

PostgreSQL Cheat Sheet

Sélection

Sélection de toutes les lignes et colonnes d'une table
SELECT * FROM *table* ;
Idem, mais limité à "colonne"
SELECT *colonne* FROM *table* ;
Limite la sélection aux lignes validant la condition
SELECT * FROM *table* WHERE *condition* ;
Limite la sélection aux n premiers résultats
SELECT * FROM *table* LIMIT *n* ;
Idem, avec un décalage o de la limite
SELECT * FROM *table* LIMIT *n* OFFSET *o* ;

Distinction

Sélectionne toutes les lignes, sans doublons
SELECT DISTINCT * FROM *table* ;
Sélectionne toutes les lignes, sans doublons
SELECT DISTINCT ON(*colonne_a*, *colonne_b*) FROM *table* ;

Agrégation de lignes

Agrège les lignes selon la valeur de "colonne"
SELECT *fonc_agg*(*colonne*) FROM *table* GROUP BY *colonne* ;
Principales fonctions d'agrégation
max, min, count, sum, avg, array_agg, string_agg, bool_and, bool_or

Ordonnancement

Trie les résultats selon la valeur de "colonne"
SELECT * FROM *table* ORDER BY *colonne* ;
Trie les résultats selon la valeur de colonne_a, puis par la valeur de colonne_b par ordre décroissant
SELECT * FROM *table* ORDER BY *colonne_a*, *colonne_b* DESC ;

CTE

Sélectionne toutes les valeurs de "table"
WITH *tableTemporaire* as (
 SELECT * FROM *table*
)
SELECT * FROM *tableTemporaire* ;

Signature synthétique

```
[ WITH requête with [, ...] ]  
SELECT [DISTINCT [ON (expression [, ...])]  
* | expression [[AS] nom [, ...]]  
[FROM table [, ...]]  
[WHERE condition [, ...]]  
[GROUP BY expression [, ...]]  
[ { UNION | INTERSECT | EXCEPT } [ALL] select ]  
[ORBER BY condition [DESC] [, ...]]  
[LIMIT count [, ...]]  
[OFFSET start [, ...]] ;
```

Opérateurs

Opérateurs logiques
AND, OR, NOT
Opérateurs génériques
=, != ou <>, <, >, <=, >=, BETWEEN

Pattern matching

Avec LIKE
_ : n'importe quel caractère, apparaissant une fois
% : n'importe quelle séquence
Avec SIMILAR TO
| : "ou" logique
+ : une ou plusieurs apparitions
* : zéro, une ou plusieurs apparitions ? : zéro ou une apparition
{*m*} : *m* apparitions
{*m*, } : *m* ou plus apparitions
{*m*, *n*} : entre *m* et *n* apparitions
Les opérateurs de LIKE restent valables
Exemples :

Sélectionne les lignes où "colonne" commence par "Pari"
SELECT * FROM *table* WHERE *colonne* LIKE 'Pari%';
Sélectionne les lignes où les deuxièmes et troisièmes caractères de "colonne" sont des voyelles
SELECT * FROM *table*
WHERE *colonne* SIMILAR TO '_(a|e|i|o|u|y){2}%';

Jointures

Ancienne syntaxe (dépréciée)
SELECT * FROM *table_a*, *table_b*
WHERE a.*colonne_jointure* = b.*colonne_jointure* ;
Syntaxe standard
SELECT * FROM *table_a*
JOIN *table_b* ON a.*colonne_jointure* = b.*colonne_jointure* ;
Opérateurs de jointure de la syntaxe standard
CROSS JOIN : produit cartésien des deux tables
[INNER] JOIN : revoie les lignes vérifiant la condition de jointure
LEFT JOIN [OUTER] : revoie toutes les lignes de la table de gauche, ainsi que celles vérifiant la condition de jointure
RIGHT JOIN [OUTER] : idem, mais renvoie toutes les lignes de la table de droite
FULL JOIN [OUTER] : renvoie les lignes jointes, auxquelles s'ajoutent toutes les lignes non jointes des deux tables

Insertion de données

Insertion de valeurs
INSET INTO *table* VALUES (*valeurs*) ;
Insertion de valeurs, colonnes choisies
INSET INTO *table* (*colonnes* [, ...])
VALUES (*valeurs* [, ...]) ;
Insertion de valeurs à partir d'une sélection
INSET INTO *table select* ;
Mise à jour de valeurs
UPDATE *table* SET *colonne* = *expression* ;
Mise à jour de valeurs à partir d'une autre table
UPDATE *table* SET *colonne* = *expression*

FROM *table* WHERE *jointure* ;

Manipulation de tables

Création d'une table
CREATE TABLE *table* (*colonne type* [, ...]) ;
Création d'une table avec une sélection
CREATE TABLE *table* AS *requête_select* ;
Renommer une table
ALTER TABLE *table* RENAME TO *nom* ;
Suppression d'une table
DROP TABLE *table* ;
Ajout d'une colonne
ALTER TABLE *table* ADD COLUMN *colonne type* ;
Suppression d'une colonne
ALTER TABLE *table* DROP COLUMN *colonne* ;
Renommer une colonne
ALTER TABLE *table* RENAME COLUMN *colonne* TO *nom* ;
Modification du type d'une colonne
ALTER TABLE *table* ALTER COLUMN *colonne* TYPE *type* ;

Manipulation de vues

Création d'une vue avec une sélection
CREATE VIEW *vue* AS *requête_select* ;
Suppression d'une colonne
ALTER DROP *vue* DROP COLUMN *colonne* ;
Suppression d'une vue
DROP VIEW *vue* ;

Indexes

Création d'un index
CREATE INDEX [*nom*] ON *table* (*colonne*) ;
Création d'un index de type donné
CREATE INDEX [*nom*] ON *table* USING *type* (*colonne*) ;
Renommer un index
ALTER INDEX *nom* RENAME TO *nom* ;
Supprimer un index
DROP INDEX *nom* ;
Principaux types d'index
GIST, BTREE, BRIN, HASH

Analyse des performances

Calcule et affiche la plan de requête
EXPLAIN *requête* ;
Calcul le plan de requête et le compare à la requête exécutée
EXPLAIN ANALYSE *requête* ;

Entretien de la base

Supprime les tuples morts
VACUUM ;
Supprime les tuples morts, met à jour les statistiques des tables
VACUUM ANALYSE ;

_____ CC-BY-SA, Mattia Bunel, 2018

<https://github.com/MBunel/CheatSheets>

Réalisé pour la version 9.5 de PostgreSQL

NB : Dans un souci de concision ce document ne respecte pas l'indentation traditionnelle.