if none of the previous conditions match.	`\else` *This block executes if none of the above conditions were true.*
\endif	Ends the conditional block.	`\endif` *Marks the end of the conditional structure.*

INFORMATIONAL COMMANDS

Command	Description	Example
\d[S+]	Lists tables, views, and sequences.	`\d+` *Displays all tables, views, and sequences with extra detail.*
\d[S+] NAME	Describes a specific table, view, sequence, or index.	`\d+ employees` *Shows detailed information about the `employees` table.*
\da[S] [PATTERN]	Lists aggregate functions.	`\da+` *Lists all aggregates with extra detail.*
\dA[+] [PATTERN]	Lists access methods.	`\dA` *Displays available access methods.*
\dAc[+] [AMPTRN [TYPEPTRN]]	Lists operator classes.	`\dAc+` *Lists operator classes with additional details.*
\dAf[+] [AMPTRN [TYPEPTRN]]	Lists operator families.	`\dAf` *Lists all operator families.*
\dAo[+] [AMPTRN [OPFPTRN]]	Lists operators of operator families.	`\dAo` *Shows operators in operator families.*
\dAp[+] [AMPTRN [OPFPTRN]]	Lists support functions of operator families.	`\dAp` *Displays support functions of operator families.*
\db[+] [PATTERN]	Lists tablespaces.	`\db+` *Shows tablespaces with additional details.*
\dc[S+] [PATTERN]	Lists conversions.	`\dc` *Displays conversions.*

\dconfig[+] [PATTERN]	Lists configuration parameters.	\dconfig+ *Displays configuration parameters with details.*
\dC[+] [PATTERN]	Lists casts.	\dC *Shows all available casts.*
\dd[S] [PATTERN]	Shows object descriptions not displayed elsewhere.	\dd *Lists object descriptions.*
\dD[S+] [PATTERN]	Lists domains.	\dD+ *Lists domains with additional information.*
\ddp [PATTERN]	Lists default privileges.	\ddp *Shows default privileges.*
\dE[S+] [PATTERN]	Lists foreign tables.	\dE *Displays foreign tables.*
\des[+] [PATTERN]	Lists foreign servers.	\des+ *Displays foreign servers with details.*
\det[+] [PATTERN]	Lists foreign tables.	\det *Shows foreign tables.*
\deu[+] [PATTERN]	Lists user mappings.	\deu+ *Displays user mappings with additional details.*
\dew[+] [PATTERN]	Lists foreign-data wrappers.	\dew *Lists foreign-data wrappers.*
\df[anptw][S+] [FUNCPTRN [TYPEPTRN ...]]	Lists functions filtered by type: aggregate, normal, procedure, trigger, or window.	\df+ *Lists functions with detailed descriptions.*
\dF[+] [PATTERN]	Lists text search configurations.	\dF *Displays text search configurations.*
\dFd[+] [PATTERN]	Lists text search dictionaries.	\dFd *Lists available text search dictionaries.*
\dFp[+] [PATTERN]	Lists text search parsers.	\dFp *Displays text search parsers.*
\dFt[+] [PATTERN]	Lists text search templates.	\dFt *Shows text search templates.*
\dg[S+] [PATTERN]	Lists roles.	\dg *Lists all roles.*
\di[S+] [PATTERN]	Lists indexes.	\di+ *Shows all indexes with details.*
\dl[+]	Lists large objects, same as \lo_list.	\dl+ *Displays large objects with additional information.*
\dL[S+] [PATTERN]	Lists procedural languages.	\dL *Shows procedural languages.*

\dm[S+] [PATTERN]	Lists materialized views.	\dm *Displays materialized views.*
\dn[S+] [PATTERN]	Lists schemas.	\dn+ *Shows schemas with additional details.*
\do[S+] [OPPTRN [TYPEPTRN [TYPEPTRN]]]	Lists operators.	\do *Displays operators.*
\dO[S+] [PATTERN]	Lists collations.	\dO+ *Lists collations with extra detail.*
\dp[S] [PATTERN]	Lists access privileges for tables, views, and sequences.	\dp *Shows access privileges for specified relations.*
\dP[itn+] [PATTERN]	Lists partitioned relations (tables or indexes).	\dP+ *Displays partitioned relations.*
\drds [ROLEPTRN [DBPTRN]]	Lists per-database role settings.	\drds *Shows role settings per database.*
\drg[S] [PATTERN]	Lists role grants.	\drg *Displays role grants.*
\dRp[+] [PATTERN]	Lists replication publications.	\dRp *Shows replication publications.*
\dRs[+] [PATTERN]	Lists replication subscriptions.	\dRs+ *Displays replication subscriptions.*
\ds[S+] [PATTERN]	Lists sequences.	\ds *Lists all sequences.*
\dt[S+] [PATTERN]	Lists tables.	\dt+ *Shows tables with additional details.*
\dT[S+] [PATTERN]	Lists data types.	\dT+ *Displays all data types with details.*
\du[S+] [PATTERN]	Lists roles.	\du *Lists roles.*
\dv[S+] [PATTERN]	Lists views.	\dv+ *Shows views with additional information.*
\dx[+] [PATTERN]	Lists extensions.	\dx+ *Displays extensions.*
\dX [PATTERN]	Lists extended statistics.	\dX *Lists extended statistics.*
\dy[+] [PATTERN]	Lists event triggers.	\dy+ *Shows event triggers.*

\l[+] [PATTERN]	Lists databases.	`\l+` *Displays databases with additional information.*
\sf[+] FUNCNAME	Shows a function's definition.	`\sf+ my_function` *Displays the definition of `my_function` with additional details.*
\sv[+] VIEWNAME	Shows a view's definition.	`\sv+ my_view` *Displays the definition of `my_view` with details.*
\z[S] [PATTERN]	Same as `\dp`; lists access privileges.	`\z+` *Shows access privileges with additional details.*

LARGE OBJECTS COMMANDS

Command	Description	Example
\lo_export LOBOID FILE	Writes (exports) a large object to a specified file.	`\lo_export 12345 '/path/to/file'` *Exports large object with ID 12345.*
\lo_import FILE [COMMENT]	Reads (imports) a file as a large object, with an optional comment.	`\lo_import '/path/to/file' 'Sample Comment'` *Imports a file as a large object with a comment.*
\lo_list[+]	Lists large objects, with + for additional details.	`\lo_list+` *Displays large objects with extra detail.*
\lo_unlink LOBOID	Deletes a specified large object.	`\lo_unlink 12345` *Deletes the large object with ID 12345.*

FORMATTING COMMANDS

Command	Description	Example
\a	Toggles between unaligned (plain text) and aligned (table format) output mode.	`\a` *Switches between unaligned and aligned output.*
\C [STRING]	Sets a title for the table output or unsets it if none is provided.	`\C "Sales Report"` *Sets the title "Sales Report" for the output.*
\f [STRING]	Sets or shows the field separator for unaligned output.	`\f ','` *Sets comma as the field separator in unaligned output.*
\H	Toggles HTML output mode on or off.	`\H` *Switches to HTML output format.*

\pset [NAME [VALUE]]	Sets table output options like border, format, and title.	`\pset border 2` *Sets a thicker border around the table.*
`\t [on	off]`	Toggles visibility of row-only output, hiding or showing column headers.
\T [STRING]	Sets HTML `<table>` tag attributes, such as class or style, or unsets them if none are provided.	`\T 'class="myTable"'` *Adds a CSS class to the HTML table.*
`\x [on	off	auto]`

CONNECTIONS COMMANDS

Command	Description	Example
`\c[onnect] {[DBNAME] [USER] [HOST] [PORT]	conninfo}`	Connects to a new database or changes connection parameters. Use - to skip parameters.
\conninfo	Displays information about the current database connection, including database name, user, host, and port.	`\conninfo` *Shows current connection details.*
\encoding [ENCODING]	Displays or sets the client encoding for character data, like UTF-8 or LATIN1.	`\encoding UTF8` *Sets client encoding to UTF-8.*
\password [USERNAME]	Securely prompts to change the password for the specified user.	`\password user123` *Prompts to change the password for "user123" securely.*

OPERATING SYSTEM COMMANDS

Command	Description	Example
\cd [DIR]	Changes the current working directory to the specified directory `DIR`.	`\cd /home/user/documents` *Changes directory to documents.*
\getenv PSQLVAR ENVVAR	Fetches the value of an environment variable `ENVVAR` and stores it in a psql variable `PSQLVAR`.	`\getenv mypath PATH` *Sets `mypath` to the system PATH.*
\setenv NAME [VALUE]	Sets or unsets an environment variable `NAME` with an optional value `VALUE`.	`\setenv GREETING Hello` *Sets GREETING to "Hello".*
`\timing [on	off]`	Toggles the timing of SQL commands, showing execution time in the output.
\! [COMMAND]	Executes a shell command from within `psql` or starts an interactive shell if no command is given.	`\! ls -la` *Lists files in the current directory.*

VARIABLE COMMANDS

Command	Description	Example
\prompt [TEXT] NAME	Prompts the user with the specified TEXT to set the internal variable NAME.	\prompt 'Enter your name: ' USERNAME *Prompts for username.*
\set [NAME [VALUE]]	Sets the internal variable NAME to VALUE. If no parameters are provided, it lists all variables.	\set DBNAME my_database *Sets the DBNAME variable.* \set *Lists all internal variables.*
\unset NAME	Unsets (deletes) the specified internal variable NAME.	\unset DBNAME *Removes the DBNAME variable.*