

## Postgres : base de donnée relationnelle-objet



Elephants can remember...

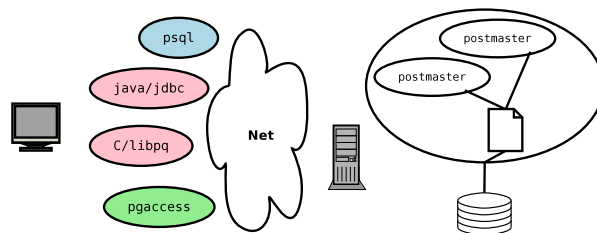
**logiciel libre** gratuit, source  
**architecture** client-serveur  
**serveur** portable **Unix** et **Windows**  
 Linux, \*BSD, Solaris, AIX, MacOS  
**client texte** Unix et Windows  
**clients graphiques** Unix et Windows  
**librairies** applicatives jdbc odbc...

Id: postgresql.tex 3765 2018-04-03 20:22:142 fabien

1

## Postgres : clients et serveur

— accès par protocole réseau (TCP/IP) ou *socket* système



2

### Michael Stonebraker

USA 1943-

- CS Dpt, UC Berkeley
- puis MIT (voir bio)
- Turing Award 2014



**Ingres 1976** 1973-1979/1989

première **base relationnelle**

**Postgres** 1985-1993

recherche **base relationnelle-objet**

financements : DARPA, ARO, NSF, ESL Inc., commercialisation

projets : finances, astéroïdes, SIG...

**Postgres v4.2** 1993 fin du support de Berkeley... logiciel libre

3

### Évolution comme logiciel libre...

**Postgres95 1.0 1994-1996** SQL (Andrew Yu, Jolly Chen)

le nom *Postgres95* vieillit mal...

**PostgreSQL 6.0 à 6.5 1997-1999** doc sub group trigger union... JDBC

PL/pgSQL MVCC *hot backups*

**7.0 à 7.4 2000-2003** intégrité référentielle, *write-ahead log*, jointure

externe, i18n, schéma, req. prép., opts, *information.schema*

**8.0 à 8.4 2004-2009** Win, save point, PITR, tablespace, pgxs, 2PC,

rôles, auto vacuum, performances, types, FTS, column perms...

**9.0 à 10 2010-2017** replicationS (sync, async, cascade, logical), ORM

optimisations, JSON\*, extensions, parallelism...

4

### Politique de gestion des versions (*versioning*)

**release** 9.2 9.3 9.4 9.5 9.6 10 11...

- progrès fonctionnels, typiquement annuel
- changement du format de stockage (*dump/restore*)  
ou bien outil de migration *pg\_upgrade*
- changement de pratique entre 9 et 10

**bugfix** 9.6.0, 9.6.1, 9.6.2, 9.6.3 10.0, 10.1, 10.2, 10.3...

- résolutions de bugs, tous les 2-4 mois
- même format de stockage, sauf si bug critique
- à suivre!

**Support sur 5 ans**, actuellement : 9.3 – 10

5

### Caractéristiques de Postgres

**modèle relationnel** langage SQL très complet, proche du standard...

vues, jointures, triggers, blob, sous requêtes, CTE...

**intégrité** référentielle (vérification des contraintes)

**textes** unicode et autres codages, localisation des opérateurs

**optimiseur** de requête avancé (décisionnel...)

indexation spaciale, textuelle, fonctionnelle, partielle...

**extensions** rules, fonctions, objets (héritage, champs multivalués...)

nombreuses fonctions, types, opérateurs disponibles, MySQL compat

**transactions** ACID, rollback, MVCC, WAL, savepoint, 2-phase commit

**sécurité** via SSL, Kerberos... accès, rôles, permissions fines

**administration** outils, sauvegardes...

6

### Postgres avancé...

**tablespace** gestion de espace et performance disques...

**PL** C SQL pgSQL pl py rb tcl PHP java js sh R scheme PSM V8...

**pgpool-II pgbouncer** partage de connexions et de charge

**duplication** synchrone ou asynchrone...

**postGIS** système d'information géographique

**PITR** reconstruction de l'état à une instant précis

**pgxn** *Postgres eXtension Network*

### Ce qui lui manque ?

**transactions** autonomes ou nichées

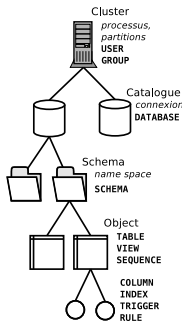
**multi-master** partage des écriture de la duplication

**sauvegarde** incrémentale, bas-niveau, à chaud (FS?)

7

8

### Hierarchie des données



- cluster** installation de la base
- commandes `initdb`, `tablespaces...`
- démarrage de `postmaster` avec `pg_ctl`
- catalogue** une base de donnée isolée
- unité de connexion, d'accès...
- schema** groupe de tables, système ou appli.
- espace de nommage, par défaut `public`
- `SET search_path TO public,pg-catalog;`
- table view séquence** objets relationnels
- column index...** objets dépendants

### Interfaces d'accès à Postgres

- psql** textuelle interactive à *la shell*
- pgadmin\*** graphique à *la access*
- PhpPgAdmin, PostgreSQL Studio** interface web
- java/jdbc C/libpq C/odbc perl/dbi python/db** librairies
- langages : java, C, C++, Perl, Python, PHP...
- fonctions ou classes, spécifiques ou standardisées
- ORM** génération de classes *MyBatis, Hibernate...*
- ecpg** SQL *intégré* à C, traitement par préprocesseur

### Démonstration de psql

- terminal textuel interactif à *la shell*... `man psql`
- commandes SQL et méta commandes `\... : aide SQL \h, cmd \?`
- fichier de configuration `.psqlrc`, historique...
- options : `-h host, -U user, -l...`

```
sh> psql -h pagode comics
psql (10.0)
...
comics> SELECT COUNT(*) FROM auteur;
count
-----
      80
(1 row)

comics> \q
```

### Commandes interactives de psql

- `\c database` connection à un catalogue
- `\?` aide sur les commandes instractives
- `\h...` aide sur les commandes SQL
- `\d...` description des objets
- `t isvsNup` tables, indexes, séquences, vues, système, schéma...
- `\x auto` ajustement automatique de l'affichage
- `\pset linestyle unicode` options diverses...
- `\q` quitter `psql`

Configuration dans fichier `.psqlrc`

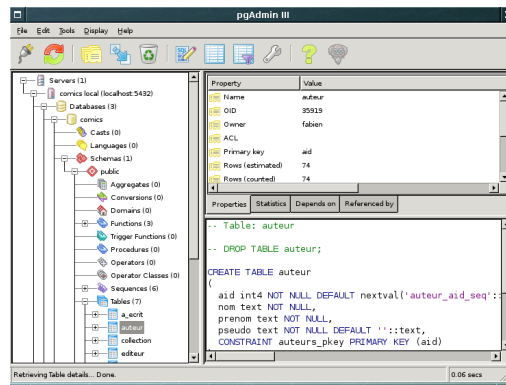
### Intérêt d'une interface texte vs graphique

- administration d'une base à distance
- tests de requêtes (développement, performance)
- pipe | : chargements, bases de tests générées

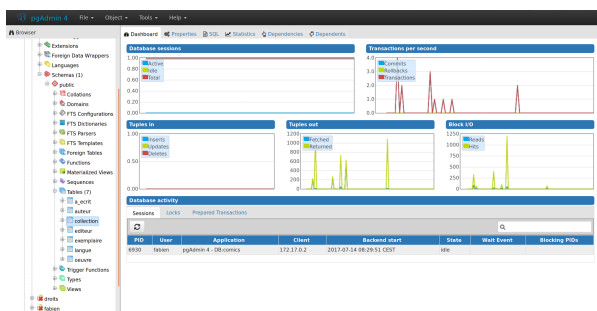
### Inconvénients

- mémorisation des commandes (aide, complétion)
- pas très joli... quoique :-)

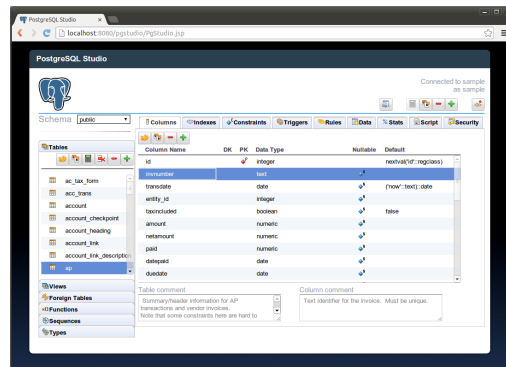
### Illustration de pgadmin3, basé sur wx/gtk



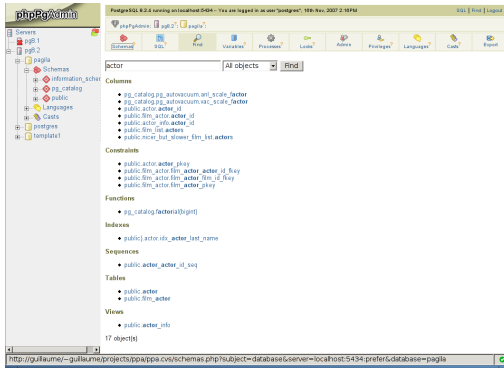
### Illustration de pgadmin4, web/wsgi/python



### PostgreSQL Studio – web / java



## PhpPgAdmin – web/php



17

## Illustration de Python/DB

```
#!/usr/bin/env python3
import psycopg2 as pg
# connect
cnn = pg.connect("")
cur = cnn.cursor()
# query & result
cur.execute("SELECT firstname, lastname, email FROM Students")
for i in range(cur.rowcount):
    row = cur.fetchone()
    print("%d: %s %s <%s>" % (i, row[0], row[1], row[2]))
# cleanup
cur.close()
cnn.close()
```

19

## Illustration de C/libpq

```
#include <stdio.h>
#include <libpq-fe.h>
int main(int argc, char * argv[]) {
    PGconn * conn =
        PQconnectdb("host=/tmp dbname=comics user=coelho");
    PGresult * res =
        PQexec(conn, "SELECT nom FROM auteurs");
    int i, n = PQntuples(res);
    for (i=0; i<n; i++)
        fprintf(stdout, "nom=%s\n", PQgetvalue(res, i, 0));
    PQclear(res);
    PQfinish(conn);
    return 0;
}
```

21

## Illustration de PHP/Postgres

```
<HTML><HEAD><TITLE>Auteurs de BD</TITLE></HEAD>
<BODY><H1>Auteurs de BD</H1>
<?
    $conn = pg_pconnect (
        "dbname=comics user=calvin host=localhost");
    $res = pg_exec($conn, "SELECT id, pseudo FROM auteurs");
    for ($i = 0; $i < pg_numrows($res); $i++) {
        $id = pg_result($res, $i, 0);
        $pseudo = pg_result($res, $i, 1);
        print("id=$id, pseudo=$pseudo ");
    }
?>
</BODY></HTML>
```

23

## Illustration de java/jdbc

```
import java.sql.*;
public class JavaJdbc {
    static public void main(String[] args) throws Exception {
        Class.forName("org.postgresql.Driver");
        Connection c = DriverManager.getConnection
            ("jdbc:postgresql://localhost/comics",
             args[0], args[1]);
        Statement s = c.createStatement();
        ResultSet r =
            s.executeQuery("SELECT id, nom FROM auteurs");
        while (r.next())
            System.out.println(r.getInt(1)+"\t"+r.getString(2));
        r.close(); s.close(); c.close();
    }
}
```

18

## Illustration de Python/anosql

### Fichier comics.sql *définition des requêtes en SQL*

```
-- name: get-auteur
-- récupère les informations d'un auteur
SELECT nom, prenom, pseudo
FROM auteur
WHERE aid = :aid;
```

### Fichier comics.py *utilisation des requêtes*

```
#!/usr/bin/env python3
import anosql as sql
q = sql.load_queries('postgres', 'comics.sql')
import psycopg2 as pg
c = pg.connect('dbname=comics')
print("info: %s" % q.get_auteur(c, aid=3))
```

### Résultat

### *exécution du script*

```
info: [('Chabosy', 'Laurent', 'Lewis Trondheim')]
```

20

## Illustration de ecpg

```
#include <stdio.h>
int main(int argc, char * argv[])
{
    EXEC SQL BEGIN DECLARE SECTION;
    char nom[64];
    int id;
    EXEC SQL END DECLARE SECTION;
    id = atoi(argv[1]);
    EXEC SQL CONNECT TO comics;
    EXEC SQL SELECT nom INTO :nom FROM auteurs WHERE id=:id;
    fprintf(stdout, "id=%d nom=%s\n", id, nom);
    EXEC SQL DISCONNECT;
    return 0;
}
```

22

## Documentation (html, pdf, epub) – +3300 pages A4

<https://www.postgresql.org/docs/current/>

1. **tutoriel** survol rapide
2. **langage SQL** par le menu...
3. **administration** d'un serveur Postgres
4. **interfaces client** texte, librairies...
5. **programmation serveur** pour développement d'extensions
6. **références SQL**, clients, commandes admin
7. **internes** protocole, fonctionnement...
8. **appendixes**, en particulier notes de versions

24

## Publications Supports Dalibo

<https://www.dalibo.com/formations>

- SQL1 SQL2 SQL3 SQL4 SQL5
- DBA1 DBA2 DBA3 DBA4 DBA5
- SIG1 SIG2

## Memento Loxodata

<https://www.loxodata.com/post/memento/>

25

## Test de performance – documentation SQLite

- description <http://www.sqlite.org/speed.html>
- transactionnel, non préparé, HDD *sauf MyISAM et SQLite NO*
- un seul client *pas de blocages...*
- un peu trafiqué pour InnoDB *sinon timeout...*
- configurations par défaut... *probablement bof*

logiciel	SQLite		Postgres	MySQL	
	version	taille		InnoDB	MyISAM
	3.8.6		9.4.1	5.5.41	
	sync	NO			
0.1	29.5	0.3	4.2	11.4	1.5
1.0	208.3	4.1	27.0	80.9	15.2

27

## Exemple d'entreprises utilisatrices

**AIR FRANCE KLM**



**BNP PARIBAS**



**SOCIÉTÉ GÉNÉRALE**



29

- 12 Commandes interactives de `psql`
- 13 Intérêt d'une interface texte vs graphique
- 13 Inconvénients
- 14 Illustration de `pgadmin3`, basé sur `wx/gtk`
- 15 Illustration de `pgadmin4`, `web/wsgi/python`
- 16 PostgreSQL Studio – `web/java`
- 17 `PhpPgAdmin` – `web/php`
- 18 Illustration de `java/jdbc`
- 19 Illustration de `Python/DB`
- 20 Illustration de `Python/anosql`
- 21 Illustration de `C/libpq`
- 22 Illustration de `ecpg`
- 23 Illustration de `PHP/Postgres`

## Postgres vs MySQL (Oracle...)

- Postgres** proche du standard, très complet
- MySQL + MyISAM** pas une base de donnée !  
rapide (?) mais pas de transactions, pas d'intégrité...
- MySQL + InnoDB/DBD** ok, mais plutôt lent ?  
transactions partielles : pas DDL, pas de check...  
performances en progrès sur les versions récentes

26

## Déploiement de Postgres dans les administrations

Circulaire 35837 de JM Ayrault du 19 septembre 2012

- PM/SGG/DISIC
- exemple +400 applications METL et MEDDE en 2013 :  
70% avec PostgreSQL, 3 To de données  
25% avec Oracle, 2 To de données
- Apache/Weblogic/Oracle → Apache/Tomcat/PostgreSQL
- déploiement de PostGIS : agriculture, intérieur, équipement, IGN
- guide d'usage et de migration vers PostgreSQL

28

## List of Slides

- 1 Postgres : base de donnée relationnelle-objet
- 2 Postgres : clients et serveur
- 3 Michael Stonebraker
- 4 Évolution comme logiciel libre...
- 5 Politique de gestion des versions (*versioning*)
- 6 Caractéristiques de Postgres
- 7 Postgres avancé...
- 8 Ce qui lui manque ?
- 9 Hiérarchie des données
- 10 Interfaces d'accès à PostgreSQL
- 11 Démonstration de `psql`
- 24 Documentation (html, pdf, epub) – +3300 pages A4
- 25 Publications Supports Dalibo
- 25 Memento Loxodata
- 26 Postgres vs MySQL (Oracle...)
- 27 Test de performance – documentation SQLite
- 28 Déploiement de Postgres dans les administrations
- 29 Exemple d'entreprises utilisatrices