

SCHEMA COMPLETO POUPE AGORA v1 (Supabase/Postgres)

1.1. profiles

Extensão do usuário com dados funcionais.

```
create table profiles (
    id uuid primary key references auth.users(id) on delete cascade,
    full_name text,
    email text unique,
    phone text,
    whatsapp_number text unique,
    avatar_url text,
    role text default 'user', -- user | admin
    plan_id uuid references plans(id),
    subscription_id uuid references subscriptions(id),
    family_id uuid references families(id),
    is_active boolean default true,
    created_at timestamptz default now(),
    updated_at timestamptz default now()
);
```

Índices recomendados:

```
create index idx_profiles_whatsapp on profiles(whatsapp_number);
create index idx_profiles_family on profiles(family_id);
```

1.2. families

Para contas do plano Família/Empresa.

```
create table families (
    id uuid primary key default gen_random_uuid(),
    owner_id uuid references profiles(id),
```

```
    name text,  
    created_at timestampz default now()  
);
```

1.3. family_members

Os “dependentes” / membros adicionais.

```
create table family_members (  
    id uuid primary key default gen_random_uuid(),  
    family_id uuid references families(id) on delete cascade,  
    profile_id uuid references profiles(id),  
    role text default 'member', -- member | viewer | editor  
    created_at timestampz default now()  
);
```

2. BILLING (PLANOS & ASSINATURAS)

2.1. plans

```
create table plans (  
    id uuid primary key default gen_random_uuid(),  
    name text not null,  
    description text,  
    price numeric,  
    periodicity text, -- monthly | yearly  
    ai_limit int default 1000,  
    members_limit int default 1,  
    accounts_limit int default 3,  
    automations_limit int default 5,  
    created_at timestampz default now()  
);
```

2.2. subscriptions

```
create table subscriptions (
    id uuid primary key default gen_random_uuid(),
    profile_id uuid references profiles(id),
    plan_id uuid references plans(id),
    external_id text, -- ID da PerfectPay/Asaas
    status text, -- active | pending | cancelled | trial
    start_date date,
    end_date date,
    next_billing_date date,
    created_at timestamptz default now(),
    updated_at timestamptz default now()
);
```

2.3. billing_events

Logs de webhooks recebidos.

```
create table billing_events (
    id uuid primary key default gen_random_uuid(),
    provider text, -- perfectpay | asaas
    payload jsonb,
    created_at timestamptz default now()
);
```

📌 3. WHATSAPP & INTEGRAÇÕES

3.1. whatsapp_channels

Controla instâncias Evolution/API oficial.

```
create table whatsapp_channels (
    id uuid primary key default gen_random_uuid(),
    provider text, -- evolution | meta
    instance_name text,
    api_key text,
```

```
    is_active boolean default true,  
    created_at timestamptz default now()  
);
```

3.2. whatsapp_events_log

Log bruto de mensagens recebidas.

```
create table whatsapp_events_log (  
    id uuid primary key default gen_random_uuid(),  
    profile_id uuid references profiles(id),  
    raw_payload jsonb,  
    direction text, -- inbound | outbound  
    message_type text, -- text | audio | image | document  
    message_id text,  
    created_at timestamptz default now()  
);
```



4. FINANCE CORE (CORAÇÃO DO SISTEMA)

4.1. accounts

Carteiras/contas bancárias.

```
create table accounts (  
    id uuid primary key default gen_random_uuid(),  
    profile_id uuid references profiles(id),  
    family_id uuid references families(id),  
    name text,  
    type text, -- carteira, conta_bancaria, credito, etc.  
    created_at timestamptz default now()  
);
```

4.2. categories

```
create table categories (
    id uuid primary key default gen_random_uuid(),
    profile_id uuid references profiles(id),
    family_id uuid references families(id),
    name text,
    parent_id uuid references categories(id),
    created_at timestamptz default now()
);
```

4.3. transactions

```
create table transactions (
    id uuid primary key default gen_random_uuid(),
    profile_id uuid references profiles(id),
    family_id uuid references families(id),
    account_id uuid references accounts(id),
    category_id uuid references categories(id),
    owner_member_id uuid references family_members(id),

    type text not null, -- expense | income | transfer
    description text,
    amount numeric not null,
    date date not null,
    source text, -- whatsapp | manual | import
    raw_input text, -- mensagem original
    ai_session_id uuid references ai_sessions(id),

    created_at timestamptz default now(),
    updated_at timestamptz default now()
);
```

Índices importantes:

```
create index idx_transactions_profile on transactions(profile_id);
create index idx_transactions_date on transactions(date);
create index idx_transactions_category on transactions(category_id);
```

4.4. goals (metas)

```
create table goals (
    id uuid primary key default gen_random_uuid(),
    profile_id uuid references profiles(id),
    family_id uuid references families(id),
    name text,
    target_amount numeric,
    current_amount numeric default 0,
    deadline date,
    status text default 'in_progress', -- in_progress | completed | failed
    created_at timestamptz default now()
);
```

4.5. debts (dívidas)

```
create table debts (
    id uuid primary key default gen_random_uuid(),
    profile_id uuid references profiles(id),
    family_id uuid references families(id),
    name text,
    principal numeric,
    interest_rate numeric,
    minimum_payment numeric,
    due_day int,
    status text default 'active',
    created_at timestamptz default now()
);
```

4.6. financial_score (Nota POP)

```
create table financial_score (
    id uuid primary key default gen_random_uuid(),
    profile_id uuid references profiles(id),
```

```
score int, -- 0 a 1000
level text, -- bronze | prata | ouro | diamante
factors jsonb,
calculated_at timestampz default now()
);
```

5. LEMBRETES & AGENDA

5.1. reminders

```
create table reminders (
id uuid primary key default gen_random_uuid(),
profile_id uuid references profiles(id),
family_id uuid references families(id),
description text,
date date,
amount numeric,
status text default 'pending', -- pending | done
created_at timestampz default now()
);
```

6. AUTOMATIONS (RODINAS DO USUÁRIO)

6.1. user_automations

```
create table user_automations (
id uuid primary key default gen_random_uuid(),
profile_id uuid references profiles(id),
name text,
trigger_type text, -- new_transaction | weekly_summary | limit_spent
| custom
conditions jsonb,
actions jsonb,
is_active boolean default true,
```

```
    created_at timestampz default now()
);
```



7. AGENTES DE IA (núcleo da inteligência)

7.1. ai_agents

```
create table ai_agents (
    id uuid primary key default gen_random_uuid(),
    name text,
    slug text unique,
    description text,
    model text, -- gpt-4o, gpt-4o-mini
    temperature numeric default 0.2,
    max_tokens int default 2048,
    is_active boolean default true,
    created_at timestampz default now()
);
```

7.2. ai_agent_prompts

Versionamento de prompts.

```
create table ai_agent_prompts (
    id uuid primary key default gen_random_uuid(),
    agent_id uuid references ai_agents(id),
    version int,
    system_prompt text,
    examples jsonb,
    is_active boolean default true,
    created_at timestampz default now()
);
```

7.3. ai_sessions

Memória de conversa por usuário/intenção.

```
create table ai_sessions (
    id uuid primary key default gen_random_uuid(),
    profile_id uuid references profiles(id),
    agent_id uuid references ai_agents(id),
    current_intent text,
    context jsonb,
    created_at timestamptz default now(),
    updated_at timestamptz default now()
);
```

7.4. ai_logs

Super importante para debug e admin.

```
create table ai_logs (
    id uuid primary key default gen_random_uuid(),
    profile_id uuid references profiles(id),
    agent_id uuid references ai_agents(id),
    request jsonb,
    response jsonb,
    tools_used jsonb,
    duration_ms int,
    created_at timestamptz default now()
);
```



8. OBSERVABILITY & SISTEMA

8.1. webhook_logs

```
create table webhook_logs (
    id uuid primary key default gen_random_uuid(),
    provider text,
    endpoint text,
    payload jsonb,
```

```
status text,  
error_message text,  
created_at timestampz default now()  
);
```