

Postgresql

Administration
Matthias COLIN

History

Michael Stonebraker, University of California, Berkeley

Developed by PostgreSQL Global Development Group

- 1985 : post-ingres project
- 1989 : postgres v1
- 1994 : postgres95 with SQL instead of POSTQUEL, client psql
- 1996 : project renamed PostgreSQL + website postgresql.org
- 2016 : PostgreSQL 9.6
- 2017 : PostgreSQL 10
- 2018 : PostgreSQL 11
- 2019 : PostgreSQL 12
- 2020 : PostgreSQL 13

Assets

- RDBMS
- Compliance standard SQL (99.99%)
- Open-source (C/C++)
- Multi-platform
- Object, XML, JSon, Array
- Programming Languages
- Extensions : PostGIS, ...
- Clients : PgAdmin, psql

Architecture PostgreSQL

- Logic:
 - Database(s) : postgres by default
 - schema(s) : public (default, transparent, non secure)
 - Variable search_path = “\$user”, public (by default)
 - Objects: table, view, sequence, index, ...
 - stored code : function, trigger (2 parts): language plpgsql, python, C, perl, ...
 - types
 - materialized view, foreign table
 - NB: no synonym (schema+view)
 - User/role : postgres (user admin)
 - Physique : processus, système de fichiers

Types, functions, operators

- Numeric types :
 - smallint/smallserial, integer/serial, bigint/bigserial : intervalles
 - numeric/decimal : integer or decimal with number of digits fixed
 - real/double precision : float numbers IEEE754
- temporal : timestamp, date, time, interval, with/without timezone
- boolean
- char(n), varchar(n), text
- xml, json/jsonb
- bytea : binary data
- geometric types (and more with PostGis)
- array version of each type : integer[]

Max size of data : 1GB (clob/blob : provided by an extension)

Functions and operators

Complete list here:

<https://www.postgresql.org/docs/11/functions.html>

In particular :

- Temporal : CURRENT_DATE, CURRENT_TIME, CURRENT_TIMESTAMP
- User : CURRENT_USER
- Conditional : case, coalesce, nullif
- Formats de types temporels et numériques :

<https://www.postgresql.org/docs/11/functions-formatting.html>

Server PostgreSQL

- 1st process :
 - postgres.exe -D directory_data (windows)
 - postgres -D directory_data (unix/linux)
- Multi-process server with shared memory
- Life cycle controlled by a service or tool pg_ctl : start/stop/restart/reload

Data directory

Files :

- postgresql.conf : main configuration
- pg_hba.conf : access filters (user, host, database, authentication)

Directories :

- base : data of each database (oid as its name)
- global : global catalog (databases, users, tablespaces)
- log
- pg_tblspc : links to external tablespaces
- pg_wal : journaling files

Backup/Restore

<https://www.postgresql.org/docs/11/backup.html>

- Dump: backup, extraction
 - pg_dump (1 database), pg_dumpall (server, multi-databases, users)
 - psql or pg_restore to restore dump
- External File System Tool (zip, rar, tar, ..., cp, ...) : système arrêté
- PITR : Point In Time Recovery (uses files WAL)
 - Setup archiving WAL files
 - Backup database : pg_backup (copy data directory + tablespaces + tag WAL)
 - Recover database : 9 points process to follow

Tools : pg_dump/pg_dumpall

- Format : plain
 - Backup with pg_dump => restore with psql
- Format : custom, directory, tar
 - Backup with pg_dump => restore with pg_restore
- Main switches :
 - -F format : plain, tar, directory, custom
 - -E encoding
 - -c --clean + --if-exists : drop objects before recreating
 - Data only : -a --data-only + --disable-triggers (--inserts/--column-inserts for insert vs copy)
 - DDL only : -s --schema-only
 - Selection : -n -N (schema) -t -T (table), include or Exclude

Continuous Archiving Logs

<https://www.postgresql.org/docs/11/wal-configuration.html>

- WAL : Write Ahead Log, journalization files
 - Switch on archiving in postgresql.conf
 - Choose command or script to archive : copy, cp, scp, ftp, ...
 - Choose command or script to restore archives
 - pg_switch_wal function
- Current WAL is tagged backup while backuping database
- File recovery.conf => recovery.done
 - Control recover process

Vacuum

- <https://www.postgresql.org/docs/12/routine-vacuuming.html>
- Each table, index, sequence has at least:
 - 1 file of data (relfilenode)
 - 1 file FSM : free space map (fragmentation)
 - 1 file VM : visibility map (current tuples used in transactions)
- Big table, index can use:
 - Several files
 - Toast technique
- Vacuum : simple or combined with analyze and full mode
- Autovacuum: parameters in postgresql.conf

Indexation

<https://www.postgresql.org/docs/11/indexes.html>

<https://www.postgresql.org/docs/11/sql-createindex.html>

<https://www.postgresql.org/docs/11/sql-reindex.html>

- B-tree: =, <, between, like, ~ (begin pattern fix: 'Star%', '^Star')
- hash: =
- GiST: GiS index with GiS operators
- SP-GiST: more complex GiS index
- GIN: inverted index (for array, ...)
- BRIN: Block Range INdexes

Supervision

- Database:
 - service or pg_ctl tool or signal: start|stop|restart|reload
 - pg_reload_conf function
 - pg_rotate_logfile function
- Transactions : pg_lock
- Sessions : pg_stat_activity
 - Kill session : pg_terminate_backend
 - Kill request : pg_cancel_backend
 - Tutorial: https://wiki.postgresql.org/wiki/Lock_Monitoring
- Disk : <https://www.postgresql.org/docs/12/functions-admin.html#FUNCTIONS-ADMIN-DBOBJECT>
 - functions computing size of database, schema, tablespace, table, index
 - functions to obtain filepath or oid from object name (table, index, sequence, ...)
 -
 -