

# Guide de gestion de base de données PostgreSQL

Voici un guide étape par étape pour utiliser PostgreSQL, un puissant système de gestion de bases de données relationnelles open-source :

## 1. Installation

- **Windows** : Téléchargez depuis [postgresql.org/download](https://www.postgresql.org/download) (inclut pgAdmin GUI)
- **macOS** : Utilisez Homebrew (`brew install postgresql`) ou téléchargez l'installateur
- **Linux** (Debian/Ubuntu) :

```
sudo apt update
sudo apt install postgresql postgresql-contrib
sudo systemctl start postgresql.service
```

## 2. Connexion à PostgreSQL

- Utilisateur admin par défaut : postgres

```
sudo -u postgres psql # Linux/macOS
psql -U postgres      # Windows (via PSQL command line)
```

- Connexion à une base de données spécifique :

```
psql -U username -d dbname -h localhost -p 5432
```

## 3. Opérations de base de données

- **Créer un utilisateur/role** :

```
CREATE USER myuser WITH PASSWORD 'mypassword';
ALTER ROLE myuser WITH CREATEDB;
```

- **Créer une base de données** :

```
CREATE DATABASE mydb;
GRANT ALL PRIVILEGES ON DATABASE mydb TO myuser;
```

- **Commandes SQL de base** :

```
-- Créer une table
CREATE TABLE users (
    id SERIAL PRIMARY KEY,
```

```

    name VARCHAR(50),
    email VARCHAR(100) UNIQUE
);

-- Insérer des données
INSERT INTO users (name, email) VALUES ('Alice', 'alice@example.com');

-- Interroger des données
SELECT * FROM users WHERE name LIKE 'A%';

-- Mettre à jour des données
UPDATE users SET email = 'new@email.com' WHERE id = 1;

-- Supprimer des données
DELETE FROM users WHERE id = 2;

```

## 4. Fonctionnalités avancées

- **Jointures :**

```

SELECT orders.id, users.name
FROM orders
INNER JOIN users ON orders.user_id = users.id;

```

- **Support JSON :**

```

CREATE TABLE products (
    id SERIAL PRIMARY KEY,
    details JSONB
);

```

- **Recherche en texte intégral :**

```

SELECT * FROM documents
WHERE to_tsvector('english', content) @@ to_tsquery('search & term');

```

## 5. Maintenance

- **Sauvegarde :**

```

pg_dump mydb > backup.sql
pg_dumpall > all_databases.sql

```

- **Restauration :**

```
psql -U username -d mydb -f backup.sql
```

- **Vacuum** (réclamer de l'espace) :

```
VACUUM ANALYZE;
```

## 6. Sécurité

- Créer un utilisateur en lecture seule :

```
CREATE USER readonly WITH PASSWORD 'readpass';  
GRANT CONNECT ON DATABASE mydb TO readonly;  
GRANT USAGE ON SCHEMA public TO readonly;  
GRANT SELECT ON ALL TABLES IN SCHEMA public TO readonly;
```

## 7. Commandes utiles

- Lister les bases de données : \l
- Basculer de base de données : \c dbname
- Lister les tables : \dt
- Décrire une table : \d tablename
- Afficher les utilisateurs : \du
- Quitter psql : \q

## 8. Outils GUI

- **pgAdmin** : Inclus avec l'installateur Windows
- **DBeaver** : GUI populaire multiplateforme
- **TablePlus** : Interface moderne avec un aspect natif

## 9. Ressources d'apprentissage

- Documentation officielle : [postgresql.org/docs](https://postgresql.org/docs)
- Tutoriel interactif : [postgresqltutorial.com](https://postgresqltutorial.com)
- Pratique : [pgexercises.com](https://pgexercises.com)

**Astuce :** Toujours tester les modifications dans un environnement de développement avant de les appliquer en production. Utilisez des transactions (BEGIN/COMMIT) pour les opérations critiques.