



# Conception Avancée de Bases de Données

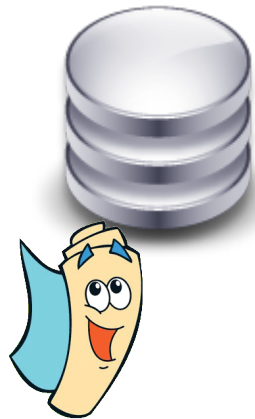
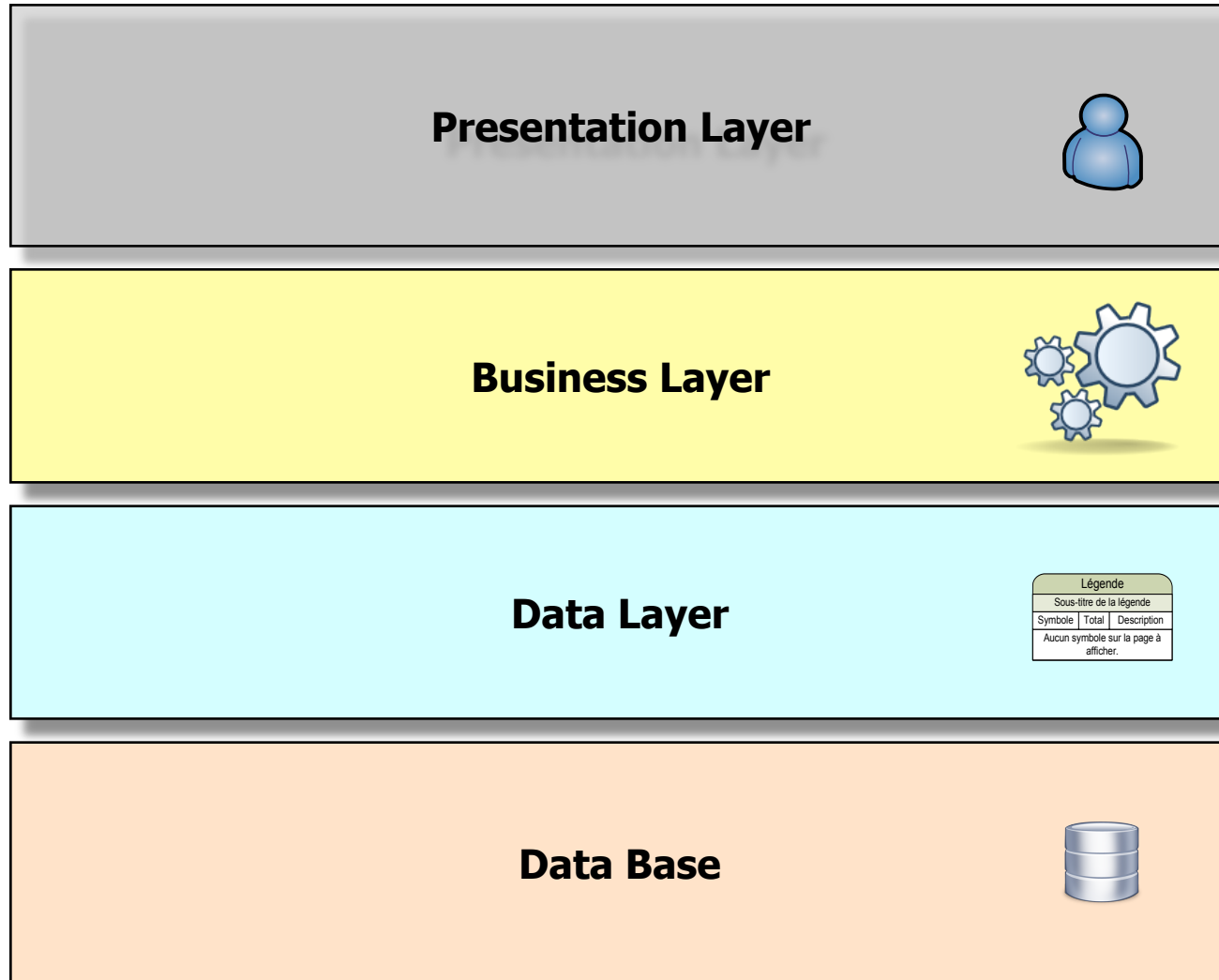
psql



**Traduction en cours**



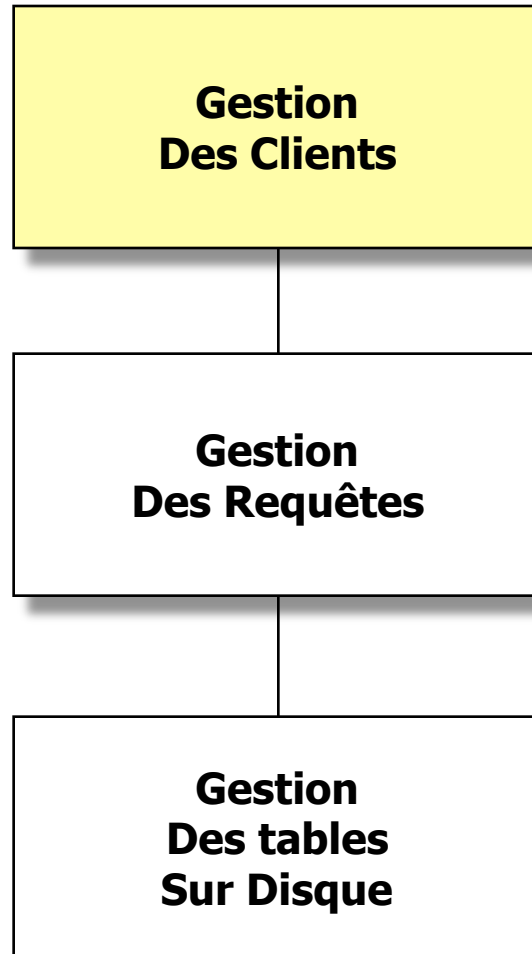
# Layered Architecture

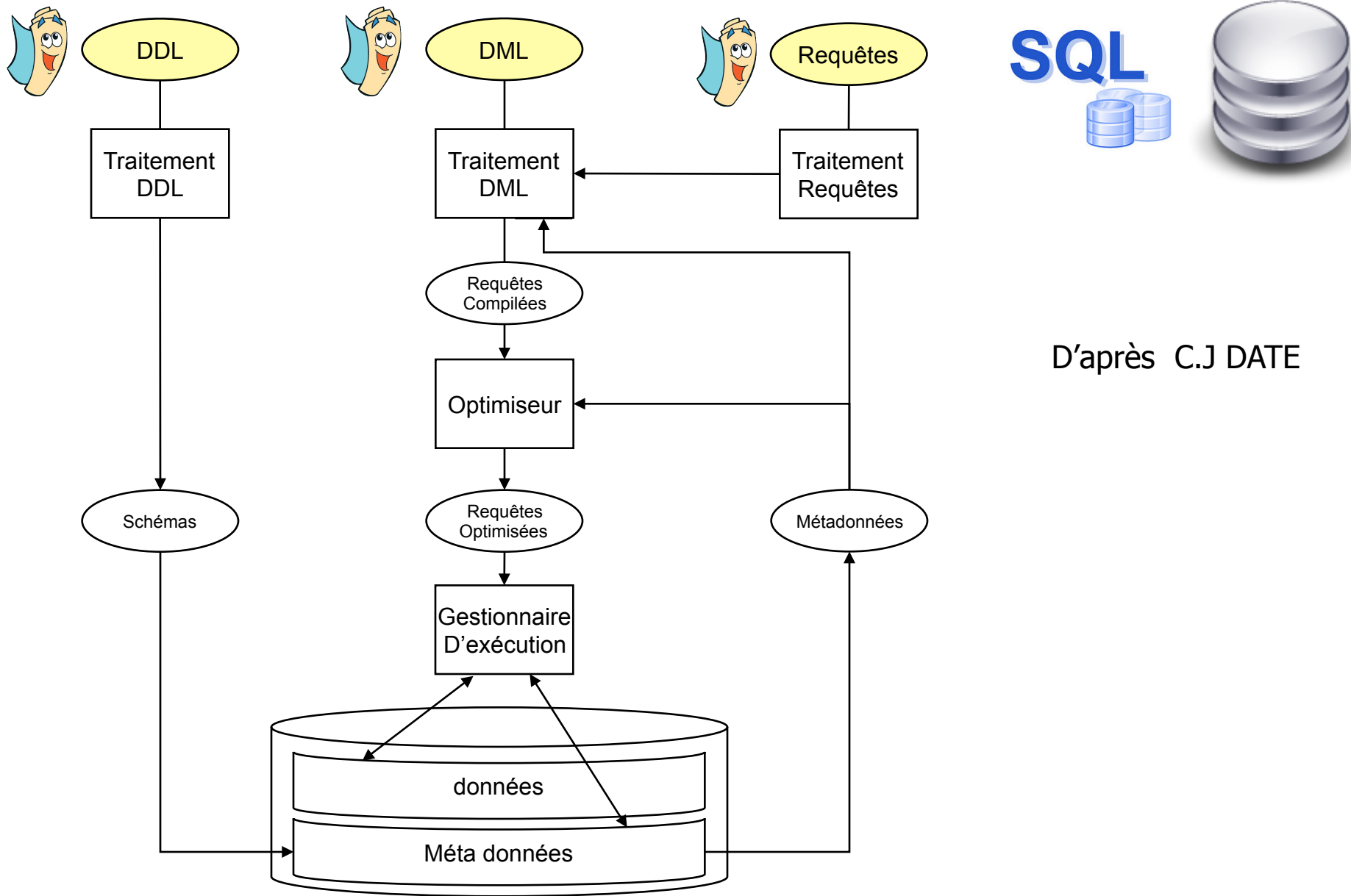


Légende		
Sous-titre de la légende		
Symbole	Total	Description
Aucun symbole sur la page à afficher.		



# Big Picture

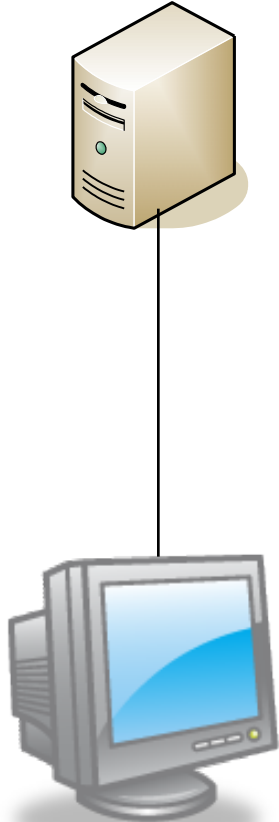




D'après C.J DATE

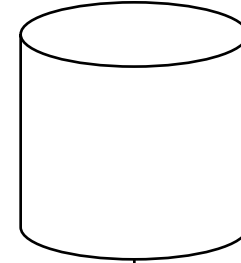
*DDL : langage de définition des données; DML : langage de manipulation des données*

# Client Server



**Server**

**Client**

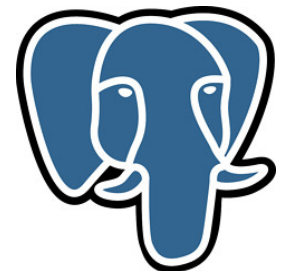
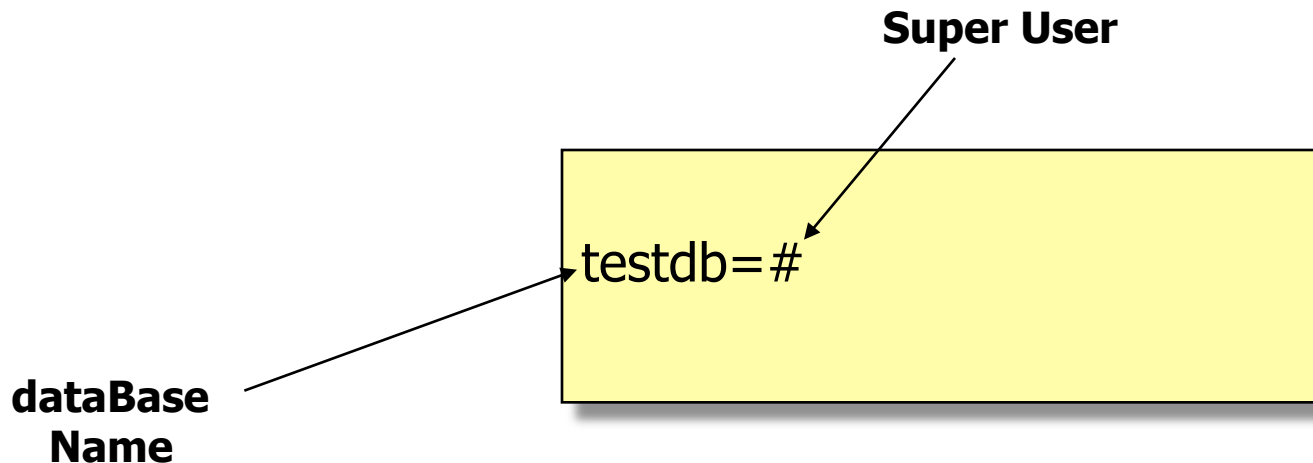


**PSQL**

# PSQL prompt



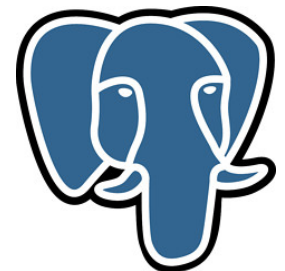
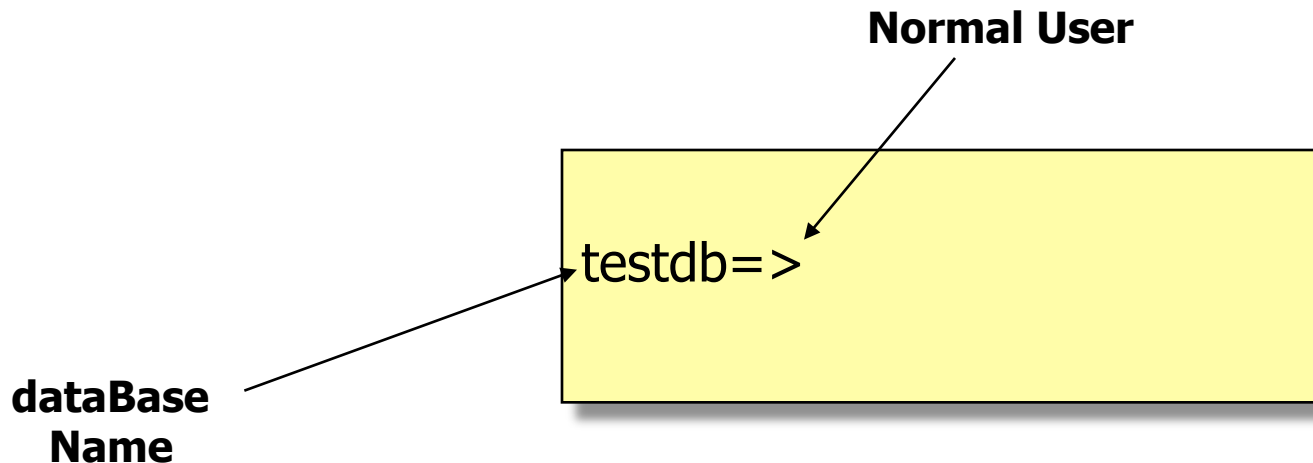
- You will be presented with a prompt that, by default, is set to display the **name of the database** you are currently connected to.
- The prompt will look like this:



# PSQL prompt



- You will be presented with a prompt that, by default, is set to display the **name of the database** you are currently connected to.
- The prompt will look like this:

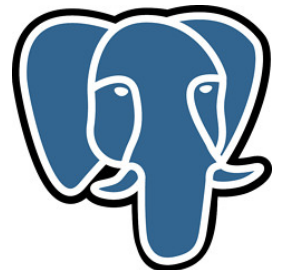


# Command queuing



- Anything you type (except for a slash command) will be queued until you terminate the query with a semicolon.
- Current query buffer.

```
test-> SELECT * FROM employees  
test-> WHERE firstname = 'Michael';
```

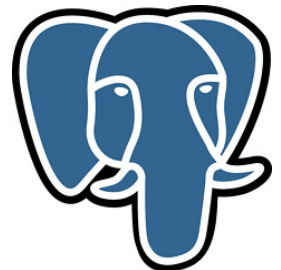




# SQL commands autocompletion



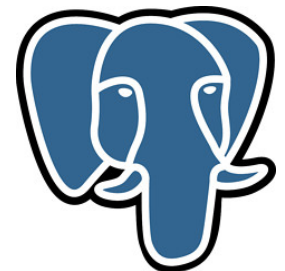
- Inside psql [TAB] can be used for autocompletion of SQL commands.



# PSQL internal commands



**\h -- help on the syntax of various PostgreSQL SQL**  
**\? -- show internal psql commands like**  
**\r -- reset (clear) the query buffer**  
**\dT -- show all data types**  
**\l -- list all databases**  
**\dt -- list tables**  
**\dS -- list system tables**  
**\z -- check user rights for all tables in DB**  
**\connect databasename user -- connect to other DB**  
**\q -- quit psql**

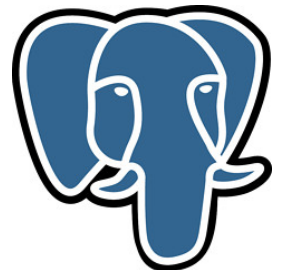


## closing character



- The prompt will show the end-character of a previous line if the character requires a closing character,

```
test=> test (  
test(>  
test(>
```



# closing character



rs=# CREATE TABLE R10 (  
rs(# TheChar VARCHAR(1)  
rs(# );  
CREATE TABLE  
rs=#



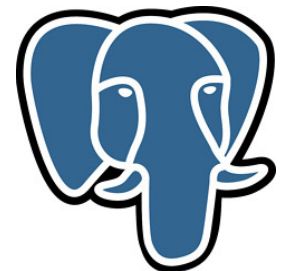
# Prompt changing



```
rs=# SELECT *  
rs-# FROM r10,s10  
rs-# WHERE r10.thechar=s10.thechar;  
thechar | thechar
```

```
-----+-----  
B        | B  
U        | U  
U        | U  
K        | K  
E        | E  
V        | V  
B        | B  
(7 lignes)
```

```
rs=#
```



# bagnoles



bagnoles=> select \*  
bagnoles-> from vendeur;

vendeur	prenom	ville	departement	telephone
germain	alain	Meudon	92	03 54 35 43 45
fayard	icham	Versailles	78	05 45 43 54 55
gentil	idir	tuchan	10	08 46 66 76 67
renard	jacques	paris	75	45 64 56 45 64
blanc	jamel	paris	75	06 77 53 65 36
boucher	jean	Nimes	30	56 45 64 56 54
girard	paul	armentieres	27	04 55 46 65 65
grosjean	philipe	montlus	28	07 56 45 64 56
legoff	pierre	paris	75	06 75 46 46 46
lego	pierre	gerbla	98	06 75 46 46 46
gros	jean	Nimes	10	08 46 75 46 67

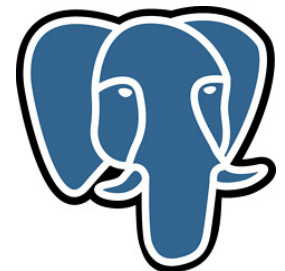
(11 lignes)

bagnoles=>

# PSQL Meta Command



- Anything you enter in psql that begins with an unquoted backslash is a psql meta-command.
- Anything else is SQL and simply goes into the current query buffer (and once you have at least one complete query, it gets automatically submitted to the backend).
- psql meta-commands are also called slash commands.



# PSQL Meta command



`\i <file>` read and execute queries from <file>

`\l` list all databases

`\d{t|i|s|v}` list tables/indices/sequences/views

`\d{p|S|l}` list permissions/system tables/objects

`\q` quit psql

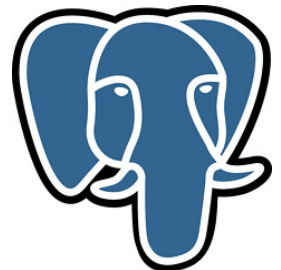




# PSQL Meta command



```
\c[onnect] [dbname]- [user]] connect to new database
```



# Meta Commande : Exécution d'un fichier SQL



```
test=> \i D:/workingDir/req.sql
```



# Prints relation properties



```
postgres=# \l
Alg2SQL      | postgres      | UTF8      | French_France.1252 | French_France.1252 |
CD           | postgres      | UTF8      | French_France.1252 | French_France.1252 |
Dico         | postgres      | UTF8      | French_France.1252 | French_France.1252 |
EMP          | postgres      | UTF8      | French_France.1252 | French_France.1252 |
TEST         | postgres      | UTF8      | French_France.1252 | French_France.1252 | =CTc/postgres
bagnoles    | test         | UTF8      | French_France.1252 | French_France.1252 |
postgres    | postgres      | UTF8      | French_France.1252 | French_France.1252 |
rs           | postgres      | UTF8      | French_France.1252 | French_France.1252 | =CTc/postgres
: postgres=CTc/postgres
stones      | postgres      | UTF8      | French_France.1252 | French_France.1252 |
template0   | postgres      | UTF8      | French_France.1252 | French_France.1252 | =c/postgres
: postgres=CTc/postgres
template1   | postgres      | UTF8      | French_France.1252 | French_France.1252 | =c/postgres
: postgres=CTc/postgres
test        | postgres      | UTF8      | French_France.1252 | French_France.1252 | =CTc/postgres

postgres=#
```



# Meta Commandes



Méta-commandes	Description
<code>\?</code>	Afficher toutes les méta-commandes
<code>\h</code>	Afficher toutes les commandes SQL
<code>\h nom_commande</code>	Aide concernant une commande SQL particulière
<code>\df</code>	Afficher toutes les fonctions postgresql
<code>\cd nom_repertoire</code>	Changer de répertoire courant
<code>\! nom_commande</code>	Exécuter une commande shell
<code>\i nom_fichier</code>	Lire et exécuter un script SQL
<code>\d</code>	Afficher la liste des tables créées
<code>\d nom_table</code>	Information concernant une table créée
<code>\copy nom_table from nom_fichier</code>	Remplissage d'une table à partir d'un fichier texte

## Delimits columns



- If you are using an ASCII formatted input file, a delimiter value may be passed to the DELIMITERS clause,
- which defines the character which delimits columns on a single line in the filename.
- If omitted, PostgreSQL will assume that the ASCII file is **tab-delimited**.

# SQL script execution

- Data base « world »



# PGsql sample Data Set

[http://wiki.postgresql.org/wiki/Sample\\_Databases](http://wiki.postgresql.org/wiki/Sample_Databases)



Sample Databases - PostgreSQL wiki - Mozilla Firefox

Eichier Édition Affichage Historique Marque-pages Outils ?

Sample Databases - PostgreSQL wiki

wiki.postgresql.org/wiki/Sample\_Databases

Google

Les plus visités Démarrage Dernières nouvelles Plombés par la Grèce,...

Log in

## Sample Databases

Many database systems provide sample databases with the product. A good intro to popular ones that includes discussion of samples available for other databases is [Sample Databases for PostgreSQL and More](#). One trivial sample that PostgreSQL ships with is the [Pgbench](#). This has the advantage of being built-in and supporting a scalable data generator—you can make databases of any size ranging from 16MB to 600GB (approximately) with the current version.

### PgFoundry Samples

The latest collection of PostgreSQL compatible database samples is at [PgFoundry Sample Databases](#). It includes three commonly used benchmark databases:

- World: Based on the [MySQL World](#) sample. Has a list of Cities, Countries, and what language they speak.
- dellstore2: PostgreSQL port of a database-neutral e-commerce test application developed by Dell. The original code supports three size scales in their data generator (10MB, 1GB, 100GB); currently only the normal, smallest sized data set has been ported to PostgreSQL. [PostgreSQL 8.4: Windowing Functions](#) uses this test data to show some advanced queries.
- Pagilia: Based on MySQL's replacement for World, [Sakila](#), which is itself inspired by the Dell DVD Store.

There are some other sample databases there as well, such as a USDA Food database and a large list of country data via ISO-3166 standards.

### Other Samples

- [Book Town](#) - used for the examples in [Practical PostgreSQL](#)
- Benchmarking databases such as [DBT-2](#) or [TPC-H](#) can be used as samples.
- [Freebase](#) - Various wiki style data on places/people/things - ~600MB compressed
- [IMDB](#) - the IMDB database - see also <http://code.google.com/p/imbi/>
- [\[1\]](#) - US federal government data collection see also [sunlightlabs](#)
- [DBpedia](#) - wikipedia data export project
- [eoddata](#) - historic stock market data (requires registration - licence?)
- [RITA](#) - Airline On-Time Performance Data
- [Openstreetmap](#) - Openstreetmap source data

Category: Howto

navigation

- Main Page
- Random page
- Recent changes
- Help

# World



- **PgFoundry Samples**

- World: Based on the MySQL World sample. Has a list of Cities, Countries, and what language they speak.
- world.sql

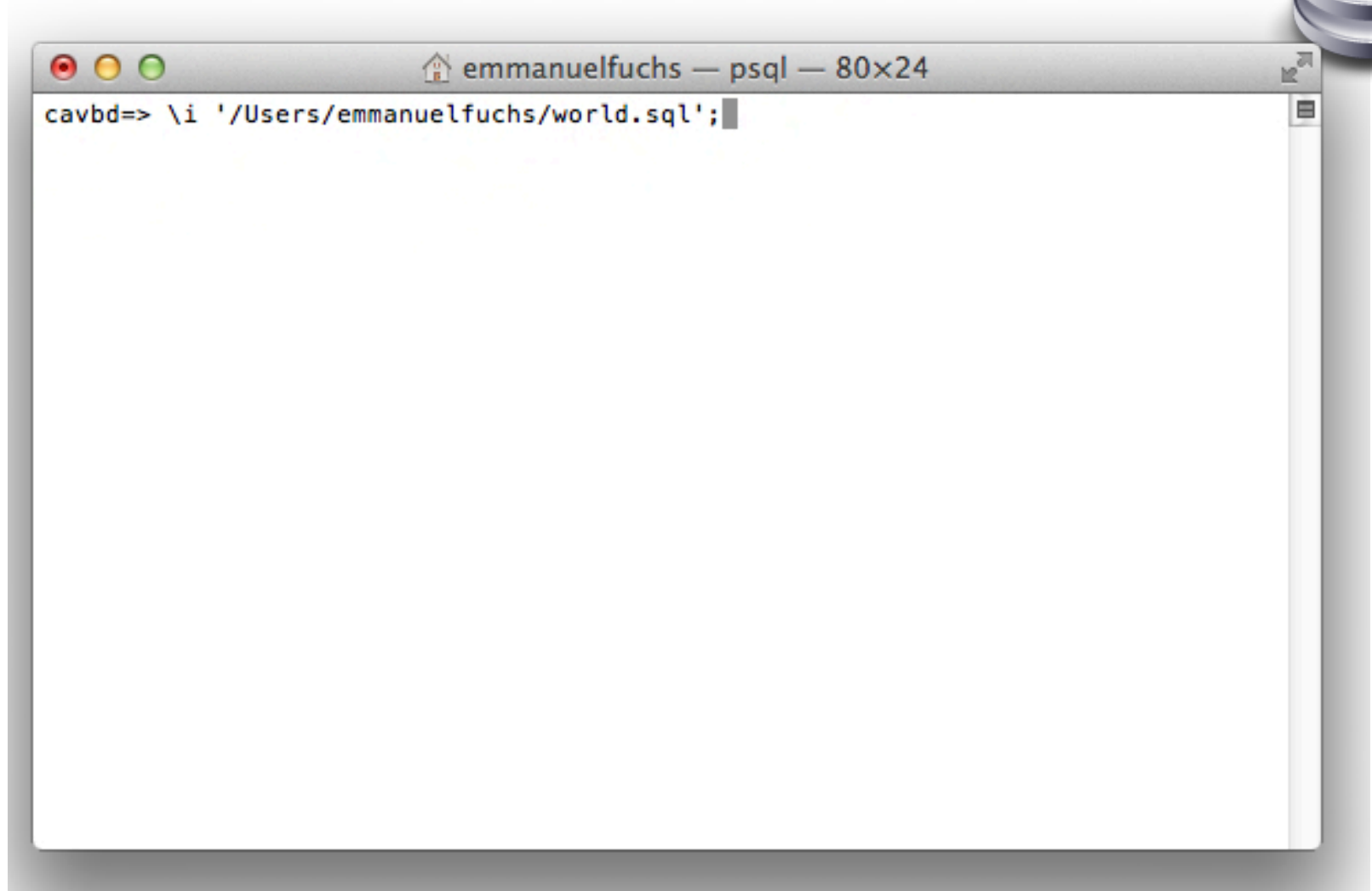


# Meta Commandes : SQL script execution



Méta-commandes	Description
<code>\?</code>	Afficher toutes les méta-commandes
<code>\h</code>	Afficher toutes les commandes SQL
<code>\h nom_commande</code>	Aide concernant une commande SQL particulière
<code>\df</code>	Afficher toutes les fonctions postgresql
<code>\cd nom_repertoire</code>	Changer de répertoire courant
<code>\! nom_commande</code>	Exécuter une commande shell
<code>\i nom_fichier</code>	Lire et exécuter un script SQL
<code>\d</code>	Afficher la liste des tables créées
<code>\d nom_table</code>	Information concernant une table créée
<code>\copy nom_table from nom_fichier</code>	Remplissage d'une table à partir d'un fichier texte

# WORLD.SQL

A screenshot of a terminal window with a grey title bar. The title bar contains three window control buttons (red, yellow, green) on the left, a home icon followed by the text 'emmanuelfuchs — psql — 80x24' in the center, and a close button on the right. The terminal area is white and contains the text 'cavbd=> \i '/Users/emmanuelfuchs/world.sql';' followed by a black cursor. A vertical scrollbar is visible on the right side of the terminal window.

```
cavbd=> \i '/Users/emmanuelfuchs/world.sql';
```

# WORLD.SQL



```
emmanuelfuchs — psql — 80x24
cavbd=> \i '/Users/emmanuelfuchs/world.sql';
BEGIN
SET
CREATE TABLE
CREATE TABLE
CREATE TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
COMMIT
ANALYZE
ANALYZE
ANALYZE
cavbd=> █
```

# TABLES creation



```
emmanuelfuchs — psql — 80x24
cavbd=> \d
 public | city          | table | cavbd
 public | country       | table | cavbd
 public | countrylanguage | table | cavbd
cavbd=> 
```



```
SQL Shell (psql)
Server [localhost]:
Database [postgres]: world
Port [5433]:
Username [postgres]: elfuchs
Mot de passe pour l'utilisateur elfuchs :
Bienvenue dans psql 8.3.17, l'interface interactive de PostgreSQL.

Saisissez:
  \copyright pour les termes de distribution
  \h pour l'aide-mémoire des commandes SQL
  \? pour l'aide-mémoire des commandes psql
  \g ou point-virgule en fin d'instruction pour exécuter la requête
  \q pour quitter

Attention : l'encodage console (850) diffère de l'encodage Windows (1252).
            Les caractères 8 bits peuvent ne pas fonctionner correctement.
            Voir la section « Notes aux utilisateurs de Windows » de la page
            référence de psql pour les détails.

world=> \cd 'E:/5. Home/12. PAST/16. PG2012/1. S1/TOPIC/24. DataBase Sample/world-1.0.tar/dbsamples-0.1/world'
world=> \i world.sql
BEGIN
SET
CREATE TABLE
CREATE TABLE
CREATE TABLE
psql:world.sql:5369: NOTICE: ALTER TABLE / ADD PRIMARY KEY will create implicit
index "city_pkey" for table "city"
ALTER TABLE
psql:world.sql:5372: NOTICE: ALTER TABLE / ADD PRIMARY KEY will create implicit
index "country_pkey" for table "country"
ALTER TABLE
psql:world.sql:5375: NOTICE: ALTER TABLE / ADD PRIMARY KEY will create implicit
index "countrylanguage_pkey" for table "countrylanguage"
ALTER TABLE
ALTER TABLE
ALTER TABLE
COMMIT
ANALYZE
ANALYZE
ANALYZE
world=>
```

# city



```
CREATE TABLE city (  
    id integer NOT NULL,  
    name text NOT NULL,  
    countrycode character(3) NOT NULL,  
    district text NOT NULL,  
    population integer NOT NULL  
);
```

# country

```
CREATE TABLE country (  
    code character(3) NOT NULL,  
    name text NOT NULL,  
    continent text NOT NULL,  
    region text NOT NULL,  
    surfacearea real NOT NULL,  
    indepyear smallint,  
    population integer NOT NULL,  
    lifeexpectancy real,  
    gnp numeric(10,2),  
    gnpold numeric(10,2),  
    localname text NOT NULL,  
    governmentform text NOT NULL,  
    headofstate text,  
    capital integer,  
    code2 character(2) NOT NULL,  
);
```



# language



```
CREATE TABLE countrylanguage (  
    countrycode character(3) NOT NULL,  
    "language" text NOT NULL,  
    isofficial boolean NOT NULL,  
    percentage real NOT NULL  
);
```



# Country Data (PGADMIN)



Édition des données – PostgreSQL 9.3 (localhost:5432) – cavbd – country

Pas de limite

	code [PK] ch	name text	continent text	region text	surfacearea real	indepyear smallint	population integer	lifeexpectanc real	gnp numeric(10,2)	gnpold numeric(10,2)	localname text	governmentfo text	headofstate text	capit integ
1	ABW	Aruba	North America	Caribbean	193		103000	78.4	828.00	793.00	Aruba	Nonmetropol	Beatrix	129
2	AFG	Afghanistan	Asia	Southern ar	652090	1919	22720000	45.9	5976.00		Afganistan	Islamic Emi	Mohammad On	1
3	AGO	Angola	Africa	Central Afr	1.2467e+06	1975	12878000	38.3	6648.00	7984.00	Angola	Republic	José Eduarc	56
4	AIA	Anguilla	North America	Caribbean	96		8000	76.1	63.20		Anguilla	Dependent T	Elisabeth J	62
5	ALB	Albania	Europe	Southern Eu	28748	1912	3401200	71.6	3205.00	2500.00	Shqipëria	Republic	Rexhep Mejç	34
6	AND	Andorra	Europe	Southern Eu	468	1278	78000	83.5	1630.00		Andorra	Parliamenta	'	55
7	ANT	Netherlands Antilles	North America	Caribbean	800		217000	74.7	1941.00		Nederlands	Nonmetropol	Beatrix	33
8	ARE	United Arab Emirates	Asia	Middle East	83600	1971	2441000	74.1	37966.00	36846.00	Al-Imarat	Emirate Fed	Zayid bin	565
9	ARG	Argentina	South America	South Ameri	2.7804e+06	1816	37032000	75.1	340238.00	323310.00	Argentina	Federal Rep	Fernando de	69
10	ARM	Armenia	Asia	Middle East	29800	1991	3520000	66.4	1813.00	1627.00	Hajastan	Republic	Robert Kots	126
11	ASM	American Samoa	Oceania	Polynesia	199		68000	75.1	334.00		Amerika San	US Territor	George W. B	54
12	ATA	Antarctica	Antarctica	Antarctica	1.312e+07		0		0.00		-	Co-administ	'	
13	ATF	French Southern territories	Antarctica	Antarctica	7780		0		0.00		Terres aust	Nonmetropol	Jacques Chi	
14	ATG	Antigua and Barbuda	North America	Caribbean	442	1981	68000	70.5	612.00	584.00	Antigua and	Constitutio	Elisabeth J	63
15	AUS	Australia	Oceania	Australia	7.74122e+06	1901	18886000	79.8	351182.00	392911.00	Australia	Constitutio	Elisabeth J	135
16	AUT	Austria	Europe	Western Eur	83859	1918	8091800	77.7	211860.00	206025.00	Österreich	Federal Rep	Thomas Kle	1523
17	AZE	Azerbaijan	Asia	Middle East	86600	1991	7734000	62.9	4127.00	4100.00	Azərbaycan	Federal Rep	Heydār Əli	144
18	BDI	Burundi	Africa	Eastern Afr	27834	1962	6695000	46.2	903.00	982.00	Burundi/Ubu	Republic	Pierre Buye	552
19	BEL	Belgium	Europe	Western Eur	30518	1830	10239000	77.8	249704.00	243948.00	België/Belg	Constitutio	Albert II	179
20	BEN	Benin	Africa	Western Afr	112622	1960	6097000	50.2	2357.00	2141.00	Bénin	Republic	Mathieu Kér	187
21	BFA	Burkina Faso	Africa	Western Afr	274000	1960	11937000	46.7	2425.00	2201.00	Burkina Fas	Republic	Blaise Com	549
22	BGD	Bangladesh	Asia	Southern ar	143998	1971	129155000	60.2	32852.00	31966.00	Bangladesh	Republic	Shahabuddi	150
23	BGR	Bulgaria	Europe	Eastern Eur	110994	1908	8190900	70.9	12178.00	10169.00	Balgarija	Republic	Petar Stoj	539

