# Fulfillment Backend Development Plan (Supabase) — Version 5

**Owner:** Jonathan Anderson (President Anderson)  
**Context:** Aligns to *Fulfillment Display v5* and the Jonathan Anderson — Master Brief (Oct 7, 2025).  
**Stack:** Supabase (Postgres + Auth + Storage + Edge Functions), TypeScript, Drizzle/Prisma (optional), pg audit, pg\_cron, Stripe (optional), GitHub Actions CI/CD.

## 1) Objectives & Scope

* Provide a production‑ready backend and “super‑based databasing” (Supabase) for Fulfillment v5.
* Model core entities: Areas of Fulfillment, Dimensions, Metrics, Journals (incl. Autobiography), Integrity Recovery Log, Forgiveness & Reconciliation Journal, Rituals & Practices, Relationships Graph, Work & Finance Integration, Holistic Stability Dashboard, and Interpretation Keys.
* Ship secure APIs with granular RLS, migrations, seed, and observability so agents can begin coding immediately.

**Non‑Goals (for v1.0):** UI/UX frontends; heavy analytics ML; third‑party syncs beyond basic webhooks.

## 2) High‑Level Architecture

* **Supabase Project**: Postgres 15+, GoTrue Auth (email + OAuth), Row Level Security enforced by default.
* **Edge Functions**: TypeScript functions for heavy/secure ops (webhooks, batch compute, journaling NLP hooks).
* **API Surface**:
  + **REST** via Supabase auto‑generated endpoints (tables & views).
  + **RPC** via Postgres functions for complex actions (e.g., rollups, snapshotting).
* **Storage**: Buckets for attachments (evidence, media, PDFs), per‑resource prefixes, signed URLs.
* **Jobs**: pg\_cron for scheduled rollups (daily/weekly/monthly) and backups; optional Queue using pgmq.

## 3) Environments & Operational Standards

* **Envs:** dev, staging, prod — isolated Supabase projects; seed data only in dev/staging.
* **Secrets:** Managed via Supabase dashboard + GitHub Actions OIDC; no secrets in repo.
* **Migrations:** SQL first (db/supabase/migrations). Use supabase db push + codegen.
* **Branching:** main (prod), develop (staging), feature branches via PRs.
* **Observability:** pg audit enabled; logs shipped to Logflare/Datadog; structured JSON logs from edge functions.

## 4) Data Model (ER Overview)

users (supabase.auth) → profiles (1:1)  
profiles → areas (1:N)  
areas → dimensions (1:N)  
dimensions → metrics (1:N)  
metrics → metric\_entries (1:N)  
profiles → journals (1:N) → journal\_entries (1:N)  
journal\_entries ↔ tags (M:N) via journal\_entry\_tags  
profiles → rituals (1:N) → ritual\_sessions (1:N)  
profiles → forgiveness\_logs (1:N) / reconciliation\_logs (1:N)  
profiles → integrity\_events (1:N)  
profiles → relationships (1:N) & relationship\_events (1:N)  
profiles → goals (OKR-style) → goal\_updates  
profiles → interpretations (auto-interpretation keys)  
profiles → dashboards (1:N) → dashboard\_snapshots  
work\_finance\_integrations (per profile) → wfi\_periods (monthly/quarterly)  
attachments (polymorphic) linked to journal\_entries, events, goals, etc.  
  
Cross-cutting: audit\_log, notifications, webhooks, api\_keys.

## 5) SQL Schema (Core) — **Run as migrations**

**Note:** All tables ship with RLS = ON and policies below. UUID primary keys. Timestamps in UTC with created\_at, updated\_at. Soft‑delete via deleted\_at where noted.

-- 5.1 Profiles  
create table public.profiles (  
 id uuid primary key references auth.users(id) on delete cascade,  
 display\_name text,  
 locale text default 'en-CA',  
 timezone text default 'America/Toronto',  
 created\_at timestamptz default now(),  
 updated\_at timestamptz default now()  
);  
  
-- 5.2 Areas of Fulfillment (aligned to v5)  
create type area\_kind as enum (  
 'WORK\_PURPOSE',  
 'MUSIC',  
 'WRITING',  
 'PUBLIC\_SPEAKING',  
 'LEARNING\_GROWTH',  
 'FRIENDSHIP',  
 'COMMUNITY',  
 'FAMILY',  
 'HEALTH',  
 'SPIRITUAL',  
 'FINANCE',  
 'LAW\_JUSTICE'  
);  
  
create table public.areas (  
 id uuid primary key default gen\_random\_uuid(),  
 profile\_id uuid not null references public.profiles(id) on delete cascade,  
 kind area\_kind not null,  
 title text not null,  
 emoji text,  
 color text,  
 position int default 0,  
 created\_at timestamptz default now(),  
 updated\_at timestamptz default now(),  
 unique(profile\_id, kind)  
);  
  
-- 5.3 Dimensions (per area)  
create table public.dimensions (  
 id uuid primary key default gen\_random\_uuid(),  
 area\_id uuid not null references public.areas(id) on delete cascade,  
 title text not null,  
 description text,  
 emoji text,  
 color text,  
 position int default 0,  
 created\_at timestamptz default now(),  
 updated\_at timestamptz default now()  
);  
  
-- 5.4 Metrics & Entries  
create table public.metrics (  
 id uuid primary key default gen\_random\_uuid(),  
 dimension\_id uuid not null references public.dimensions(id) on delete cascade,  
 key text not null,  
 unit text default 'score',  
 kind text check (kind in ('scalar','boolean','percent','enum','duration','money')) default 'scalar',  
 min\_value numeric,  
 max\_value numeric,  
 enum\_options text[],  
 target numeric,  
 created\_at timestamptz default now(),  
 updated\_at timestamptz default now(),  
 unique(dimension\_id, key)  
);  
  
create table public.metric\_entries (  
 id uuid primary key default gen\_random\_uuid(),  
 metric\_id uuid not null references public.metrics(id) on delete cascade,  
 value\_numeric numeric,  
 value\_text text,  
 value\_bool boolean,  
 as\_of date not null,  
 note text,  
 created\_at timestamptz default now(),  
 updated\_at timestamptz default now(),  
 unique(metric\_id, as\_of)  
);  
  
-- 5.5 Journals (incl. Autobiography linkage)  
create type journal\_kind as enum ('GENERAL','AUTOBIOGRAPHY','INTEGRITY','FORGIVENESS','RECONCILIATION');  
  
create table public.journals (  
 id uuid primary key default gen\_random\_uuid(),  
 profile\_id uuid not null references public.profiles(id) on delete cascade,  
 kind journal\_kind not null default 'GENERAL',  
 title text,  
 created\_at timestamptz default now(),  
 updated\_at timestamptz default now()  
);  
  
create table public.journal\_entries (  
 id uuid primary key default gen\_random\_uuid(),  
 journal\_id uuid not null references public.journals(id) on delete cascade,  
 content markdown not null,  
 mood int check (mood between 1 and 10),  
 related\_area uuid references public.areas(id),  
 related\_dimension uuid references public.dimensions(id),  
 occurred\_at timestamptz not null default now(),  
 created\_at timestamptz default now(),  
 updated\_at timestamptz default now(),  
 deleted\_at timestamptz  
);  
  
create table public.tags (  
 id uuid primary key default gen\_random\_uuid(),  
 name citext unique  
);  
  
create table public.journal\_entry\_tags (  
 journal\_entry\_id uuid references public.journal\_entries(id) on delete cascade,  
 tag\_id uuid references public.tags(id) on delete cascade,  
 primary key (journal\_entry\_id, tag\_id)  
);  
  
-- 5.6 Integrity / Forgiveness / Reconciliation Events  
create table public.integrity\_events (  
 id uuid primary key default gen\_random\_uuid(),  
 profile\_id uuid not null references public.profiles(id) on delete cascade,  
 title text not null,  
 description text,  
 category text check (category in ('breach','repair','commitment','boundary')) not null,  
 status text check (status in ('open','in\_progress','resolved')) default 'open',  
 occurred\_at timestamptz not null,  
 resolved\_at timestamptz,  
 created\_at timestamptz default now(),  
 updated\_at timestamptz default now()  
);  
  
create table public.forgiveness\_logs (  
 id uuid primary key default gen\_random\_uuid(),  
 profile\_id uuid not null references public.profiles(id) on delete cascade,  
 subject text not null,  
 narrative markdown,  
 stage text check (stage in ('acknowledge','feel','release','recommit')),  
 occurred\_at timestamptz default now(),  
 created\_at timestamptz default now(),  
 updated\_at timestamptz default now()  
);  
  
create table public.reconciliation\_logs (  
 id uuid primary key default gen\_random\_uuid(),  
 profile\_id uuid not null references public.profiles(id) on delete cascade,  
 counterparty text not null,  
 intent markdown,  
 outcome markdown,  
 occurred\_at timestamptz default now(),  
 created\_at timestamptz default now(),  
 updated\_at timestamptz default now()  
);  
  
-- 5.7 Rituals & Practices  
create table public.rituals (  
 id uuid primary key default gen\_random\_uuid(),  
 profile\_id uuid not null references public.profiles(id) on delete cascade,  
 title text not null,  
 cadence text check (cadence in ('daily','weekly','monthly','custom')) default 'daily',  
 instructions markdown,  
 created\_at timestamptz default now(),  
 updated\_at timestamptz default now()  
);  
  
create table public.ritual\_sessions (  
 id uuid primary key default gen\_random\_uuid(),  
 ritual\_id uuid not null references public.rituals(id) on delete cascade,  
 did\_happen boolean default true,  
 notes text,  
 occurred\_at timestamptz default now(),  
 created\_at timestamptz default now()  
);  
  
-- 5.8 Relationships Graph  
create type relation\_kind as enum ('mother\_child','father\_child','sibling\_playmate','admired','admiring','partner','friend','colleague');  
  
create table public.relationships (  
 id uuid primary key default gen\_random\_uuid(),  
 profile\_id uuid not null references public.profiles(id) on delete cascade,  
 person\_name text not null,  
 kind relation\_kind not null,  
 started\_on date,  
 ended\_on date,  
 notes markdown,  
 created\_at timestamptz default now(),  
 updated\_at timestamptz default now()  
);  
  
create table public.relationship\_events (  
 id uuid primary key default gen\_random\_uuid(),  
 relationship\_id uuid not null references public.relationships(id) on delete cascade,  
 title text,  
 detail markdown,  
 occurred\_at timestamptz not null,  
 created\_at timestamptz default now()  
);  
  
-- 5.9 Goals (OKR) & Updates  
create table public.goals (  
 id uuid primary key default gen\_random\_uuid(),  
 profile\_id uuid not null references public.profiles(id) on delete cascade,  
 title text not null,  
 area\_id uuid references public.areas(id),  
 dimension\_id uuid references public.dimensions(id),  
 target\_date date,  
 status text check (status in ('planned','active','paused','done','dropped')) default 'planned',  
 created\_at timestamptz default now(),  
 updated\_at timestamptz default now()  
);  
  
create table public.goal\_updates (  
 id uuid primary key default gen\_random\_uuid(),  
 goal\_id uuid not null references public.goals(id) on delete cascade,  
 summary markdown,  
 progress numeric,  
 created\_at timestamptz default now()  
);  
  
-- 5.10 Work & Finance Integration (monthly/quarterly)  
create table public.work\_finance\_integrations (  
 id uuid primary key default gen\_random\_uuid(),  
 profile\_id uuid not null references public.profiles(id) on delete cascade,  
 framework\_version text default 'v5',  
 created\_at timesta