

Datenbank Migration

Wie läuft die Migration ab?

Die Migration erfolgt mit `pg_upgrade` – dem Standardwerkzeug für In-Place-Upgrades von PostgreSQL-Instanzen.

→ [Offizielle Dokumentation zu pg_upgrade](#)

Aufbau der Migration

Die Migration wird bei jeder Benutzeranmeldung ausgeführt. Dabei wird geprüft, ob neue `.sql`-Skripte in `sql/Pg-upgrade2` oder `sql/Pg-upgrade2-auth` vorhanden sind, die noch nicht ausgeführt wurden. Bereits angewendete Migrationen werden in der Datenbanktabelle `schema_info` protokolliert, sodass sichergestellt ist, dass jedes Skript nur einmal ausgeführt wird.

Idempotente Skripte

Der Begriff *idempotent* stammt aus dem Lateinischen: *idem* („gleich“) und *potens* („wirksam“).

Ein idempotentes Skript führt bei mehrfacher Ausführung stets zum selben Ergebnis. Das bedeutet, dass sich der Zustand der Instanz nach der ersten erfolgreichen Ausführung durch weitere Ausführungen nicht mehr ändert. Wiederholte Ausführungen haben somit keinen zusätzlichen Effekt.

Beispiel

Betrachten wir folgendes einfache PostgreSQL-Skript:

```
CREATE TABLE users (  
    id SERIAL PRIMARY KEY,  
    name TEXT NOT NULL,  
    email TEXT UNIQUE  
);  
  
CREATE INDEX idx_users_email ON users(email);
```

Wenn dieses Skript zweimal ausgeführt wird, tritt bei der zweiten Ausführung ein Fehler auf. Dies kann die Migration sowie das Update des ERP-Systems verlangsamen oder sogar vollständig verhindern.

Im Folgenden eine sichere, moderne und idempotente Alternative:

```
CREATE TABLE IF NOT EXISTS users (  
    id SERIAL PRIMARY KEY,  
    name TEXT NOT NULL,  
    email TEXT UNIQUE  
);  
  
CREATE INDEX IF NOT EXISTS idx_users_email ON users(email);
```

Vorteile von idempotenter Skripte

Idempotenz bietet den offensichtlichen Vorteil, dass ein Skript mehrfach ausgeführt werden kann, ohne dass sich das Ergebnis verändert. Zwar wirken idempotente Skripte oft komplexer, dennoch lohnt sich der Einsatz dieses Standards.

Weitere, weniger offensichtliche Vorteile sind:

1. Sie sind ideal für komplexe CI/CD-Umgebungen, in denen Software auf mehreren SQL-Instanzen ausgeführt wird.
2. Sie vereinfachen Automatisierung und Rollouts.
3. Sie verhindern Duplikate und Fehler.

Obwohl unser ERP-System durch eine Protokollierung bereits gewährleistet, dass jedes Skript auf einer Datenbankinstanz nur einmal ausgeführt wird, ist Idempotenz in bestimmten Fällen besonders wichtig:

1. Ein Kundenbranch enthält ein sehr ähnliches Migrationsskript.
2. Die Einträge zur Durchführung eines Skripts gehen verloren.
3. Zwei Migrationsskripte führen zum gleichen Ergebnis.

Revision #2

Created 2026-01-14 10:18:11 CET by Nikolay Hadzhiev

Updated 2026-01-14 11:25:15 CET by Nikolay Hadzhiev