

# Database Migratie Instructies - AI Contentplan Generator

---

## Probleem Analyse

---

### Foutmelding:

```
foreign key constraint "ContentPlan_clientId_fkey" cannot be implemented.  
Key columns "clientId" and "id" are of incompatible types: uuid and text.
```

### Root Cause:

- De originele migratie ( `20251212_content_plans_tables.sql` ) gebruikte **UUID** datatypes voor alle ID kolommen
- De bestaande database gebruikt **TEXT** datatypes voor alle IDs (Client.id, BlogPost.id, etc.)
- Dit zorgt voor incompatibele foreign key constraints

### Oplossing:

De gefixte migratie ( `20251212_content_plans_tables_FIXED.sql` ) gebruikt nu **TEXT** voor alle ID kolommen, consistent met de bestaande database structuur.

---

## Stap-voor-Stap Migratie Instructies

---

### Optie A: Als de migratie NOG NIET is uitgevoerd


#### 1. Open Supabase SQL Editor

- Ga naar je Supabase dashboard
- Navigeer naar: `SQL Editor` in het linker menu

#### 2. Run de gefixte migratie

- Open het bestand: `/supabase/migrations/20251212_content_plans_tables_FIXED.sql`
- Kopieer de VOLLEDIGE inhoud
- Plak in de Supabase SQL Editor
- Klik op **"Run"**

#### 3. Verifieer de migratie

- Run de verificatie queries (zie sectie hieronder)
  - Als alles groen is:  Klaar!
- 

### Optie B: Als de migratie AL GEDEELTELIJK is uitgevoerd

#### 1. Run eerst het cleanup script

- Open `/database_cleanup_script.sql`
- Kopieer de inhoud
- Plak in de Supabase SQL Editor

- Klik op **“Run”**
- Dit verwijdert de half-aangemaakte tabellen

## 2. Verifieer cleanup

```
sql
SELECT table_name
FROM information_schema.tables
WHERE table_schema = 'public'
AND table_name IN ('ContentPlan', 'ContentPlanItem');
```

- **Verwacht resultaat:** Geen rijen (tabellen zijn verwijderd)

## 3. Run de gefixte migratie

- Volg stap 2 van Optie A hierboven

## Verificatie Queries

Na het runnen van de gefixte migratie, run deze queries om te verifiëren:

### 1. Check of tabellen bestaan

```
SELECT table_name
FROM information_schema.tables
WHERE table_schema = 'public'
AND table_name IN ('ContentPlan', 'ContentPlanItem')
ORDER BY table_name;
```

**Verwacht:** 2 rijen (ContentPlan, ContentPlanItem)

### 2. Check datatypes (KRITIEK!)

```
SELECT column_name, data_type
FROM information_schema.columns
WHERE table_name = 'ContentPlan'
AND column_name IN ('id', 'clientId');
```

**Verwacht resultaat:**

```
| column_name | data_type |
|-----|-----|
| id | text |
| clientId | text |
```

 **BEIDE moeten ‘text’ zijn!**

### 3. Check foreign key constraints

```
SELECT
  tc.table_name,
  kcu.column_name,
  ccu.table_name AS references_table,
  ccu.column_name AS references_column
FROM information_schema.table_constraints AS tc
JOIN information_schema.key_column_usage AS kcu
  ON tc.constraint_name = kcu.constraint_name
JOIN information_schema.constraint_column_usage AS ccu
  ON ccu.constraint_name = tc.constraint_name
WHERE tc.constraint_type = 'FOREIGN KEY'
AND tc.table_name IN ('ContentPlan', 'ContentPlanItem');
```

**Verwacht:** 3 foreign keys zonder errors



### Wat is er gefixed?

#### Veranderingen in de gefixte migratie:

Kolom	Origineel (FOUT)	Gefixed (CORRECT)
ContentPlan.id	UUID	TEXT
ContentPlan.clientId	UUID	TEXT
ContentPlanItem.id	UUID	TEXT
ContentPlanItem.planId	UUID	TEXT
ContentPlanItem.blogPostId	UUID	TEXT

#### Waarom TEXT?

- De bestaande `Client` tabel gebruikt: `"id" TEXT PRIMARY KEY DEFAULT gen_random_uuid()::TEXT`
- De bestaande `BlogPost` tabel gebruikt: `"id" TEXT PRIMARY KEY DEFAULT gen_random_uuid()::TEXT`
- Alle tabellen in de database volgen dit patroon voor consistentie



### Troubleshooting

#### Error: "relation ContentPlan already exists"

**Oplossing:** Run het cleanup script (Optie B hierboven)

#### Error: "foreign key constraint still fails"

##### Mogelijke oorzaken:

1. Je hebt de oude migratie gerund in plaats van de gefixte versie
2. De `Client` tabel heeft een ander datatype (onwaarschijnlijk)

**Check dit:**

```
SELECT data_type
FROM information_schema.columns
WHERE table_name = 'Client'
AND column_name = 'id';
```

Moet **text** teruggeven.

**Error: “permission denied”**

**Oplossing:** Zorg dat je ingelogd bent als Supabase admin user (service role)

---



## Testing de Migratie

---

Na succesvolle migratie, test de functionaliteit:

**1. Test via de UI:**

- Ga naar `/admin/blog`
- Klik op “AI Contentplan”
- Vul de gegevens in
- Controleer of het plan wordt aangemaakt

**2. Test via SQL (optioneel):**

- Open `/database_verification_queries.sql`
  - Run de test insert queries onderaan
  - Verifieer dat data zonder errors wordt toegevoegd
- 



## Checklist

---

- ☐ Cleanup script gerund (indien nodig)
  - ☐ Gefixte migratie succesvol gerund
  - ☐ Verificatie query 1: Tabellen bestaan
  - ☐ Verificatie query 2: Datatypes zijn TEXT (niet UUID!)
  - ☐ Verificatie query 3: Foreign keys werken
  - ☐ UI test: Content plan generator werkt
  - ☐ Commit de gefixte migratie naar GitHub
-

## Bestand Overzicht

Bestand	Doel
20251212_content_plans_tables_FIXED.sql	✅ Gefixte migratie - RUN THIS!
20251212_content_plans_tables.sql	❌ Origineel (niet gebruiken)
database_cleanup_script.sql	🧹 Cleanup voor gefaalde migratie
database_verification_queries.sql	🔍 Verificatie queries
DATABASE_MIGRATION_INSTRUCTIONS.md	📖 Deze handleiding

## Volgende Stappen

Na succesvolle migratie:

### 1. Test de AI Content Plan Generator in de UI

### 2. Commit de gefixte bestanden:

```
bash
git add supabase/migrations/20251212_content_plans_tables_FIXED.sql
git add database/*.sql DATABASE_MIGRATION_INSTRUCTIONS.md
git commit -m "fix: Correct data types in content plans migration (UUID -> TEXT)"
git push
```

### 3. Update de documentatie:

- Voeg deze migratie toe aan je project README
- Documenteer welke migratie files zijn gebruikt

## Belangrijke Notities

### ⚠️ LET OP:

- Gebruik ALTIJD de **FIXED** versie van de migratie
- De originele migratie file ( 20251212\_content\_plans\_tables.sql ) kan worden verwijderd of hernoemd
- Alle toekomstige migraties moeten TEXT gebruiken voor ID kolommen (niet UUID)

### ✅ Best Practice:

Wanneer je nieuwe tabellen aanmaakt met foreign keys:

1. Check eerst het datatype van de parent table
2. Gebruik HETZELFDE datatype voor de foreign key kolom
3. In deze database: gebruik altijd TEXT voor ID kolommen

**Hulp nodig?** Check de error logs in Supabase en vergelijk met de troubleshooting sectie hierboven.