

Corrigindo o Erro de Dependência de Funções RLS

O Problema

Ao tentar dropar as funções RLS (`is_org_member`, `is_org_admin`, `get_user_org_id`), você recebeu o erro:

```
ERROR: 2BP01: cannot drop function is_org_member(uuid) because other objects depend on it
```

Isso acontece porque **políticas RLS dependem dessas funções**. O PostgreSQL protege contra a remoção acidental de objetos que são usados por outros objetos.

A Solução: CASCADE

O modificador `CASCADE` instrui o PostgreSQL a **automaticamente dropar todos os objetos dependentes** junto com a função.

```
DROP FUNCTION IF EXISTS public.is_org_member(uuid) CASCADE;
```

O que CASCADE faz:

1. **Identifica dependências**: PostgreSQL encontra todas as políticas RLS que usam a função
2. **Remove dependências**: As políticas são automaticamente dropadas
3. **Remove a função**: Finalmente, a função é removida

Políticas que serão dropadas:

Tabela	Política
<code>organizations</code>	“Members can view their org”
<code>profiles</code>	“Users can view profiles in their org”
<code>org_members</code>	“Members can view org members”, “Admins can manage org members”
<code>processes</code>	“Members can view org processes”, “Admins can insert/update/delete processes”
<code>process_events</code>	“Members can view org process events”, “Admins can insert/update process events”

Como Executar

Passo 1: Abra o Supabase SQL Editor

1. Acesse o [Supabase Dashboard](https://supabase.com/dashboard) (<https://supabase.com/dashboard>)
2. Selecione seu projeto
3. Vá para **SQL Editor**

Passo 2: Execute o Script Completo

1. Abra o arquivo `supabase/complete_reset.sql`
2. Copie todo o conteúdo
3. Cole no SQL Editor
4. Clique em **Run**

Passo 3: Verifique o Resultado

Execute as queries de verificação:

```
-- Verificar funções
SELECT proname FROM pg_proc
WHERE proname IN ('is_org_member', 'is_org_admin', 'get_user_org_id');

-- Verificar políticas
SELECT tablename, policymename FROM pg_policies
WHERE tablename IN ('organizations', 'profiles', 'org_members', 'processes', 'process_events')
ORDER BY tablename, policymename;
```

É Seguro?

Sim, porque:

1. O script recria **todas** as funções dropadas
2. O script recria **todas** as políticas RLS removidas pelo CASCADE
3. Usa `IF EXISTS` para evitar erros se objetos não existirem
4. Usa blocos `DO $$` para verificar tabelas opcionais (`processes` , `process_events`)

Estrutura do Script

```
complete_reset.sql
└── Step 1: DROP FUNCTION ... CASCADE (3 funções)
└── Step 2: CREATE FUNCTION (3 funções recriadas)
└── Step 3: CREATE POLICY para organizations (1 política)
└── Step 4: CREATE POLICY para profiles (3 políticas)
└── Step 5: CREATE POLICY para org_members (3 políticas)
└── Step 6: CREATE POLICY para processes (4 políticas, se tabela existir)
└── Step 7: CREATE POLICY para process_events (3 políticas, se tabela existir)
```

Após Executar

1. **Recarregue a aplicação** no navegador
2. **Faça login novamente** (se necessário)

3. **Acesse o Dashboard** - os dados devem carregar corretamente

Troubleshooting

Erro: “policy already exists”

O script já inclui `DROP POLICY IF EXISTS` para tabelas opcionais. Se ainda ocorrer:

```
-- Drole a política manualmente antes de criar
DROP POLICY IF EXISTS "nome_da_politica" ON nome_da_tabela;
```

Erro: “relation does not exist”

As tabelas base ainda não foram criadas. Execute primeiro:

1. `001_multempresa.sql`
2. `004_processes.sql` (se precisar de processos)
3. Depois execute `complete_reset.sql`

Dados desapareceram

CASCADE não remove dados, apenas objetos de esquema (funções, políticas, views). Seus dados estão seguros.