

# PostgreSQL: Datenbankverwaltungshandbuch

Hier ist eine Schritt-für-Schritt-Anleitung zur Verwendung von PostgreSQL, einem leistungsstarken Open-Source-Relationalen Datenbankverwaltungssystem:

## 1. Installation

- **Windows:** Laden Sie von [postgresql.org/download](https://www.postgresql.org/download) herunter (enthält pgAdmin GUI)
- **macOS:** Verwenden Sie Homebrew (`brew install postgresql`) oder laden Sie den Installer herunter
- **Linux** (Debian/Ubuntu):

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

## 2. Verbindung zu PostgreSQL herstellen

- Standard-Administratorbenutzer: postgres

```
sudo -u postgres psql # Linux/macOS
psql -U postgres      # Windows (über PSQL-Befehlszeile)
```

- Verbindung zu einer bestimmten Datenbank herstellen:

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

## 3. Grundlegende Datenbankoperationen

- **Benutzer/Rolle erstellen:**

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

- **Datenbank erstellen:**

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

- **Grundlegende SQL-Befehle:**

```
-- Tabelle erstellen
CREATE TABLE users (
    id SERIAL PRIMARY KEY,
```

```

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

-- Daten einfügen
INSERT INTO users (name, email) VALUES ('Alice', 'alice@example.com');

-- Daten abfragen
SELECT * FROM users WHERE name LIKE 'A%';

-- Daten aktualisieren
UPDATE users SET email = 'new@email.com' WHERE id = 1;

-- Daten löschen
DELETE FROM users WHERE id = 2;

```

## 4. Erweiterte Funktionen

- **Joins:**

```

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

```

- **JSON-Unterstützung:**

```

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

```

- **Volltextsuche:**

```

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

```

## 5. Wartung

- **Sicherung:**

```

pg_dump mydb > backup.sql
pg_dumpall > all_databases.sql

```

- **Wiederherstellung:**

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

- **Vacuum** (Speicher zurückfordern):

```
VACUUM ANALYZE;
```

## 6. Sicherheit

- Erstellen Sie einen schreibgeschützten Benutzer:

```
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. Nützliche Befehle

- Datenbanken auflisten: \l
- Datenbank wechseln: \c dbname
- Tabellen auflisten: \dt
- Tabelle beschreiben: \d tablename
- Benutzer anzeigen: \du
- psql beenden: \q

## 8. GUI-Tools

- **pgAdmin:** Im Windows-Installer enthalten
- **DBeaver:** Beliebtes plattformübergreifendes GUI
- **TablePlus:** Modernes Interface mit nativem Gefühl

## 9. Lernressourcen

- Offizielle Dokumentation: [postgresql.org/docs](https://www.postgresql.org/docs/)
- Interaktives Tutorial: [postgresqltutorial.com](https://www.postgresqltutorial.com/)
- Üben: [pgexercises.com](https://www.pgexercises.com/)

**Tipp:** Testen Sie Änderungen immer in einer Entwicklungsumgebung, bevor Sie sie in die Produktion übernehmen. Verwenden Sie Transaktionen (BEGIN/COMMIT) für kritische Operationen.