

Fiche procédure PostgreSQL

Table des matières

- Installation de PostgreSQL
- Création d'une base de données
- Utilisation de PostgreSQL
- Bonnes pratiques
- Exemple de code SQL PostgreSQL

Installation de PostgreSQL

Sous Windows

1. Télécharge l'installateur sur le site officiel PostgreSQL.
2. Lance l'installateur et choisis le répertoire d'installation.
3. Laisse les composants par défaut cochés (serveur PostgreSQL, pgAdmin, Stack Builder).
4. Indique un mot de passe fort pour l'utilisateur super-administrateur `postgres` : retiens-le bien.
5. Laisse le port par défaut (5432) ou modifie-le si besoin.
6. Termine l'installation. Tu peux utiliser pgAdmin (interface graphique) pour gérer tes bases ou la ligne de commande (`psql`).

Sous Linux (exemple Debian/Ubuntu)

1. Mets à jour le système :
2. `bash`

```
sudo apt update && sudo apt upgrade -y
```

- 3.
4. Installe PostgreSQL :
5. `bash`

```
sudo apt install postgresql -y
```

- 6.
7. Vérifie que le service est actif :
8. `bash`

```
sudo systemctl status postgresql
```

9.

Création d'un utilisateur et d'une base

1. Passe à l'utilisateur système `postgres` :
2. `bash`

```
sudo -i -u postgres
```

- 3.
4. Crée un utilisateur PostgreSQL :
5. `bash`

```
createuser --interactive
```

- 6.
7. Crée une base de données :
8. `bash`

```
createdb nom_utilisateur
```

- 9.
10. Connecte-toi à la base :
11. `bash`

```
psql
```

- 12.

Création d'une base de données

Dans `psql` ou via pgAdmin, crée une base de données :

```
sql
```

```
CREATE DATABASE ma_base;
```

Crée une table :

```
sql
```

```
CREATE TABLE utilisateurs (  
  id SERIAL PRIMARY KEY,  
  nom VARCHAR(50),  
  email VARCHAR(100)  
);
```

Utilisation de PostgreSQL

- PostgreSQL est un SGBD relationnel robuste et open source, compatible SQL standard et riche en fonctionnalités.
- Utilise `psql` (ligne de commande) ou pgAdmin (interface graphique) pour gérer tes bases.
- Les opérations principales :
 - `SELECT` : lire des données
 - `INSERT` : ajouter des données
 - `UPDATE` : modifier des données
 - `DELETE` : supprimer des données

Exemple de requête `SELECT` :

```
sql
```

```
SELECT nom, email FROM utilisateurs WHERE nom = 'Dupont';
```

Bonnes pratiques

- Utilise des mots de passe forts pour les comptes administrateurs.
- Nomme clairement tes bases, tables et colonnes.
- Indente et commente tes requêtes pour la lisibilité.
- Indexe les colonnes fréquemment utilisées dans les filtres ou jointures.
- Évite les jointures inutiles pour de meilleures performances.
- Teste et optimise régulièrement tes requêtes (utilise `EXPLAIN`).
- Utilise les transactions (`BEGIN`, `COMMIT`, `ROLLBACK`) pour garantir la cohérence des données.
- Sépare les droits utilisateurs selon les besoins (principe du moindre privilège).
- Surveille l'utilisation des ressources (disque, mémoire, connexions) et ajuste la configuration si besoin.

Exemple de code SQL PostgreSQL

```
sql
```

```
-- Création d'une table
```

```
CREATE TABLE produits (  
  id SERIAL PRIMARY KEY,  
  nom VARCHAR(100),  
  prix NUMERIC(10,2)  
);
```

```
-- Insertion de données
INSERT INTO produits (nom, prix) VALUES ('Clavier', 49.99);

-- Lecture de données
SELECT * FROM produits WHERE prix < 100;

-- Mise à jour
UPDATE produits SET prix = 44.99 WHERE nom = 'Clavier';

-- Suppression
DELETE FROM produits WHERE id = 1;
```